BIOMEDICAL DATA VISUALIZATION: METHODS AND APPLICATIONS

EDITED BY: Guangchuang Yu, Tommy Tsan-Yuk Lam, Chuan-Le Xiao and Meng Zhou

PUBLISHED IN: Frontiers in Genetics

frontiers Research Topics

ISSN 1664-8714
ISBN 978-2-88976-190-6
DOI 10.3389/978-2-88976-190-6

About Frontiers

Frontiers is more than just an open-access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers Journal Series

The Frontiers Journal Series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the Frontiers Journal Series operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to Quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews.
Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the Frontiers Journals Series: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area! Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers Editorial Office: frontiersin.org/about/contact

BIOMEDICAL DATA VISUALIZATION: METHODS AND APPLICATIONS

Topic Editors:
Guangchuang Yu, Southern Medical University, China
Tommy Tsan-Yuk Lam, The University of Hong Kong, Hong Kong, SAR China
Chuan-Le Xiao, Sun Yat-sen University, China
Meng Zhou, Wenzhou Medical University, China

Citation: Yu, G., Lam, T. T.-Y., Xiao, C.-L., Zhou, M., eds. (2022). Biomedical Data Visualization: Methods and Applications. Lausanne: Frontiers Media SA.
doi: 10.3389/978-2-88976-190-6

Table of Contents

frontiers | Frontiers in Genetics

EDITORIAL
published: 27 April 2022
doi: 10.3389/fgene.2022.890775

Editorial: Biomedical Data Visualization: Methods and Applications

Tianzhi Wu[1], *Chuan-Le Xiao*[2], *Tommy Tsan-Yuk Lam*[3,4] *and Guangchuang Yu*[1]*

[1]Department of Bioinformatics, School of Basic Medical Sciences, Southern Medical University, Guangzhou, China, [2]State Key Laboratory of Ophthalmology, Zhongshan Ophthalmic Center, Sun Yat-sen University, Guangzhou, China, [3]State Key Laboratory of Emerging Infectious Diseases, School of Public Health, The University of Hong Kong, Pokfulam, Hong Kong SAR, China, [4]Laboratory of Data Discovery for Health Limited, Hong Kong, Hong Kong SAR, China

Keywords: biomedical data, visualization, transcriptomics, virtual reality, network visualiztion

Editorial on the Research Topic

Biomedical Data Visualization: Methods and Applications

"A picture tells a thousand words." This is very true in many circumstances but particularly for modern academia as a nicely illustrative figure grabs our attention and helps us explain scientific findings. Data visualization is the most effective way to explain and convey rich information, especially for complex biomedical data. The rapid growth of biomedical data in both volume and complexity creates new challenges in presenting data effectively and accurately (O'Donoghue et al., 2018). This includes exploring data to reveal hidden information and presenting analysis findings. New visualization methods are being developed to address new emerging problems or provide new insights into old data types. Innovations in visualization will continue to revolutionize how we learn from our data, and it is extremely important for biomedical research.

Scatter plots and line charts are the most basic form of data visualization through simply mapping variables to data points. Bar charts, histograms, boxplots, and heatmaps are also widely used. These methods and their combination solve most of the visualization needs, including presenting data overview or summary and helping us to identify patterns or trends. With the rapid growth data volume, effectively utilizing plotting space becomes much in demand. For example, how to truncate an overly long chart, rearrange and display the key parts of a graph in the limited space for publication, and how to select parts of them to zoom in or zoom out while keeping the panorama of the current chart. These are the details that need to be resolved. The ggbreak package can be used as an example in this respect (Xu et al.). It was designed to solve the above issues, by increasing the available visual space for a better presentation of the data and detailed annotations and thus improves our ability to interpret the data. The ggbreak package is consistent with the ggplot2 package by following the syntax of the grammar of graphics (Wilkinson, 2010) and implementing such syntax. There is no additional learning cost to use ggbreak if users are familiar with the ggplot2 syntax. Another package we introduce here is smplot (Min and Zhou), which is also based on R and ggplot2, it simplifies the plotting process of commonly statistical graphs for easy visualize, such as violin plot, slope chart, raincloud plot, and so on. These tools reflect a trend in the development of basic tools: solving practical problems while no additional cost is added and keeping it a good user experience.

It is always right to choose the appropriate visualization method according to specific needs. The Venn diagram can efficiently reflect the relationship between multiple sets. Therefore, it is often used to distinguish members of gene sets, pathways, species, etc. When the number of sets is less than 5, the Venn diagram is a more intuitive form of data visualization than heatmap or

Edited and reviewed by:
Richard D. Emes, University of Nottingham, United Kingdom

***Correspondence:**
Guangchuang Yu
gcyu1@smu.edu.cn

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

Received: *06 March 2022*
Accepted: *28 March 2022*
Published: *27 April 2022*

Citation:
Wu T, Xiao C-L, Lam TT-Y and Yu G (2022) Editorial: Biomedical Data Visualization: Methods and Applications. Front. Genet. 13:890775. doi: 10.3389/fgene.2022.890775

tables. This is also the reason why the Venn diagram appears frequently in biomedical research publications. The ggVennDiagram package (Gao et al.) has been developed as a systematic and easy-to-use method for calculating overlapping members in different sets and visualizing such intersection information in Venn diagrams based on the ggplot2 syntax. It has some features that are not available in general tools, including novel shapes and color filling of different proportion regions. When we can provide new features and perspectives, it is sometimes necessary to reinvent the wheel. The tools have been continuously improved during such overthrows.

When discussing biomedical visualization, it is more than a direct display of given data. Usually, the data sets in the modern biomedical field are complex. Before coming to results and conclusions, scientists spend most of their time processing and exploring data. Therefore, many tailored visualization tools have been developed to meet the needs of data exploration. For example, there is a growing demand in different biomedical research scenarios for network visualization of the relationship between different types of nodes with complex metadata. Integrating different attribute information of nodes and edges in a network may inspire new insights. The CrossLink package (Liu et al.) was designed to plot a network diagram with node attributes as graph annotation aligned to the network. The HandyCNV package (Zhou et al.) is also developed for a specific need. It provides common functions for CNV (copy number variant) and ROH (runs of homozygosity) research, including basic data processing and also essential visualization. Designed as a one-stop tool, HandyCNV aims to make analysis easier and more efficient. Similarly, in the field of microbiome, an easy-to-use tool called EasyMicroPlot (Liu et al.) provides analysis and visualization for clinical microbial studies. Overall, these tools reflect the current needs of visualization methods: to develop standardized, time-saving, and user-friendly one-stop tools for corresponding scenarios.

Data from clinical registrations can provide more insights into patients' treatments and their outcomes. This is what we called real-world data, which goes beyond the controlled clinical trial, and allows us to test the results in an uncontrolled reality world (Rudrapatna and Butte, 2020). However, to produce credible conclusions, there are still many aspects that need to be improved. The most basic problem is the missing data. It is hard to guarantee the completeness of real-world data collection, and the incomplete data will pose challenges for further data analysis tasks. Therefore, it is important to handle these missing data in an appropriate way. ImputEHR (Zhou and Saghapour) evaluates the influence of various imputation approaches in real-world data sets such as EHR (Electronic Health Record) and provides a practical and fast imputation tool. This tool provides a web application, which makes the operation and visualization interactive. Interactivity is a new trend in the development of visualization tools. It can significantly improve the user experience of exploring data.

Among the emerging technologies in recent years, virtual reality (VR) is more and more widely used in medical fields, typical include in medical education and training. The advantage is that VR can dynamically explore complex biomedical data. But on the other hand, it is also limited by expensive hardware and complex data preprocessing steps. SinglecellVR is a web application that utilizes VR to visually explore single-cell data and common sequencing data, including transcriptome, epigenome, and proteome data (Stein et al.). It is designed for cheap and easily available virtual reality hardware, such as Google Cardboard. As a new solution of single-cell visualization, SinglecellVR has been reported within several media, which reflects the interest and concern of researchers on the use of low-cost VR in biomedicine. With the increase of data dimensions, VR will have a wide range of applications in exploring biomedical data.

Spatial transcriptomic is a popular molecular measurement technology used in the biomedical field recently. It can measure transcriptional information while retaining tissue spatial information. A review on the analysis and visualization of spatial transcriptomic data is presented in this research topic (Liu et al.). It covers the latest status of spatial transcriptome technology, and mainly focus on the current analysis and visualization tools in the preprocessing of data, the identification of spatially related gene patterns, and the visualization in expression domain, spatial domain, and cell-to-cell communication. As the latest research field of biomedicine, spatial transcriptomic is expected to reveal the complex transcriptional structure of heterogeneous tissues and enhance our understanding of the cellular mechanism of the disease (Burgess, 2019). Thus, more and more visualization methods and tools suitable for this field should be developed.

We have screened parts of the current biomedical visualization tools and their focus points cover different scenarios, from basic visual representation to specific field applications. They could be presented as part of the status of biomedical data visualization and their design concept also reflect the current trend of visualization. Moreover, biomedical data visualization methods play important roles in exploring and visualizing large-scale omics data, including genomics, transcriptomics, epigenetics, quantitative imaging, and so on. Regardless of the changes in data generation techniques, visualization tools, application scenarios, and research areas, the main tasks and development trends of data visualization remain unchanged—to better explore data, to better represent results, to improve the efficiency of information sharing and communication, and to improve user experience.

AUTHOR CONTRIBUTIONS

GY conceptualized the idea. TW and GY drafted the manuscript. C-LX and TT-YL helped in revising the manuscript.

REFERENCES

Burgess, D. J. (2019). Spatial Transcriptomics Coming of Age. *Nat. Rev. Genet.* 20, 317. doi:10.1038/s41576-019-0129-z

O′Donoghue, S. I., Baldi, B. F., Clark, S. J., Darling, A. E., Hogan, J. M., Kaur, S., et al. (2018). Visualization of Biomedical Data. *Annu. Rev. Biomed. Data Sci.* 1, 275–304. doi:10.1146/annurev-biodatasci-080917-013424

Rudrapatna, V. A., and Butte, A. J. (2020). Opportunities and Challenges in Using Real-World Data for Health Care. *J. Clin. Invest.* 130, 565–574. doi:10.1172/JCI129197

Wilkinson, L. (2010). The Grammar of Graphics. *Wires Comp. Stat.* 2, 673–677. doi:10.1002/wics.118

Conflict of Interest: Author TT-YL was employed by the company Laboratory of Data Discovery for Health Limited.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers in Genetics

METHODS
published: 02 July 2021
doi: 10.3389/fgene.2021.691274

ImputEHR: A Visualization Tool of Imputation for the Prediction of Biomedical Data

Yi-Hui Zhou[1,2]* *and Ehsan Saghapour*[1]

[1] Department of Biological Science, North Carolina State University, Raleigh, NC, United States, [2] Bioinformatics Research Center, North Carolina State University, Raleigh, NC, United States

Electronic health records (EHRs) have been widely adopted in recent years, but often include a high proportion of missing data, which can create difficulties in implementing machine learning and other tools of personalized medicine. Completed datasets are preferred for a number of analysis methods, and successful imputation of missing EHR data can improve interpretation and increase our power to predict health outcomes. However, use of the most popular imputation methods mainly require scripting skills, and are implemented using various packages and syntax. Thus, the implementation of a full suite of methods is generally out of reach to all except experienced data scientists. Moreover, imputation is often considered as a separate exercise from exploratory data analysis, but should be considered as art of the data exploration process. We have created a new graphical tool, ImputEHR, that is based on a Python base and allows implementation of a range of simple and sophisticated (e.g., gradient-boosted tree-based and neural network) data imputation approaches. In addition to imputation, the tool enables data exploration for informed decision-making, as well as implementing machine learning prediction tools for response data selected by the user. Although the approach works for any missing data problem, the tool is primarily motivated by problems encountered for EHR and other biomedical data. We illustrate the tool using multiple real datasets, providing performance measures of imputation and downstream predictive analysis.

Keywords: electronic health records, imputation, gradient boosting, prediction, decision trees

OPEN ACCESS

Edited by:
Guangchuang Yu,
Southern Medical University, China

Reviewed by:
Khanh N. Q. Le,
Taipei Medical University, Taiwan
Hao Zhu,
Southern Medical University, China

***Correspondence:**
Yi-Hui Zhou
yihui_zhou@ncsu.edu

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

Received: *06 April 2021*
Accepted: *25 May 2021*
Published: *02 July 2021*

Citation:
Zhou Y-H and Saghapour E (2021) ImputEHR: A Visualization Tool of Imputation for the Prediction of Biomedical Data. Front. Genet. 12:691274. doi: 10.3389/fgene.2021.691274

1. INTRODUCTION

Recently, hospitals in the United States have made a concerted effort to transition their health records from paper to digital, the proportion of which has dramatically increased, from 9.4% in 2008 to 75.5% in 2014 (Charles et al., 2013). Although we are seeing improvements in the overall quality of EHR-derived datasets, data missingness remains a substantial and unavoidable issue (Chan et al., 2010; Weiskopf and Weng, 2013). Missing EHR data could be caused by a lack of collection or a lack of documentation (Wells et al., 2013), and it could be missing at random or not at random (Hu et al., 2017). Researchers have noted the problems posed by missing data and are developing strategies to address it (Haukoos and Newgard, 2007; Newgard and Haukoos, 2007), as EHR systems become more relevant and adopted worldwide.

The expectation of collecting real-world data without missingness is unrealistic. Even the most detailed protocols for data collection cannot guarantee that every subject will have a record at each observation. Missing data present a challenge for analysts, as it can introduce a substantial amount of bias, makes the handling and analysis of the data more arduous, and creates reductions in efficiency (Barnard and Meng, 1999). Many standard analysis methods, including regression, are defeated by even a single missing value from among many potential predictors. Thus, it is possible that standard analysis may essentially "throw away" large portions of the data, even though a small fraction of the data may actually be missing. Ultimately, data missingness decreases our ability to discern the deeper structures and relationships underlying the observations, causing a significant negative impact on scientific research (McKnight et al., 2007). Many important scientific and business decisions are based on results from data analyses, and so dealing with missing data in an appropriate manner is recognized as a crucial step.

The process of data *imputation* (artificially replacing missing data with an estimated value) offers a practical work-around so that many downstream data handling steps become feasible. This process preserves all observations by replacing missing data with an estimated value based on other available information. Once all missing values have been imputed, datasets can then be analyzed using standard techniques for complete data (Gelman and Hill, 2006). Many advanced analysis methods, such as machine learning, require a complete dataset, so imputing missing data enables researchers to apply statistical and computational association methods that would otherwise be unavailable. Missing data imputation methods are considered standard in areas such as genetic association (Schurz et al., 2019) and proteomics (Jin et al., 2021), where correlation structures are strong. For electronic health records, the need for imputation methods have more recently realized (Jazayeri et al., 2020), and the use of imputation shown to improve prediction accuracy (Beaulieu-Jones et al., 2017). However, use of many of these methods requires purpose-built scripting pipelines (Hu et al., 2017), while we aim in this paper to provide a variety of tools using a very simple interface.

When imputation is performed, issues of bias and correct handling of variability/uncertainty arise (Rubin, 2003), depending on the imputation accuracy. Much of the traditional statistical literature on handling missing data has dealt with likelihood inference for low-dimensional problems (Rubin, 1976), or resampling techniques such as multiple imputation, which can mimic and account for imputation uncertainty. However, our focus here is on the practical impact of imputation for downstream analysis, such as EHR-based prediction of important health measures. For such efforts, the emphasis is placed on the success of machine-learning methods, which themselves may involve penalization techniques and estimation known to be biased. Thus, we consider imputation as a possibly essential pre-processing step to serve a larger goal, and it should be judged accordingly. Machine-learning methods have reached a high degree of sophistication in biology and genomics (Le and Huynh, 2019; Le et al., 2019), but for electronic health records, which tend to be less structured, a variety of approaches must be considered. In this work, we evaluate the effectiveness of various imputation methods on EHR and other real-world datasets, and proposed a practical and fast imputation method as a hybrid of existing methods.

2. DATASETS

2.1. MIMIC-III

The Medical Information Mart for Intensive Care III (MIMIC-III) is a large database comprising de-identified health-related data associated with over 40,000 patients who stayed in ICUs at the Beth Israel Deaconess Medical Center between 2001 and 2012 (Johnson et al., 2016). MIMIC-III is freely available on PhysioNet (https://mimic.physionet.org). The database includes information such as demographics, hourly vital sign measurements, laboratory test results, procedures, medications, caregiver notes, imaging reports, and mortality (including post-hospital discharge).

MIMIC-III is disseminated as a relational database consisting of 26 tables containing many categorical and continuous features. We extracted ICD-9 codes from the "DIAGNOSES_ICD" table, demographics and discharge time or time of death from the "ADMISSIONS" table, and laboratory measurements from the "LABEVENTS" table with <30% missing, totaling 603 features. ICD-9 is the actual code corresponding to the diagnosis assigned to the patient. However, it is often unclear whether a negative value indicates that the patient does not have a specific code, or the code is truly missing. The laboratory measurements are continuous values for 726 unique items. The missing proportion of laboratory tests can be as high as 90%, which significantly impacts any downstream analysis of these data. Therefore, it is important to study the appropriate missing data imputation methods for laboratory tests.

2.2. Datasets From the UCI Machine Learning Repository

The UCI Machine Learning Repository is a collection of datasets that are used by researchers for the empirical analysis of machine learning algorithms (Dua and Graff, 2017). Although these datasets are largely complete, we can effectively evaluate our imputation under complete missing at random assumptions by artificially masking individual observations and recording the imputation accuracy. Datasets are maintained on their website (https://archive.ics.uci.edu/ml/index.php). We selected the following four datasets for imputation testing: (1) "Boston," information for predicting the value of house prices (Harrison and Rubinfeld, 1978); (2) "Spam," attributes to determine whether e-mails were spam (Cranor and LaMacchia, 1998), (3) "Letter," character image features to identify a letter of the alphabet (Frey and Slate, 1991), and (4) "Breast Cancer," numerical features of cell images for tumor diagnosis in 357 malignant and 212 benign samples (Street et al., 1993). These datasets have varying numbers of samples and features, with both continuous and categorical data, as summarized in **Table 1**.

3. METHODS

ImputEHR is designed to provide several existing imputation methods in easy-to-use interface, as described below. In addition, we have noted that tree-based imputation has been relatively under-represented, and we propose some novel enhancements here in order to provide effective tree-based imputations with reasonable computational burden. Gradient boosted trees are an effective machine learning algorithm that iteratively combines decision trees in order to make predictions. In Python, we modified the MissForest algorithm (Stekhoven and Bühlmann, 2012), which imputes missing values using random forests (Liaw and Wiener, 2002), by applying the *LightGBM* module, a gradient boosting framework known for its light computational burden and better performance than previous decision tree-based algorithms (Ke et al., 2017), in the *missingpy* Python library for missing data imputation. Pseudocode for the ImputeEHR1 algorithm is shown in **Table 2**. The ImputeEHR2 approach is using the *XGBoost* (Extreme Gradient Boosting) module (Chen et al., 2015), a common boosting algorithm, in the *missingpy* library. The performance of ImputeEHR was validated using MIMIC-III and the four repository datasets.

3.1. Imputing Missing Data

We compared our proposed ImputeEHR1, ImputeEHR2, and five state-of-the-art imputation methods in Python: MissForest, MICE (Buuren and Groothuis-Oudshoorn, 2010), KNNImputer (Troyanskaya et al., 2001), SoftImpute (Mazumder et al., 2010), and GAIN (Yoon et al., 2018). In addition, we also performed simple feature-mean and feature-median replacement as the most basic and simple imputation method. KNNImputer is based on k-nearest neighbors algorithm. GAIN adapts the generative adversarial nets framework. The MICE and SoftImpute methods are implemented in the *fancyimpute* Python library. SoftImpute uses an iterative soft-thresholded SVD algorithm and MICE uses chained equations to impute missing values. We used default parameter settings for each method, and parameters for the two ImputeEHR methods are listed in **Supplementary Table 1**.

In each dataset, we generated missing data (missing completely at random), with rates from 10 to 90% in increments of 10% by randomly removing data and ran the imputation methods. The Root Mean Squared Error (RMSE) was then calculated at each missingness rate in comparison of the values between the real and imputed data. We ran 10 iterations in order to obtain average RMSEs.

Supplementary Tables 2–5 show the average RMSEs for each dataset, with the lowest RMSE at each missingness rate highlighted. Overall, our proposed method significantly outperforms all of the state-of-the-art models. ImputeEHR has the lowest RMSE in 24 out of a possible 36 comparisons, followed by MICE and MissForest methods having 6 and 3, respectively.

3.2. Testing Runtimes Between Methods

We evaluated the speeds of ImputeEHR1, ImputeEHR2, and MissForest method, since they are each tree-based learning algorithms, using the *scikit-learn* Python library (Pedregosa et al., 2011). We set the number of trees at 100, and used default values for the remaining parameter settings. **Figure 1** shows the runtimes by missingness rate in each dataset. Our experiments show that both ImputeEHR1 and ImputeEHR2 can accelerate the imputation process 20–25 times faster than MissForest while achieving lower RMSEs. Moreover, ImputeEHR1 is faster than ImputeEHR2 for the largest dataset. We performed this experiment on a desktop computer with Windows 10, Intel(R) Xenon CPU E5-2687W v4@3.00 GHz CPU, 128 GB RAM and GeForce GTX 1080, 8 GB.

TABLE 2 | Pseudocode of the ImputeEHR algorithm.

Algorithm: ImputeEHR algorithm

Require: X is $n \times m$ -dimensional data matrix, with stopping criterion γ

1. Make initial guess using mean or median imputation for missing values;
2. $k \leftarrow A$ sorted indices vector according to t he amount of missing values of column X;

w.r.t. increasing amount of missing values;

3. **While** not γ **do**
4. $X_{old}^{imp} \leftarrow$ Store previously imputed matrix;
5. **for** s in k **do**
6. Fit a LightGBM or Xgboost : $y_{obs}^{(s)} \sim X_{obs}^{(s)}$;
7. Predict $y_{miss}^{(s)}$ using $X_{miss}^{(s)}$;
8. $X_{new}^{imp} \leftarrow$ update imputed matrix from $y_{miss}^{(s)}$;
9. **end for**
10. Update γ
11. **end** while
12. **Return** Matrix X;

TABLE 1 | The Boston data have information for predicting the value of house prices; the spam data contain the attributes to determine whether e-mails spam; the letter data have character image features to identify a letter of the alphabet; the breast cancer data gathered the numerical features of cell images for tumor diagnosis.

Dataset	Download link	# Sample	# Features	Attribute type
Boston	https://archive.ics.uci.edu/ml/machine-learning-databases/housing	506	13	Both
Spam	https://archive.ics.uci.edu/ml/datasets/Spambase	4,601	57	Continuous
Letter	https://archive.ics.uci.edu/ml/datasets/Letter+Recognition	20,000	16	Categorical
Breast cancer	https://archieve.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29	569	30	Continuous

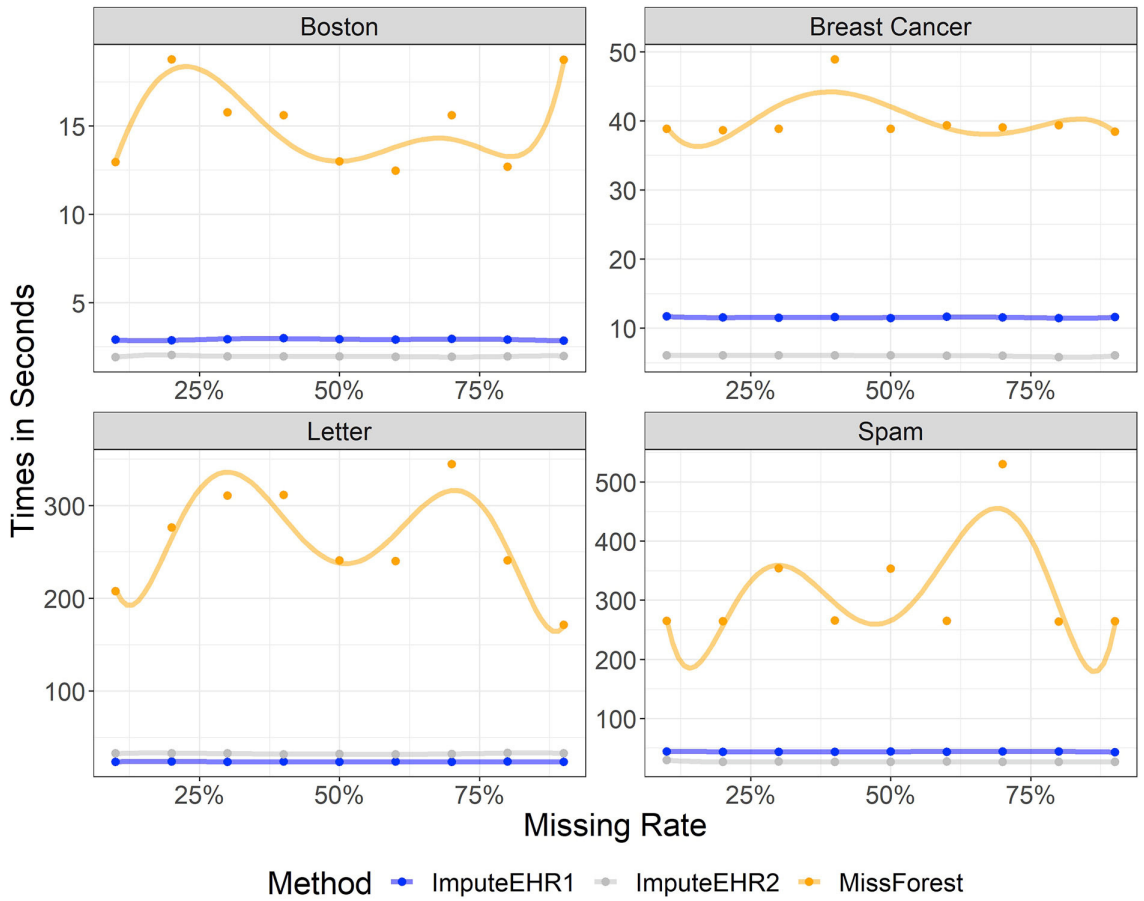

FIGURE 1 | Running time of ImputeEHR1 (blue), MissForest (orange), and ImputeEHR2 (gray) for each dataset.

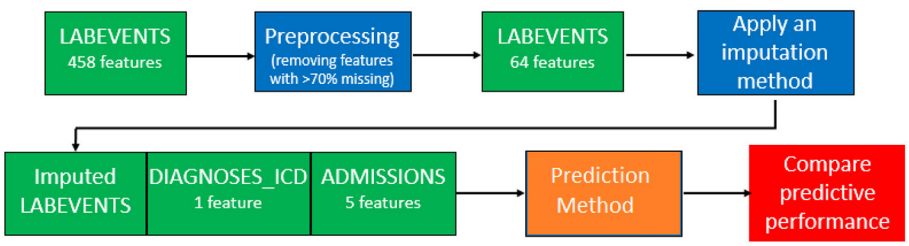

FIGURE 2 | Our pipeline of the MIMIC-III data imputation and prediction.

3.3. Evaluating Predictive Performance for a Variable of Interest, After Imputation

We attempted to predict the mortality for ICU patients in the MIMIC-III database. **Figure 2** provides an illustration of our pipeline. First, we aggregated the laboratory tests in the "LABEVENTS" table by averaging the values taken within the first 24 h of a patient's first admission to ICU. After removing laboratory tests which are >70% missing, 64 items remained. Then, we selected patients with complete records for the 64 laboratory tests, resulting in 714 patients. So our filtered "LABEVENTS" data have dimension 714 patients × 64 laboratory tests, which we used as input for each imputation method.

Then, we combined the imputed "LABEVENTS" data with the ICD-9 codes from the "DIAGNOSIS_ICD" table and the demographics and mortality outcome from the "ADMISSIONS" table into a model matrix and applied lasso regression (Tibshirani, 1996) with five-fold cross-validation. This process involves randomly splitting the samples into five groups, keeping four groups as a training set, so the model can predict the outcomes for samples in the fifth group. This process was run five times so outcomes are predicted in all samples. The area under

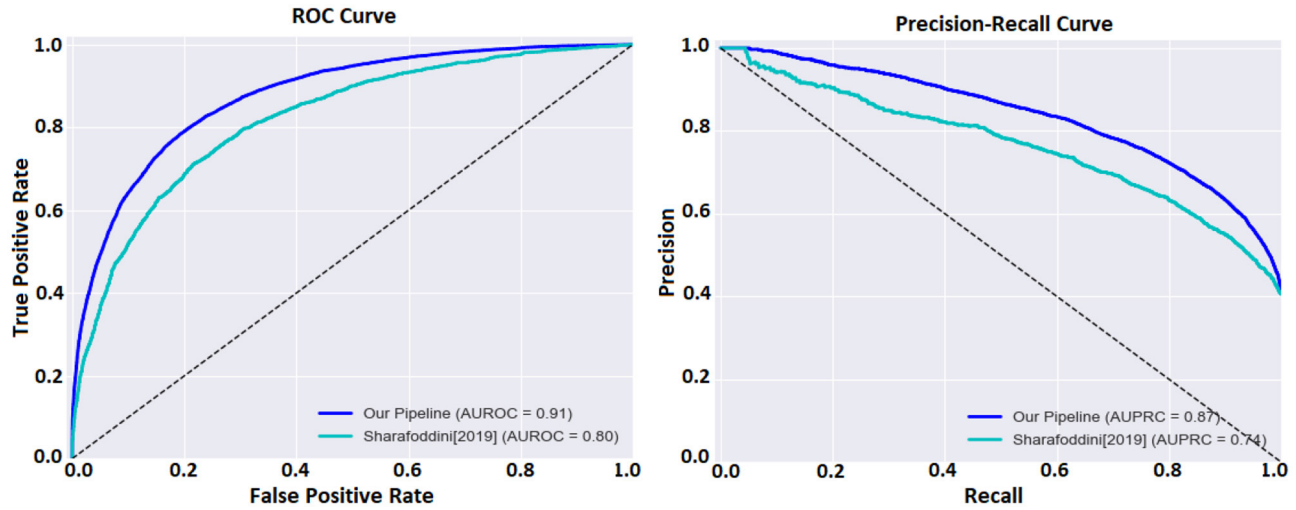

FIGURE 3 | (Left) Receiver operating characteristic curve (ROC) comparison between our pipeline and the method (Sharafoddini et al., 2019) on the mortality prediction of the MIMIC-III data. **(Right)** Precision recall curve comparison.

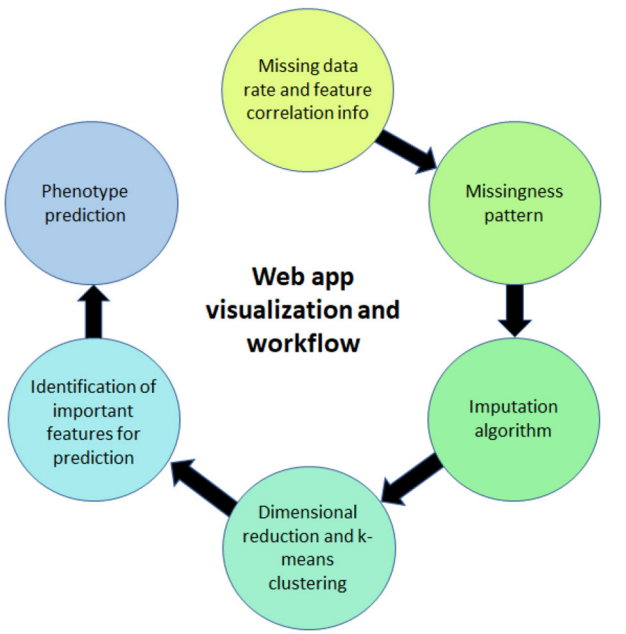

FIGURE 4 | Illustration of the web app for visualization.

the curve (AUC) is the metric we used to compare the predicted vs. the actual outcomes. The ImputeEHR method has the highest AUC 0.91, and the tree-based algorithms perform better than other methods. Our pipeline provides the highest prediction accuracy comparing the historical mortality prediction in the literature (Sharafoddini et al., 2019), which reached the best AUC 0.80 (**Figure 3**). Both receiver operating characteristic curve and precision recall curve show that our pipeline provides the best prediction of mortality.

4. WEB APPLICATION

The web application (ImputEHR app), available as a scikit-learn package in Python, allows users to apply our pre-processing, feature engineering, and prediction methods on their dataset, and to visualize the results. Below we briefly describe the six major components of the web app, illustrated in **Figure 4**, and show its capabilities by presenting results of our implementation, using the "Breast Cancer" dataset from the UC Irvine Machine Learning Repository as an example.

4.1. Percentage of Missing Rate and Correlation Features Information

Users can obtain initial information about the missing rates of each feature in their dataset. **Supplementary Figure 1** shows the percentage of missing values in our example. Since the breast cancer dataset in **Table 1** (Street et al., 1993) does not have missing values, we randomly set 35–45% of the values as missing and continue to use it as the toy example for our ImputEHR app.

In addition, the app has the option for users to plot the correlation between any two features (factors). It also helps the users to decide if they need to include these factors that might be highly correlated with each. If the dataset has missing values, users can show the scatterplot before imputing, removing the missing values. Three parameters to better visualize the scatterplot are the color, size, and clarity of the data points (**Supplementary Figure 2**).

4.2. Visualization of Missingness Patterns

As an optional feature in our app, the missingness patterns can be checked by users via the black/white image plot, in which black is for missing values. The user can also hover mouse around the Dendrogram and zoom in to check the information for the grouped factors due to the missingness. **Supplementary Figure 3**

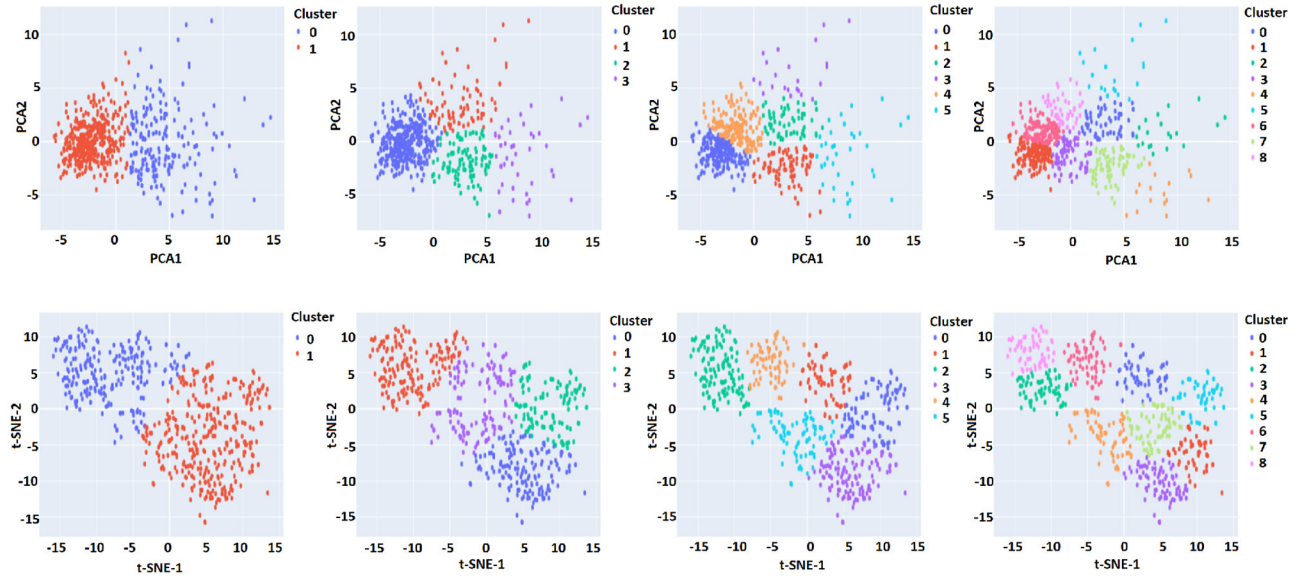

FIGURE 5 | Visualization of patterns in the imputed dataset. User has the option to use the number of cluster and dimension reduction method.

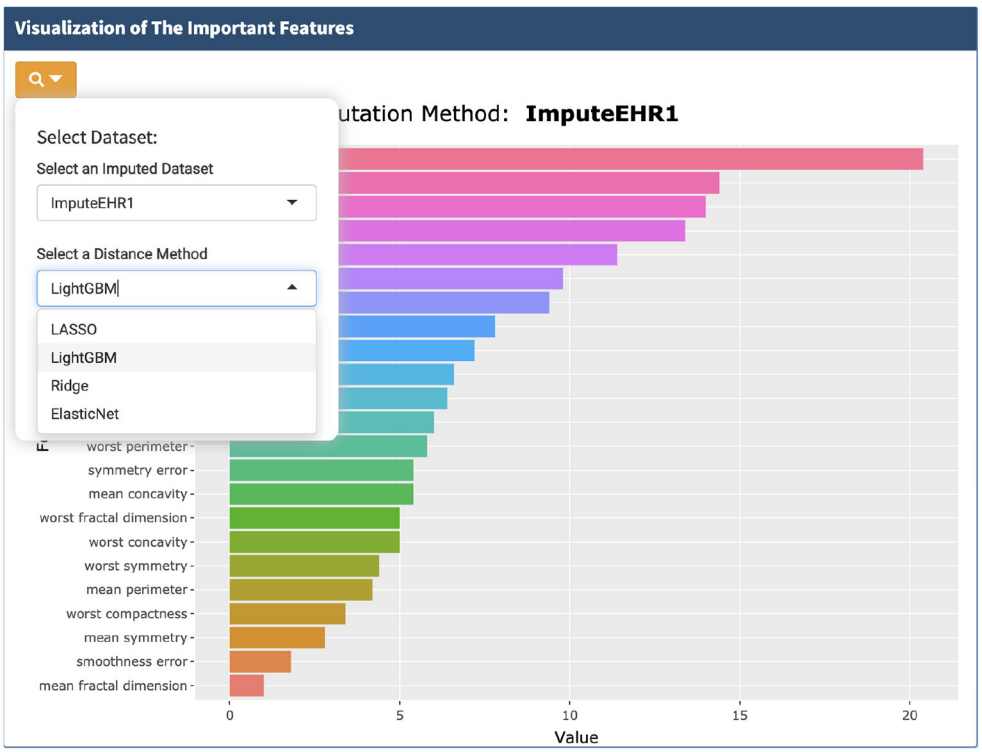

FIGURE 6 | Visualization of the important features selected by the four methods.

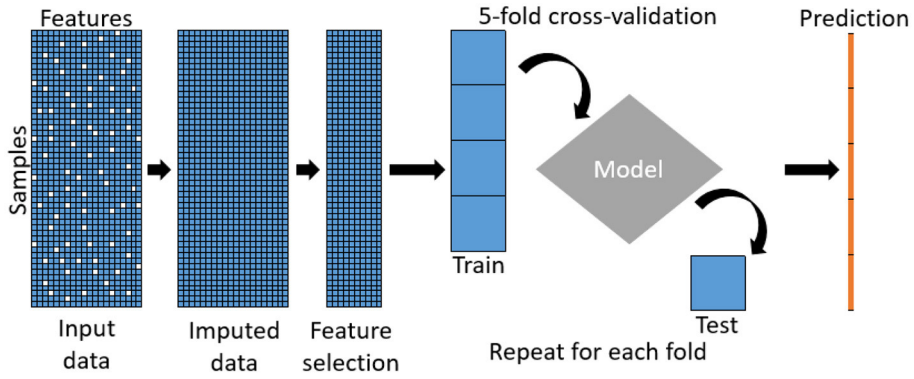

FIGURE 7 | Pipeline of the predictive model.

includes the visualization of Dendrogram on missingness pattern based on the toy data.

4.3. Imputation Algorithm

Within the app, the nine imputation methods listed in section 3.1 are available: ImputeEHR1, ImputeEHR2, MissForest, MICE, KNNImputer, SoftImpute, GAIN, mean, and median. **Supplementary Table 6** provides the important parameters' selection for the toy example via ImputeEHR1 and ImputeEHR2 methods.

Some methods have their own hyperparameters. For KNNImputer, we set $k = 5$, which is considered the default number of nearest neighbors. Four parameters, "batch_size," "hint_rate," "alpha," and "iteration," are embedded for the GAIN method. The "batch_size" defines the number of training samples present in a single batch. The "hint_rate" reveals the discriminator partial information about the missingness of the original sample. The "alpha" is a hyperparameter, and "iteration" describes the number of times a batch of data passes through the algorithm to update its parameters.

4.4. Visualization From Combining Dimensional Reduction Algorithms and K-Means Clustering

ImputEHR makes it easy for users to visualize patterns in their imputed dataset. Principal component analysis (PCA) Pearson (1901) and t-distributed stochastic neighbor embedding (t-SNE) (Van der Maaten and Hinton, 2008) methods are embedded for dimension reduction. Users can plot the result of either method, partitioning the observations into k clusters. Our ImputEHR app suggests the number of optimal clusters using the Elbow method (Syakur et al., 2018), which runs k-means clustering on the imputed dataset for a range of values for k between 1 and 9. For the visualization purpose, the green line in **Supplementary Figure 4** indicates the best choice of k plot on the toy example. Three parameters considered for the t-SNE method are "learning rate," "n_iter" (number of iterations), and "perplexity." Perplexity defines the number of close neighbors at each point, and learning rate affects the convergence of the embedding. In **Figure 5** and **Supplementary Figure 5**, we applied k-means method with different numbers of clusters on the outcome of the PCA and t-SNE methods. In our app, user can also mouse over the point and see which variable it is.

4.5. Visualization of the Important Features

A very useful feature of our app is that it helps users to nail down the most important features for further investigation. We provide the users four methods for feature selection from the imputed dataset: LightGBM (Ke et al., 2017), lasso (Tibshirani, 1996), ridge (Hoerl and Kennard, 1970), and elastic net (Zou and Hastie, 2005) (**Figure 6**). Users can decide how many important features to visualize.

4.6. Visualization of the Phenotype Prediction

When performing imputation, if downstream prediction is intended, then the response variable should be removed from the imputation process to avoid overtraining datasets in which cross-validation for prediction of the response must be used. Accordingly, ImputEHR enables the user to select a response variable to be excluded from the imputation process. We also provide the author the visualization of the correlation between the imputed value and the masked 5% non-missing data for each variable (**Supplementary Figure 4**).

Important features from an imputed dataset are selected as input to predict the phenotype, illustrated in **Figure 7**, using five-fold cross-validation to avoid overfitting. Users can select from a suite of prediction methods including random forests, lasso, LightGBM, and KNN.

The running time for a job depends largely on the size of dataset, the missing rate, and the computer hardware. All analyses were performed in Python 3.6.

5. CONCLUSIONS

ImputeEHR can quickly and accurately impute missing data, implementing a variety of methods. The ease of performing imputation can lead to better predictive performance, as many methods are made feasible by imputation. We have created a tool covering a range of imputation options, including novel and fast tree-based methods. We have also included a variety

of basic phenotype prediction methods, although the user can easily output the imputed dataset for import into other prediction routines. As with any imputation tools, the accuracy will be limited by the correlation structures, and in general the number of features relative to the sample size. For these and other reasons, this tool is not designed for genomic imputation (Schurz et al., 2019) or for proteomics data (Jin et al., 2021), or other areas with well-understood biological correlation structures. However, the ease of use and seamless interface for using multiple imputation methods makes our approach a useful approach in a variety of analysis pipelines.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**. The toydata for the ImputEHR app is located at https://github.com/zhouLabNCSU/ImputEHR/tree/main/Demo%20File, further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

Y-HZ is the leader of this project. Her contribution includes writing the manuscript, designing the data analysis, summarizing the results, and software management. ES contributed to the Python code underneath the ImputEHR app. Both authors contributed to the article and approved the submitted version.

FUNDING

This work was supported by Y-HZ's startup funding at NCSU and Cystic Fibrosis Foundation KNOWLE18XX0.

ACKNOWLEDGMENTS

Thanks to Mr. Gallins' effort in reformatting the manuscript into the Latex format. Thanks for Kuncheng Song's contribution to the new **Figure 1**, ImputEHR Rshiny app and software maintenance, Yang Sun's contribution to **Figure 3**, Paul Gallins' contribution to **Figure 7** and draft reformatting.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2021.691274/full#supplementary-material

The Rshiny link, supplementary documents, and breast cancer toy example dataset are available at: https://github.com/zhouLabNCSU/ImputEHR.

REFERENCES

Barnard, J., and Meng, X. L. (1999). Applications of multiple imputation in medical studies: from aids to nhanes. *Stat. Methods Med. Res.* 8, 17–36. doi: 10.1177/096228029900800103

Beaulieu-Jones, B. K., and Moore, J. H., (2017). CONSORTIUM PROAACT. Missing data imputation in the electronic health record using deeply learned autoencoders. *Pac. Symp. Biocomput.* 2017, 207–218. doi: 10.1142/9789813207813_0021

Buuren, S., and Groothuis-Oudshoorn, K. (2010). mice: Multivariate imputation by chained equations in R. *J. Stat. Softw.* 45, 1–68. doi: 10.18637/jss.v045.i03

Chan, K. S., Fowles, J. B., and Weiner, J. P. (2010). Electronic health records and the reliability and validity of quality measures: a review of the literature. *Med. Care Res. Rev.* 67, 503–527. doi: 10.1177/1077558709359007

Charles, D., Gabriel, M., and Furukawa, M. F. (2013). Adoption of electronic health record systems among us non-federal acute care hospitals: 2008-2012. *ONC Data Brief* 9, 1–9. Available online at: https://www.healthit.gov/sites/default/files/oncdatabrief16.pdf

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., et al. (2015). *Xgboost: Extreme Gradient Boosting. R Package Version 0.4-2 1*.

Cranor, L. F., and LaMacchia, B. A. (1998). Spam! *Commun. ACM* 41, 74–83. doi: 10.1145/280324.280336

Dua, D., and Graff, C. (2017). *UCI machine learning repository*. Irvine: University of California. Available online at: https://archive.ics.uci.edu/ml/citation_policy.html

Frey, P. W., and Slate, D. J. (1991). Letter recognition using holland-style adaptive classifiers. *Mach. Learn.* 6, 161–182. doi: 10.1007/BF00114162

Gelman, A., and Hill, J. (2006). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press. doi: 10.1017/CBO9780511790942

Harrison, D. Jr., and Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean air. *J. Environ. Econ. Manage.* 5, 81–102. doi: 10.1016/0095-0696(78)90006-2

Haukoos, J. S., and Newgard, C. D. (2007). Advanced statistics: missing data in clinical research?part 1: an introduction and conceptual framework. *Acad. Emerg. Med.* 14, 662–668. doi: 10.1197/j.aem.2006.11.037

Hoerl, A. E., and Kennard, R. W. (1970). Ridge regression: biased estimation for nonorthogonal problems. *Technometrics* 12, 55–67. doi: 10.1080/00401706.1970.10488634

Hu, Z., Melton, G. B., Arsoniadis, E. G., Wang, Y., Kwaan, M. R., and Simon, G. J. (2017). Strategies for handling missing clinical data for automated surgical site infection detection from the electronic health record. *J. Biomed. Informatics* 68, 112–120. doi: 10.1016/j.jbi.2017.03.009

Jazayeri, A., Liang, O. S., and Yang, C. C. (2020). Imputation of missing data in electronic health records based on patients? similarities. *J. Healthc. Informatics Res.* 4, 295–307. doi: 10.1007/s41666-020-00073-5

Jin, L., Bi, Y., Hu, C., Qu, J., Shen, S., Wang, X., et al. (2021). A comparative study of evaluating missing value imputation methods in label-free proteomics. *Sci. Rep.* 11, 1–11. doi: 10.1038/s41598-021-81279-4

Johnson, A. E., Pollard, T. J., Shen, L., Li-Wei, H. L., Feng, M., Ghassemi, M., et al. (2016). MIMIC-III, a freely accessible critical care database. *Sci. Data* 3, 1–9. doi: 10.1038/sdata.2016.35

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., et al. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Adv. Neural Inform. Process. Syst.* 3146–3154. Available online at: https://proceedings.neurips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html

Le, N. Q. K., and Huynh, T. T. (2019). Identifying snares by incorporating deep learning architecture and amino acid embedding representation. *Front. Physiol.* 10:1501. doi: 10.3389/fphys.2019.01501

Le, N. Q. K., Yapp, E. K. Y., Nagasundaram, N., and Yeh, H. Y. (2019). Classifying promoters by interpreting the hidden information of DNA sequences via deep learning and combination of continuous fasttext n-grams. *Front. Bioeng. Biotechnol.* 7:305. doi: 10.3389/fbioe.2019.00305

Liaw, A., and Wiener, M. (2002). Classification and regression by randomforest. *R News* 2, 18–22. Available online at: https://cogns.northwestern.edu/cbmg/LiawAndWiener2002.pdf

Mazumder, R., Hastie, T., and Tibshirani, R. (2010). Spectral regularization algorithms for learning large incomplete matrices. *J. Mach. Learn. Res*. 11, 2287–2322. Available online at: https://jmlr.csail.mit.edu/papers/volume11/mazumder10a/mazumder10a.pdf

McKnight, P. E., McKnight, K. M., Sidani, S., and Figueredo, A. J. (2007). *Missing Data: A Gentle Introduction*. Guilford Press.

Newgard, C. D., and Haukoos, J. S. (2007). Advanced statistics: missing data in clinical research?part 2: multiple imputation. *Acad. Emerg. Med*. 14, 669–678. doi: 10.1111/j.1553-2712.2007.tb01856.x

Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *London Edinburgh Dublin Philos. Mag. J. Sci*. 2, 559–572. doi: 10.1080/14786440109462720

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in python. *J. Mach. Learn. Res*. 12, 2825–2830. Available online at: https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf

Rubin, D. B. (1976). Inference and missing data. *Biometrika* 63, 581–592. doi: 10.1093/biomet/63.3.581

Rubin, D. B. (2003). Discussion on multiple imputation. *Int. Stat. Rev*. 71, 619–625. doi: 10.1111/j.1751-5823.2003.tb00216.x

Schurz, H., Müller SJ, Van Helden, P. D., Tromp, G., Hoal, E. G., Kinnear, C. J., et al. (2019). Evaluating the accuracy of imputation methods in a five-way admixed population. *Front. Genet*. 10:34. doi: 10.3389/fgene.2019.00034

Sharafoddini, A., Dubin, J. A., Maslove, D. M., and Lee, J. (2019). A new insight into missing data in intensive care unit patient profiles: observational study. *JMIR Med. Informatics* 7:e11605. doi: 10.2196/11605

Stekhoven, D. J., and Bühlmann, P. (2012). Missforest?non-parametric missing value imputation for mixed-type data. *Bioinformatics* 28, 112–118. doi: 10.1093/bioinformatics/btr597

Street, W. N., Wolberg, W. H., and Mangasarian, O. L. (1993). Nuclear feature extraction for breast tumor diagnosis. *Biomed. Image Process. Biomed. Visual*. 1905, 861–870. doi: 10.1117/12.148698

Syakur, M., Khotimah, B., Rochman, E., and Satoto, B. (2018). Integration k-means clustering method and elbow method for identification of the best customer profile cluster. *IOP Conf. Ser*. 336:012017. doi: 10.1088/1757-899X/336/1/012017

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B* 58, 267–288. doi: 10.1111/j.2517-6161.1996.tb02080.x

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., et al. (2001). Missing value estimation methods for dna microarrays. *Bioinformatics* 17, 520–525. doi: 10.1093/bioinformatics/17.6.520

Van der Maaten, L., and Hinton, G. (2008). Visualizing data using t-SNE. *J. Mach. Learn. Res*. 9, 2579–2605. Available online at: https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf

Weiskopf, N. G., and Weng, C. (2013). Methods and dimensions of electronic health record data quality assessment: enabling reuse for clinical research. *J. Am. Med. Informatics Assoc*. 20, 144–151. doi: 10.1136/amiajnl-2011-000681

Wells, B. J., Chagin, K. M., Nowacki, A. S., and Kattan, M. W. (2013). Strategies for handling missing data in electronic health record derived data. *Egems* 1:1035. doi: 10.13063/2327-9214.1035

Yoon, J., Jordon, J., and Van Der Schaar, M. (2018). Gain: Missing data imputation using generative adversarial nets. *arXiv preprint arXiv:1806.02920*. Available online at: https://arxiv.org/abs/1806.02920

Zou, H., and Hastie, T. (2005). Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B* 67, 301–320. doi: 10.1111/j.1467-9868.2005.00503.x

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Genetics

BRIEF RESEARCH REPORT
published: 16 July 2021
doi: 10.3389/fgene.2021.706854

Crosslink: An R Package for Network Visualization of Grouped Nodes

Di Liu[1], Zhijie Bai[2], Bing Liu[2,3,4] and Zongcheng Li[3]**

[1] Peking-Tsinghua Center for Life Sciences, Peking University, Beijing, China, [2] State Key Laboratory of Proteomics, Academy of Military Medical Sciences, Academy of Military Sciences, Beijing, China, [3] State Key Laboratory of Experimental Hematology, Institute of Hematology, Fifth Medical Center of Chinese PLA General Hospital, Beijing, China, [4] Key Laboratory for Regenerative Medicine of Ministry of Education, Institute of Hematology, School of Medicine, Jinan University, Guangzhou, China

The demand for network visualization of relationships between nodes attributed to different categories grows in various biomedical research scenarios, such as gene regulatory networks, drug-target networks, ligand-receptor interactions and association networks of multi-omics elements. Elegantly visualizing the relationships between nodes with complex metadata of nodes and edges appended may inspire new insights. Here, we developed the crosslink R package, tailored for network visualization of grouped nodes, to provide a series of flexible functions for generating network diagrams. We first designed a CrossLink class for storage of metadata about nodes and edges and manipulation of node coordinates. Then affine transformation and function mapping transformation are implemented to perform fundamental node coordinates transformation by groups, based on which various network layouts can be defined easily. For convenience, we predefined several commonly used layouts, including row, column, arc, polygon and hive, which also can be combined in one layout. Finally, we designed a user-friendly wrapper function to draw network connections, aesthetic mappings of metadata and decoration with related annotation graphs in one interface by taking advantage of the powerful ggplot2 system. Overall, the crosslink R package is easy-to-use for achieving complex visualization of a network diagram of grouped nodes surrounded by associated annotation graphs.

Availability and Implementation: Cosslink is an open-source R package, freely available from github: https://github.com/zzwch/crosslink; A detailed user documentation can be found in https://zzwch.github.io/crosslink/.

Keywords: R package, network, visualization, grouped data, crosslink

OPEN ACCESS

Edited by:
Guangchuang Yu,
Southern Medical University, China

Reviewed by:
Matthew N. Bernstein,
Morgridge Institute for Research, United States
Kira Vyatkina,
Saint Petersburg Academic University (RAS), Russia

****Correspondence:***
Bing Liu
bingliu17@yahoo.com
Zongcheng Li
lizc07@vip.qq.com

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

Received: *08 May 2021*
Accepted: *25 June 2021*
Published: *16 July 2021*

Citation:
Liu D, Bai Z, Liu B and Li Z (2021) Crosslink: An R Package for Network Visualization of Grouped Nodes. Front. Genet. 12:706854. doi: 10.3389/fgene.2021.706854

INTRODUCTION

With the rapid development of multi-omic technologies, intricate relationships between different categories of *biomedical* molecules were established, which brought huge opportunities and challenges to network visualization. Visualization of relationships between various biomolecules from different layers is helpful to explain and extract comprehensive biological information. For instance, Youqiong Ye etc. presented the network among the identified molecular alterations and the sensitivity of anticancer drugs to directly display a multi-omic molecular feature landscape of

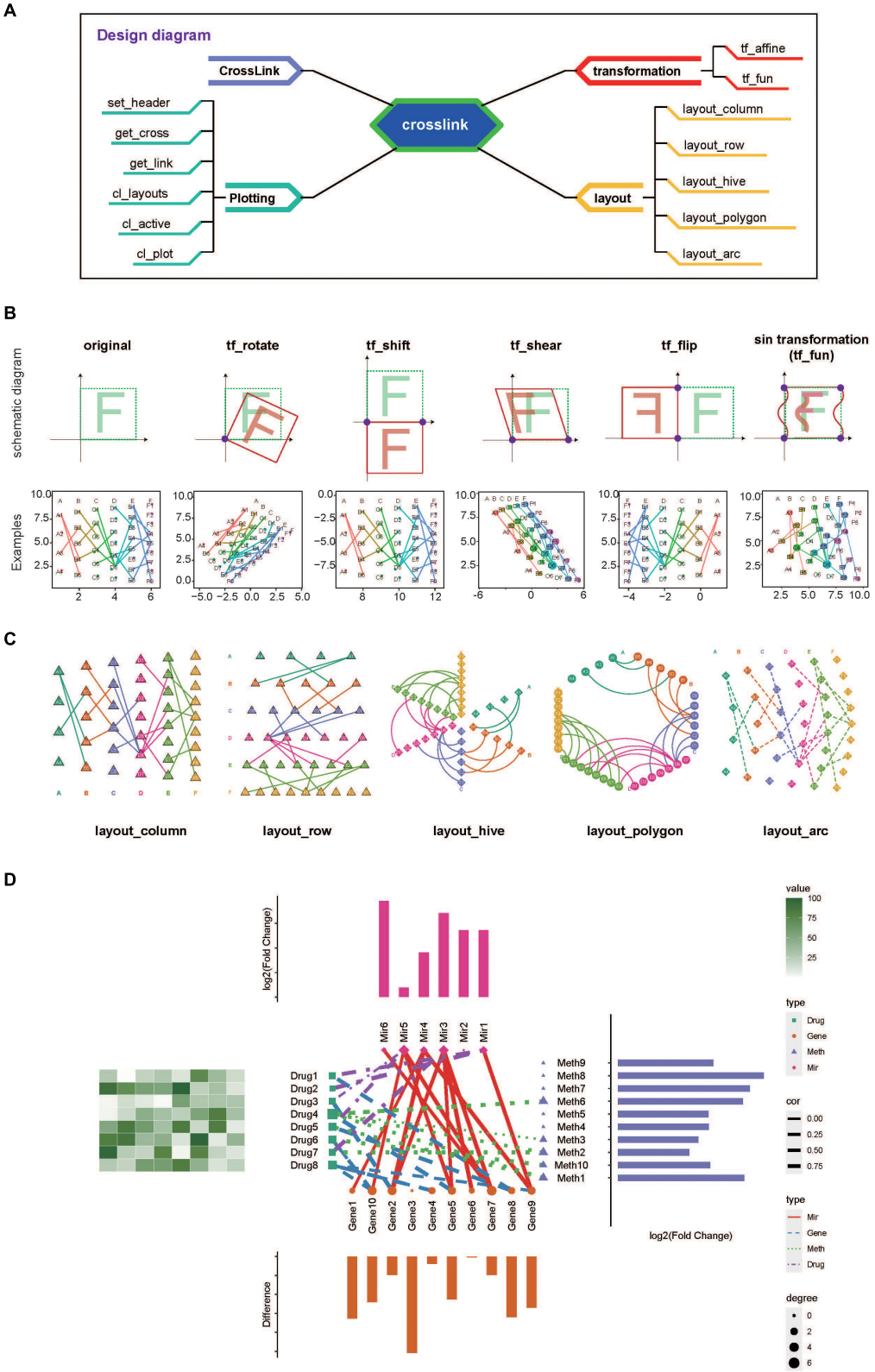

FIGURE 1 | Overviw and usage examples of crosslink. **(A)**. A schematic diagram of crosslink showing four modules and associated functions. **(B)** Schematic diagram and examples showing transformation effects after using the coordinate transformation functions as indicated. **(C)** Examples of five predefined layout styles. **(D)** A typical application of combination network visualization by using crosslink.

tumor hypoxia (Ye et al., 2019). And recently, there is a study characterizing the network among the expression of altered m6A regulators and cancer related pathways to illustrate the role of m6A in carcinogenesis (Li et al., 2019). Besides, researches in brain disease and plant development often provide an intuitive correlation network diagram to explain the influence of key regulators on other related layers (Shahan et al., 2018; Gilson et al., 2020). These cases show the common elements required for network visualization in many *biomedical* researches: (1) connections between multiple groups of biomolecules (i.e., grouped nodes), (2) mapping of additional biological information onto biomolecules and connections (i.e., nodes and edges), (3) arrangement of biomolecules in columns according to their categories, and (4) combination of annotation graphs around the network diagram.

A number of tools have been developed for visualization of various complex network, such as Cytoscape (Shannon et al., 2003), igraph (Csardi and Nepusz, 2006), ggraph (Pedersen, 2020) and Gelphi (Bastian et al., 2009). Recently, CellChat (Jin et al., 2021) was released to specifically analyze and visualize cell-cell communication network. Importantly, none of the tools above offer the function to combine the network diagram with the corresponding annotation graphs for grouped nodes. For the present, a tool specially designed for network visualization of grouped nodes that supports nodes decoration with annotation plots is still lacking.

Therefore, the user-friendly R package crosslink is developed here to arrange nodes by group, map metadata onto aesthetics of nodes and edges and align annotation graphs with the network. This package would hopefully meet various specific demands on network visualization of grouped biomolecules in *biomedical* research.

MATERIALS AND METHODS

The crosslink is developed in R language and mainly includes four modules, which is CrossLink class, coordinate transformation methods, layout modules and the plotting function, as shown in **Figure 1A**. The CrossLink class is the basic module, storing the metadata of nodes and edges, node coordinates and other parameters. The other three modules are operated on the data structure of CrossLink class. Here, we termed the group of nodes as "cross" and the edge between groups as "link".

First, the function "crosslink" is used to generate a CrossLink object. With this function users can easily initialize a default network by inputting nodes and edges information. Several adjustments including spaces between nodes and gaps between crosses (groups) are also available for fine-tuning the default layout.

Second, coordinate transformation module, consisting of several affine transformation methods and the method to define the function for mapping transformation, is then applied for node coordinate transforming by crosses. The "tf_affine" function is designed for coordinate transforming of grouped nodes in the network. It requires a CrossLink object as the input and returns the object with transformed coordinates. This function provides several designed modes including rotating, shifting, shearing, flipping and scaling (**Figure 1B**), which would be useful when adjusting node coordinates in one or all groups to beautify presentation of complex relationships among multiple types of data, as shown in **Figure 1D**. The "tf_fun" interface allows users to customize transforming function according to specific needs. Here, as an example, we designed a "sin" transformation method using "tf_fun" interface to illustrate its usage (**Figure 1B**).

Third, *commonly used styles are predefined in the* layout module*, including row, column, arc, polygon and hive as shown in* **Figure 1C**. Users can specify a predefined network layout or combine multiple predefined layouts to design a diverse network.

Fourth, the plotting function "cl_plot" allows various aesthetic settings for nodes, edges, node labels and headers by taking advantage of "ggplot2" system (Ito and Murphy, 2013). In particular, this function provides the annotation interface to achieve the combination of the network diagram and corresponding annotation graphs, with node coordinates aligned (**Figure 1D**). Additionally, the plotting module also includes several data extraction functions, such as "get_cross" and "get_link", which can be used to obtain the coordinate and metadata information. The "set_header" function is provided to place cross (group) headers.

In summary, crosslink provides a friendly interface for users to realize diverse network plotting of grouped nodes. This package can be applied to various biomedical studies for visualizing complex information and relationships between biomolecules in different categories (Goh et al., 2007; Neph et al., 2012; Chen and Wu, 2013; Shahan et al., 2018).

DISCUSSION

This work presented the first network visualization R package tailored for grouped nodes that implements a series of functions to store network data, manipulate node coordinates, and plot network diagram with supports for aesthetic mappings for nodes and edges and aligned graph annotation.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

ZL and BL conceived and designed the study. ZL completed the R package "crosslink" and wrote the manuscript. DL performed the figure test and wrote the user guide and the manuscript. ZB proofread and corrected the manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

This work was supported by the National Natural Science Foundation of China (81900115 and 31930054).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2021.706854/full#supplementary-material

REFERENCES

Bastian, M., Sebastien, H., and Mathieu, J. (2009). "Gephi: an open source software for exploring and manipulating networks," in *Proceeding of the International AAAI Conference on Web and Social Media*.

Chen, B. S., and Wu, C. C. (2013). Systems biology as an integrated platform for bioinformatics, systems synthetic biology, and systems metabolic engineering. *Cells* 2, 635–688. doi: 10.3390/cells2040635

Csardi, G., and Nepusz, T. (2006). The igraph software package for complex network research. *InterJ. Comp. Syst.* 1695. Available online at: https://igraph.org

Gilson, M., Zamora-López, G., Pallarés, V., Adhikari, M. H., Senden, M., Campo, A. T., et al. (2020). Model-based whole-brain effective connectivity to study distributed cognition in health and disease. *Netw Neurosci.* 4, 338–373. doi: 10.1162/netn_a_00117

Goh, K. I., Cusick, M. E., Valle, D., Childs, B., Vidal, M., and Barabási, A. L. (2007). The human disease network. *Proc. Natl. Acad. Sci. U.S.A.* 104, 8685–8690. doi: 10.1073/pnas.0701361104

Ito, K., and Murphy, D. (2013). Application of ggplot2 to pharmacometric graphics. *CPT Pharmacometrics Syst. Pharmacol.* 2:e79. doi: 10.1038/psp.2013.56

Jin, S., Guerrero-Juarez, C. F., Zhang, L., Chang, I., Ramos, R., Kuan, C. H., et al. (2021). Inference and analysis of cell-cell communication using CellChat. *Nat. Commun.* 12:1088. doi: 10.1038/s41467-021-21246-9

Li, Y., Xiao, J., Bai, J., Tian, Y., Qu, Y., Chen, X., et al. (2019). Molecular characterization and clinical relevance of m(6)a regulators across 33 cancer types. *Mol. Cancer* 18:137. doi: 10.1186/s12943-019-1066-3

Neph, S., Stergachis, A. B., Reynolds, A., Sandstrom, R., Borenstein, E., and Stamatoyannopoulos, J. A. (2012). Circuitry and dynamics of human transcription factor regulatory networks. *Cell* 150, 1274–1286. doi: 10.1016/j.cell.2012.04.040

Pedersen, T. L. (2020). *ggraph: An Implementation of Grammar of Graphics for Graphs and Networks*. R package version 2.0.4. Available online at: https://CRAN.R-project.org/package=ggraph

Shahan, R., Zawora, C., Wight, H., Sittmann, J., Wang, W., Mount, S. M., et al. (2018). Consensus coexpression network analysis identifies key regulators of flower and fruit development in wild strawberry. *Plant Physiol.* 178, 202–216. doi: 10.1104/pp.18.00086

Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., et al. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.* 13, 2498–2504. doi: 10.1101/gr.1239303

Ye, Y., Hu, Q., Chen, H., Liang, K., Yuan, Y., Xiang, Y., et al. (2019). Characterization of hypoxia-associated molecular features to aid hypoxia-targeted therapy. *Nat. Metab* 1, 431–444. doi: 10.1038/s42255-019-0045-8

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Genetics

TECHNOLOGY AND CODE
published: 07 September 2021
doi: 10.3389/fgene.2021.706907

ggVennDiagram: An Intuitive, Easy-to-Use, and Highly Customizable R Package to Generate Venn Diagram

*Chun-Hui Gao[1], Guangchuang Yu[2] and Peng Cai[1]**

[1] State Key Laboratory of Agricultural Microbiology, State Environmental Protection Key Laboratory of Soil Health and Green Remediation, College of Resources and Environment, Huazhong Agricultural University, Wuhan, China, [2] Department of Bioinformatics, School of Basic Medical Sciences, Southern Medical University, Guangzhou, China

Venn diagrams are widely used diagrams to show the set relationships in biomedical studies. In this study, we developed ggVennDiagram, an R package that could automatically generate high-quality Venn diagrams with two to seven sets. The ggVennDiagram is built based on ggplot2, and it integrates the advantages of existing packages, such as venn, RVenn, VennDiagram, and sf. Satisfactory results can be obtained with minimal configurations. Furthermore, we designed comprehensive objects to store the entire data of the Venn diagram, which allowed free access to both intersection values and Venn plot sub-elements, such as set label/edge and region label/filling. Therefore, high customization of every Venn plot sub-element can be fulfilled without increasing the cost of learning when the user is familiar with ggplot2 methods. To date, ggVennDiagram has been cited in more than 10 publications, and its source code repository has been starred by more than 140 GitHub users, suggesting a great potential in applications. The package is an open-source software released under the GPL-3 license, and it is freely available through CRAN (https://cran.r-project.org/package=ggVennDiagram).

Keywords: Venn diagram, grammar of graphic, data visualization, R software, ggplot2

OPEN ACCESS

Edited by:
Alfredo Pulvirenti, University of Catania, Italy

Reviewed by:
Gregorio Iraola, Institut Pasteur de Montevideo, Uruguay
Rifat Hamoudi, University of Sharjah, United Arab Emirates

****Correspondence:***
Peng Cai
cp@mail.hzau.edu.cn

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

Received: *08 May 2021*
Accepted: *06 August 2021*
Published: *07 September 2021*

Citation:
Gao C-H, Yu G and Cai P (2021) ggVennDiagram: An Intuitive, Easy-to-Use, and Highly Customizable R Package to Generate Venn Diagram. Front. Genet. 12:706907. doi: 10.3389/fgene.2021.706907

INTRODUCTION

A Venn diagram is a widely used diagram that shows the relationships between multiple sets. In biomedical studies, a Venn diagram is frequently used in distinguishing the membership of various types of data, such as compounds, genes, pathways, and species. When the number of sets is less than five, Venn diagrams are probably the most intuitive form of data visualization, superior to heat maps and tables.

In the R environment, one of the most popular platforms in biomedical data visualizations, many packages are available to plot a Venn diagram including VennDiagram (Chen and Boutros, 2011), colorfulVennPlot (Noma and Manvae, 2013), venn (Dusa, 2020), nVennR (Quesada, 2021), eulerr (Larsson, 2020), venneuler (Wilkinson, 2011), RVenn (Akyol, 2019), and gplots

(Warnes et al., 2020), to name a few (see **Table 1** for a feature comparison of these packages). As one of the most popular software, VennDiagram supports multiple input formats, and it can also generate Euler diagrams in addition to Venn. In addition, venn supports the drawing of Venn diagrams with up to seven sets. RVenn has been developed as a systematic and easy-to-use method for calculating intersecting and overlapping members in Venn diagrams. It is impossible to develop a state-of-the-art Venn tool without absorbing the strengths of the above-mentioned tools.

However, the above-mentioned software packages also have their disadvantages. First of all, these packages have limitations in displaying the difference between various regions in a Venn diagram in spite of the capability of exhibiting the original sets. ColorfulVennPlot and venn do support region filling, but users need to manually specify colors for every region, making it too complicated to be used by ordinary users. Besides, most of these packages lack full support for grammar of graphics, resulting in the failure of adequate integration into the popular ggplot2 ecosystem. In addition, the inputs of some packages are very obscure; thus, it is time-consuming to obtain a qualified input data.

Considering this, we developed ggVennDiagram, an intuitive, easy-to-use, and customizable R package to generate Venn diagrams, which supports a two- to seven-set Venn plot and generates publication-quality figure with minimal input. Furthermore, we also developed a comprehensive Venn data structure to simplify the expansion of Venn diagrams and make the new presentation of the diagram easy in the future.

RESULTS AND DISCUSSION

Workflow of ggVennDiagram

The main function "ggVennDiagram()" accepts a list input and outputs a *ggplot* object. By measuring the length of input list, it automatically applies internal functions to build a plot in two steps: data pre-processing and visualization. The second step relies on ggplot2's functions; therefore, we mainly focus on explaining the first step as follows.

Data pre-processing then can be divided into two procedures: shape generation, which defines the edges of Venn sets and regions and region value calculation which calculates the region items and performs necessary statistics, such as counting and calculating percentages.

Since the returned data after data pre-processing are compatible with the *sf* object, these data are directly passed into "geom_sf()"/"geom_sf_label()"/"geom_sf_text()" functions intrinsically provided by ggplot2. Filling colors are mapped to the counts of region items, and a color bar legend is generated automatically to show the difference between different regions (**Figure 1A**).

Shape Generation

In ggVennDiagram, we treated all the edges, labels, and polygons as simple features, which refer to a standard to describe how the objects in the real world can be presented in computers, with emphasis on the spatial geometry of these objects. A total of 15 types of simple features are implemented in R, three of which are used to describe all the components of a Venn diagram.

Firstly, the edges of sets are inherited from *LINESTRING*, which is a sequence of points connected by straight non-self-intersecting lines. Secondly, all the possible intersecting regions are inherited from *POLYGON*, which is formed by a sequence of closed points. Thirdly, the labels of sets are inherited from *POINT*, which is a single point used to anchor a short text. Simple features are to define the coordinates of Venn plot components. It is the first time for simple features to be employed in a Venn diagram. Such a design enhances the ability to describe Venn diagram components, making it possible to calculate intersection and overlapping regions between different sets.

To simplify the calculation of simple features, we introduce an S4 class *Polygon* object which expands the S4 class *Venn* object derived from RVenn. As those methods are implemented in RVenn, set operation methods are implemented for *Polygon* object, resulting in the unified set operation functions for the set object *Venn* and the shape object *Polygon*.

The shape used in the Venn diagram with less than four sets can be a simple structure, such as a circle or an ellipse, but when the Venn diagram has more than four sets, irregular polygons are required. It is hard to generate irregular polygons with simple geometric functions. Therefore, ggVennDiagram is designed to bear a built-in preprocessed shape data set imported from venn, VennDiagram, and some online materials, which undoubtedly increases the efficiency of shape generation on the user side.

Region Value Calculation

Region value calculation depends on the RVenn package and new functions written on its defined *Venn* object. There are a total of 2^n - 1 regions in a Venn diagram, in which n indicates the number of sets. The member and its number in each region are stored with region IDs in a *tibble* and joined with the region shape object through unique IDs. Likewise, the member and its number in a set are assigned to the *SetEdge* through unique IDs in parallel. By doing this, a complete *VennPlotData* object is generated for subsequent plotting (**Figure1B**).

Stepwise Self-Customization of Venn Diagrams

After data pre-processing, ggVennDiagram calls native ggplot2 functions to draw Venn diagrams in four layers (**Figures 2A,B**). The first layer is to show the number of members in each region, with gradient color filling exhibiting the differences in member number among various regions. The second layer is to show set edges. When an irregular polygon rather than an ellipse and circle is used to draw a Venn diagram, set edges are essential for distinguishing the boundary between different sets. The third layer is to display set labels, and the fourth layer is to exhibit region labels. The data pre-processing function is accessible to users. Thus, it is easy for those familiar with the ggplot2 syntax to revise the details of the image including

TABLE 1 | Feature comparisons of currently available Venn plot tools (R packages and web tools).

	Grammar of graphics	Data processing			Visualization				References
		Access to region members	Input format	Structured data storage	Region filling[a]	Shapes	No. of sets	Element control (set and region)	
R packages									
ggVennDiagram	Fully support	Yes	List	Yes	Yes	Circle, ellipse, and others	2–7	Set edge/label, region filling/label	Gao, 2021
VennDiagram	No	Yes	List	No	No	Circle, ellipse	2–5	Set edge/label/filling/area, region label	Chen and Boutros, 2011
colorfulVennPlot	No	No	Named vector	No	Yes	Circle, ellipse	2–4	Set label, region filling/label	Noma and Manvae, 2013
venn	No[b]	No	List, formula, set number, Boolean values	No	Yes	Circle, ellipse, and others[c]	2–7[c]	Set edge/label, region filling/label	Dusa, 2020
nVennR	Partial	Yes	List	Yes	No	Irregular polygon (calculated)	2–many	Set edge/filling/area, region label	Quesada, 2021
eulerr	No	No	List, data frame, table, matrix, named vector	No	No	Circle, ellipse	2–4, maybe many[d]	Set label/filling/area, region label	Larsson, 2020
venneuler	No	No	Formula, matrix, character vector	No	No	Circle	2–4, maybe many[d]	Set label/filling/area	Wilkinson, 2011
RVenn	No	Yes[e]	Venn object (derived from list)	No	No	Circle	2–3	Set filling/edge	Akyol, 2019
gplots	No	Yes	List, data frame	No	No	Circle, ellipse	2–5	Set label, region label	Warnes et al., 2020
Online webtool									
InteractiVenn	na	Yes	List (web interface)	na	No	Circle, ellipse, and Edwards	2–6	Set label/filling, region label	Heberle et al., 2015
Venny	na	Yes	List (web interface)	na	Yes	Circle, ellipse	2–4	Set label, region label/filling	Oliveros, 2007

[a]*Region filling indicates that every single part of set intersections/overlapping can be specified separately.*
[b]*venn has a parameter ("ggplot") to enable the output of a ggplot object in plotting.*
[c]*The five- to seven-set Venn diagram is plotted by ggVennDiagram on the basis of venn.*
[d]*When the relationship of different sets is simple enough, eulerr and venneuler can produce an area-proportional Euler plot with more than four sets.*
[e]*Set operation of RVenn is expanded in ggVennDiagram to calculate the shapes in different regions. na, not applicable.*

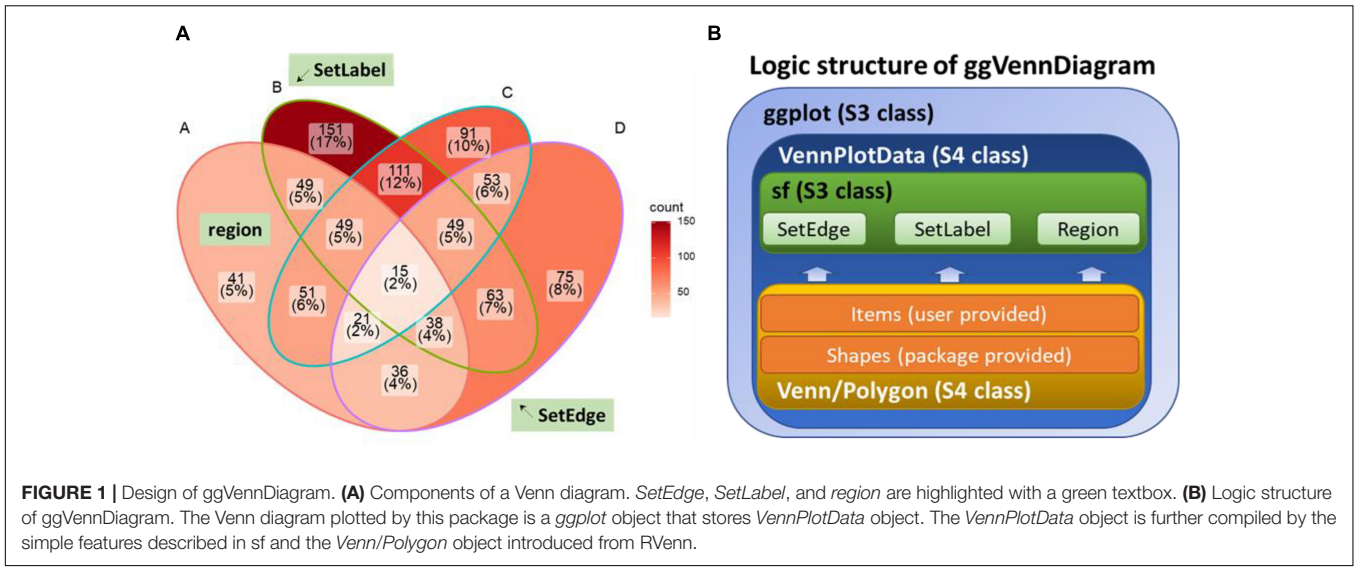

FIGURE 1 | Design of ggVennDiagram. **(A)** Components of a Venn diagram. *SetEdge*, *SetLabel*, and *region* are highlighted with a green textbox. **(B)** Logic structure of ggVennDiagram. The Venn diagram plotted by this package is a *ggplot* object that stores *VennPlotData* object. The *VennPlotData* object is further compiled by the simple features described in sf and the *Venn/Polygon* object introduced from RVenn.

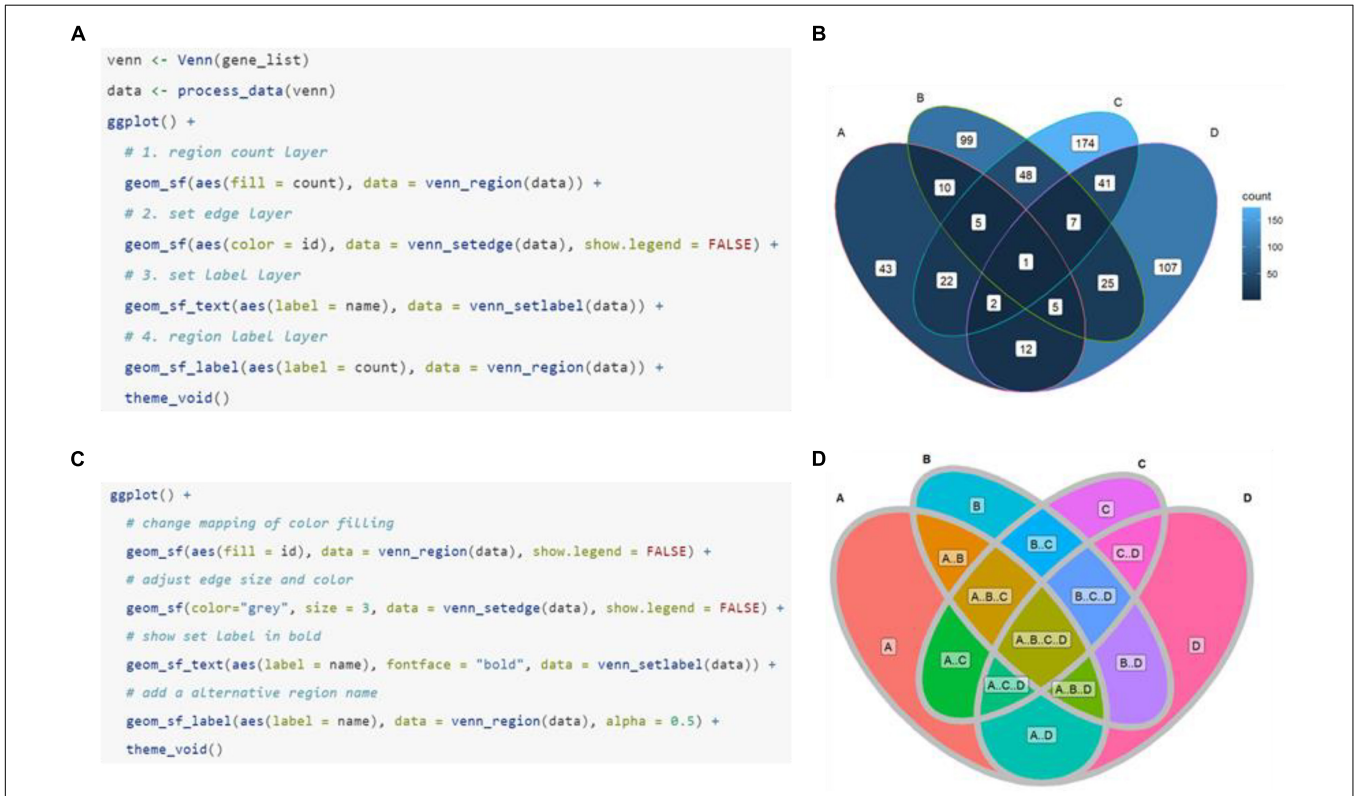

FIGURE 2 | Plotting method of ggVennDiagram. The default manner **(A)** and returned plot **(B)** when user calls "ggVennDiagram()" with a four-set list of genes ("gene_list"). **(C,D)** Stepwise self-customization of the Venn plot by using ordinary ggplot2 functions.

the region fill color, line color/thickness, text style, and so on (**Figures 2C,D**).

Novel Shapes in Venn Diagrams

As has been noted above, a set of built-in shapes from ggVennDiagram is used to plot the Venn diagram. By default, only the most appropriate shape is used when the main function "ggVennDiagram()" is called. However, other applicable shapes can be specified in a stepwise plot, which has been described in the previous section (**Figure 3A**). In addition, ggVennDiagram provides a series of functions to help users with a novel shape when they know shape coordinates. For example, a six-set Venn diagram can be made up of only six triangles (**Figure 3B**). To this end, we just need to pass the vertex coordinates and set label

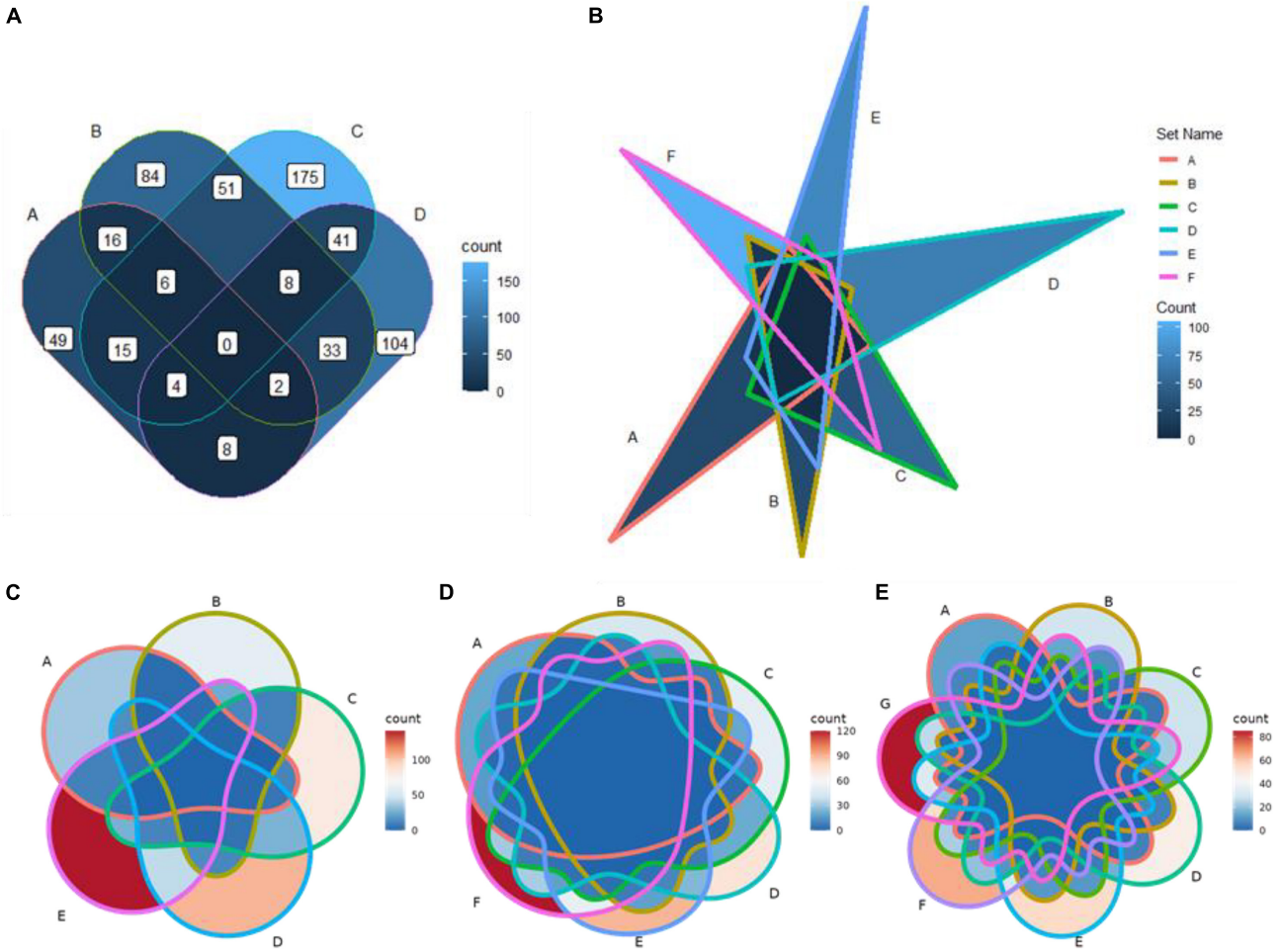

FIGURE 3 | Application of new shapes and support for a Venn diagram of up to seven sets in ggVennDiagram. **(A)** A rounded rectangle is used to plot a four-set Venn diagram. **(B)** A triangle is used to plot a six-set Venn diagram. **(C–E)** Five-, six-, and seven-set Venn diagrams plotted based on ggVennDiagram. The reproducible examples are available from the vignettes attached to the package.

coordinates to the "triangle()" function and "label_position()" function, respectively, and then construct a *VennPlotData* object with the constructor function "VennPlotData()" (**Figure 1B**). The generated *VennPlotData* object now can join with set and calculated region values through "plotData_add_venn()" function, and the resultant data can be used in stepwise customization of the Venn diagram (**Figure 3B**).

Venn Diagram With More Than Four Sets

From version 1.0, ggVennDiagram supports Venn diagrams with up to seven sets (**Figures 3C–E**). This feature is dependent on the shapes imported from another R package venn (Dusa, 2020). However, we insist that Venn diagrams with more than four sets may not be a good choice to display their relationships.

To date, there are three major methods to display set relationships: Venn diagram, Euler diagram, and UpSet plot (Conway et al., 2017). The UpSet plot is a state-of-the-art visualization technique for the quantitative analysis of sets (Lex et al., 2014), and it supports an unlimited number of sets. When the number of sets is very large, it is more justified to choose the UpSet plot.

Integration of ggVennDiagram Into Bioinformatics Analysis Pipelines

The first version of ggVennDiagram was released on October 9th, 2019 (version 0.3). Since then, it has been applied to many biomedical research fields. For example, Cook et al. (2020) used ggVennDiagram to show overlapping differentially expressed genes across three sample times (days 1, 3, and 5) in both the root and the shoot of canola. Besides, Harris et al. (2020) used ggVennDiagram to display that 22.5% of differentially expressed genes were shared by treated mice and human patients. Furthermore, Maguire et al. (2020) used ggVennDiagram to confirm that their novel method has low bias and is more sensitive than three other methods for small RNA library preparation. In addition, ggVennDiagram is also used for analyzing the differences between several spatially varied oral metabolomics samples (Ciurli et al., 2021) and for comparing

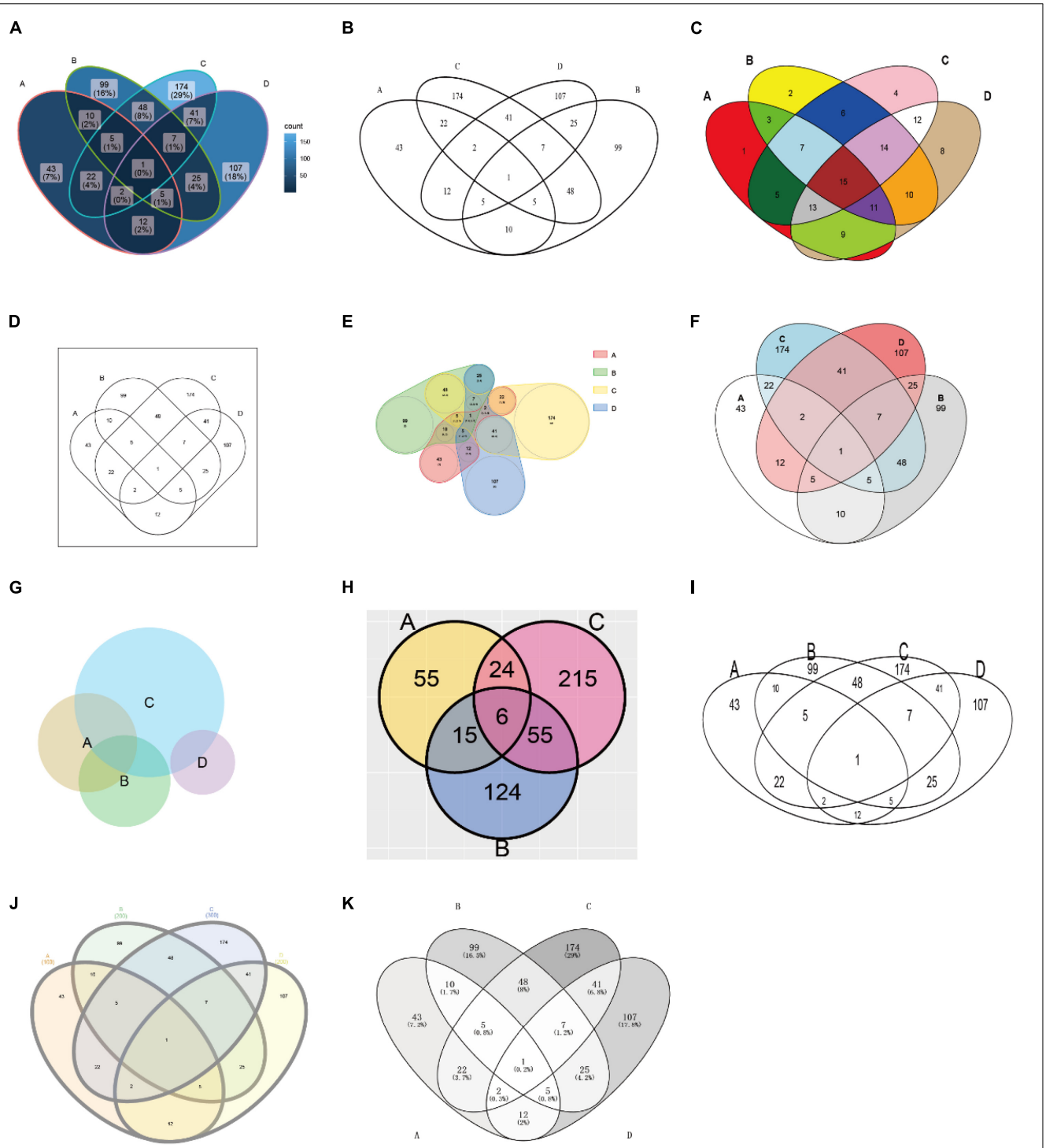

FIGURE 4 | Plots generated by the tools listed in **Table 1**. The used tools are ggVennDiagram, VennDiagram, colorfulVennPlot, venn, nVennR, eulerr, RVenn, gplots, interactiVenn, and Venny **(A–K)**, respectively. Except for venn **(D)**, venneuler **(G)**, and RVenn **(H)**, all plots are generated with minimal configuration (using default parameters) and the same input data, which is a simulated four-set gene list. The input for venn is a named vector, while the input for venneuler is a qualified expression. Since RVenn do not support four-set Venn plots, only three sets of the gene list are used **(H)**.

single-nucleotide variants between tumor and non-tumor tissues (Horny et al., 2021). So far, ggVennDiagram has been cited in more than 20 peer-reviewed articles and open-access preprints, as retrieved by Google Scholar. It could be speculated that ggVennDiagram has a very wide range of application scenarios in biomedical studies.

Feature Comparisons of Currently Available Venn Plot Tools

Table 1 presents the features of currently available Venn plot tools (see also **Figure 4** for the comparison of the generated plots by these tools). First of all, the support for grammar of graphics by nine R packages and two web tools was assessed. Grammar of graphics is a general scheme for data visualization, which breaks up graphs into semantic components, such as scales and layers. Except ggVennDiagram, none of these tools fully support this feature in plotting Venn diagrams.

Additionally, ggVennDiagram takes the lead in the following three aspects of data processing capacity. (1) We can get access to region members by querying the *VennPlotData* object. (2) It should be noted that we only implement the input of list (as input format). This design is simple enough to understand and prepare, and it is easy to store set members, which is essential for the calculation of region members. (3) *Via* the design of a layered object, ggVennDiagram can store plotting data into the *VennPlotData* object (**Figure 1B**), thus making it possible to query and reuse the target data.

Furthermore, ggVennDiagram is superior in four aspects of visualization. (1) Region filling allows the user to easily identify the differences between various parts of the Venn diagram, and this is one of the key features of ggVennDiagram. Although several other tools have this feature, only ggVennDiagram is fully automatic since it is driven by ggplot2's aesthetic mapping. (2) The ggVennDiagram has built-in shapes consisting of circles, ellipses, and others. Besides, we also provide functions to help users to import self-defined shapes (**Figures 3A,B**). (3) The ggVennDiagram supports two- to seven-set Venn diagrams, which is adequate for daily use. (4) Element control in ggVennDiagram can be applied for set edge/label and region filling/label, so that it is convenient to set their color/line type/size, and so on (**Figures 1A**, **2B,D**, **3A–E**).

Notably, several tools support both Venn and Euler diagrams. However, an Euler diagram has two shortages: firstly, it is area proportional, but the human eye is less sensitive to area than to color; secondly, it only shows relevant relationships, but sometimes, it is impossible to show all intersection regions merely by using simple geometric shapes, such as circles and ellipses. Therefore, we assume that it is more appropriate to use color filling for displaying the difference between different regions in ordinary biomedical studies.

Overall, ggVennDiagram integrates and optimizes a Venn diagram plotting method, exhibiting multiple advantages in performance over current existing tools. Compared with webtool, R scripts are easier to integrate into the existing bioinformatics analysis pipelines to realize automation and batch drawing of Venn diagrams. Therefore, it is necessary and useful to develop ggVennDiagram.

DATA AVAILABILITY STATEMENT

The ggVennDiagram R package is open source and freely available on CRAN (https://cran.r-project.org/package=ggVennDiagram) and GitHub (https://github.com/gaospecial/ggVennDiagram). ggVennDiagram mainly requires R (>3.5.0), the ggplot2, and sf packages, and its full function also depends on the plotly package.

AUTHOR CONTRIBUTIONS

C-HG, GY, and PC wrote this manuscript. C-HG implemented this package with the help of GY. PC supervised the project. All authors contributed to the article and approved the submitted version.

FUNDING

This work was supported by the National Natural Science Foundation of China (32100090, 41877029, and 41961130383), Royal Society-Newton Advanced Fellowship (NAFR1191017), the National Key Research Program of China (2020YFC1806803), Wuhan Applied Foundational Frontier Project (2019020701011469), and Fundamental Research Funds for the Central Universities (2662021JC012).

ACKNOWLEDGMENTS

We thank Adrian Dușa for letting us reuse the "venn:::sets" data in his venn package, and this is critical to enable five- to seven-set Venn diagrams in ggVennDiagram. We also thank the GitHub user Yi Liu (@liuyigh) for his contribution on code curation. Great gratitude goes to linguistics Ping Liu from Huazhong Agriculture University, Wuhan, China, for her work on English editing and language polishing.

REFERENCES

Akyol, T. Y. (2019). *RVenn: set Operations for Many Sets*. Available online at: https://CRAN.R-project.org/package=RVenn (accessed May 1, 2021).

Chen, H., and Boutros, P. C. (2011). VennDiagram: a package for the generation of highly-customizable Venn and Euler diagrams in R. *BMC Bioinform.* 12:35. doi: 10.1186/1471-21\05-12-35

Ciurli, A., Liebl, M., Derks, Rico. J. E, Neefjes, J. J. C., and Giera, M. (2021). Spatially resolved sampling for untargeted metabolomics: a new tool for salivomics. *iScience* 24:102768. doi: 10.1016/j.isci.2021.102768

Conway, J. R., Lex, A., and Gehlenborg, N. (2017). UpSetR: an R package for the visualization of intersecting sets and their properties. *Bioinformatics* 33, 2938–2940. doi: 10.1093/bioinformatics/btx364

Cook, J., Douglas, G. M., Zhang, J., Glick, B. R., Langille, M. G. I., Liu, K.-H., et al. (2020). Transcriptomic profiling of Brassica napus responses to *Pseudomonas aeruginosa*. *Innate Immun.* 27, 143–157. doi: 10.1177/1753425920980512

Dusa, A. (2020). *venn: draw venn diagrams*. Available online at: https://CRAN.R-project.org/package=venn (accessed May 1, 2021).

Gao, C.-H. (2021). *ggVennDiagram: a ggplot2 implement of venn diagram*. Available online at: https://github.com/gaospecial/ggVennDiagram (accessed May 1, 2021).

Harris, S. E., Poolman, T. M., Arvaniti, A., Cox, R. D., Gathercole, L. L., and Tomlinson, J. W. (2020). The American lifestyle-induced obesity syndrome diet in male and female rodents recapitulates the clinical and transcriptomic features of nonalcoholic fatty liver disease and nonalcoholic steatohepatitis. *Am. J. Physiol. Gastrointest. Liver Physiol.* 319, G345–G360. doi: 10.1152/ajpgi.00055.2020

Heberle, H., Meirelles, G. V., da Silva, F. R., Telles, G. P., and Minghim, R. (2015). InteractiVenn: a web-based tool for the analysis of sets through Venn diagrams. *BMC Bioinform.* 16:169. doi: 10.1186/s12859-015-0611-3

Horny, K., Gerhardt, P., Hebel-Cherouny, A., Wülbeck, C., Utikal, J., and Becker, J. C. (2021). Mutational Landscape of Virus- and UV-Associated Merkel Cell Carcinoma Cell Lines Is Comparable to Tumor Tissue. *Cancers* 13:649. doi: 10.3390/cancers13040649

Larsson, J. (2020). *eulerr: area-proportional Euler and Venn Diagrams With Ellipses*. Available online at: https://CRAN.R-project.org/package=eulerr (accessed May 1, 2021).

Lex, A., Gehlenborg, N., Strobelt, H., Vuillemot, R., and Pfister, H. (2014). UpSet: visualization of Intersecting Sets. *IEEE Trans. Vis. Comput. Graph.* 20, 1983–1992. doi: 10.1109/TVCG.2014.2346248

Maguire, S., Lohman, G. J. S., and Guan, S. (2020). A low-bias and sensitive small RNA library preparation method using randomized splint ligation. *Nucleic Acids Res.* 48:e80. doi: 10.1093/nar/gkaa480

Noma, E., and Manvae, A. (2013). colorfulVennPlot: plot and Add Custom Coloring to Venn Diagrams for 2-Dimensional, 3-Dimensional and 4-Dimensional data. Available Online at: https://CRAN.R-project.org/package=colorfulVennPlot (accessed May 1, 2021).

Oliveros, J. C. (2007). *Venny. an Interactive Tool for Comparing Lists with Venn's Diagrams*. Available Online at: https://bioinfogp.cnb.csic.es/tools/venny/index.html [Accessed July 6, 2021].

Quesada, V. (2021). *nVennR: create n-Dimensional, Quasi-Proportional Venn Diagrams*. Available online at: https://CRAN.R-project.org/package=nVennR [Accessed March 3, 2021].

Warnes, G. R., Bolker, B., Bonebakker, L., Gentleman, R., Huber, W., Liaw, A., et al. (2020). *gplots: various r programming tools for plotting data*. Available online at: https://github.com/talgalili/gplots (accessed May 1, 2021).

Wilkinson, L. (2011). *venneuler: venn and Euler Diagrams*. Available online at: https://CRAN.R-project.org/package=venneuler (accessed May 1, 2021).

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers in Genetics

TECHNOLOGY AND CODE
published: 17 September 2021
doi: 10.3389/fgene.2021.731355

HandyCNV: Standardized Summary, Annotation, Comparison, and Visualization of Copy Number Variant, Copy Number Variation Region, and Runs of Homozygosity

Jinghang Zhou[1,2], Liyuan Liu[1,2], Thomas J. Lopdell[3], Dorian J. Garrick[2] and Yuangang Shi[1]**

[1] School of Agriculture, Ningxia University, Yinchuan, China, [2] AL Rae Centre for Genetics and Breeding, Massey University, Hamilton, New Zealand, [3] Research and Development, Livestock Improvement Corporation, Hamilton, New Zealand

Detection of CNVs (copy number variants) and ROH (runs of homozygosity) from SNP (single nucleotide polymorphism) genotyping data is often required in genomic studies. The post-analysis of CNV and ROH generally involves many steps, potentially across multiple computing platforms, which requires the researchers to be familiar with many different tools. In order to get around this problem and improve research efficiency, we present an R package that integrates the summarization, annotation, map conversion, comparison and visualization functions involved in studies of CNV and ROH. This one-stop post-analysis system is standardized, comprehensive, reproducible, timesaving, and user-friendly for researchers in humans and most diploid livestock species.

Keywords: copy number variant, run of homozygosity, haplotype, SNP, CNVR

OPEN ACCESS

Edited by:
Guangchuang Yu,
Southern Medical University, China

Reviewed by:
Xiaofeng Huang,
Cornell University, United States
Max Robinson,
Institute for Systems Biology, United States

****Correspondence:***
Dorian J. Garrick
D.Garrick@massey.ac.nz
Yuangang Shi
shyga818@126.com

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

Received: *26 June 2021*
Accepted: *25 August 2021*
Published: *17 September 2021*

Citation:
Zhou J, Liu L, Lopdell TJ, Garrick DJ and Shi Y (2021) HandyCNV: Standardized Summary, Annotation, Comparison, and Visualization of Copy Number Variant, Copy Number Variation Region, and Runs of Homozygosity. Front. Genet. 12:731355. doi: 10.3389/fgene.2021.731355

INTRODUCTION

Genome-wide data have been accumulated for large numbers of individuals of various species as the cost of single nucleotide polymorphism (SNP) genotyping continues to decrease. In addition to using these data for GWAS (genome wide association study) or GS (genomic selection), interesting genomic information about copy number variant (CNV) and runs of homozygosity (ROH) can be inferred from these genotypes, and a range of software products [such as PennCNV (Wang et al., 2007), CNVPartition (Illumina, 2021), SNP and Variation Suite (Bozeman and Golden Helix, 2020)] have been developed to detect CNV and ROH for SNP data. However, few tools can integrate the summary data with annotations, comparisons, and visualizations of these results. As a result, extracting useful information from CNV and ROH data sets is time consuming, especially when it requires processing multiple results from different models and software. In order to get more comprehensive results, researchers often implement their own pipelines to switch back and forth between different tools, an approach that is prone to introducing bugs and thereby producing spurious results.

There are several common "pitfalls" we have observed when conducting CNV analyses using SNP genotyping data. The most frequent is to annotate the candidate genes in a CNVR (copy number variation region) without considering the frequency of the CNVs: this can result in undue weight being given to rare CNVs that affect only one or two samples. A second issue is comparing CNVs between different studies, and making comparisons only at the population level, and not at the individual sample level. Comparison at the population level could reflect the ubiquitous nature

of CNVs, but at the individual level it also provides information about the robustness of CNV detection algorithms. A third issue arises when comparing CNVRs that have been detected using different reference genomes, which requires converting the coordinates of the regions between the two genomes. Making these conversions requires careful consideration, as the order of SNPs on chromosomes might differ between two different reference assemblies, such that the lengths or even chromosomal orders of CNVs can change, which might lead to meaningless comparisons between CNVRs. A fourth common problem is get the incorrect number of overlapping CNVRs when presenting comparison results via Venn diagram. Since the number of overlapping regions is relative to the results, and a single long interval generated using one approach might overlap multiple shorter intervals detected using another approach, in which case representing the results via Venn diagram requires special annotation.

There are also some steps that may be easily forgotten performing ROH analysis on SNP genotyping data. For example, the SNP density distributions may not have been carefully examined prior to inference of ROH. The density of SNPs may differ across the chromosome on different SNP chips, but ROH detection methods are highly affected by characteristics such as SNP density, window size, tolerance of occasional heterozygosity in the run, and the presence of missing values in the detection window. Knowing SNP density can therefore help us to select better parameters when performing ROH detection. Moreover, while reporting the candidate genes by functional annotation of genes that located in ROH regions, we may not examine the frequencies of haplotypes within these interesting genes, but this step could provide valuable information about the high frequency genotypes of these genes, which is useful on designing the further validation experiments and can provide the valuable reference to others when they comparing the genes using the same SNP chips on different populations.

There are several common requirements in studying CNV and ROH patterns in a new species or population. These include: the need for preparing summary tables, making summary figures, generating CNVRs and plotting CNVR distribution maps with gene annotations, comparing CNVs and CNVRs between studies, converting genome coordinates and map files from one reference to another, finding high frequency abnormal genomic regions, creating consensus gene lists, producing custom visualization of results, and identifying haplotypes in regions of interest. Therefore, we built this open-source tool to provide a standardized, reproducible, time-saving and widely available one-stop post-analysis system to make research more simple, practical and efficient while avoiding common "pitfalls" that can affect the accuracy and interpretability of these studies.

METHOD

Brief Introduction of Main Functions

The functions provided by this package can be categorized into five sections: Conversion; Summary; Annotation; Comparison; and Visualization. The most useful features provided are: integrating summarized results, generating lists of CNVRs, annotating the results with known gene positions, plotting CNVR distribution maps, and producing customized visualizations of CNVs and ROHs with gene and other related information on one plot (**Figure 1**). This package supports a range of customizations, including the color, size of high-resolution figures, and choice of output folder to avoid conflict between the results of different runs. Where applicable, output files are compatible with other software such as PennCNV (Wang et al., 2007), Plink (Chang et al., 2015), or DAVID annotation tools (Jiao et al., 2012).

The conversion section handles the conversions of genomic positions between two reference genomes, and provides two functions. *convert_map* is designed to compare SNP map files for two different reference genomes, matching by SNP name, and produce SNP maps in a format suitable for use by *convert_coord*. The function also reports the density of SNPs by chromosome. *convert_coord* is designed to convert the physical positions of genomic intervals based on a given SNP map file. Currently, the function is limited to inputs generated by *convert_map*, and can only convert the coordinates for intervals on the same type of SNP chip. Converting coordinates may change the total length of the intervals, as the positions and orders of the SNPs on the chromosome will potentially differ between various reference genomes; therefore, the function produces a table that summarizes how many intervals were converted successfully, and reports on the differences in length between the converted and original intervals.

The summary section contains a group of functions to summarize CNV results, generate CNVRs, and make CNVR distribution maps from CNV results. There is also a collection of functions to summarize ROH results, report frequencies of ROH regions, inbreeding coefficient by different length groups and to generate haplotypes on interesting ROH regions.

The functions used for reporting CNV results include *clean_cnv*, *summary_cnv_plot*, and *call_cnvr*. *clean_cnv* takes a CNV list from PennCNV and CNVPartition and reformats it into a standard format for use in the functions listed below. *cnv_summary_plot* generates a range of summary plots, aggregating CNV results by length group, CNV type, chromosome, and individual. *call_cnvr* generates CNV regions as the union of sets of CNVs that overlap by at least one base pair (Redon et al., 2006). This function will output three tables: (a) the list of CNVRs, containing the number of CNVs and number of samples in each CNVR that can reflect the frequency of CNVRs; (b) a brief summary table showing numbers of CNVRs by length and type (Deletion, Duplication, and Mixed, where Mixed indicates that both duplications and deletions are found within the CNVR); and (c) the total length and number of CNVRs on each chromosome.

roh_window will report: a table of high frequency ROH regions on the autosomes that passed the common frequency threshold, a table containing inbreeding coefficients by different length groups of each individual, a brief summary of the total numbers and lengths of ROHs in length groups, and a plot of high frequency ROH regions by chromosome. The inbreeding coefficients are calculated as $F_{roh} = \left(\sum L_{roh}\right)/\left(\sum L_{auto}\right)$

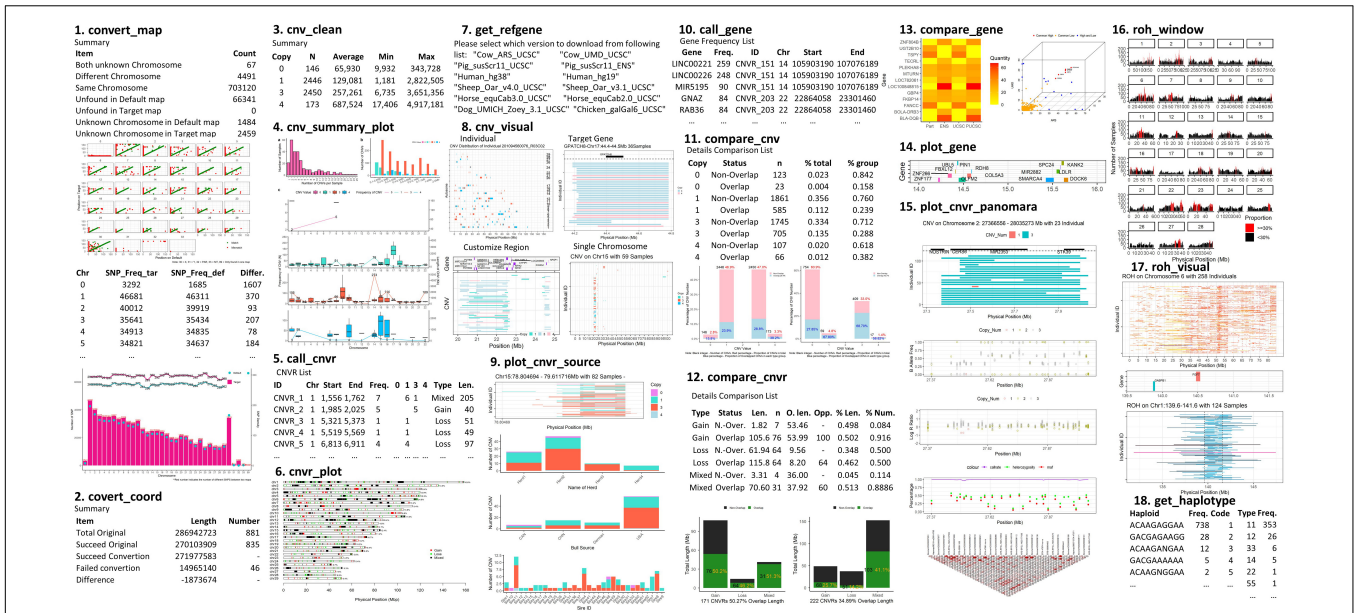

FIGURE 1 | Example plots illustrating the main functions and output from the HandyCNV package.

(McQuillan et al., 2008), where $\sum L_{roh}$ is the total length of ROH, and $\sum L_{auto}$ is the total length of autosomes. Other functions in this group include *prep_phased*, *closer_snp*, and *get_haplotype*; see the package vignette for more information (Jinghang et al., 2021).

The annotation section facilitates downloading and formatting reference gene lists, and annotating genes on genomic intervals. *get_refgene* will automatically download a reference gene list and invoke *clean_ucsc* and *clean_ensgene* from UCSC (Navarro Gonzalez et al., 2021) websites for human, cow, sheep, pig, horse, chicken or dog species, then remove the duplicated genes and report the standard format as output. *call_gene* is used to report how many genes are located in the given genomic intervals. The frequency of genes is calculated from the number of samples that has the same gene annotated in its CNVs.

The comparison section consists of functions for comparing sets of CNVs (*compare_cnv*), CNVRs (*compare_cnvr*), gene frequency lists (*compare_gene*), and other intervals (*compare_interval*). These functions were implemented using the *foverlaps* function in the *data.table* R package (Dowle et al., 2019). *compare_gene* can produce consensus gene lists, given lists of genes present in CNVRs in multiple studies. The remaining functions report numbers, lengths, and proportions of overlapping intervals (CNVs, CNVRs, etc.) on a population and individual basis.

Finally, twelve functions in HandyCNV are included in the visualization section; of these, five produce plots as a subset of their output, and have been mentioned previously: *cnv_summary_plot*, *roh_window*, *compare_cnv*, *compare_cnvr*, and *convert_map*. The remaining visualization functions mainly focus on customizing and integrating the plotting of all information related to CNV, ROH, and high frequency CNVR: these are *cnvr_plot*, *plot_gene*, *cnv_visual*, *roh_visual*, *plot_cnvr_panorama*, *plot_snp_density*, and *plot_cnvr_source*. These functions are described in the package vignette (Jinghang et al., 2021).

Pipelines for the Post Analysis of CNVs and ROHs

Post-analysis of CNVs and CNVRs

The recommended pipeline contains 14 basic steps depending on the study purposes (**Figure 2**), although usage is not limited to these basic steps, and users are free to explore their data by customizing the functions. By running through this pipeline, users can produce a wide range of results, such as summary tables and plots of CNV results, the CNVR list and its brief summary information and CNVR distribution plot, the frequency of CNVs and CNVRs within annotated genes, and comparison results between CNVs, CNVR, and annotated genes.

Post-analysis of ROHs

The pipeline for the post analysis of ROHs contains eight basic steps (**Figure 3**). The main results produced by running through this pipeline are the high frequency ROH regions list, ROH-based inbreeding coefficients, a list of genes that are located in the ROH regions, and the frequency of haplotypes within genes or regions of interest.

APPLICATION EXAMPLES OF CNV AND ROH

We now provide two example runs of the pipeline, using two previously published data sets: the first is a CNV list produced for a human population in Brazil (de Godoy et al., 2020), and the second is genotype data for an inbred breed of horses

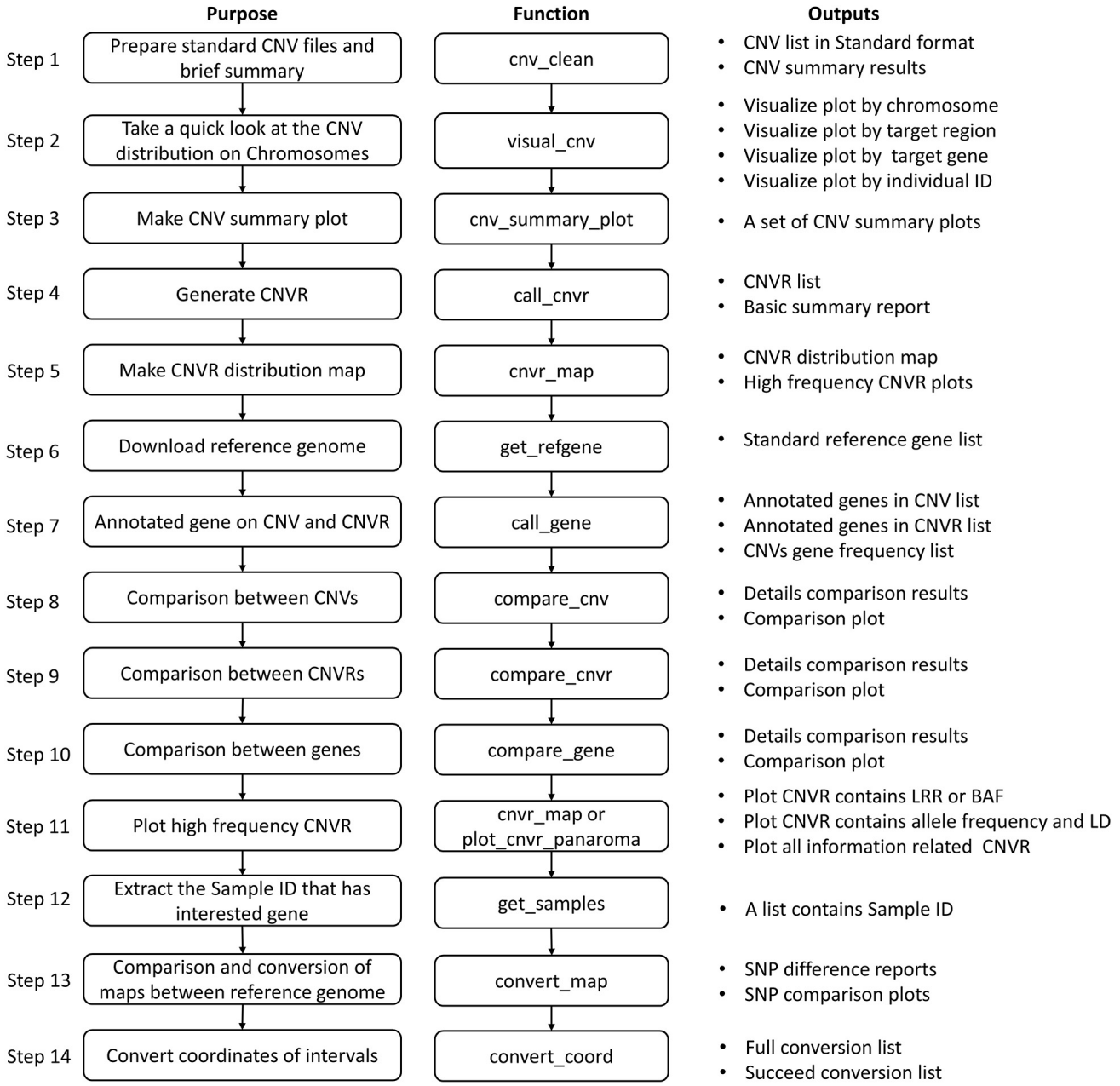

FIGURE 2 | Pipeline of post analysis of CNV results using HandyCNV.

(Velie et al., 2016). The purpose of these examples is to introduce how to use the functions in this package; therefore, further interpretation of the results is not included.

Example 1. the Post-analysis of CNVs in a Human Dataset

The CNV result in this example was cited from a study published in 2020 which comprised 268 microarrays samples in a human population in Brazil (de Godoy et al., 2020). In this example, we will introduce how to prepare the standard CNV list, then produce brief summary, generate CNVRs, annotate genes and visualize CNVs. **Figure 4** presents the code used in example 1, the R script can be found in **Supplementary File 1**.

To replicate this example, we first need to download the dataset "Table S1 – Detailed information about all CNVs analyzed in our sample" (de Godoy et al., 2020) and save the sheet "All array platforms' CNVs" as.*csv* format file. Then use *read.csv* to load the CNV list and select the columns required by *cnv_clean* (see **Figure 5C**).

A formatted clean CNV list will return as an object named "clean_cnv" in working environment, and a brief summary table

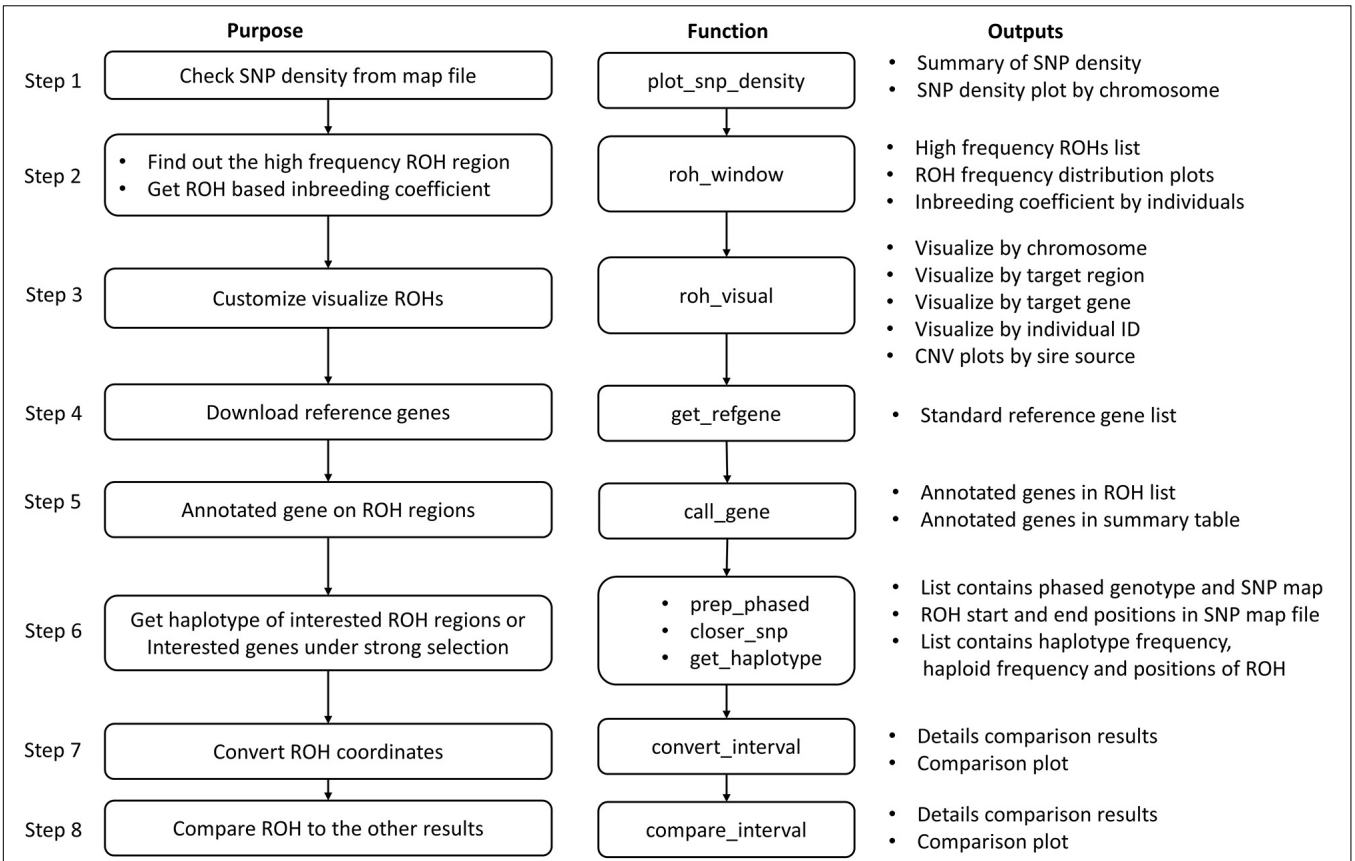

FIGURE 3 | Pipeline of post-analysis of ROH in HandyCNV.

of CNV (see **Figure 5D**) will be written out after executing *cnv_clean*.

We then take a quick look at the CNV distribution by reading the "clean_cnv" list as input and customizing parameters in *cnv_visual*. In example, we first set "chr_id = 14" to visualize CNVs distribution on chromosome 14 (see **Figure 5E**), then zoom into the region with higher frequency CNVs (see **Figure 5I**) by setting "start_position = 105" and "end_position = 110." Visualizing other chromosomes or regions and changing the colors of copy numbers can easily be done by adjusting the relevant arguments.

The CNV summary plot (see **Figure 5A**) can be plotted via *cnv_summary_plot* by taking "clean_cnv" as input. The CNVR list (see **Figure 5F**) is generated using call_cnvr by taking the "clean_cnv" file as input, producing a brief summary table of CNVR (see **Figure 5G**) that will be saved in the working directory in the meantime. The CNVR distribution map (see **Figure 5B**) is generated via *cnvr_plot* by loading the CNVR list.

For gene annotation steps, the reference gene list can be downloaded and formatted by assigning the genome version argument in *get_refgene*. Then the genes annotation list of CNV or CNVR are generated by running *call_gene*. Three input files need be assigned in the function: the clean CNV file ("clean_cnv"), the CNVR list ("cnvr"), and the reference gene list ("human_hg19"); the gene frequency list (see **Figure 5J**) will be returned as an object in the R environment. We can plot all the high frequency CNVRs with gene annotation results (see one example plot in **Figure 5H**) at the same time through cnvr_plot by reading "cnvr," "clean_cnv" and reference gene list ("human_hg19") and setting the "sample_size" and "common_cnv_threshold" arguments.

Finally, we can extract Sample IDs of CNVs that contain genes of interest (see **Figure 5K**) using *get_samples*, by loading the CNV annotation list generated by *call_gene* and assigning the gene name to the "gene_name" argument.

Since this example only contains one CNV result in one reference genome, the functions in the comparison and conversion sections are not applicable in this example. Users of these functions can browse the vignette of this package from the Github repository (Jinghang et al., 2021).

Example 2. the Post-analysis of ROH Using Horse Genotype Samples

The genotype data used to detect ROH in this example is from the work of Velie et al. (2016) and contains 285 horse samples. This example aims to present how to use the functions in HandyCNV to analyze ROHs. This example includes ROH detection by Plink 1.9 (Chang et al., 2015) and genotype phasing by Beagle 5.1 (Browning et al., 2018). **Figure 6** presents

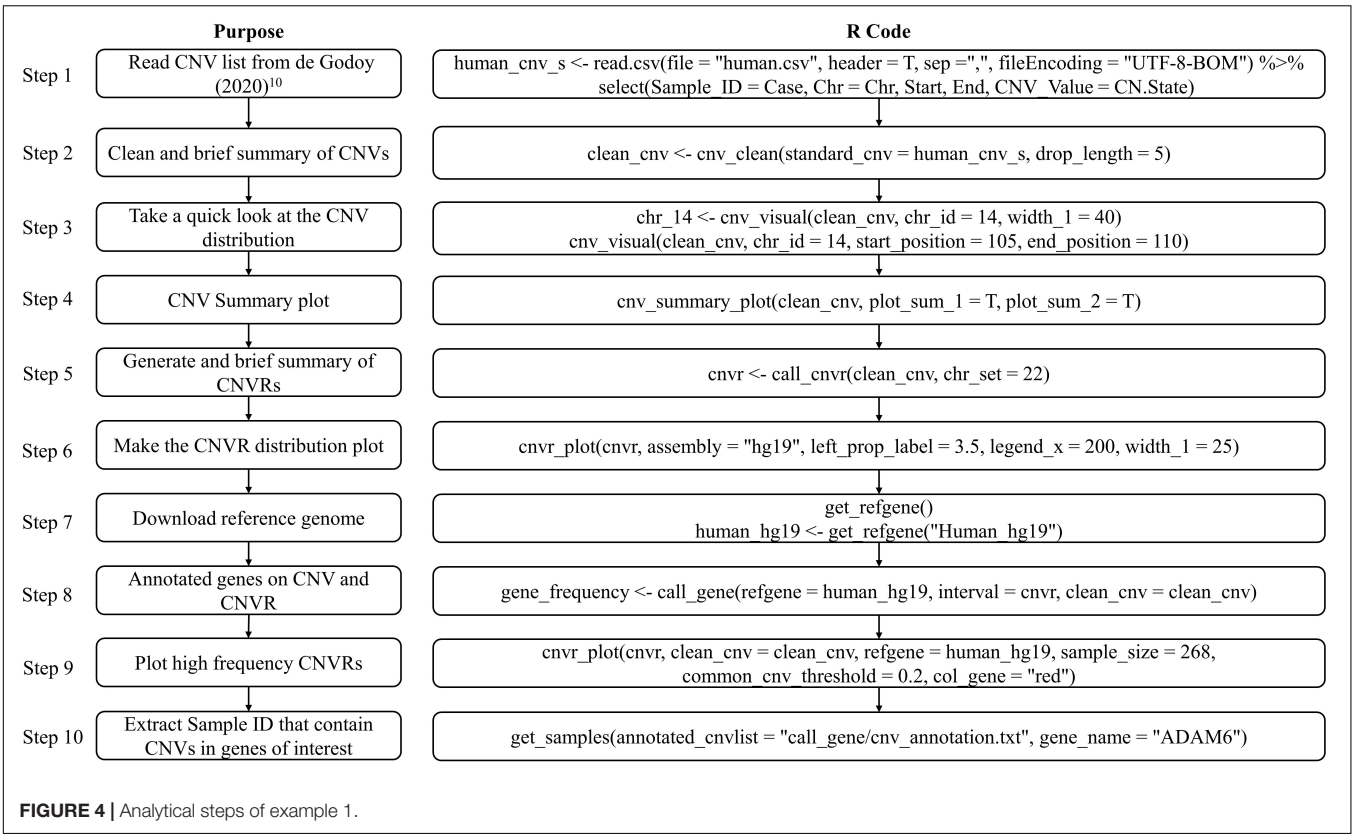

FIGURE 4 | Analytical steps of example 1.

the code used in example 2; the R script can be found in **Supplementary File 2**.

To run this example, we first need to prepare the genotype data. The genotype files are read using the *fread* function (Dowle et al., 2019). Because the original ped file does not match the format required by Plink 1.9, we insert a sequential column of family IDs, plus placeholder columns of zeroes for the father, mother, and sex code by using *data.frame* and *cbind* functions (R Core Team, 2020). Before testing the ROH, the map file was loaded as the input file in *plot_snp_density* to get a brief summary and visualization of SNP density (**Figure 7A**). The *jpeg* and *dev.off* functions (R Core Team, 2020) are used to save the plot.

Then, we invoke Plink 1.9 (Chang et al., 2015) by *shell* (R Core Team, 2020) from R Studio (Team, 2021) to generate binary genotype files and call ROH. For Windows operating systems, ensure that the plink.exe file is either in the current directory or accessible via the PATH system variable. To run Plink 1.9 on other operation system, please refer to the Plink website (Chang et al., 2015).

Once we get ROH results, we can run *roh_window*, which takes a "plink.hom" file as input to report the brief summary of ROH by length group (see **Figure 7B**), high frequency ROH regions (see **Figure 7D**), ROH frequency distribution plot (see **Figure 7G**), and to calculate the ROH based inbreeding coefficient (**Figure 7K**).

In this example, we present visualizations of ROH on the whole of chromosome 22 (see **Figure 7C**) and on the 22.81–23.22 Mb region on chromosome 22 (see **Figure 7E**) via *roh_visual*, which needs to load the "plink.hom" data set as input. The "chr_id" or "target_region" arguments are available to customize visualization, alongside additional arguments to customize the colors of ROHs.

The horse reference gene list ("quaCab2") was downloaded from the UCSC website (Navarro Gonzalez et al., 2021) by *get_refgene*. The genes located in the high frequency ROH regions (see **Figure 7F**) were annotated via *call_gene*, which requires loading the reference gene list ("quaCab2") and the high frequency ROH regions file that was generated by *roh_window*. Since we have the reference gene list, we can visualize ROH region with genes (see **Figure 7H**) via roh_visual by assigning the clean ROH file ("clean_roh = clean_roh"), target ROH region ["target_region = c (1, 139.6, 141.6)"] and reference gene lists ("refgene = equaCab2"). We can also visualize ROHs in terms of the gene we are interested in: here, we are looking at the *GABPB1* gene, first, exacting the physical position of this gene from the reference gene list ("equaCab2") using the "*filter*" and "*select*" functions (Wickham et al., 2019), then using *visual_roh* to load the ROH file ("plink.hom") as input and assigning the gene position to the "target_region" argument to present the plot (see **Figure 7E**). We can write a loop (R Core Team, 2020) of *visual_roh* to plot all regions with genes annotated by iterating over the high frequency ROHs that contain genes.

To get the haplotype of the genes need the phased genotype files. Here, we take chromosome 1 as example to present how to use Plink 1.9 (Chang et al., 2015) and Beagle 5.1 (Browning et al., 2018) to phase the genotypes. The *shell* (R Core Team, 2020)

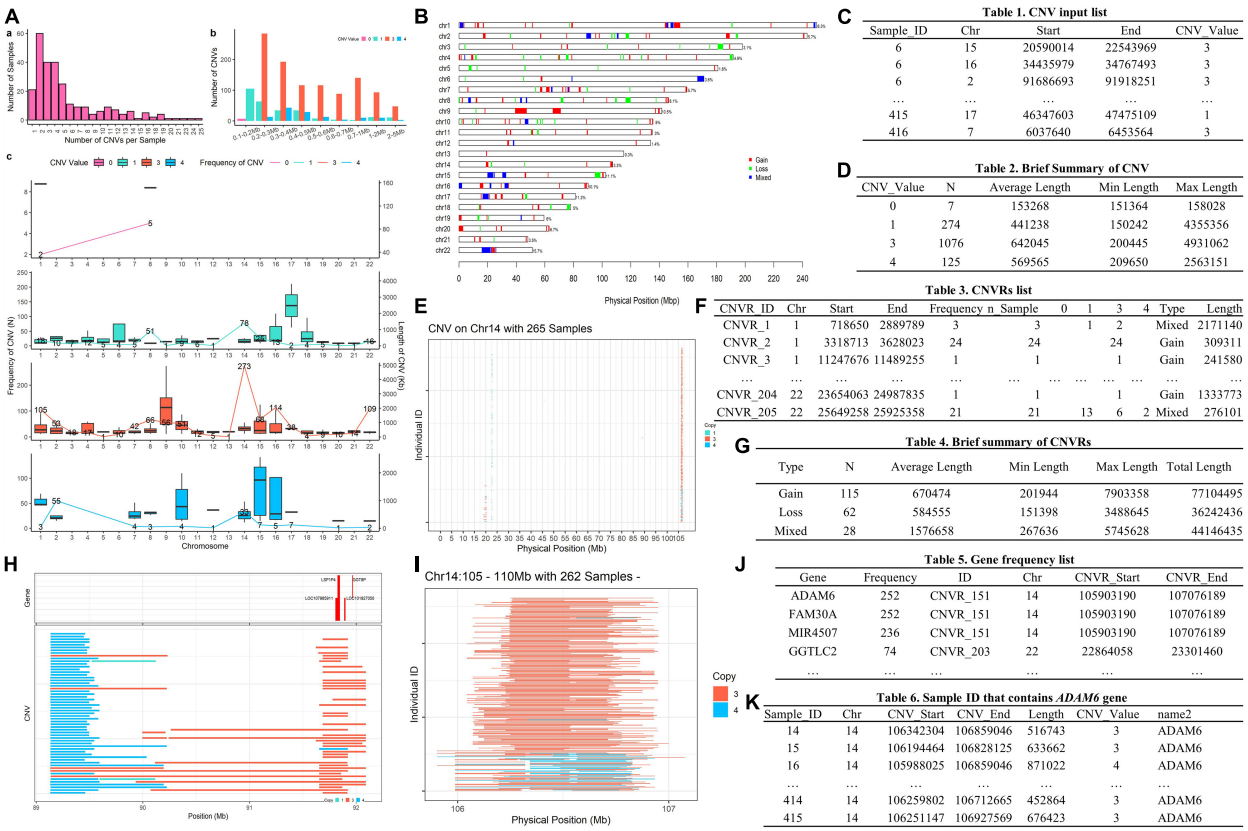

Table 1. CNV input list

Sample_ID	Chr	Start	End	CNV_Value
6	15	20590014	22543969	3
6	16	34435979	34767493	3
6	2	91686693	91918251	3
…	…	…	…	…
415	17	46347603	47475109	1
416	7	6037640	6453564	3

Table 2. Brief Summary of CNV

CNV_Value	N	Average Length	Min Length	Max Length
0	7	153268	151364	158028
1	274	441238	150242	4355356
3	1076	642045	200445	4931062
4	125	569565	209650	2563151

Table 3. CNVRs list

CNVR_ID	Chr	Start	End	Frequency	n_Sample	0	1	3	4	Type	Length
CNVR_1	1	718650	2889789	3	3		1	2		Mixed	2171140
CNVR_2	1	3318713	3628023	24	24			24		Gain	309311
CNVR_3	1	11247676	11489255	1	1			1		Gain	241580
…	…	…	…	…	…	…	…	…	…	…	…
CNVR_204	22	23654063	24987835	1	1			1		Gain	1333773
CNVR_205	22	25649258	25925358	21	21		13	6	2	Mixed	276101

Table 4. Brief summary of CNVRs

Type	N	Average Length	Min Length	Max Length	Total Length
Gain	115	670474	201944	7903358	77104495
Loss	62	584555	151398	3488645	36242436
Mixed	28	1576658	267636	5745628	44146435

Table 5. Gene frequency list

Gene	Frequency	ID	Chr	CNVR_Start	CNVR_End
ADAM6	252	CNVR_151	14	105903190	107076189
FAM30A	252	CNVR_151	14	105903190	107076189
MIR4507	236	CNVR_151	14	105903190	107076189
GGTLC2	74	CNVR_203	22	22864058	23301460
…	…	…	…	…	…

Table 6. Sample ID that contains *ADAM6* gene

Sample_ID	Chr	CNV_Start	CNV_End	Length	CNV_Value	name2
14	14	106342304	106859046	516743	3	ADAM6
15	14	106194464	106828125	633662	3	ADAM6
16	14	105988025	106859046	871022	4	ADAM6
…	…	…	…	…	…	…
414	14	106259802	106712665	452864	3	ADAM6
415	14	106251147	106927569	676423	3	ADAM6

FIGURE 5 | The main outputs of example 1. Panel **(A)** is CNV summary plot; panel **(**B**)** is CNVR distribution map; panel **(C)** is CNV input list; panel **(D)** is the brief summary table of CNV; panel **(E)** is a plot of CNVs on Chromosome 14; panel **(F)** is CNVR list; panel **(G)** is the brief summary table of CNVRs; panel **(H)** is an example plot of the high frequency CNVR; panel **(I)** is a plot of CNVs on Chr14:105-110 Mb; panel **(J)** is the gene frequency list; and panel **(K)** is the sample list that contain CNVs in the LINC00221 gene.

function is used to invoke plink (Chang et al., 2015) to generate the VCF format genotype file, then to invoke beagle (Browning et al., 2018) to phase the genotypes from Rstudio (Team, 2021). For Windows operating systems, ensure that the plink and java executables are either in the current directory or accessible via the PATH system variable. Likewise, adjust the path to the Beagle JAR file as required for your operating system. For instructions on installing and running Beagle 5.1, refer to their manual (Browning et al., 2018).

Finally, we take *GABPB1* as an example to show how to get the haplotypes. First, we use *prep_phased* to load the phased genotype file (phased_geno = "orse_chr1_phased.vcf.gz") that was generated by Beagle, and set the "convert_letter" argument as "TRUE" to convert the genotype file into the standard format used by HandyCNV (returned as "geno_chr1"). Second, we use *closer_snp* to extract the gene's position (returned as "GABPB1_pos") from the SNP map file, which requires the SNP map file (provided using the "phased_input" argument), and to assign the gene's physical position we got from reference gene list to the "chr," "start," and "end" arguments, respectively. Finally, we use *get_haplotype* to get the haplotype information (see **Figures 7I,J**) for the *GABPB1* gene by assigning the formatted phased genotype list ("geno_chr1") to the "geno" argument and assigning the gene's position ("GABPB1_pos") to the "pos" argument.

DISCUSSION

Here we present a freely available and open source R package called HandyCNV, which provides a comprehensive set of functions to summarize and visualize the CNVs and run of homozygosity results detected from SNP genotyping data.

Many good software packages have been developed for the detection of CNV and ROH from SNP chip data [such as PennCNV (Wang et al., 2007), CNVPartition (Illumina, 2021), SNP and Variation Suite (Bozeman and Golden Helix, 2020), and Plink (Chang et al., 2015)], and some well-designed tools for CNV-based association analysis [such as CNVRuler (Kim et al., 2012), CNVRanger (da Silva et al., 2019), and CNVassoc (Subirana et al., 2011)]. However, while they do include some basic data summary and visualization functions, they do not contain any features to customize visualization of CNV or ROH results, or to report the haplotype information for target genomic regions. In contrast to these tools, the HandyCNV package is focused on the detailed summarization and custom

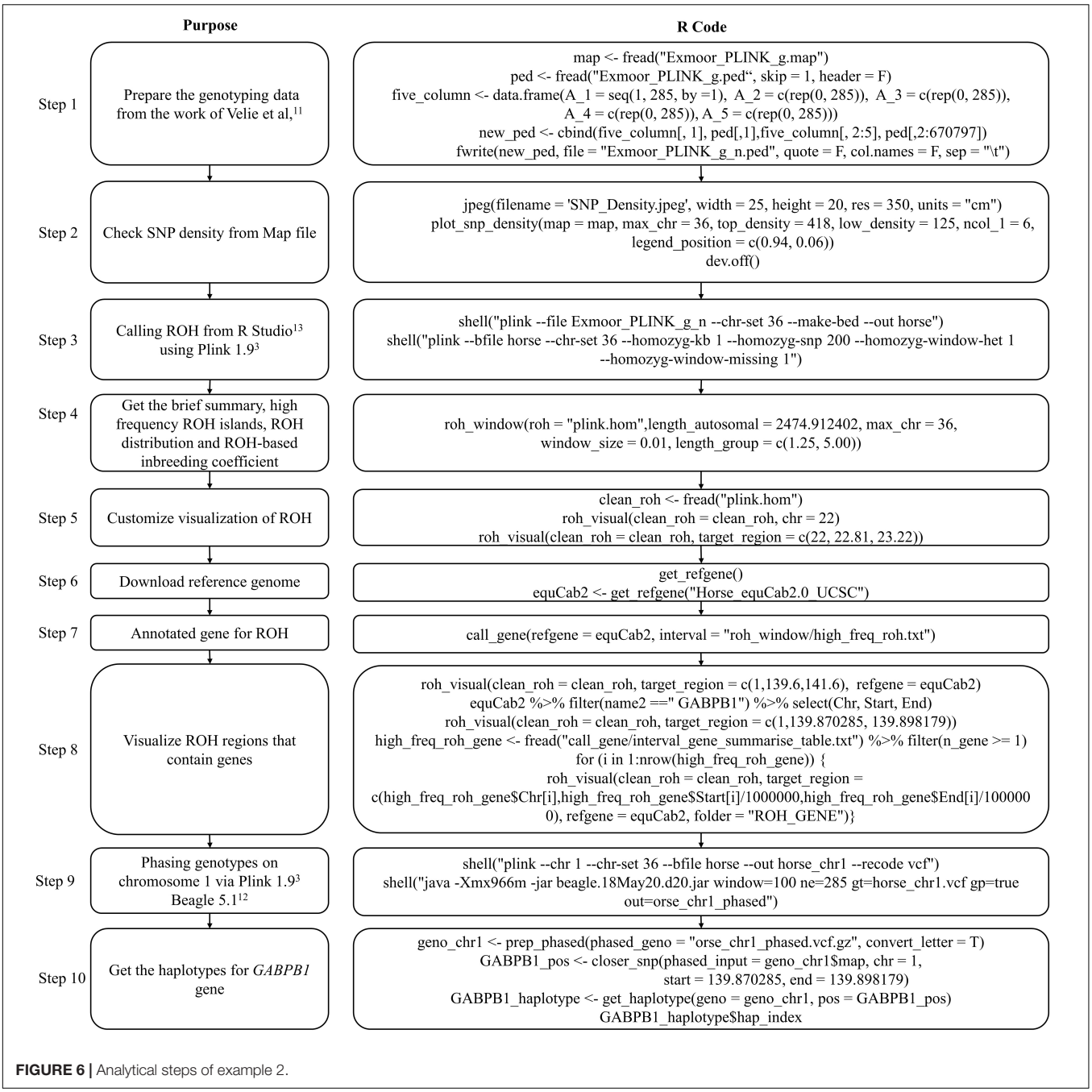

FIGURE 6 | Analytical steps of example 2.

visualization of CNV and ROH results, facilitating tasks such as converting SNP maps, identifying CNVRs from lists of CNVs, genome annotation, comparing and visualizing CNV, CNVR, and ROH, reporting summary results and processing haplotypes of genomic regions of interest. The integration of multiple tasks into a single package provides a standardizable, reproducible and timesaving post-analysis of CNV and ROH, which can help researchers to produce comprehensive tables and figures, and easily identify the samples that contains the genomic regions or genes of most interest for the further validation of experiment designs.

There are some limitations to this package. For example, the *plot_cnvr_panorama* function needs to read genotype data to plot BAF and LRR information: this can require larger amounts of storage. We have tested it on 150 k SNP chip with 2,100 samples on a desktop windows system and it performs well; however, it may not be suitable for higher density chips and very large data sets. The *get_haplotype* function is also limited, as it currently only accepts phased genotypes produced by Beagle 5.1 (Browning et al., 2018) with physical position. In addition, the functions in the conversion section require users provide the target and default map files.

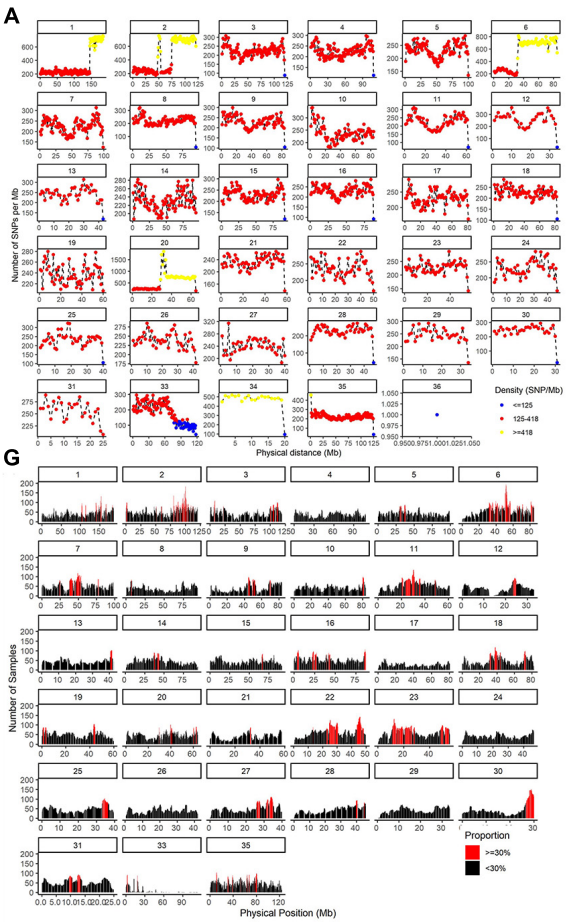

Table 1. Summary of ROH by length group

Length_group	number_roh	Mean	SD	Min	Max	Proportion	N_per_animal
<1Mb	19296	638748	252174	57697	999996	0.313	73.4
1-1.5Mb	16913	1222962	142834	1000012	1499981	0.275	64.3
1.5-2.5Mb	14140	1912588	281786	1500005	2499897	0.23	53.8
2.5-3.5Mb	5544	2928551	286654	2500198	3499414	0.09	21.1
3.5-5Mb	3320	4100763	414280	3500090	4999955	0.054	12.6
>=5Mb	2359	7112022	2383523	5000099	22861009	0.038	9

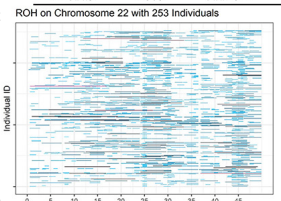

Table 2. High frequency ROH regions

Interval_ID	Chr	Start	End	Length
ROH_1	1	103400000	104390000	990001
ROH_2	1	112780000	113810000	1030001
ROH_3	1	139680000	139730000	50001
…	…	…	…	…
ROH_295	35	76510000	76940000	430001
ROH_296	35	78850000	81170000	2320001

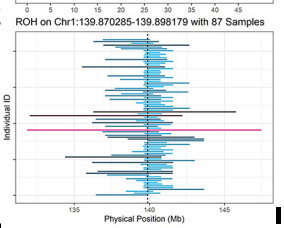

Table 3. Genes located in ROH regions

ID	Chr	Start	End	n_gene	gene_name
ROH_3	1	139600000	142000000	2	*GABPB1,FGF7*
ROH_4	1	146800000	148000000	2	*PLA2G4B,JMJD7*
ROH_13	1	184600000	185000000	1	*PYGL*
…	…	…	…	…	…
ROH_141	30	26400000	30200000	5	*NR5A2,MIR9124,TIMM17A,MYOG,CHIT1*
ROH_143	31	90000000	10800000	1	*THBS2*

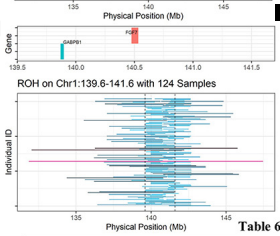

Table 3. Frequency of Haplotypes

Order	Recode - Haplotype	Freq
1	1-1	94
5	1-2	59
6	1-3	57
14	2-2	12
15	2-3	10
21	3-3	9
…	…	…

Table 4. Frequency of Haploid

Order	Haploid	Freq	Re_code
1	CGATTTA	328	1
2	TGATCTA	106	2
3	TGATTTA	89	3
4	TGATTTG	15	4
…	…	…	…

Table 6. Summary of ROH by length group

Sample_ID	n	Total_length	Mean	SD	Min_length	Max_length	Froh_1Mb
ID_1	306	639.83808	2.09	2.11	0.251949	13.260542	0.259
ID_2	217	475.57072	2.19	1.84	0.274626	12.631464	0.192
ID_3	204	345.17248	1.69	1.2	0.267131	6.703111	0.139
…	…	…	…	…	…	…	…
ID_284	49	50.557911	1.03	0.45	0.266325	2.542705	0.02
ID_285	244	316.91682	1.3	0.97	0.124275	6.736802	0.128

FIGURE 7 | The main outputs of example 2. Panel **(A)** is SNP density distribution plot; panel **(B)** is brief summary of ROH by length group; panel **(C)** is plot of ROH on Chromosome 22; panel **(D)** is the high frequency ROH regions list; panel **(E)** is plot of ROHs on Chr1:139.6-141.6 Mb; panel **(F)** is genes annotation list of ROH regions; panel **(G)** is the ROH frequency distribution plot; panel **(H)** is plot of ROHs that overlap to the *GABPB1* gene; panel **(I)** is the frequency of haplotypes on *GABPB1* Gene; panel **(J)** is the frequency of haploids on the *GABPB1* gene; and panel **(K)** is the list of ROHs-based inbreeding coefficient.

SOFTWARE INFORMATION

The current release of HandyCNV is version 1.1.6, which can be installed in the R environment using the following code: "remotes::install_github (repo = 'JH-Zhou/HandyCNV@v.1.1.6')." The current development version can be found at the GitHub repository (github.com/JH-Zhou/HandyCNV).

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: The human CNV lists used in Example 1 can be found in "Table S1 – Detailed information about all CNVs analyzed" at **Supplementary Material section** in Victória Cabral Silveira Monteiro de Godoy's study (doi: 10.1590/1678-4685-GMB-2019-0218). The genotype data used in Example 2 can be found in Brandon D. Velie's study which was public available via Figshare (doi: 10.6084/m9.figshare.3145759).

ETHICS STATEMENT

Ethical review and approval was not required for the study on human participants in accordance with the local legislation and institutional requirements. Written informed consent for participation was not required for this study in accordance with the national legislation and the institutional requirements. Ethical review and approval was not required for the animal study because no animal sampling, experiments or phenotype measurement applied in this study. The genotype data used in this analysis are from previous studies.

AUTHOR CONTRIBUTIONS

JZ conceived the analysis, compiled the package, and wrote the manuscript. LL contributed to code writing and testing, and reviewed the manuscript. TL contributed to package testing, proofreading of the manuscript, and vignette. DG and YS provided instruction for analysis, reviewed the manuscript,

manual, and vignette. All authors contributed to the article and approved the submitted version.

FUNDING

JZ was funded by the China Scholarship Council. YS was supported by the China Agricultural Research System of MOF and MARA.

ACKNOWLEDGMENTS

We thank the two reviewers for their valuable comments, which have improved the scalability of the functions and structural integrity of this paper. We also thank BioRxiv for accepting an earlier version of this manuscript as a pre-print, and the Github platform for providing a place to store open source code, which helped to promote our study to more users in the early stage. This package depends on several independently developed R packages, such as the Tidyverse family (Wickham et al., 2019) and data.table (Dowle et al., 2019), et al. We appreciate all related contributors to the open source R language.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2021.731355/full#supplementary-material

REFERENCES

Bozeman, M. T., and Golden Helix, I. (2020). *SNP & Variation Suite TM (Version 8.x)*.

Browning, B. L., Zhou, Y., and Browning, S. R. (2018). A one-penny imputed genome from next-generation reference panels. *Am. J. Hum. Genet.* 103, 338–348. doi: 10.1016/j.ajhg.2018.07.015

Chang, C. C., Chow, C. C., Tellier, L. C., Vattikuti, S., Purcell, S. M., and Lee, J. J. (2015). Second-generation PLINK: rising to the challenge of larger and richer datasets. *Gigascience* 4:7. doi: 10.1186/s13742-015-0047-8

da Silva, V., Ramos, M., Groenen, M., Crooijmans, R., Johansson, A., Regitano, L., et al. (2019). CNVRanger: association analysis of CNVs with gene expression and quantitative phenotypes. *Bioinformatics* 36, 972–973. doi: 10.1093/bioinformatics/btz632

de Godoy, V. C. S. M., Bellucco, F. T., Colovati, M., de Oliveira, H. R. Jr., Moysés-Oliveira, M., and Melaragno, M. I. (2020). Copy number variation (CNV) identification, interpretation, and database from Brazilian patients. *Genet. Mol. Biol.* 43:218. doi: 10.1590/1678-4685-gmb-2019-0218

Dowle, M., Srinivasan, A., Gorecki, J., Chirico, M., Stetsenko, P., Short, T., et al. (2019). *Package 'Data.Table'Extension of 'Data-Frame'. CRAN Repository Version:1.14.0*.

Illumina (2021). *GenomeStudio*. https://www.illumina.com/techniques/microarrays/array-data-analysis-experimental-design/genomestudio.html (accessed June 10, 2021).

Jiao, X., Sherman, B. T., Huang da, W., Stephens, R., Baseler, M. W., Lane, H. C., et al. (2012). DAVID-WS: a stateful web service to facilitate gene/protein list analysis. *Bioinformatics* 28, 1805–1806. doi: 10.1093/bioinformatics/bts251

Jinghang, Z., Liyuan, L., Thomas, L., Dorian, G., and Yuangang, S. (2021). *Vignettes and Manual of HandyCNV*. https://jh-zhou.github.io/HandyCNV/ (accessed September 1, 2021).

Kim, J.-H., Hu, H. J., Yim, S. H., Bae, J. S., Kim, S. Y., and Chung, Y. J. (2012). CNVRuler: a copy number variation-based case–control association analysis tool. *Bioinformatics* 28, 1790–1792. doi: 10.1093/bioinformatics/bts239

McQuillan, R., Leutenegger, A. L., Abdel-Rahman, R., Franklin, C. S., Pericic, M., Barac-Lauc, L., et al. (2008). Runs of homozygosity in european populations. *Am. J. Hum. Genet.* 83, 359–372. doi: 10.1016/j.ajhg.2008.08.007

Navarro Gonzalez, J., Zweig, A. S., Speir, M. L., Schmelter, D., Rosenbloom, K. R., Raney, B. J., et al. (2021). The UCSC genome browser database: 2021 update. *Nucleic Acids Res.* 49, D1046–D1057. doi: 10.1093/nar/gkaa1070

R Core Team (2020). R: A Language and Environment for Statistical Computing. Vienna, Austria.

Redon, R., Ishikawa, S., Fitch, K. R., Feuk, L., Perry, G. H., Andrews, T. D., et al. (2006). Global variation in copy number in the human genome. *Nature* 444, 444–454. doi: 10.1038/nature05329

Subirana, I., Diaz-Uriarte, R., Lucas, G., and Gonzalez, J. R. (2011). CNVassoc: association analysis of CNV data using R. *BMC Med. Genomics* 4:47. doi: 10.1186/1755-8794-4-47

Team, Rs (2021). *RStudio: Integrated Development Environment for R*. Boston, MA: RStudio.

Velie, B. D., Shrestha, M., François, L., Schurink, A., Tesfayonas, Y. G., Stinckens, A., et al. (2016). Using an inbred horse breed in a high density genome-wide scan for genetic risk factors of insect bite hypersensitivity (IBH). *PLoS One* 11:e0152966. doi: 10.1371/journal.pone.0152966

Wang, K., Li, M., Hadley, D., Liu, R., Glessner, J., Grant, S. F., et al. (2007). PennCNV: an integrated hidden Markov model designed for high-resolution copy number variation detection in whole-genome SNP genotyping data. *Genome Res.* 17, 1665–1674. doi: 10.1101/gr.6861907

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L., François, R., et al. (2019). Welcome to the tidyverse. *J. Open Source Softw.* 4:1686. doi: 10.21105/joss.01686

Conflict of Interest: TL is employed by Livestock Improvement Corporation.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers in Genetics

ORIGINAL RESEARCH
published: 28 October 2021
doi: 10.3389/fgene.2021.764170

singlecellVR: Interactive Visualization of Single-Cell Data in Virtual Reality

David F. Stein[1,2†], *Huidong Chen*[3,4,5,6†], *Michael E. Vinyard*[3,4,5,6,7†], *Qian Qin*[3,4,5,6†], *Rebecca D. Combs*[4,8], *Qian Zhang*[3,4,5,6] *and Luca Pinello*[3,4,5,6*]

[1]Graduate School of Biomedical Sciences, Icahn School of Medicine at Mount Sinai, New York, NY, United States, [2]Khoury College of Computer Sciences, Northeastern University, Boston, MA, United States, [3]Molecular Pathology Unit, Massachusetts General Hospital, Charlestown, MA, United States, [4]Center for Cancer Research, Massachusetts General Hospital, Charlestown, MA, United States, [5]Department of Pathology, Harvard Medical School, Boston, MA, United States, [6]Broad Institute of Harvard and MIT, Cambridge, MA, United States, [7]Department of Chemistry and Chemical Biology, Harvard University, Cambridge, MA, United States, [8]Winsor School, Boston, MA, United States

Single-cell assays have transformed our ability to model heterogeneity within cell populations. As these assays have advanced in their ability to measure various aspects of molecular processes in cells, computational methods to analyze and meaningfully visualize such data have required matched innovation. Independently, Virtual Reality (VR) has recently emerged as a powerful technology to dynamically explore complex data and shows promise for adaptation to challenges in single-cell data visualization. However, adopting VR for single-cell data visualization has thus far been hindered by expensive prerequisite hardware or advanced data preprocessing skills. To address current shortcomings, we present *singlecellVR*, a user-friendly web application for visualizing single-cell data, designed for cheap and easily available virtual reality hardware (e.g., Google Cardboard, ~$8). *singlecellVR* can visualize data from a variety of sequencing-based technologies including transcriptomic, epigenomic, and proteomic data as well as combinations thereof. Analysis modalities supported include approaches to clustering as well as trajectory inference and visualization of dynamical changes discovered through modelling RNA velocity. We provide a companion software package, *scvr* to streamline data conversion from the most widely-adopted single-cell analysis tools as well as a growing database of pre-analyzed datasets to which users can contribute.

Keywords: single-cell, scRNA-seq, scATAC-seq, virtual reality, VR, data visualization, clustering, trajectory inference

OPEN ACCESS

Edited by:
Guangchuang Yu,
Southern Medical University, China

Reviewed by:
Dake Zhang,
Beihang University, China
Indu Khatri,
Leiden University Medical Center, Netherlands

****Correspondence:***
Luca Pinello
lpinello@mgh.harvard.edu

†These authors have contributed equally to this work

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

Received: *25 August 2021*
Accepted: *27 September 2021*
Published: *28 October 2021*

Citation:
Stein DF, Chen H, Vinyard ME, Qin Q, Combs RD, Zhang Q and Pinello L (2021) singlecellVR: Interactive Visualization of Single-Cell Data in Virtual Reality. Front. Genet. 12:764170. doi: 10.3389/fgene.2021.764170

1 INTRODUCTION

Characterization of cell type, while once dominated by pathological description, has over the past decade shifted towards a more quantitative and molecular approach. As such, molecular measurements in single cells have emerged as the centerpiece of the current paradigm of mechanistic biological investigation (Trapnell, 2015). Technological advancements have enabled researchers to measure all aspects of the central dogma of molecular biology at the single-cell level (Stuart and Satija, 2019). Single-cell RNA sequencing (scRNA-seq), a technique that profiles the relative expression of genes in individual cells and single-cell Assay for Transposase Accessible Chromatin using sequencing (scATAC-seq), a technique that surveys genome-wide chromatin accessibility are the most well-established and widely-used of these methods (Buenrostro et al., 2015;

Lähnemann et al., 2020). In fact, combined scRNA-seq + scATAC-seq assays are now routine (Perkel, 2021). Additionally, assays to profile DNA methylation (Luo et al., 2018) or protein levels are now maturing and becoming more widely-accessible (Specht et al., 2019; Labib and Kelley, 2020). Most recently, combinations of various data modalities can now routinely be collected in parallel from the same cell (Chen S. et al., 2019; Zhu et al., 2019; Ma et al., 2020; Xing et al., 2020; Swanson et al., 2021).

scRNA-seq experiments generate on the order of millions of sequencing reads that sample the relative expression of approximately 20,000–30,000 transcribed features (e.g., genes) in each cell of the sample. Normalized read counts for each feature can then be compared to discern differences between cells. scATAC-seq samples comprise a larger feature space wherein cells are characterized by the genomic coordinates of chromatin accessible regions and sequence-features derived from these regions (e.g. transcription factor motifs, *k*-mer frequencies, etc.). Initially performed in dozens to hundreds of cells, these experiments are now performed on the order of millions of cells. With a high dimensional feature space as a result of thousands of features being considered for each cell and large (in cell number) experiments, analysis methods for this data have been required to advance concurrently with the development of these technologies (Chen et al., 2019c; Tian et al., 2019).

With the exception of proof-of-concept methods still too nascent to be widely applied (Chen et al., 2021), omics measurements of single-cells are generally destructive, preventing measurement of a cell at more than a single time point. As a result, most single-cell measurements for studying dynamic processes are of a "snapshot" nature, imposing inherent limitations on the study of such processes from this data (Weinreb et al., 2018). In light of this, transcription rates can be informative of ongoing processes in cells. The recent advent of *RNA velocity* quantifies and models the ratios of spliced and unspliced RNA (mRNA and pre-mRNA, respectively) such that they indicate the temporal derivative of gene expression patterns and thereby reflect dynamic cellular processes, allowing predictions of past and future cell states (La Manno et al., 2018).

Among others, PCA, t-SNE, and UMAP are dimensional reduction methods that have become common choices for enabling the visualization of high-dimensional single-cell datasets. Dimensionally reduced datasets are plotted such that similar cells cluster together and those with highly differing features are likewise clustered apart. In addition to the visualization and clustering of cells, trajectory inference methods have been proposed to learn a latent topological structure to reconstruct the putative time-ordering (*pseudotime*) by which cells may progress along a dynamic biological process (Saelens et al., 2018). As single-cell technologies have advanced, techniques to cluster and organize cells based on single-cell assays have advanced alongside them, allowing key insights toward cell type and state characterization. Combined with RNA velocity information, trajectory inference can offer key insights on dynamical changes to cell states. Once in press however, representation of these dimensionally-reduced visualizations is limited to just two or three dimensions. Even using three-dimensional plots from published studies, one cannot dynamically adjust or rotate the visualization to better understand the data from another angle. In addition, cells are typically annotated by features (e.g. time points, cell type or clusters) to investigate stratification along an axis of some biological process. To change the annotations presented in publication, one must often reprocess the raw data, which is time- and skill-intensive, highlighting the need for more dynamical visualization tools. While such current data representations are often limited and static, single-cell omic datasets are information-rich and, in many cases, important biological heterogeneity cannot be easily investigated or visualized outside the scope of the original publication, without spending considerable cost and time to reanalyze the datasets from scratch.

VR visualization methods for single cell data have been recently proposed (Yang et al., 2018; Legetth et al., 2019; Bressan et al., 2021). However, these methods require either expensive hardware or specific data inputs that mandate intermediate to advanced computational skills. Thus, tools and clear protocols are required to enable researchers, especially those who are not able to efficiently reprocess the raw data, to explore the richness of published datasets (or their own unpublished data) through a simple, easy and affordable VR platform. Importantly, this platform must be flexible enough to accept all types of omics data from established and emerging technologies and processing tools currently employed by the single-cell community.

At the time of this writing, three non-peer-reviewed methods employing VR technology that produce two- and three-dimensional visualizations of single-cell data have recently been reported. *CellexalVR* enables the visualization of standard scRNA-seq data though requires users to preprocess their data through scripting (Legetth et al., 2019). Unfortunately, this tool also requires expensive and dedicated VR hardware to operate. Another recent method for visualizing single-cell data in VR is *Theia* (Bressan et al., 2021), which has been designed with a focus on the exploration of spatial datasets for both RNA and protein measurements. Similar to *CellexalVR*, expensive computing power and VR hardware required to use *Theia* creates a barrier to entry. An alternative to these high-performance methods for VR visualization of single-cell data is *starmap* (Yang et al., 2018), which allows the use of inexpensive cardboard visor hardware. However, *starmap* lacks the advanced portability of outputs from commonly-used scRNA-seq analysis tools and limits cell annotation to clustering results of transcriptomic data. Of note, there are currently no peer-reviewed tools available for the visualization of single-cell data in VR illustrating the novelty in this area of research. To overcome the limitations of these existing methods as well as build on their qualities and initial progress, we present *singlecellVR*, an interactive web application, which implements a flexible, innovative visualization for various modalities of single-cell data built on VR technology. *singlecellVR* supports clustering, trajectory inference and abstract graph analysis for transcriptomic as well as epigenomic and proteomic single cell data. Importantly, *singlecellVR* supports visualization of cell

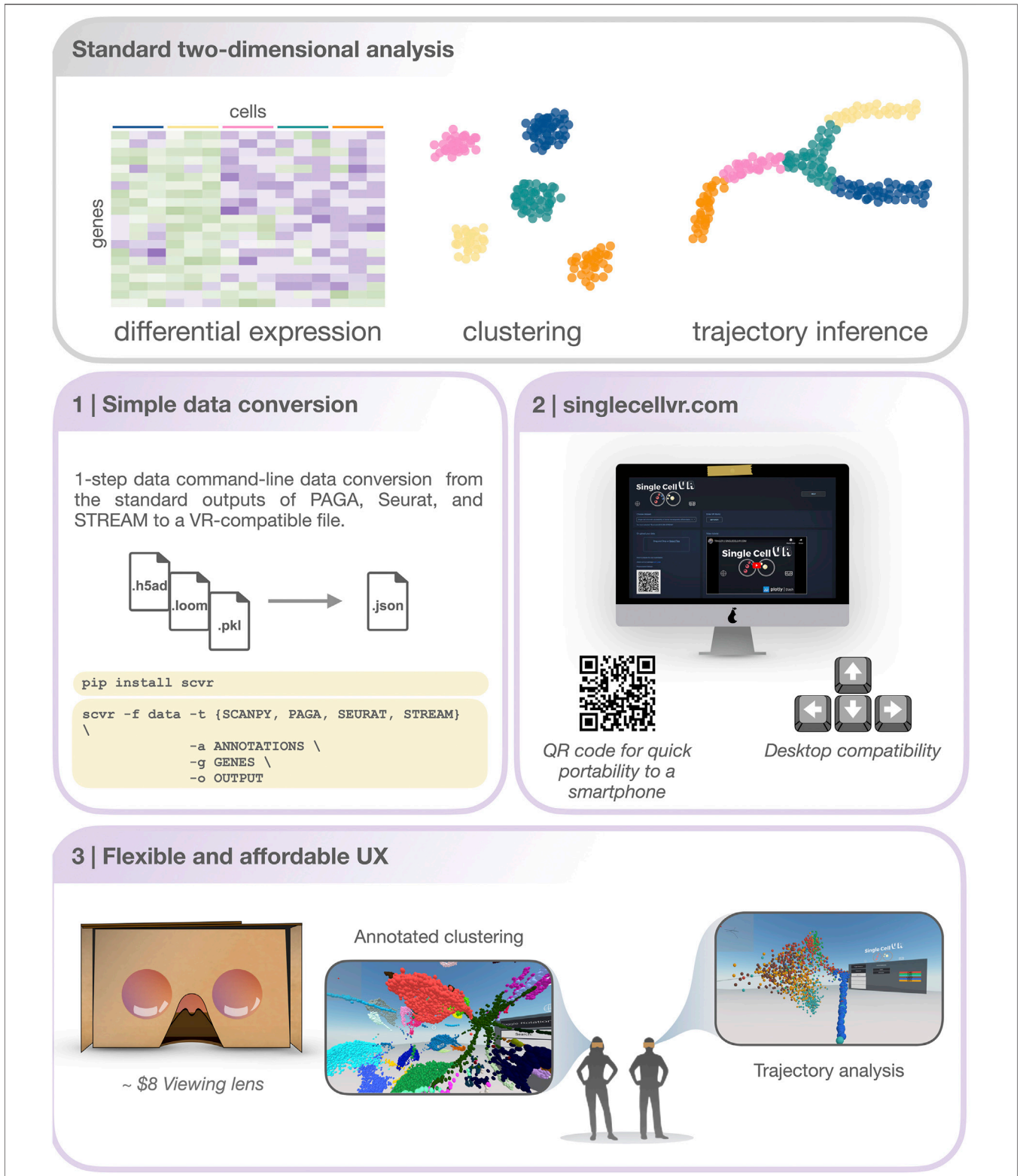

FIGURE 1 | An overview of the *singlecellVR* user experience. Top, grey: The outputs of a standard 2-dimensional scRNA-seq analysis. Middle and bottom, purple: a step-by-step overview of the *singlecellVR* workflow: 1 Schematic of *flexible* data conversion. One command to install (*via* the Python pip package manager) and one command to convert the data to be VR-compatible. 2.Webpage for uploading and exploring VR data. 3 VR mode visualization using a cheap smartphone enabled headset.

dynamics as described by RNA velocity, a recent milestone in the sequence-based analysis of single cells (La Manno et al., 2018; Bergen et al., 2020). *singlecellVR* is a browser-contained, free, and open-access tool. Notably, we have developed a one-command conversion tool, *scvr* to directly prepare the results of commonly-used single-cell analysis tools for visualization using *singlecellVR*.

2 RESULTS

2.1 SinglecellVR User Experience and Overview

SinglecellVR is an easy-to-use web platform and database that can be operated from inexpensive, cardboard visor hardware that costs as little as ~$8 and is available online from popular vendors including Google and Amazon. The webpage, available at http://www.singlecellvr.com enables users to explore several preloaded datasets or upload their own datasets for VR visualization. Visualization can be done either on a personal computer or smartphone. To facilitate the transition between the personal computer browser view and the phone-enabled VR visor (VR mode), we have implemented an easy way to transition between these two visualizations as described in the next sections. In VR mode an interactive visualization is presented to the user, allowing them to manipulate and visualize single-cell data using an array of annotations through the cardboard visor. Additionally, *singlecellVR* features the ability to receive as inputs, the standard output files of commonly-used tools for standard single-cell analysis: *Seurat* (Hao et al., 2021), *Scanpy (along with EpiScanpy)* (Wolf et al., 2018; Danese et al., 2019), *STREAM* (Chen et al., 2019a), *PAGA* (Wolf et al., 2019), and *scVelo* (Bergen et al., 2020). A companion package, *scvr* enables the conversion of these standard outputs to VR-compatible objects in a single command.

In the sections below, we will describe the basis for this VR visualization platform as well as provide descriptive examples of the visualization that can be performed using *singlecellVR*. We will compare *singlecellVR* to existing methods and describe its unique advantages that build on the early progress of single-cell data visualization in VR. We include a detailed protocol and quick-start guide that describe how the web platform enables researchers to explore their own data and dually functions as a database for preformatted datasets that can be explored immediately in VR (**Supplementary Note 1**).

2.2 VR Database and scvr Preprocessing Tool

SinglecellVR provides a growing database of several datasets processed for VR visualization. Initialization and future growth of this database is enabled, scale-free through the streamlined *scvr* utility. As shown in **Figure 1**, to use *singlecellVR*, the user may select a precomputed dataset or convert their data from commonly used single-cell workflows. This conversion can be easily accomplished by using *scvr*, a simple one-line command tool for performing data conversion and produces a simple zipped .json file with all the information required for visualizing cells and their annotations in VR. Additionally, datasets for which RNA velocity information has been calculated may be submitted directly for visualization of velocity in VR without prior conversion (**Supplementary Note 2**; **Supplementary Notebook 4**).

Conversion from the standard output of any single-cell analysis tool to this format would normally pose a significant methodological roadblock to most users, especially non-computational biologists. To bridge this gap, *scvr* parses and converts the outputs of *Scanpy*, *EpiScanpy*, *Seurat*, *PAGA*, and *STREAM* (respectively *.loom*, *.h5ad* and *.pkl*) and creates the required zipped *.json* file (**Supplementary Note 2**). This file contains the 3-D coordinates of cells in a specified space (e.g. UMAP, LLE, etc.), cell annotations (e.g. FACS-sorting labels, clustering solutions, sampling time or pseudotime, etc.), and feature quantification (gene expression levels, transcription factor deviation, etc.). It also contains the graph structure (the coordinates of nodes and edges) obtained from supported trajectory inference methods. Users interested in visualizing scRNA-seq dynamics using RNA velocity generated from spliced and unspliced read counts can likewise prepare this information for visualization in *singlecellVR* using the *scvr* companion utility. Users can follow established workflows for obtaining these insights from the raw read file inputs as well as make use of the tutorials available at the *singlecellVR* GitHub Repository (**Section 5**; **Supplementary Notebook 4**).

Importantly, *scvr* has been made available as a Python pip package to streamline its installation and can convert a processed dataset for VR visualization with a simple command. To install, one can simply open their command line utility and run: "pip install scvr." Once installation is completed, the user can navigate to https://github.com/pinellolab/singlecellvr, to copy and customize the example commands provided to execute the one-step process for converting their data to a VR-compatible format. In addition to the documentation of *scvr* we have filmed a short video tutorial found on the homepage of *singlecellVR* to further assist less experienced users in preparing their data for visualization.

To showcase the functionality and generalizability of *scvr* across data types, we have preprocessed a collection of 17 published datasets, which includes both scRNA-seq as well as scATAC-seq and single cell proteomic data and made them available for immediate VR visualization. Taken together we believe this step addresses a key limitation of previously-developed VR tools mentioned above, and a formal comparison is presented in **Section 2.5** (Yang et al., 2018; Legetth et al., 2019; Bressan et al., 2021).

Excitingly, given the small footprint of the files obtained with *scvr*, we are offering users the ability to easily submit their processed data to the *singlecellVR* GitHub Repository (see **Supplementary Figure S1**) to make the tool a general resource for the field. In this way, we hope to even further extend the ability of biologists to visualize once static datasets and easily generate new hypotheses through manipulation of a large number of rich datasets. Therefore, we envision that our website will function as a repository for VR visualization data of single cell biological annotations.

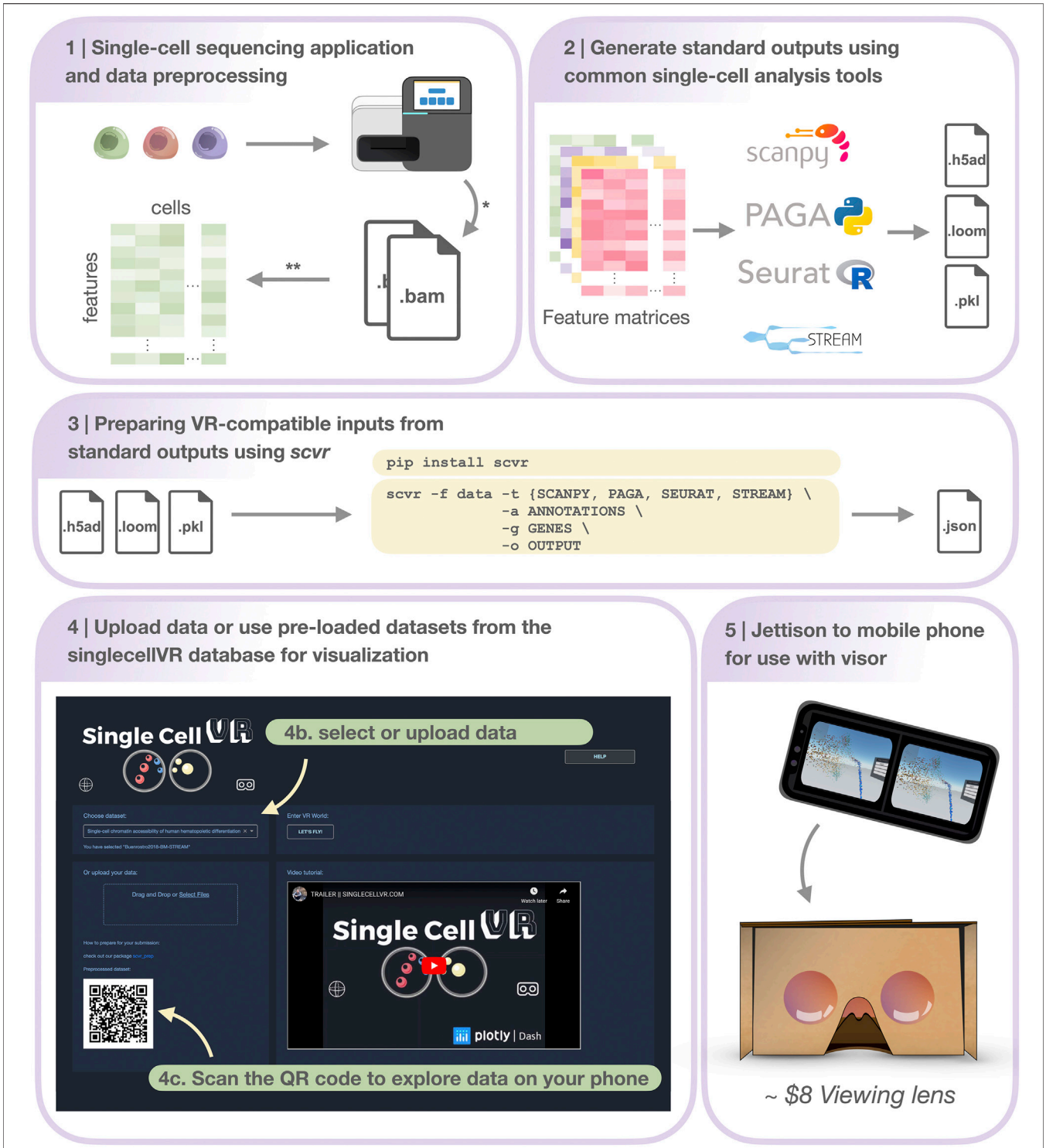

FIGURE 2 | Step-by-step protocol for data processing and using *singlecellVR*. Step 1 Single-cell data can be generated using a variety of technologies or downloaded from online repositories*. Data can then be preprocessed and prepared for use (most often as a feature matrix) with common single-cell analysis tools**. Step 2 Pre-processed data can be analyzed using common single-cell analysis tools (listed here). Step 3 Users can process their data for use with *singlecellVR* from any of the standard outputs created by analysis tools listed in ***Step 2***. listed at the top in a single command. Step 4 Users can select from pre-processed data or upload their own data (Step 4b) and scan the dynamically generated QR code with their phone to begin the VR visualization (***Step 4c***). Step 5 Users can use the QR code on the website to transfer their data to their phone for use with simple hardware. * and ** are explained in the **Section 5** section, Step 1.

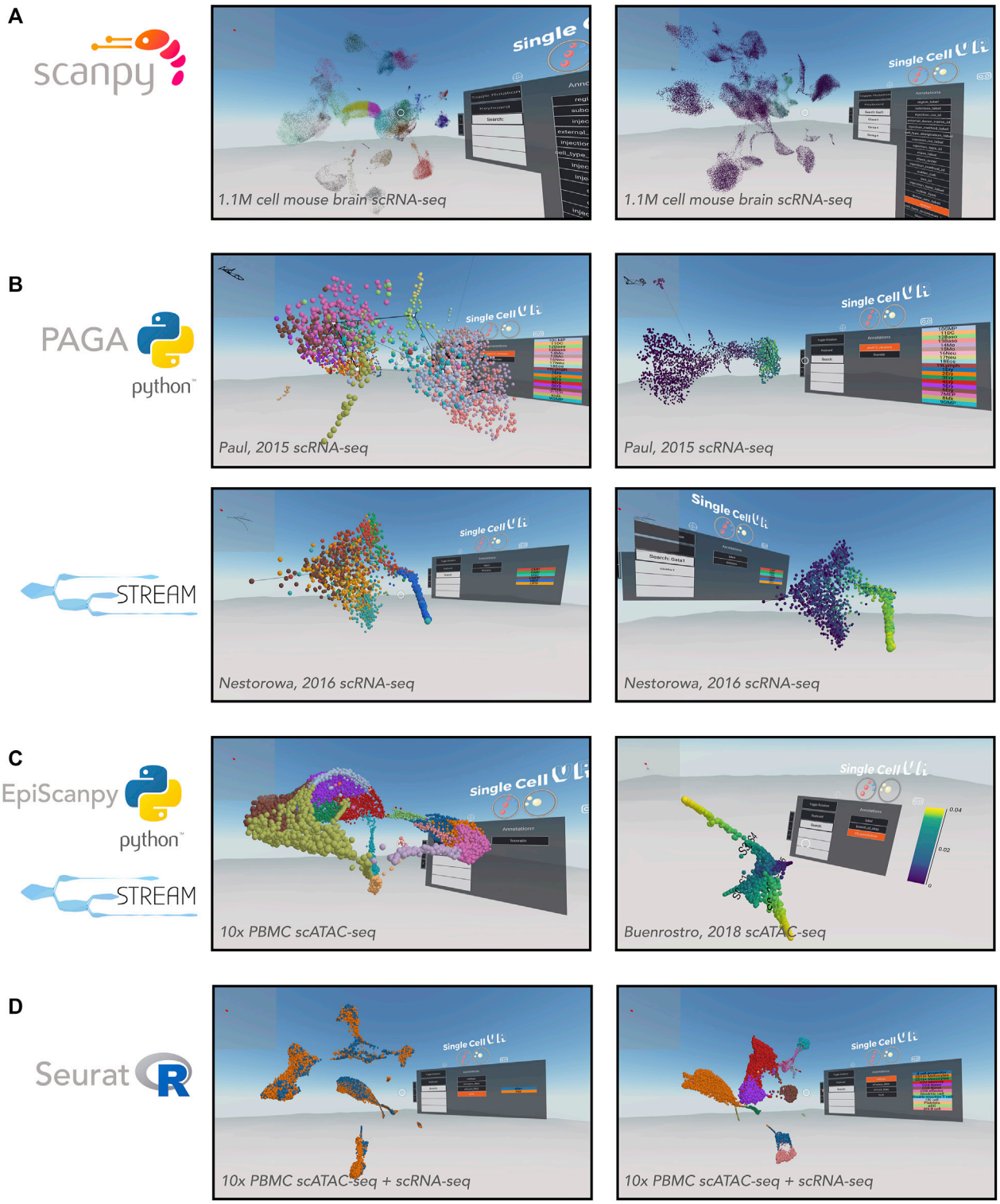

FIGURE 3 | VR visualization of single-cell processed datasets profiled by different technologies and analyzed by various computational tools. **(A)** Scanpy offers solutions for clustering single-cell data. Shown is a UMAP of the Allen Brain Atlas mouse brain scRNA-seq dataset from Yao (2020) and processed by Scanpy. Leiden clustering solution **(left)** and expression of *Gad1* **(right)**. **(B)** Trajectory inference applications. *PAGA* offers a partition-based graph abstraction to uncover potential trajectories (edges) between group of cells (nodes) (top-left) relative gene expression (e.g., *Klf1*, *top-right*), amongst other annotations. The *PAGA*-analyzed dataset shown here is from Paul, et al. (2015). *STREAM* offers the visualization of developmental trajectories, which can be visualized by cell identity **(bottom-left)** or by relative gene expression (e.g., *Gata1*, *bottom-right*), amongst other annotations. The STREAM-analyzed dataset shown here is from Nestorowa, et al. (2016). **(C)** Epigenomic

(*Continued*)

FIGURE 3 | applications. *EpiScanpy* enables the clustering and visualization of scATAC-seq data **(left)**. PBMC (healthy donor) 10,000 cells dataset analyzed by *EpiScanpy* and with colors corresponding to clustering solutions (Louvain clustering). *STREAM* was used to perform trajectory inference on th scATAC-seq dataset Buenrostro et al. (2018) **(right)**. **(D)** Seurat offers solutions for clustering single-cell data as well as integrating datasets across experiments. Shown is a Seurat-integrated scRNA-seq and scATAC-seq PBMC dataset from 10x Genomics, colored by technology **(left)** and cell type **(right)**.

2.3 A Simple, Cloud-Based Web Tool for VR Visualization

SinglecellVR is available as a webapp at http://www.singlecellvr.com. This website enables users to explore several preloaded datasets or upload their own datasets for VR visualization. To build *singlecellVR* we have adopted recent web technologies, *Dash* by *Plotly* and *A-FRAME*, a recently-developed JavaScript framework for VR/AR. This allowed us to create a tool that is portable and does not require any installation. The website can be reached through any web browser and browser compatibility was tested against Google Chrome, Apple Safari, and Mozilla Firefox. Visualization can be done either on a personal computer or smartphone (both Android and Apple smartphones).

Once the users have uploaded their data to *singlecellVR*, they have the option to view and explore the data in 3-D directly in their web browser or to quickly jettison the data to their mobile device for visualization in a VR headset (**Figure 2** and **Supplementary Figure S2**). A key challenge associated with developing a method for visualization of single-cell data is transporting data that is typically processed in desktop settings to the smartphone-based VR visualization. In fact, we predict that in most cases, users will prefer to upload their data through a computer in which they may have run their analyses. To overcome this challenge and enable a seamless transition to a smartphone for VR view, our website dynamically generates a QR code that enables users to open the VR view on their phone to view data uploaded through a personal computer. This mixed approach is particularly useful because, as mentioned before, most users are not processing single-cell data analysis from a phone nor would they keep the files on a mobile device.

2.4 Supported Tools and Analysis

2.4.1 Visualizing Single-Cell Clustering Solutions in VR

As previously mentioned, *Scanpy* and *Seurat* are two commonly-used tools for performing cell clustering as well as differential expression analysis. Here we demonstrate the utility of *singlecellVR* to visualize the common outputs of these tools, showcasing both the clustering solutions as well as differentially expressed genes or other technical or biological features that are visualized easily through the VR interface (**Figure 3**). A key advantage of our tool is the ability to supply multiple annotations to cells to visualize various attributes of the measured data, for example based on a biological query of interest or experimental design. This may include stratification by cluster identity, time points, tissues, or FACS-based labels. In **Figure 3**, we demonstrate the ability to select visualizations by various cluster identifications, which are user-customizable. With the advent of cross-experiment integration methods that can integrate not only multiple scRNA-seq experiments but experiments across modalities of single-cell data collection, this flexible labelling strategy should enable the user in the future to visualize even the most novel and complicated experiments in rich detail.

In addition to flexibility for visualizing complex experimental setups, *singlecellVR* is able to visualize large experiments. To demonstrate this utility, we first processed (using *Scanpy* and *scvr*) and visualized on *singlecellVR*, scRNA-seq data from the Chan-Zuckerberg Biohub *Tabula Muris* project, a dataset consisting of 44,949 cells and 20 tissues from seven mice (Schaum et al., 2018). In **Supplementary Figure S3A**, clustering analyses of this dataset are projected into VR, colored by mouse tissue (left) and Louvain cluster identity (right). With a quick rendering time (<1 s) for the *Tabula Muris* dataset, we next explored the realm of visualization for a modern, large atlas-scale dataset (>1 M cells). Using *Scanpy* and *scvr*, we successfully processed and visualized on our website, cells from the Allen Brain Institute that capture cortical and hippocampal development inside the mouse brain (**Figure 3A**) (Yao, 2020). This dataset consists of 1,093,785 cells and is among the largest scRNA-seq datasets created, to date. Visualization of this dataset in a dynamic VR setting creates the opportunity for more in-depth study of sub-sections of the data, which is particularly valuable for such a large dataset. While the datasets visualized in this manuscript were obtained in their pre-processed state, we have created IPython notebook tutorials for integrating datasets from multiple transcriptomic experiments as is performed in Seurat; these may be accessed in the associated GitHub repository.

2.4.2 Visualizing Single-Cell Trajectory Inference Results in VR

Single-cell measurements are particularly useful for capturing cross-section snapshots of a biological process. With dense cell sampling, one can often observe transient cell states that exist between two, more stable states. However, without an intrinsic understanding of the process being studied, it may be difficult to order these cells along a time axis of a biological process. To enable ordering cells by transcriptional (or epigenomic) states, pseudotemporal ordering, based on trajectory inference and machine learning algorithms has become a useful technique for the single-cell field. Trajectory inference, like clustering, describes a high-dimensional biological process and being limited to a two/three-dimensional static visualization on paper, with a limited selection of genes or annotations is not ideal. Thus, we intend for our tool to leverage the richness of these datasets and make their general usefulness to the field more widespread. We therefore wanted to extend our VR visualization to the results of common trajectory inference tools (**Figure 3B**). *singlecellVR* supports two trajectory inference tools: *PAGA*, a partition-based graph abstraction trajectory inference method and *STREAM,* a method based on principal graphs (Albergante

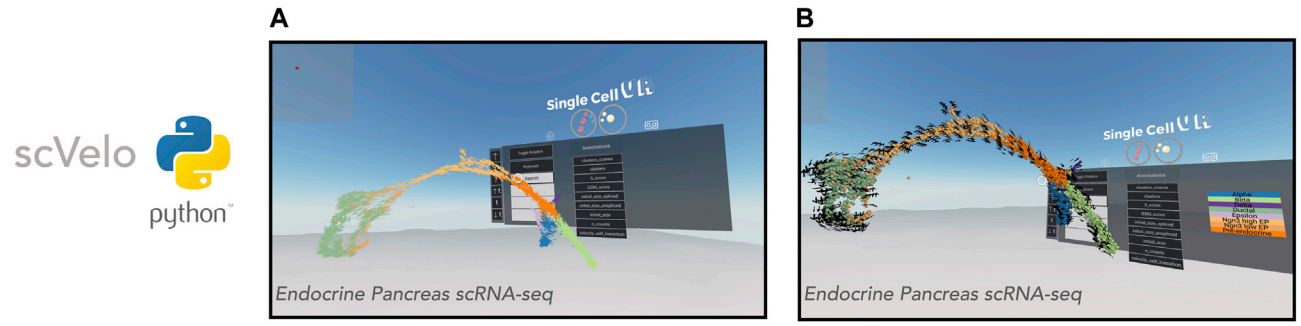

FIGURE 4 | VR visualization of single-cell datasets with RNA Velocity. *scVelo* enables efficient analysis of the RNA velocity attributes of single-cell data. Shown is a 3-D UMAP of an endocrine pancreas dataset (Bastidas-Ponce et al., 2019). **(A)**: Cells are displayed as their corresponding 3-D velocity vectors and colored according to cluster annotation. **(B)**: Cells are displayed as 3-D orbs surrounded by a corresponding grid of velocity vectors. Cells are colored according to cluster annotation.

et al., 2020) that recovers a tree like structure to summarize developmental trajectories and to visualize the relative densities of cell populations along each branch.

To showcase the ability of *singlecellVR* to visualize trajectory inference results, we reprocessed a popular myeloid and erythroid differentiation dataset (Paul et al., 2015), performing trajectory inference using *PAGA*. *PAGA* is designed specifically to preserve relative cell topology in constructing the trajectory along a pseudotime axis. In the depiction of the *PAGA*-generated trajectory, nodes (gray) correspond to cell groups, and edges (black lines between nodes) connecting the groups quantify their connectivity and confidence (thickness) (**Figure 3B**, top). To showcase the VR output of *STREAM* we reprocessed a popular mouse blood dataset (Nestorowa et al., 2016). In STREAM, a set of smooth curves, termed *principal graph*, are fitted to the data and each curve represents a developmental branch. Within *singlecellVR*, we are able to easily explore these trajectories and observe qualitatively, the distribution of cells along each branch in the UMAP space (**Figure 3B**, bottom). The branches of these trajectories are represented by the curves that cut through the cells.

SinglecellVR and *scvr* also support processing and visualizing single-cell epigenomic data. To demonstrate this functionality, we first used the *EpiScanpy* workflow to cluster a scATAC-seq dataset from 10x Genomics containing 10,000 cell PBMC (healthy donor) (**Figure 3C**, left). Next we reprocessed with STREAM a scATAC-seq dataset profiling human hematopoiesis (Buenrostro et al., 2018) (**Figure 3C**, right). In addition, we extend *singlecellVR* to single-cell quantitative proteomics data. To this end we reprocessed data from *SCoPE2*, a recent assay to quantitate proteins in single cells using mass spectrometry (Specht et al., 2019). We performed trajectory inference using *STREAM* on one *SCoPE2* dataset profiling the transition from monocytes to macrophages in the absence of polarizing cytokines. Our analysis revealed a bifurcated branch structure as cells progress towards macrophage phenotypes (**Supplementary Figure S3B**). Importantly, such bifurcation is not readily visualized in previous reports in two dimensions. Finally, we took advantage of the recent advances in the multi-omics field, using *Seurat* to integrate and co-embed PBMC cells profiled by scRNA-seq and scATAC-seq by *10x Genomics* (**Figure 3D**).

2.4.3 Visualizing RNA Velocity Analysis in VR

Having successfully applied the *singlecellVR* framework to the visualization of trajectory inference analyses for multiple modalities of single-cell data, we sought to extend the framework further to visualize dynamical changes at single-cell resolution by way of RNA velocity. We first demonstrated this on a popular endocrine pancreas dataset (**Figure 4**), which has been previously employed to demonstrate the utility of visualizing dynamic processes using velocity.

Visualization of RNA velocity using *singlecellVR* has two modes. In the default mode, each cell is represented by an arrow where the magnitude and direction of the arrow denote the velocity of that cell (**Figure 4A**). For larger datasets, cells may be represented as spheres while a surrounding grid system of arrows denotes the predicted trajectory of a given cell (**Figure 4B**). This is particularly helpful for interpreting the overall direction of cells in various clustering regions or subsets of a given trajectory. In either mode, the arrows are animated to gravitate towards the direction of the corresponding cell trajectory. Latent time, t is a parameter of the velocity calculation for a given cell. To aid in user comprehension of observed velocity, the speed and distance of the animated velocity vector may be calibrated on the fly during the VR experience through adjustment of the t parameter using the floating VR assistance menu. These results taken together with the visualizations of clustering analyses as well as trajectory inference analyses indicate that *singlecellVR* is a robust, generalizable tool across multiple modalities of single-cell analysis.

2.4.4 Creating Reproducible Visualizations

To enable *singlecellVR* users to create visualizations that can be reproduced upon sharing, we have included a feature in the VR interface, which captures absolute x, y, and z coordinates such that one may navigate to an identical position with precision. In line with this, we have also included pitch, yaw, and roll descriptions of the camera angle view. These position descriptions of the VR viewpoint can be captured and shared as part of the visualization. Viewpoint descriptions may also be toggled on or off (**Supplementary Figure S4**).

2.5 Comparison of singlecellVR to Existing Methods

As mentioned above, there are currently three unpublished reports of VR tools created to visualize single-cell data: *CellexalVR* (Legetth et al., 2019), *starmap* (Yang et al., 2018), and *Theia* (Bressan et al., 2021). In this section, we compare these tools to *singlecellVR* on two axes: 1) ease of use and 2) overall performance for visualization and analysis in VR.

2.5.1 Ease of Use

Competing Tools

Both *CellexalVR* and *Theia* require or recommend *HTC Vive* or *HTC Vive Pro* VR hardware (~$500–1,000), an *Intel Core i7 processor* (~$300) or better, *an NVIDIA GTX1080* or *NVIDIA GeForce RTX 3080/3090 (~$1,500–3,000), 16–32 GB RAM (~$50–150) and a solid-state hard drive (SSD)* (*1 TB SSD recommended for CellexalVR*) (~$50–100). Altogether, this equipment requires a minimum investment of roughly $,2300–$4,550. These are computational equipment that most biologists will not have at their disposal within their lab, likely limiting use of this tool to more computationally-focused labs.

CellexalVR requires software and boilerplate-level to pre-process the data in preparation for VR visualization is required and therefore requires the user to perform scripting to prepare data for downstream use with the VR visualization. While *Theia* has provided a convenient python script to convert *AnnData* objects, their software is not open-source, hindering further community contribution.

A contrasting alternative to *CellexalVR* and *Theia* is *Starmap*, which is compatible with low-cost hardware such as *Google Cardboard*. However, *Starmap* takes as input comma-separated values containing information of the three-dimensional coordinates of cells in the visualization as well as annotations (e.g., cluster ID), and up to 12 features per cell. This file must be prepared entirely by the user without assistance from the *Starmap* platform, limiting the audience of this tool to experienced computational biologists.

singlecellVR

The single-command companion package for data preparation, *scvr* described above enables users to visualize their own precomputed data directly from the outputs of commonly-used single-cell RNA-seq analysis tools. Currently supported tools include *Scanpy*, *EpiScanpy, Seurat*, *PAGA*, *STREAM, and scVelo*. *singlecellVR* is the only tool of the three discussed (*CellexalVR*, *Theia*, and *Starmap*) that features a QR code to quickly transport the VR data visualization to another device.

2.5.2 VR Performance and Analysis Capabilities

Competing Tools

CellexalVR proposes a versatile, user-friendly visualization for standard scRNA-seq workflow outputs and demonstrates comparable utility on scATAC-seq data. *Theia* offers a similarly high-performance visualization of single-cell data. *Theia's* key distinguishing contribution is it's visualization of spatial transcriptomic single-cell datasets.

Starmap is only demonstrated on scRNA-seq data and lacks the ability to visualize analyses beyond clustering (such as trajectory inference or an illustration of velocity). Further, *Starmap* is only capable of displaying up to 12 features for a given cell, limiting the throughput with which users may analyze their data.

singlecellVR

In contrast to existing methods, *singlecellVR* offers both a high-performance visualization with in-depth analysis and the ability to visualize all modalities of data at scale, while at the same time offering a software that is compatible with low-cost hardware and requires minimal computational abilities. These advances, which build on the progress made by these initial methods create a tool, which offers a low-cost alternative to existing tools with virtually zero barrier to entry, while maintaining high-performance VR visualizations.

3 DISCUSSION

The amount of publicly available scRNA-seq data has exploded in recent years. With new assays to capture chromatin accessibility, DNA methylation and protein levels in single cells, we predict a second wave of dataset generation. Each of these datasets is extremely high-dimensional and thus, rich with latent information about a given biological sample. Ideally, biologists would be able to explore this treasure-trove of data from any angle and make hypotheses assisted by *in silico* analysis at little to no time cost. Often however, experimental biologists lack the advanced computational skills and/or time required to reprocess and reanalyze raw data from published experiments to gain an understanding of the data from their desired angle of interest. Additionally, biologists who wish to thoroughly explore data prior to publication may rely on a computational specialist who is less connected to the biological problem of interest, introducing a disconnect in hypothesis-driven experimental turnover.

While once primarily reserved for entertainment, VR has found utility in both industrial and academic applications. In this manuscript we present a protocol for visualizing single-cell data in VR. This protocol is based on *singlecellVR, a* VR-based visualization platform for single cell data and discusses its innovations and differences with existing methods. Importantly, we provide a simple mechanism to prepare results from commonly-used single-cell analysis tools for VR visualization with a single command to considerably increase accessibility (see **Section 5**). With this added utility, we seek to empower non-computational biologists to explore their data and employ rapid hypothesis testing that could not be made from the traditional static representations typical of communication in a scientific report on paper or a computer screen.

We anticipate that VR will become increasingly useful as a research and education tool and that the construction of software libraries will aid such advancements. VR has also recently found application in other sources of biological data, including single-neuron morphological imaging data (Wang et al., 2019), three-dimensional confocal microscopy data for fluorescent molecule localization (i.e., fluorophore-tagged

proteins) within cells (Stefani et al., 2018), and three-dimensional single-molecule localization super-resolution microscopy (Spark et al., 2020). Our scalable and flexible VR visualization framework is not limited to scRNA-seq and it can be also easily adapted to other single-cell assays and tools that already support epigenomic data and/or single-cell proteomic data (*EpiScanpy* (Danese et al., 2019), *Seurat* (Stuart et al., 2019), and *STREAM* (Chen et al., 2019a)). Finally, we extend our framework to computational methods that derive the RNA velocity of single cells for visualization in VR (La Manno et al., 2018; Bergen et al., 2020). With the recent advances in spatially-resolved transcriptomics (Welch et al., 2019) and corresponding analysis methods (Hao et al., 2021; Miller et al., 2021), visualization of such data has already been extended to a VR framework (Bressan et al., 2021). We believe this new sort of three-dimensional VR will also become especially useful once made available to the general research community via inexpensive hardware and facile data preprocessing and preparation for VR visualization. As software to analyze single cells reach their maturity, one could imagine the incorporation of such visualizations into more clinically translatable settings, such as medical devices.

4 CONCLUSION

This manuscript presents *singlecellVR, a scalable* web platform for the VR visualization of single-cell data and its associated preprocessing software, *scvr,* which streamlines the results of commonly used single-cell workflows for visualization in VR. *singlecellVR* enables any researcher to easily visualize single-cell data in VR. The platform is user-friendly, requires no advanced technical skills or dedicated hardware. Importantly, we have curated and preprocessed several recent single-cell datasets from key studies across various modalities of data generation and analysis approaches, providing the scientific community with an important resource from which they may readily explore and extract biological insight.

4.1 Key Points

- *singlecellVR* is a web platform that enables quick and easy visualization of single-cell data in virtual reality. This is highlighted by a database of pre-loaded datasets ready for exploration at a single click or via a QR code to quickly jettison the visualization to a smartphone enabled VR visor.
- *scvr* is a companion package to easily convert standard outputs of common single-cell tools in a single command
- *singlecellVR* is made for use with cheap and easily-available VR hardware such as Google Cardboard (~$8).
- *singlecellVR* can visualize both clustering solutions as well as trajectory inference models of single-cell data for transcriptomic, epigenomic, and proteomic data as well as multi-modally integrated datasets. Additionally, *singlecellVR* offers a three-dimensional VR visualization of RNA velocity dynamics.

5 MATERIALS AND METHODS

5.1 Single-Cell Data Preparation

All datasets were processed using *Scanpy* (version 1.5.1, RRID: SCR_018139), *AnnData* (version 0.7.6, RRID:SCR_018209), *EpiScanpy* (version 0.1.8), *Seurat* (version 3.1.5, RRID: SCR_007322), *PAGA* (part of *Scanpy*, version 1.5.1, RRID: SCR_018139), *STREAM* (version 1.0), and *scVelo* (version 0.2.3, RRID:SCR_018168) following their documentations. Jupyter notebooks to reproduce data processing are available at https://github.com/pinellolab/singlecellvr. Analyses were performed on a 2019 MacBook Pro (2.4 GHz Intel Core i9, 16 GB RAM).

5.2 Preparation of Processed Data for Visualization in VR

The preprocessing package, *scvr* generates a series of .json files containing the spatial coordinates representative of cell embeddings in 3D embedding (e.g. PCA, UMAP, etc.) and information including labels and features (e.g., gene expression, TF motif deviation, etc.). These .json files are zipped upon output from *scvr* into a single file that can be easily uploaded to *singlecellVR* for visualization.

5.3 SinglecellVR Webapp Construction

To build *singlecellVR*, we used *A-FRAME* (version 1.2.0), *Dash* by *Plotly* (version 1.13.3).

DATA AVAILABILITY STATEMENT

The source code and the supporting data for this study are available online on GitHub at https://github.com/pinellolab/singlecellvr. The preprocessing package, scvr is included within that repository https://pypi.org/project/scvr/. The documentation for scvr is available here: https://github.com/pinellolab/singlecellvr. Video tutorials for learning about and running visualization experiments with singlecellVR (and using scvr to prepare the data) are available on YouTube, here: https://www.youtube.com/playlist?list=PLXqLNtGqlbeMaAuiBStnBzUNE6a-ULYx8. All the analyses in this article can be reproduced using the Jupyter notebooks available at https://github.com/pinellolab/singlecellvr. Additionally, we have provided a wiki within the same repository for a more detailed guide to reproducing results from the paper as they pertain to the supplementary materials.

AUTHOR CONTRIBUTIONS

Authors DFS, HC, MEV, and QQ contributed equally to this publication. DFS, HC, MEV, and LP conceived this project and designed the experiment, which was begun at the 2019 HackSeq, at the University of British Columbia where input from collaborators mentioned in the acknowledgements was received. DFS, HC, MEV, and RDC processed the data hosted in the database as well as produced the VR image demonstrations of *singlecellVR* shown in this manuscript. DFS led the

development of the virtual reality framework. HC led the development of Dash-based website and the preprocessing module, scvr. QQ contributed to extending the VR tool with velocity and new API. QZ contributed to extending the VR tool to incorporate single-cell protein analysis. MEV led the preparation of the manuscript. All authors performed user-testing of the software. LP supervised the development of this work and provided guidance. All authors wrote and approved the final manuscript.

FUNDING

This project has been made possible in part by grant number 2019-202669 from the Chan Zuckerberg Foundation. LP is also partially supported by the National Human Genome Research Institute (NHGRI) Career Development Award (R00HG008399) and Genomic Innovator Award (R35HG010717). MEV is supported by the National Cancer Institute (NCI) Ruth L. Kirschstein NRSA Individual Predoctoral Fellowship (1F31CA257625-01).

ACKNOWLEDGMENTS

We would like to acknowledge the organizers and participants of HackSeq19 (University of British Columbia), where this project began. We would like to especially acknowledge those *HackSeq19* participants that contributed to the development of this project: Michelle Crown, Alexander Dungate, David Lin, Terry Lin, and Sepand Dyanatkar Motaghed. Additionally, we would like to thank Ashley Browne, Salma Ibrahim, and Kate McCurley for their contributions to the construction of the *singlecellVR* database through the Winsor Science Internship Program.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2021.764170/full#supplementary-material

Supplementary Figure 1 | Instructions for contributing VR-processed data to the *singlecellVR* data repository. Users can contribute to the growing repository of VR datasets by submitting a pull request to our GitHub repository: https://github.com/pinellolab/singlecellvr. To do so, first fork and clone the repository (steps 1 and 2, above). Next, add your data (step 3). Finally, create a pull request (step 4) to be submitted for approval. Once approved, your data will be incorporated into the growing repository of VR datasets. It is necessary to add the "VR Dataset" flag (purple, already added to the sidebar) to the pull request. In addition, we ask users to describe the data, methods used and available annotations (e.g. genes, timepoints, clusters labels etc.) in the commit message or comment section of the pull request. ***Note****: for velocity results, files >50 MB are too large to be shared through GitHub and must be shared via other channels. However, coordination of this sharing may proceed through GitHub as shown in this figure. For more, see* **Supplementary Note 2** *and* **Supplementary Notebook 4**.

Supplementary Figure 2 | Tips for using the VR interface. **(A)** It is not required but one can easily connect a keyboard with your smartphone using a Bluetooth-enabled keyboard (a small portable keyboard can be purchased from Amazon for ~$10). However, you can still use a normal computer with your browser and explore using your mouse and keyboard, the three-dimensional transcriptional space with cells, trajectories and graph abstractions. The full set of interactive keyboard functionalities are detailed above. **(B)** There are several similar versions of cardboard VR adapters available for ~$8. Many VR headsets such as Google Cardboard have a single button that allows a user to click the screen of their phone while immersed in a virtual reality experience. By holding down this button, users without a keyboard may move forward in the direction of their gaze. You can also simply use your computer screen to do initial exploration of the data in 2-D. **(C)** Users may navigate the VR visualization via a combination of gaze controls and keyboard inputs. A circle, centered in the user's field of view indicates the direction that a user will move through the virtual space and also acts as the appendage through which the user will interact with objects in the visualization. Additionally, users may select the "keyboard" button on the menu to render a virtual keyboard. Cardboard users may use this keyboard to search for available features to render on the display. The "Enter/Return" key on the virtual keyboard clears the current search. Subsequently selecting the "keyboard" button will hide the keyboard from view.

Supplemental Figure 3 | **(A)** Rendering the single-cell virtual reality visualization. *Scanpy* offers tools for clustering, which can be visualized using *singlecellVR*. Cells can be visualized and colored by various annotations. Shown: mouse tissue type, (left) or their cluster ID (right). The *Scanpy*-analyzed dataset shown here is from the Chan Zuckerberg Initiative's *Tabula Muris* dataset (Schaum et al., 2018). **(B)** *STREAM*-processed single-cell proteomics data from *SCoPE2* (Specht et al., 2019). These visualizations are an example of an advantage gained by trajectory analysis and three-dimensional visualization. 3-D UMAP plots (ordered left to right, top to bottom) generated by *STREAM*, respectively colored by pseudotime progression, cell type (orange: monocyte, blue: macrophage), expression of *Safb*, and expression of *Pfn1*.

Supplementary Figure 4 | Camera coordinates and angle descriptions enable reproducible visualizations. Shown is a UMAP of the Allen Brain Atlas mouse brain scRNA-seq dataset *Yao, et al.,* 2020 (Yao, 2020) and processed by Scanpy colored by the Leiden clustering solution. Close-ups of the coordinates as well as the toggle for displaying coordinates are shown.

REFERENCES

Albergante, L., Mirkes, E., Bac, J., Chen, H., Martin, A., Faure, L., et al. (2020). Robust and Scalable Learning of Complex Intrinsic Dataset Geometry via ElPiGraph. *Entropy* 22, 296. doi:10.3390/e22030296

Bastidas-Ponce, A., Tritschler, S., Dony, L., Scheibner, K., Tarquis-Medina, M., Salinno, C., et al. (2019). Massive Single-Cell mRNA Profiling Reveals a Detailed Roadmap for Pancreatic Endocrinogenesis. *Development* 146. doi:10.1242/DEV.173849

Bergen, V., Lange, M., Peidli, S., Wolf, F. A., and Theis, F. J. (2020). Generalizing RNA Velocity to Transient Cell States through Dynamical Modeling. *Nat. Biotechnol.* 38, 1408–1414. doi:10.1038/s41587-020-0591-3

Bressan, D., Mulvey, C. M., Qosaj, F., Becker, R., Grimaldi, F., Coffey, S., et al. (2021). Exploration and Analysis of Molecularly Annotated, 3D Models of Breast Cancer at Single-Cell Resolution Using Virtual Reality. *bioRxiv* 06, 448342. doi:10.1101/2021.06.28.448342

Buenrostro, J. D., Corces, M. R., Lareau, C. A., Wu, B., Schep, A. N., Aryee, M. J., et al. (2018). Integrated Single-Cell Analysis Maps the Continuous Regulatory Landscape of Human Hematopoietic Differentiation. *Cell* 173, 1535–1548. doi:10.1016/j.cell.2018.03.074

Buenrostro, J. D., Wu, B., Litzenburger, U. M., Ruff, D., Gonzales, M. L., Snyder, M. P., et al. (2015). Single-cell Chromatin Accessibility Reveals Principles of Regulatory Variation. *Nature* 523, 486–490. doi:10.1038/nature14590

Chen, H., Albergante, L., Hsu, J. Y., Lareau, C. A., Lo Bosco, G., Guan, J., et al. (2019a). Single-cell Trajectories Reconstruction, Exploration and Mapping of Omics Data with STREAM. *Nat. Commun.* 10. doi:10.1038/s41467-019-09670-4

Chen, H., Lareau, C., Andreani, T., Vinyard, M. E., Garcia, S. P., Clement, K., et al. (2019c). Assessment of Computational Methods for the Analysis of Single-Cell ATAC-Seq Data. *Genome Biol.* 20. doi:10.1186/s13059-019-1854-5

Chen, S., Lake, B. B., and Zhang, K. (2019d). High-throughput Sequencing of the Transcriptome and Chromatin Accessibility in the Same Cell. *Nat. Biotechnol.* 37, 1452–1457. doi:10.1038/s41587-019-0290-0

Chen, W., Guillaume-Gentil, O., Dainese, R., Rainer, P. Y., Zachara, M., Gäbelein, C. G., et al. (2021). Genome-wide Molecular Recording Using Live-Seq. *bioRxiv* 03, 436752. doi:10.1101/2021.03.24.436752

Danese, A., Richter, M. L., Fischer, D. S., Theis, F. J., and Colomé-Tatché, M. (2019). EpiScanpy: Integrated Single-Cell Epigenomic Analysis. *bioRxiv*. doi:10.1101/648097

Hao, Y., Hao, S., Andersen-Nissen, E., Mauck, W. M., Zheng, S., Butler, A., et al. (2021). Integrated Analysis of Multimodal Single-Cell Data. *Cell* 184, 3573–3587. doi:10.1016/J.CELL.2021.04.048

La Manno, G., Soldatov, R., Zeisel, A., Braun, E., Hochgerner, H., Petukhov, V., et al. (2018). RNA Velocity of Single Cells. *Nature* 560, 494–498. doi:10.1038/s41586-018-0414-6

Labib, M., and Kelley, S. O. (2020). Single-cell Analysis Targeting the Proteome. *Nat. Rev. Chem.* 4, 143–158. doi:10.1038/s41570-020-0162-7

Lähnemann, D., Köster, J., Szczurek, E., McCarthy, D. J., Hicks, S. C., Robinson, M. D., et al. (2020). Eleven Grand Challenges in Single-Cell Data Science. *Genome Biol.* doi:10.1186/s13059-020-1926-6

Legetth, O., Rodhe, J., Lang, S., Dhapola, P., Pålsson, J., Wallergård, M., et al. (2019). CellexalVR: A Virtual Reality Platform to Visualise and Analyse Single-Cell Data. *bioRxiv*. doi:10.1101/329102

Luo, C., Hajkova, P., and Ecker, J. R. (2018). Dynamic DNA Methylation: In the Right Place at the Right Time. *Science* 361, 1336–1340. doi:10.1126/science.aat6806

Ma, S., Zhang, B., LaFave, L. M., Earl, A. S., Chiang, Z., Hu, Y., et al. (2020). Chromatin Potential Identified by Shared Single-Cell Profiling of RNA and Chromatin. *Cell* 183, 1103–1116. doi:10.1016/J.CELL.2020.09.056

Miller, B. F., Bambah-Mukku, D., Dulac, C., Zhuang, X., and Fan, J. (2021). Characterizing Spatial Gene Expression Heterogeneity in Spatially Resolved Single-Cell Transcriptomic Data with Nonuniform Cellular Densities. *Genome Res.* 271288, 120. doi:10.1101/GR.271288.120

Nestorowa, S., Hamey, F. K., Pijuan Sala, B., Diamanti, E., Shepherd, M., Laurenti, E., et al. (2016). A Single-Cell Resolution Map of Mouse Hematopoietic Stem and Progenitor Cell Differentiation. *Blood* 128, e20–e31. doi:10.1182/blood-2016-05-716480

Paul, F., Arkin, Y. a., Giladi, A., Jaitin, D. A., Kenigsberg, E., Keren-Shaul, H., et al. (2015). Transcriptional Heterogeneity and Lineage Commitment in Myeloid Progenitors. *Cell* 163, 1663–1677. doi:10.1016/j.cell.2015.11.013

Perkel, J. M. (2021). Single-cell Analysis Enters the Multiomics Age. *Nature* 595, 614–616. doi:10.1038/D41586-021-01994-W

Saelens, W., Cannoodt, R., Todorov, H., and Saeys, Y. (2018). A Comparison of Single-Cell Trajectory Inference Methods: towards More Accurate and Robust Tools. *bioRxiv*. doi:10.1101/276907

Schaum, N., Karkanias, J., Neff, N. F., May, A. P., Quake, S. R., Wyss-Coray, T., et al. (2018). Single-cell Transcriptomics of 20 Mouse Organs Creates a Tabula Muris. *Nature* 562, 367–372. doi:10.1038/s41586-018-0590-4

Spark, A., Kitching, A., Esteban-Ferrer, D., Handa, A., Carr, A. R., Needham, L.-M., et al. (2020). vLUME: 3D Virtual Reality for Single-Molecule Localization Microscopy. *Nat. Methods* 17, 1097–1099. doi:10.1038/s41592-020-0962-1

Specht, H., Emmott, E., Petelski, A. A., Huffman, R. G., Perlman, D. H., Serra, M., et al. (2019). Single-cell Proteomic and Transcriptomic Analysis of Macrophage Heterogeneity. *bioRxiv*. doi:10.1101/665307

Stefani, C., Lacy-Hulbert, A., and Skillman, T. (2018). ConfocalVR: Immersive Visualization for Confocal Microscopy. *J. Mol. Biol.* 430, 4028–4035. doi:10.1016/J.JMB.2018.06.035

Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck, W. M., et al. (2019). Comprehensive Integration of Single-Cell Data. *Cell* 177, 1888–1902. doi:10.1016/j.cell.2019.05.031

Stuart, T., and Satija, R. (2019). Integrative Single-Cell Analysis. *Nat. Rev. Genet.* 20, 257–272. doi:10.1038/s41576-019-0093-7

Swanson, E., Lord, C., Reading, J., Heubeck, A. T., Genge, P. C., Thomson, Z., et al. (2021). Simultaneous Trimodal Single-Cell Measurement of Transcripts, Epitopes, and Chromatin Accessibility Using tea-seq. *Elife* 10. doi:10.7554/ELIFE.63632

Tian, L., Dong, X., Freytag, S., Lê Cao, K.-A., Su, S., JalalAbadi, A., et al. (2019). Benchmarking Single Cell RNA-Sequencing Analysis Pipelines Using Mixture Control Experiments. *Nat. Methods* 16, 479–487. doi:10.1038/s41592-019-0425-8

Trapnell, C. (2015). Defining Cell Types and States with Single-Cell Genomics. *Genome Res.* 25, 1491–1498. doi:10.1101/gr.190595.115

Wang, Y., Li, Q., Liu, L., Zhou, Z., Ruan, Z., Kong, L., et al. (2019). TeraVR Empowers Precise Reconstruction of Complete 3-D Neuronal Morphology in the Whole Brain. *Nat. Commun.* 10, 1–9. doi:10.1038/s41467-019-11443-y

Weinreb, C., Wolock, S., Tusi, B. K., Socolovsky, M., and Klein, A. M. (2018). Fundamental Limits on Dynamic Inference from Single-Cell Snapshots. *Proc. Natl. Acad. Sci. USA* 115, E2467–E2476. doi:10.1073/pnas.1714723115

Welch, J. D., Kozareva, V., Ferreira, A., Vanderburg, C., Martin, C., and Macosko, E. Z. (2019). Single-Cell Multi-Omic Integration Compares and Contrasts Features of Brain Cell Identity. *Cell* 177, 1873–1887. doi:10.1016/j.cell.2019.05.006

Wolf, F. A., Angerer, P., and Theis, F. J. (2018). SCANPY: Large-Scale Single-Cell Gene Expression Data Analysis. *Genome Biol.* 19. doi:10.1186/s13059-017-1382-0

Wolf, F. A., Hamey, F. K., Plass, M., Solana, J., Dahlin, J. S., Göttgens, B., et al. (2019). PAGA: Graph Abstraction Reconciles Clustering with Trajectory Inference through a Topology Preserving Map of Single Cells. *Genome Biol.* 20. doi:10.1186/s13059-019-1663-x

Xing, Q. R., Farran, C. A. E., Zeng, Y. Y., Yi, Y., Warrier, T., Gautam, P., et al. (2020). Parallel Bimodal Single-Cell Sequencing of Transcriptome and Chromatin Accessibility. *Genome Res.* 30, 1027–1039. doi:10.1101/GR.257840.119

Yang, A., Yao, Y., Li, J., and Ho, J. W. K. (2018). Starmap: Immersive Visualisation of Single Cell Data Using Smartphone-Enabled Virtual Reality. *bioRxiv*. doi:10.1101/324855

Yao, Z., Nguyen, T. N., van Velthoven, C. T. J., Goldy, J., Sedeno-Cortes, A. E., Baftizadeh, F., et al. (2020). A Taxonomy of Transcriptomic Cell Types across the Isocortex and Hippocampal Formation. *bioRxiv*. doi:10.1101/2020.03.30.015214

Zhu, C., Yu, M., Huang, H., Juric, I., Abnousi, A., Hu, R., et al. (2019). An Ultra High-Throughput Method for Single-Cell Joint Analysis of Open Chromatin and Transcriptome. *Nat. Struct. Mol. Biol.* 26, 1063–1070. doi:10.1038/s41594-019-0323-x

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers in Genetics

ORIGINAL RESEARCH
published: 02 November 2021
doi: 10.3389/fgene.2021.774846

Use *ggbreak* to Effectively Utilize Plotting Space to Deal With Large Datasets and Outliers

Shuangbin Xu[†], *Meijun Chen*[†], *Tingze Feng, Li Zhan, Lang Zhou and Guangchuang Yu* *

Department of Bioinformatics, School of Basic Medical Sciences, Southern Medical University, Guangzhou, China

With the rapid increase of large-scale datasets, biomedical data visualization is facing challenges. The data may be large, have different orders of magnitude, contain extreme values, and the data distribution is not clear. Here we present an R package *ggbreak* that allows users to create broken axes using *ggplot2* syntax. It can effectively use the plotting area to deal with large datasets (especially for long sequential data), data with different magnitudes, and contain outliers. The *ggbreak* package increases the available visual space for a better presentation of the data and detailed annotation, thus improves our ability to interpret the data. The *ggbreak* package is fully compatible with *ggplot2* and it is easy to superpose additional layers and applies scale and theme to adjust the plot using the *ggplot2* syntax. The *ggbreak* package is open-source software released under the Artistic-2.0 license, and it is freely available on CRAN (https://CRAN.R-project.org/package=ggbreak) and Github (https://github.com/YuLab-SMU/ggbreak).

Keywords: axis break, gap plot, long sequential data, outlier, ggplot2

OPEN ACCESS

Edited by:
Zhibin Lv,
Sichuan University, China

Reviewed by:
Feng Long Yang,
University of Electronic Science and Technology of China, China
Lijun Dou,
Shenzhen Polytechnic, China

***Correspondence:**
Guangchuang Yu
gcyu1@smu.edu.cn

[†]*These authors have contributed equally to this work*

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

Received: *13 September 2021*
Accepted: *12 October 2021*
Published: *02 November 2021*

Citation:
Xu S, Chen M, Feng T, Zhan L, Zhou L and Yu G (2021) Use ggbreak *to Effectively Utilize Plotting Space to Deal With Large Datasets and Outliers. Front. Genet. 12:774846. doi: 10.3389/fgene.2021.774846*

INTRODUCTION

Many visualization methods would not be able to display a graph on a print page and this limits the publication of these results. There are several reasons. For example, the amount of data is large, the data contains outliers and squeezes the main part of the graph or both. As the volume and complexity of biomedical data are growing rapidly (O'Donoghue et al., 2018), circular graphs such as chord diagrams, sunburst diagrams, and circular phylograms, are becoming popular to save space for big data applications. However, not all horizontal methods have corresponding circular counterparts. Moreover, a circular graph also has its limitations. Compared with a horizontal chart, a circular graph is not intuitive and not easy to compare. One of the approaches to explore a large dataset is to split the data into several rows of graphs, especially for long sequences of data (e.g., time-series plot). Splitting a graph into multiple rows helps to improve the identification of data trends and patterns.

Outliers are unusual values that lie outside the overall pattern of distribution. It's bad practice to simply exclude outlier data points since they are not always due to experimental errors or instrument errors. Outliers can be legitimate observations and could represent significant scientific effects. The identification of meaningful outliers can often lead to unexpected findings. Many analytical methods are looking for outliers. Such as differentially expressed gene detection, genome-wide association studies. Visualizing data with outliers can be challenging as the graph will be stretched or squeezed by the outliers. To overcome this issue, data transformation methods, such as log transformation, are often used to transform skewed data. Nonetheless, the transformation should be motivated by the data type. The normal distribution is widely used in biomedical research studies to model continuous outcomes and the log transformation is the most popular method that was used to reduce the skewness of the distribution. A previous study showed that log transformation would introduce new

TABLE 1 | Major functions of *ggbreak*.

Function	Description
scale_wrap	Wraps a 'gg' plot over multiple rows
scale_x_break	Set an *x*-axis break point
scale_y_break	Set a *y*-axis break point
scale_x_cut	Set an *x*-axis divide point
scale_y_cut	Set a *y*-axis divide point

problems that are even more difficult to deal with (Feng et al., 2014). Applying log-transformation to data sets that are not log-normal distributed does not reduce skewness. If we are looking for outliers in our data, a process like a log transformation would de-emphasize them (Metcalf and Casey, 2016). Furthermore, log-transformed data shares little in common with the original data. Some plot patterns like boxplots have been implemented to solve the visualization problem of outliers that still can't meet the requirement (Williamson et al., 1989). Broken axes have become a common feature of graphs in biomedical studies and also other research areas. Breaking the axis can simplify the outlier visualization, improve aesthetics, and save space (Amos and MedImmune, 2015). Advantages include applying to different distributions and preserving the original data scale, and thus more easy to convey the difference and variation between the low and high groups.

Displaying a plot with a gapped axis (i.e., missing range on one axis) is often used for the visualization of highly skewed data. When the bulk of the values get squeezed into a smaller region of the plot due to outliers, the gapped axis allows the plot to eliminate the open space between the outliers and the other data. Thus both data can be presented on the graph clearly. The R programming language has become one of the most popular tools for biomedical data visualization. However, creating gap plots is not well supported in R. The *plotrix* package provides *gap.plot()*, *gap.barplot()* and *gap.boxplot()* functions (Lemon, 2006), and the *gg.gap* package provides *gg.gap()* function to draw gap plots in base graphics and ggplot2 respectively. Unfortunately, these functions do not support overlay graphic layers after creating a gapped axis. Allowing further annotation on the graph is quite important because before the gapped plot is created, the graph is stretched or squeezed and it is not easy to add an annotation at the exact position. Moreover, in addition to gap plot, axis break has other applications, including displaying long sequence data in multiple rows, splitting a graph into multiple slices to zoom in and out to help interpretation of selected parts. These features are not implemented in R. To fill these gaps, we developed an R package, *ggbreak*, for creating an elegant axis break based on the grammar of graphic syntax implemented in *ggplot2*. This package provides a better solution to set axis break and can be widely applied in tailored visualization for various types of plots and data.

DESCRIPTION

Overview of the *ggbreak* Package

The *ggbreak* package was developed with the merits of *ggplot2* which is intuitive and flexible for data visualization (Wickham, 2009). The *ggbreak* package provides several scale functions including *scale_x_break()*, *scale_y_break()*, *scale_x_cut()*, *scale_y_cut()* and *scale_wrap()* to set axis break of *ggplot2* graphics (**Table 1**). The *scale_x_break()* and *scale_y_break()* functions create a gap plot with one or multiple missing ranges and allow users to adjust the relative width or height of plot slices (*i.e.*, zoom in or zoom out different parts of the plot). The *ticklabels* parameter can be used to specify customized axis labels of the plot slices. The *scale_x_cut()* and *scale_y_cut()* functions cut the plot into multiple slices to allow zoom in or zoom out of selected parts (e.g., allocating more space to display differentially expressed genes with labels in a volcano plot). The *scale_wrap()* function splits a plot over multiple rows to make the plot with a long *x*-axis (*e.g.*, time-series graphics) easier to read. The *ggbreak* package is fully compatible with *ggplot2*. After wrapping, breaking, and cutting axes of a plot, users are free to superpose multiple geometric layers from different data sources and apply theme and other scale settings. Plots created by *ggbreak* are compatible with *patchwork* and *aplot* to produce a composite plot.

Case Study

Example 1: Automatically Wrap Plot with Long *x*-Axis Scale

Graphs for long sequence data usually are squeezed and difficult to interpret due to the limited size of a print page. Wrapping plot for large-scale data into multiple panels helps users to identify sequential patterns. Here we provided an example to demonstrate the wrap plot implemented in *ggbreak*. The amino acid scales are numeric features of amino acids that are used to analyze protein sequences. Especially hydrophilicity/hydrophobicity scales are frequently used to characterize protein structures. Results of hydrophilicity/hydrophobicity scales usual are presented as a line chart. For long protein sequences, the line would be crowded in the graph because of highly divergent trends of the hydrophilicity/hydrophobicity scales. The protein sequence was downloaded from the NCBI database (PDB: 7MWE_A) and then the hydrophilicity/hydrophobicity scales were analyzed using *Expasy-ProtScale* with default parameters (Gasteiger et al., 2003). As showed in **Figure 1A**, the line is highly squeezed which makes it difficult for interpreting and understanding the sequential patterns. Splitting the plot into four rows makes the trends more clear to read (**Figure 1B**). The hydrophilicity regions and hydrophobicity regions are easier to identify through the whole sequence. Highlighted regions showed a clear division of hydrophilicity regions and hydrophobicity regions.

Example 2: Shrank Outlier Long Branch of a Phylogenetic Tree

Data outliers may have their biological meanings and are important in the studies. It is not appropriate to simply discard outliers in these scenarios. Data transformation de-emphasizes the outliers and is not always appropriate. Using broken axes is much simple and convenient for outlier data visualization since it preserves the original scale and works for known and unknown data distributions. A phylogenetic tree is widely used to model

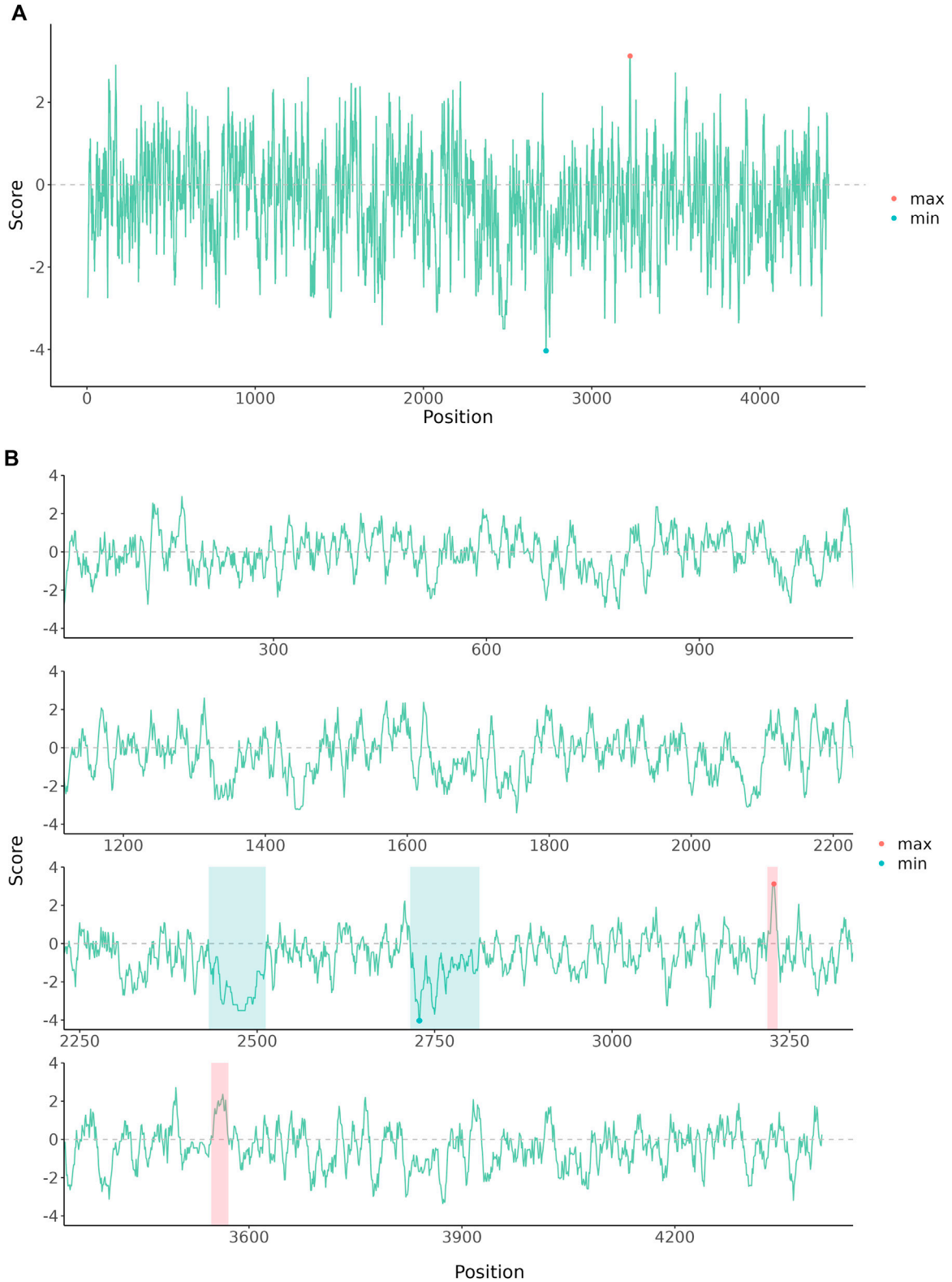

FIGURE 1 | The amino acid hydrophobicity or hydrophilicity scales of human E3 ubiquitin-protein ligase HUWE1 Chain A. The protein sequence was downloaded from the NCBI database (PDB: 7MWE_A). **(A)** The original plot of the amino acid scales. **(B)** The amino acid scales were wrapped into four rows and further annotated to highlight hydrophilicity and hydrophobicity regions.

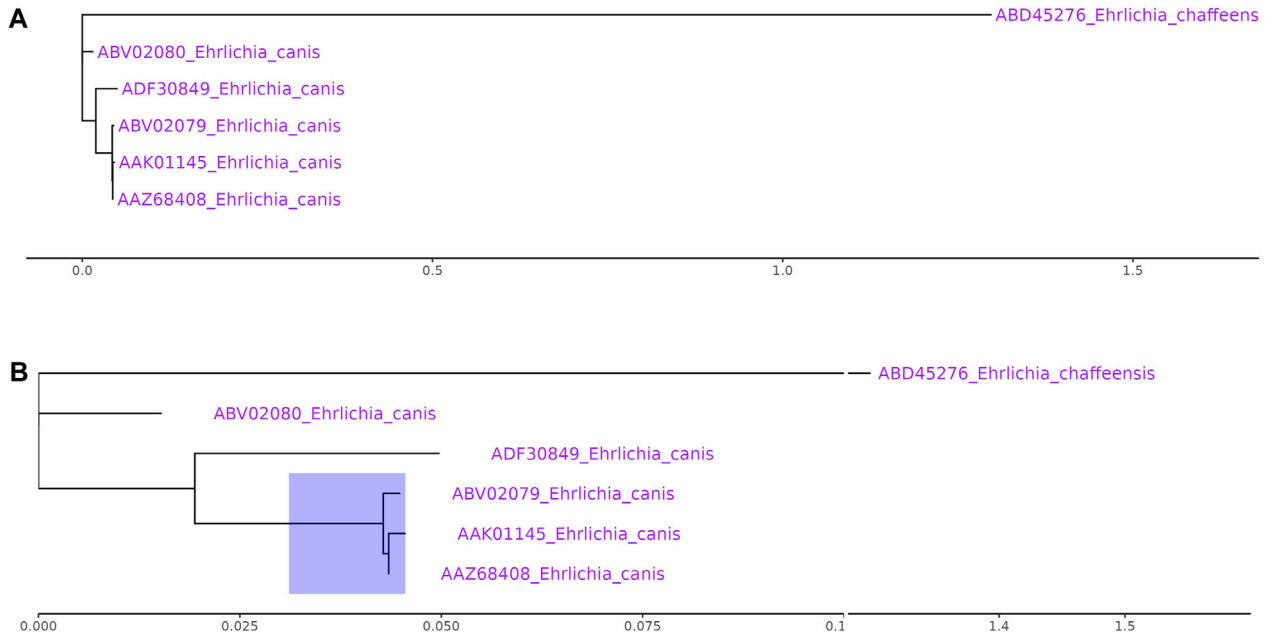

FIGURE 2 | Phylogenetic tree with outlier long branch. **(A)** The original plot fails to present the detailed topological structure of the main group. **(B)** The tree with a gapped axis to shrank the outlier long branch improves the readability of the main group. All sequences were collected from NCBI database (*Ehrlichia chaffeens*: ABD45276; *Ehrlichia canis*:ABV02080, ADF30849, ABV02079, AAK01145, AAZ68408).

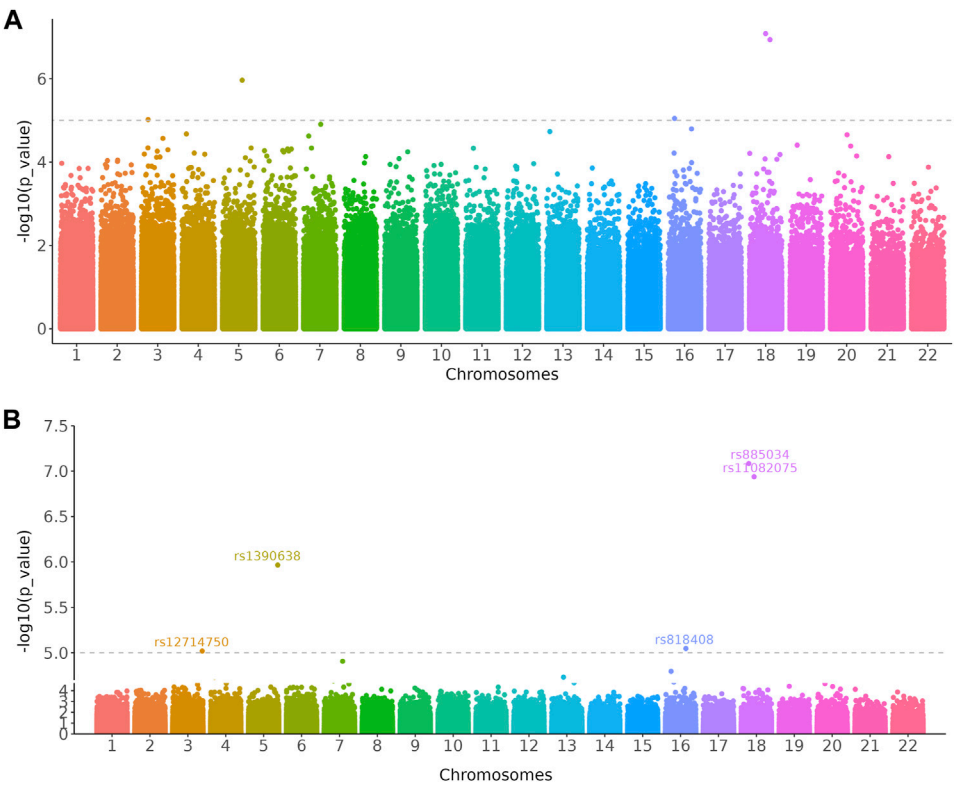

FIGURE 3 | Manhattan plots showed loci significantly associated with the relative abundance of gut bacterial diversity using Chao1 index from 541 East Asians. **(A)** Original Manhattan plot. **(B)** Manhattan plot with significant region zoomed in. The dashed line represents a genome-wide suggestive level ($p = 1 \times 10^{-5}$). The GWAS data was downloaded from the GWAS Catalog database (Study accession: GCST90007012).

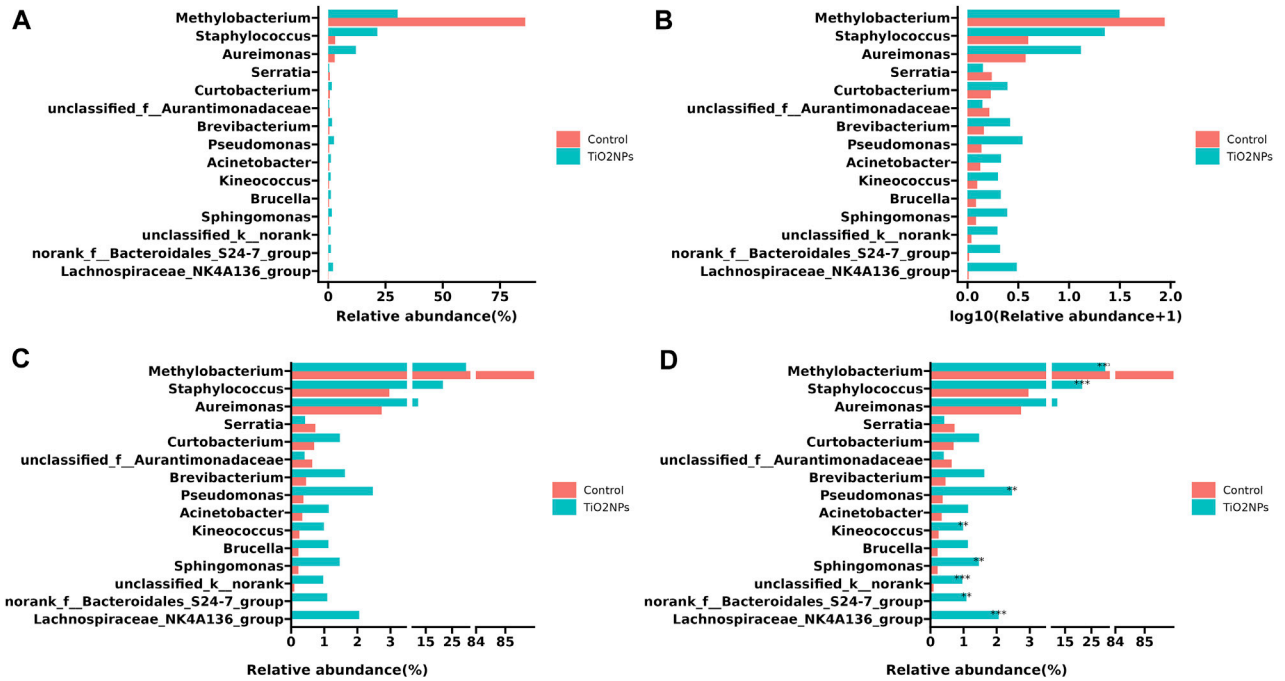

FIGURE 4 | Visualizing relative microbe abundances using bar charts to show the top 15 significant genera between two groups. **(A)** Ordinary bar chart that is difficult to deal with outliers. **(B)** Bar chart with log transformation on the *x*-axis to remove skewness. **(C)** Bar chart with a gapped axis that can effectively deal with data of different magnitude. **(D)**. Superposed significant level layer to the gapped plot **(C)**. The example data was collected from a paper published in 2020 (Li et al., 2020).

evolutionary relationships. An outgroup is usually employed to root the unrooted tree. As the outgroup is dissimilar to the main group, it may be placed on the outlier long branch. Phylogenetic tree with outlier long branch is difficult to display well as the main group will be squeezed into a smaller space (**Figure 2A**). The example data were collected from the NCBI database (Huang et al., 2010). After shrinking the outlier long branch using *ggbreak*, the detailed topological structure of the highlighted region can be displayed (**Figure 2B**).

Example 3: Cut Manhattan Plot to Create More Space for Annotation

Data presented on the graph is not equally important and researchers may want to zoom in on specific regions that are significant to the results. For instance, biologists want to focus on the differentially expressed genes (DEGs) of transcriptome data on a volcano plot. The *scale_x_cut()* and *scale_y_cut()* functions implemented in *ggbreak* allow users to zoom in significant regions of a plot. Here we use the Manhattan plot to demonstrate this feature. Manhattan plot is a kind of scatter plot and is commonly used in genome-wide association studies (GWAS) to display significant single nucleotide polymorphisms (SNPs). Researchers usually focus on the upper part of the graph that displays many significant results. It is difficult to label these significant results because these labels tend to overlap in a limited space and make it difficult to read. With the *scale_y_cut()* function, it is easy to zoom in on significant regions. The example data was collected from the GWAS Catalog database (Study accession: GCST90007012) (Buniello et al., 2019; Ishida et al., 2020). The lower part was zoomed out to save space for further annotation of the upper part and thus making it easier to highlight and interpret significant results (**Figure 3**).

Example 4: Display Discontinuous Axis on a Bar Chart

Since data have different magnitudes, visualizing data with gaps (missing ranges) is frequently used in biomedicine studies, especially for bar charts. For example, in metagenomics research, microbe abundance often has different orders of magnitude, with dominant microbes account for the major proportion, while other minor catalogs only account for a small fraction. To show microbe abundances properly, a common way is to create gaps in an axis. The data used in the following example was obtained from a published paper that has described relative abundances of the top 15 genera showing significant differences among samples from the TiO2NPs-treated group and Control group (Li et al., 2020). The value of *Methylobacterium* in the Control group is much higher than other observations (**Figure 4A**). Log transformation is a widely used method to reduce the skewness of the data (**Figure 4B**). However, the transformed data and the original data are not on the same scale, which will affect the interpretation of the data. Inserting two gaps in the axis makes it much more visible for other small observations. So that the relative abundance pattern of microbes is clear at a glance (**Figure 4C**). In addition, the gapped plot shares similar features with the log-transformed one and the result is intuitive and easy to interpret. Unlike log-transformation, a gapped plot can be applied to all data. Furthermore, it is easier to annotate the gapped plot (e.g.,

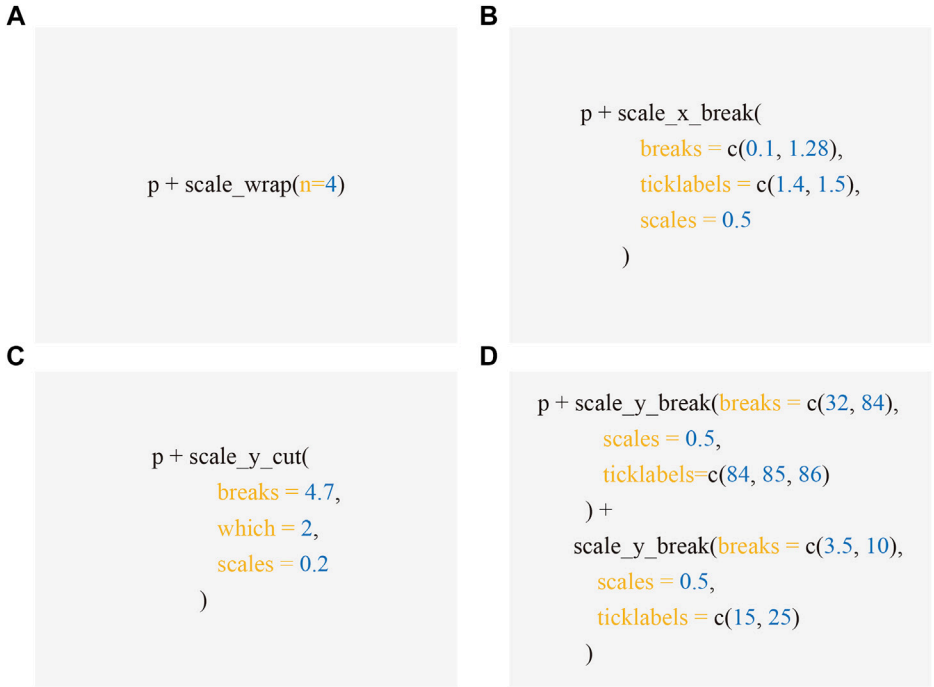

FIGURE 5 | Code excerpts to produce **Figures 1**-**4**. Applying scale functions implemented in ggbreak allows creating axis break to produced Figure 1B **(A)**, 2B **(B)**, 3B **(C)**, and 4C **(D)** respectively from the subplot A, which is represented by the p object - a graphic object produced by ggplot2.

superpose labels of significant level) since the scale of the value is the same as the original data (**Figure 4D**).

CONCLUSION

Gapped axis is quite regularly used in biomedical data visualization, but it is not well implemented in R. Here, we provide a fully functional tool, *ggbreak*, which can easily use the *ggplot2* grammar of graphics syntax to create a gapped axis. The output is still a *ggplot* object that can be further superposed annotation layers and customized by applying scale and theme settings. Unlike other software designed mainly for bar charts, *ggbreak* can be applied to all graphics generated by *ggplot2*. Moreover, *ggbreak* expands the usage of broken axes by applying it to wrap long sequential data and zoom in on important regions. The usage of axes breaks should depend on the data type. Inserting axis breaks appropriately would make the graphs much more readable and improve our ability to interpret the data.

CODE AVAILABILITY

The ggbreak package is freely available on CRAN (https://CRAN.R-project.org/package=ggbreak). The excerpts of the source code that produced **Figures 1**-**4** are presented in **Figure 5**. The complete code is available in **Supplemental Material**. R markdown file and data sets used to generate the Supplemental File are available on Github (https://github.com/YuLab-SMU/supplemental-ggbreak).

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding author.

AUTHOR CONTRIBUTIONS

GY designed the package. SX, GY, and MC implemented the package. MC and GY wrote the manuscript. TF, LiZ, and LaZ proofread and corrected the manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

This work was supported by Startup Fund from Southern Medical University.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2021.774846/full#supplementary-material

REFERENCES

Amos, S., and MedImmune, G. (2015). "Creating a Break in the Axis." in Proceedings of PharmaSUG 2015 (PharmaSUG).

Buniello, A., MacArthur, J. A. L., Cerezo, M., Harris, L. W., Hayhurst, J., Malangone, C., et al. (2019). The NHGRI-EBI GWAS Catalog of Published Genome-wide Association Studies, Targeted Arrays and Summary Statistics 2019. *Nucleic Acids Res.* 47, D1005–D1012. doi:10.1093/nar/gky1120

Feng, C., Wang, H., Lu, N., Chen, T., He, H., Lu, Y., et al. (2014). Log-transformation and its Implications for Data Analysis. *Shanghai Arch. Psychiatry* 26, 105–109. doi:10.3969/j.issn.1002-0829.2014.02.009

Gasteiger, E., Gattiker, A., Hoogland, C., Ivanyi, I., Appel, R. D., and Bairoch, A. (2003). ExPASy: The Proteomics Server for In-Depth Protein Knowledge and Analysis. *Nucleic Acids Res.* 31, 3784–3788. doi:10.1093/nar/gkg563

Huang, C.-C., Hsieh, Y.-C., Tsang, C.-L., and Chung, Y.-T. (2010). Sequence and Phylogenetic Analysis of Thegp200protein ofEhrlichia Canisfrom Dogs in Taiwan. *J. Vet. Sci.* 11, 333–340. doi:10.4142/jvs.2010.11.4.333

Ishida, S., Kato, K., Tanaka, M., Odamaki, T., Kubo, R., Mitsuyama, E., et al. (2020). Genome-wide Association Studies and Heritability Analysis Reveal the Involvement of Host Genetics in the Japanese Gut Microbiota. *Commun. Biol.* 3, 686. doi:10.1038/s42003-020-01416-z

Lemon, J. (2006). Plotrix: a Package in the Red Light District of R. R-News. *Psychology* 6, 8–12.

Li, M., Li, F., Lu, Z., Fang, Y., Qu, J., Mao, T., et al. (2020). Effects of TiO2 Nanoparticles on Intestinal Microbial Composition of Silkworm, *Bombyx mori*. *Sci. Total Environ.* 704, 135273. doi:10.1016/j.scitotenv.2019.135273

Metcalf, L., and Casey, W. (2016). "Introduction to Data Analysis," in *Cybersecurity And Applied Mathematics*. Editors L. Metcalf and W. Casey (Boston: Syngress), 43–65. doi:10.1016/B978-0-12-804452-0.00004-X

O'Donoghue, S. I., Baldi, B. F., Clark, S. J., Darling, A. E., Hogan, J. M., Kaur, S., et al. (2018). Visualization of Biomedical Data. *Annu. Rev. Biomed. Data Sci.* 1, 275–304. doi:10.1146/annurev-biodatasci-080917-013424

Wickham, H. (2009). Ggplot2: Elegant Graphics For Data Analysis. *Ggplot2: Elegant Graphics for Data Analysis*.

Williamson, D. F., Parker, R. A., and Kendrick, J. S. (1989). The Box Plot: a Simple Visual Method to Interpret Data. *Ann. Intern. Med.* 110, 916–921. doi:10.7326/0003-4819-110-11-916

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Genetics

ORIGINAL RESEARCH
published: 25 November 2021
doi: 10.3389/fgene.2021.755789

Identification of an Autophagy-Related Gene Signature for the Prediction of Prognosis in Early-Stage Colorectal Cancer

Xu-tao Lin[1,2†], Qiu-ning Wu[1,2†], Si Qin[2,3†], De-jun Fan[1,2], Min-yi Lv[2,4], Xi Chen[2,4], Jia-wei Cai[2,4], Jing-rong Weng[2,4], Yi-feng Zou[2,4], Yu-ming Rong[5] and Feng Gao[2,4]**

[1]Department of Gastrointestinal Endoscopy, The Sixth Affiliated Hospital of Sun Yat-sen University, Guangzhou, China, [2]Guangdong Provincial Key Laboratory of Colorectal and Pelvic Floor Diseases, Guangdong Institute of Gastroenterology, The Sixth Affiliated Hospital of Sun Yat-sen University, Guangzhou, China, [3]Department of Medical Ultrasonics, The Sixth Affiliated Hospital of Sun Yat-sen University, Guangzhou, China, [4]Department of Colorectal Surgery, The Sixth Affiliated Hospital of Sun Yat-sen University, Guangzhou, China, [5]Department of VIP Region, Cancer Center of Sun Yat-sen University, Guangzhou, China

OPEN ACCESS

Edited by:
Guangchuang Yu,
Southern Medical University, China

Reviewed by:
Dongguo Li,
Capital Medical University, China
Saifur Rahaman,
City University of Hong Kong, Hong Kong SAR, China
Huaichao Luo,
Sichuan Cancer Hospital, China

***Correspondence:**
Yu-ming Rong
rongym@sysucc.org.cn
Feng Gao
gaof57@mail.sysu.edu.cn

[†]These authors have contributed equally to this work

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

Received: 09 August 2021
Accepted: 19 October 2021
Published: 25 November 2021

Citation:
Lin X-t, Wu Q-n, Qin S, Fan D-j, Lv M-y, Chen X, Cai J-w, Weng J-r, Zou Y-f, Rong Y-m and Gao F (2021) Identification of an Autophagy-Related Gene Signature for the Prediction of Prognosis in Early-Stage Colorectal Cancer. Front. Genet. 12:755789. doi: 10.3389/fgene.2021.755789

Purpose: A certain number of early-stage colorectal cancer (CRC) patients suffer tumor recurrence after initial curative resection. In this context, an effective prognostic biomarker model is constantly in need. Autophagy exhibits a dual role in tumorigenesis. Our study aims to develop an autophagy-related gene (ATG) signature-based on high-throughput data analysis for disease-free survival (DFS) prognosis of patients with stage I/II CRC.

Methods: Gene expression profiles and clinical information of CRC patients extracted from four public datasets were distributed to discovery and training cohort (GSE39582), validation cohort (TCGA CRC, n = 624), and meta-validation cohort (GSE37892 and GSE14333, n = 420). Autophagy genes significantly associated with prognosis were identified.

Results: Among 655 autophagy-related genes, a 10-gene ATG signature, which was significantly associated with DFS in the training cohort (HR, 2.76[1.56–4.82]; $p = 2.06 \times 10–4$), was constructed. The ATG signature, stratifying patients into high and low autophagy risk groups, was validated in the validation (HR, 2.29[1.15–4.55]; $p = 1.5 \times 10–2$) and meta-validation cohorts (HR, 2.5[1.03–6.06]; $p = 3.63 \times 10–2$) and proved to be prognostic in a multivariate analysis. Functional analysis revealed enrichment of several immune/inflammatory pathways in the high autophagy risk group, where increased infiltration of T regulatory cells (Tregs) and decreased infiltration of M1 macrophages were observed.

Conclusion: Our study established a prognostic ATG signature that effectively predicted DFS for early-stage CRC patients. Meanwhile, the study also revealed the possible relationship among autophagy process, immune/inflammatory response, and tumorigenesis.

Keywords: colorectal cancer, prognosis, early stage, disease-free survival, autophagy-related gene

INTRODUCTION

CRC is currently the second leading cause of cancer deaths worldwide, ranking third in morbidity (Bray et al., 2018). Depending upon the tumor stage at diagnosis, relative 5-year survival rates for patients with CRC range from 65% for all stages to 91, 82, and 12% for patients diagnosed with stage I, II, and IV, respectively (Miller et al., 2019). Although increasing awareness of cancer screening and advances in technology have improved early detection (Dekker et al., 2019) and enabled treatments without chemotherapy, around one-third of patients with stages I-III CRC still encounter tumor relapse after so-called curative treatment (Van Der Stok et al., 2017). Therefore, it is particularly important to identify these high-risk patients with poor prognosis. The National Comprehensive Cancer Network (NCCN) suggested such clinicopathologic features as high-risk factors for stage II colorectal cancer, including tumor size, number of lymph nodes investigated, degree of differentiation, tumor perforation, bowel obstruction, and lymphovascular invasion. However, some studies have shown that these pathological features are inadequate to accurately identify such high-risk patients (O'Connor et al., 2011; Kannarkatt et al., 2017). Accordingly, there is a growing need for novel molecular markers for prognosis patterns, which might provide valuable information for supplementary adjuvant chemotherapy or other targeted therapy. Recently, the predictive potency of KRAS and BRAF mutations, microsatellite instability (MSI) status, and CpG island methylator phenotype (CIMP) status in CRC had been studied extensively. Kim et al. (Kim et al., 2019) investigated a novel prognostic predictor based on an 11-gene signature for identifying high-risk CRC and predicting patients who will have the worst response to adjuvant chemotherapy. However, more markers await to be discovered.

Autophagy plays a dual role in tumorigenesis (Rosich et al., 2013). It inhibits early tumor initiation by the clearance of damaged mitochondria, peroxisome, and other cytotoxic substance and also caters to the high metabolic demands from accelerated proliferating tumors by degrading intracellular organelle and macromolecule substances (Kongara and Karantza, 2012; Carroll and Martin, 2013). The autophagy process includes initiation of autophagy, biogenesis of the phagophore, expansion of the phagophore, formation of the autophagosome, fusion with the lysosome, and reformation of the lysosome (Zhi et al., 2018). A few gene mutations on the autophagy process reveal correlated with human cancer. PARK2 (Parkin), an autophagy-related gene participating in mitophagy and autophagy-independent functions that regulate the cell cycle, was identified as a potential tumor suppressor on chromosome 6q25-q26 which is frequently deleted in human cancers. Autophagy substrate p62 deficiency triggered by autophagy deficiency was found to suppress tumorigenesis in mouse liver. P62 regulates NRF2 and also mTOR and NFκB, all of which are important in cancer signaling (White et al., 2015). Recent studies have started working on autophagy-related gene model building in CRC (Huang et al., 2019; Zhou et al., 2019; Cheng et al., 2021; Zhao et al., 2021). Most of them constructed different autophagy gene signatures to monitor the CRC prognosis, regardless of tumor stages. However, early-stage CRC patients may need more accurate prognostic guidance, wondering about the need for additional chemotherapy. So, to further investigate how autophagy affects the prognosis of early-stage CRC patients, we identified an autophagy-related gene (ATG) signature from CRC-specific transcriptomes based on high-throughput data analysis. The ATG signature, which stratified the stage I/II CRC patients into distinct risk groups, together with functional analysis, might provide insights into the mechanism of CRC recurrence and targeted treatment.

METHODS

Public Datasets

Gene expression data of CRC tissue samples and corresponding clinical information obtained from the public database were retrospectively analyzed. A total of 1,610 patient samples from four independent public cohorts were included. Three hundred nine patients without adjuvant chemotherapy but with survival information in the GSE39582 dataset (n = 566) served as the discovery cohort; stage I and II patients among these 309 patients were intended for training. The Cancer Genome Atlas (TCGA) CRC dataset (n = 624) was used for independent validation, while the GSE37892 and GSE14333 datasets (n = 420) were combined for meta-validation. All datasets are from Affymetrix Human Genome U133 Plus 2.0 Array. TCGA cohort data were downloaded from Broad GDAC Firehose before transcripts per million (TPM) of level 3 RNA-Seq data in the log2 scale were applied to calibrate the gene expression levels. Other datasets were obtained directly in their processed format from the Gene Expression Omnibus database (GEO) through the Bioconductor package "GEOquery." The "combat" algorithm of the R package "sva" and the z-scores were used to correct the batch effects, so as to standardize microarray data across multiple experiments and compare them independent of the original hybridization intensities.

Construction of a Prognostic ATG Signature

The absolute median difference (MAD) was used to select the ATGs (Guinney et al., 2015). MAD is a robust statistic of statistical divergence that is more adaptable to outliers in a dataset than the standard deviation. To construct a prognostic ATG signature, we focused on 655 ATGs from eight gene sets identified *via* MSigDB (version 6.2) (Subramanian et al., 2005; Liberzon et al., 2011; Liberzon et al., 2015) with the keyword "autophagy." Only 617 genes measured on all platforms involved in this study with high variation (MAD >0.5) were selected. After 1,000 times random Cox univariate regressions, genes with repeated significance, which indicated a strong relationship between ATGs and patients' disease-free survival (DFS), were selected as candidates for the signature. A Cox proportional hazard regression model on CRC samples together with the least absolute shrinkage and selection operator (LASSO) was applied to minimize the risk of overfitting as well as to generate a risk model.

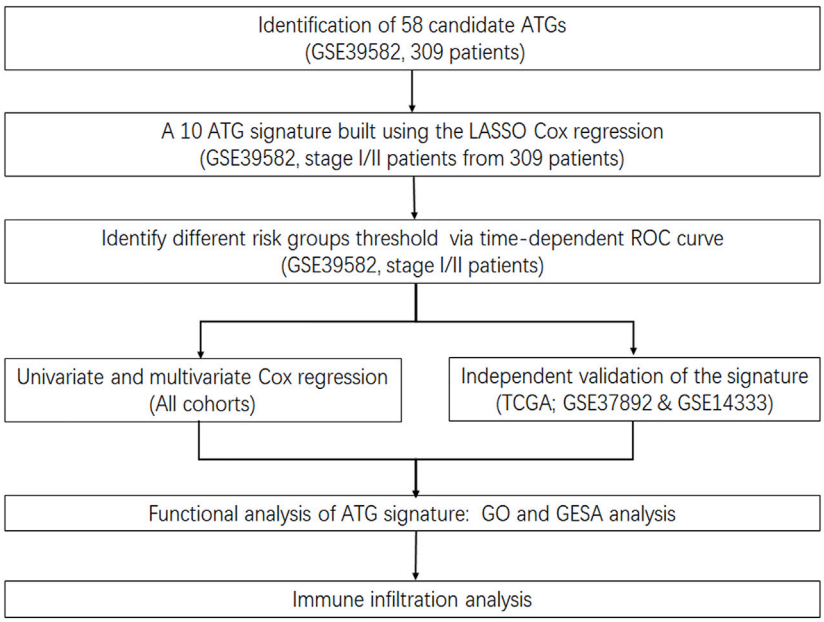

FIGURE 1 | Flowchart.

Patients were divided into low and high autophagy risk groups in accordance with the optimal ATG signature cutoff, which was defined by the time-dependent receiver operating characteristic (ROC) curve analysis at 5 years of DFS in the training cohort. The ATG signature score with the largest Youden's index in the ROC curve was deemed as the cutoff value.

Validation of ATG Signature

Oncotype DX is a quantitative multi-gene, real-time polymerase chain reaction (RT-PCR) assay that measures gene expression in paraffin-embedded tumor tissues. The C-index was employed in the GSE39582 cohort, TCGA cohorts, and meta-validation cohorts respectively in comparison to Oncotype DX colon to assess the predictive capability of the model. For further validation, the prognostic value of the ATG signature was evaluated in CRC patients with early stages (stage I and II) and all stages in different cohorts through survival analysis. Univariate and multivariate analyses of the ATG signature and available clinical parameters were performed to assess whether the ATG signature is an independent risk factor. The independent risk factors identified by multivariate Cox regression analysis were applied to construct the nomogram for estimating the DFS of 5 years in CRC.

Functional Analysis

Enrichment of potential pathways of the ATG signature by gene ontology (GO) analysis was performed on gProfiler (https://biit.cs.ut.ee/gprofiler/), and gene set enrichment analysis (GSEA) (Newman et al., 2015) was conducted using the Bioconductor package "fgsea." Gene sets of cancer hallmarks from MSigDB with statistical significance (FDR-adjusted $p<0.05$) were selected (Markle et al., 2010). CIBERSORT (Mokarram et al., 2017) was used to dissect immune cell infiltration in different risk groups.

Statistical Analysis

All statistical analyses were performed in R software (version 3.5.1). Categorical variables were reported as count. Continuous data were reported as mean with standard deviation (SD) and compared with the Student's t-test. The LASSO regression was plotted using the "glmnet" R package (version: 2.0-16). Time-dependent ROC curve analysis was done by the R package "survivalROC" (version: 1.0.3). Survival analysis was conducted using the Kaplan–Meier method and compared with the log-rank test. Univariate and multivariate analyses of ATG signature and clinical parameters were performed using the log-rank test. The statistical significance level was set at $\alpha = 0.05$, two-sided.

RESULTS

ATG Signature Establishment

After filtration with MAD >0.5, 617 genes measured on all platforms were selected for this study. By 1,000 times random Cox univariate regressions, 58 ATGs were identified to be strongly relevant to DFS and considered as candidates for the signature (**Figure 1**). A LASSO Cox regression in stage I and II patients in the training cohort (**Table 1**) revealed 10 ATGs for the risk model (**Figure 2**), with the coefficient of each ATG listed in **Table 2**. The risk model was formulated as follows: autophagy-related risk score = 0.040346631 × exp CD163L1 + 0.040346631 × exp FAM13B + 0.160103165 × exp HDAC6 + 0.05063732× exp HPR − 0.012205947 × exp NR2C2 − 0.027104325 × exp RAB12 +

TABLE 1 | Characteristics of GSE39582, TCGA, and meta-validation cohorts.

	GSE39582 (discovery and training)	TCGA (validation)	Meta-validation
Number of patients	566	624	420
Patients with survival data	557	509	356
Mean age, years	66.85 ± 13.29	66.27 ± 12.76	66.66 ± 12.60
Gender, n			
Male	310	332	233
Female	256	292	187
TNM stage, n			
Stage I	33	105	44
Stage II	264	230	167
Stage III	205	180	148
Stage IV	60	88	61
NA	4	21	-
CMS system, n			
CMS1	91	68	74
CMS2	232	207	168
CMS3	69	64	69
CMS4	127	117	97
NA	47	168	12
Tumor location			
Left	342	354	233
Right	224	270	185
NA	—	—	2
RFS event, n			
Yes	177	100	87
No	380	416	269
NA	9	108	64
OS event, n			
Yes	191	67	NA
No	371	557	NA
NA	4	—	420
DFS event, n			
Yes	248	146	87
No	314	386	269
NA	4	92	64
MMR status, n			
MSI	75	189	-
MSS	444	431	-
NA	47	4	420
CIMP status, n			
Positive	91	—	—
Negative	405	—	—
NA	70	624	420
CIN status, n			
Positive	353	—	—
Negative	110	—	—
NA	103	624	420
TP53 status, n			
Wild type	161	—	—
Mutation	190	—	—
NA	215	624	420
KRAS status, n			
Wild type	328	34	—
Mutation	217	30	—
NA	21	560	420
BRAF status, n			
Wild type	461	32	—
Mutation	51	3	—
NA	54	589	420
Chemotherapy adjuvant, n			
Yes	233	231	118
No	316	393	171
NA	17	—	131

RFS, relapse-free survival; OS, overall survival; DFS, disease-free survival; MMR, mismatch repair; CIMP, CpG island methylator phenotype; CIN, chromosomal instability; NA, not available.

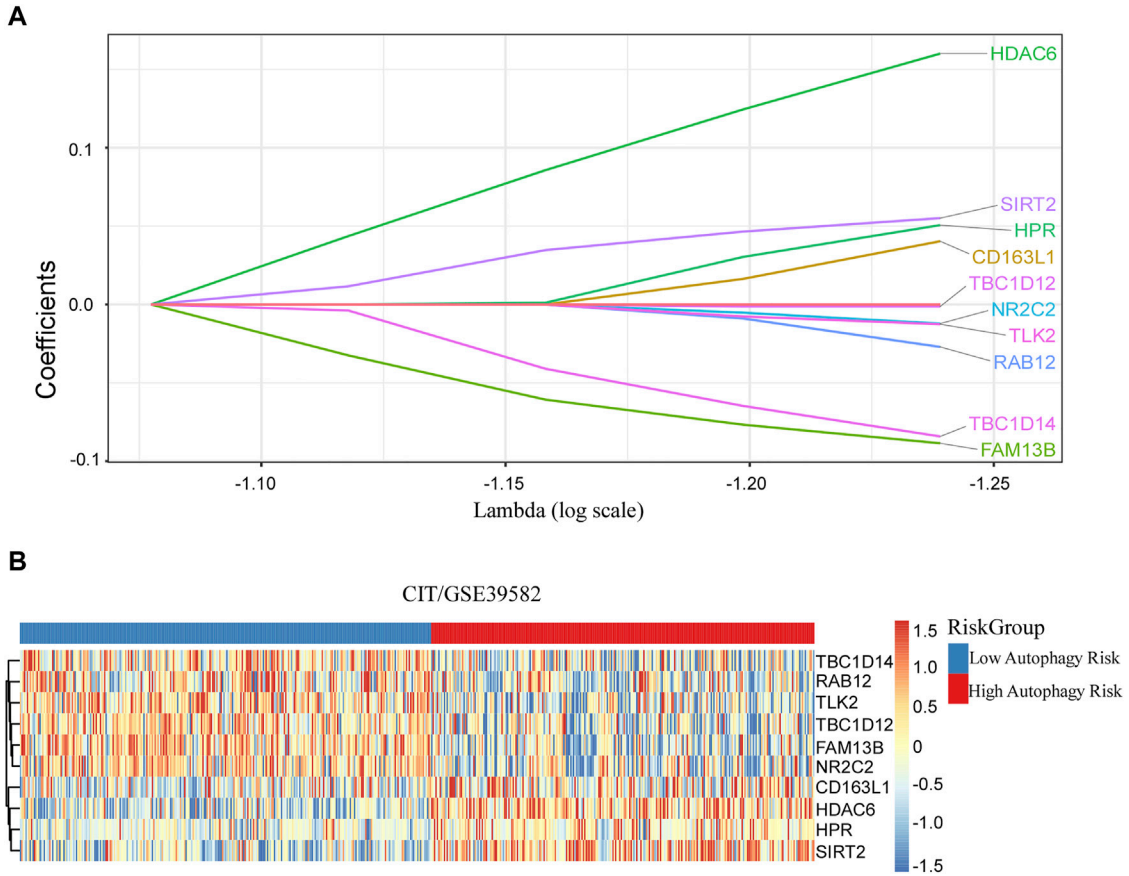

FIGURE 2 | Discovered ATG signature for prognostic prediction of colorectal cancer. **(A)** Ten ATGs signatures were found from the LASSO Cox regression. **(B)** Heatmap of the identified 10-gene signature.

TABLE 2 | Model information.

Gene	Name	Frequency in resampling	Average *p*-value	Coefficient
CD163L1	CD163 molecule-like 1	674	0.050594905	0.040346631
FAM13B	Family with sequence similarity 13 member B	625	0.050594905	0.040346631
HDAC6	Histone deacetylase 6	719	0.042438619	0.160103165
HPR	Haptoglobin-related protein	769	0.051952291	0.05063732
NR2C2	Nuclear receptor subfamily 2 group C member 2	898	0.019300065	-0.012205947
RAB12	RAB12, member RAS oncogene family	866	0.025407812	-0.027104325
SIRT2	Sirtuin 2	877	0.023493248	0.055095935
TBC1D14	TBC1 domain family member 14	998	0.002105817	-0.084226324
TLK2	Tousled like kinase 2	743	0.040418795	-0.012613054
TBC1D12	TBC1 domain family member 12	999	0.002072649	-0.001301898

0.055095935 × exp SIRT2 − 0.084226324 × exp TBC1D14 − 0.012613054 × exp TLK2 − 0.001301898 × exp TBC1D12 (each gene represents its mRNA expression level).

Based on the time-dependent ROC curve analysis of 5-year DFS in the training cohort, the optimal cutoff of ATG signature that divided the patients into high and low autophagy risk groups was −0.0087 (**Figure 3**). The risk scores of all patients are shown in **Supplementary Table S1**. Survival analysis showed that the DFS rate was higher in the low autophagy risk group compared to the high autophagy risk group for patients with early stages (stages I and II) in the training cohort (**Figures 4A–C**, HR, 2.76[1.56–4.82]; p = 2.06 ×

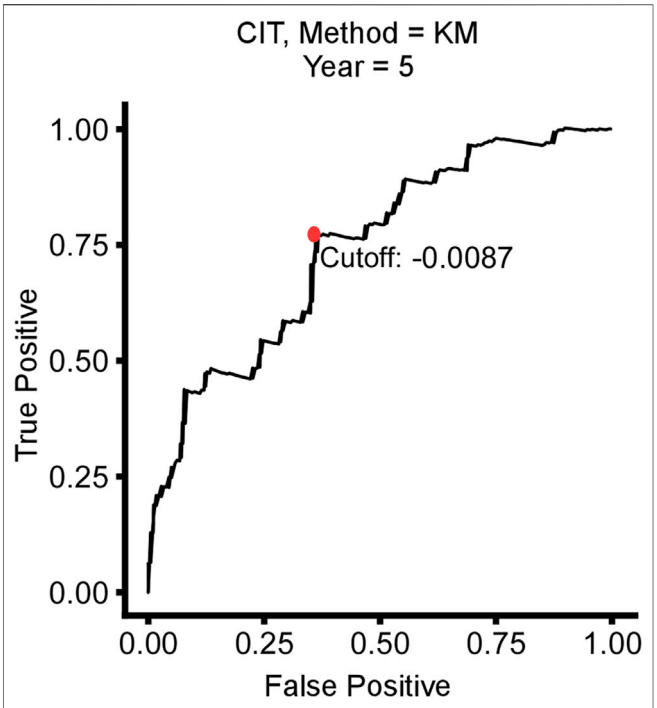

FIGURE 3 | Time-dependent ROC curve for 5-year disease-free survival (DFS) in stage I/II CRC patients in the training cohort.

10–4). So it was indicated for patients in all stages in the GSE39582 dataset (**Figures 5A–C**, HR, 1.7[1.25–2.31]; p = 5.21 × 10–4).

Validation of the ATG Signature

To assess the predictive capability of the risk model, the C-index was first applied to various cohorts which turned out to be 0.74 (95% CI, 0.63–0.85) in the GSE39582 cohort, 0.70 (95% CI, 0.54–0.85) in the TCGA cohort, and 0.70 (95% CI, 0.51–0.89) in the meta-validation cohort (**Table 3**), higher than those of Oncotype DX colon. We employed the same formula to the independent validation cohort (TCGA) and the meta-validation cohort (GSE37892 and GSE14333). Patients were significantly stratified into different risk groups by the ATG signature considering DFS. For early stages, CRC patients in the high autophagy risk group displayed poorer DFS in both the independent validation cohort (**Figures 4D–F**, HR, 2.29 [1.15–4.55]; p = 1.5 × 10–2) and the meta-validation cohort (**Figures 4G–I**, HR, 2.5[1.03–6.06]; p = 3.63 × 10–2). So it was indicated for patients with all stages in both the independent validation (**Figures 5D–F**, HR, 1.79[1.16–2.7]; p = 5.3 × 10–3) and meta-validation cohorts (**Figures 5G–I**, HR, 1.64 [1.04–2.52]; p = 3.16 × 10–2). Besides, the univariate and multivariate analyses further proved ATG as an independent prognostic factor after adjusting for clinical parameters such as sex and tumor stage (**Table 4**). Nomogram is displayed in **Supplementary Figure S1**.

Functional Analysis of the ATG Signature

GO analysis and GSEA were carried out to explore the biological function and signaling pathways of genes from the ATG signature. GO analysis revealed that the genes within the ATG signature were mostly involved in the regulation of autophagy and catabolic processes (**Figure 6A**; **Supplement Table S2**). GSEA was performed between different risk groups to further investigate the pathways that were significantly affected. We found a significant enrichment in multiple immune/inflammatory pathways in the high autophagy risk group, including the IL6/JAK/STAT3 signaling pathway, the IL2/STAT5 pathway, the IFN-α pathway, the IFN-γ pathway, and the inflammatory response pathway (**Figure 6B**, p value < 0.005). Some cell cycle/metabolism-associated pathways, including G2-M, oxidative phosphorylation, E2F, and MYC, were also significantly enriched in the high autophagy risk group, in addition to a few classic pathways like mTORC1 and epithelial–mesenchymal transmission (EMT; p value < 0.005).

As there was a significant enrichment in the immune/inflammatory pathway through GSEA analysis, we conducted immune infiltration analysis. The ESTIMATE algorithm displayed significant differences in the immune score (p = 0.02) and ESTIMATE score (p = 0.027) between the high and low autophagy risk groups in the TCGA CRC cohort (**Figure 7A**). Infiltration of plasma cells and Tregs was enriched significantly in the high autophagy risk group compared with the low autophagy risk group in the GSE39582 and TCGA cohorts (**Figure 7B–C**). By contrast, M1 macrophage infiltration turned out to be significantly lower in the high autophagy risk group (**Figure 7B–C**). Similar results are shown in **Supplementary Figure S2**.

Adjuvant Chemotherapy Effects on Different Autophagy Risk Groups

Survival analysis conducted for different autophagy risk groups with and without adjuvant chemotherapy respectively showed that for early-stage CRC patients without adjuvant chemotherapy in the high autophagy risk group displayed poorer DFS in the GSE39582 cohort (**Figures 8A**, HR, 5[2.38–10.5]; p = 2.42 × 10–6). However, DFS showed no significant difference between high and low autophagy risk groups for early CRC patients with adjuvant chemotherapy (**Figure 8B**, p = 4.46 × 10–1). Similar results were observed in the TCGA cohort (**Figure 8C**, HR, 2.05 [0.9–4.67]; p = 8.22 × 10–2 and **Figure 8D**, p = 5.84 × 10–1). In addition, according to the cutoff values, we divided 48 kinds of cell lines related to colorectal cancer into high and low autophagy risk groups to test the effects of different chemotherapy drugs on cell lines and found that the half-maximal inhibitory concentrations (IC-50) of oxaliplatin, fluorouracil, and irinotecan were lower in the low autophagy risk group (**Figures 9A–C**). These results indicate that the model can be used to predict the drug sensitivity of cell lines to chemotherapeutic drugs under different autophagy risk groups.

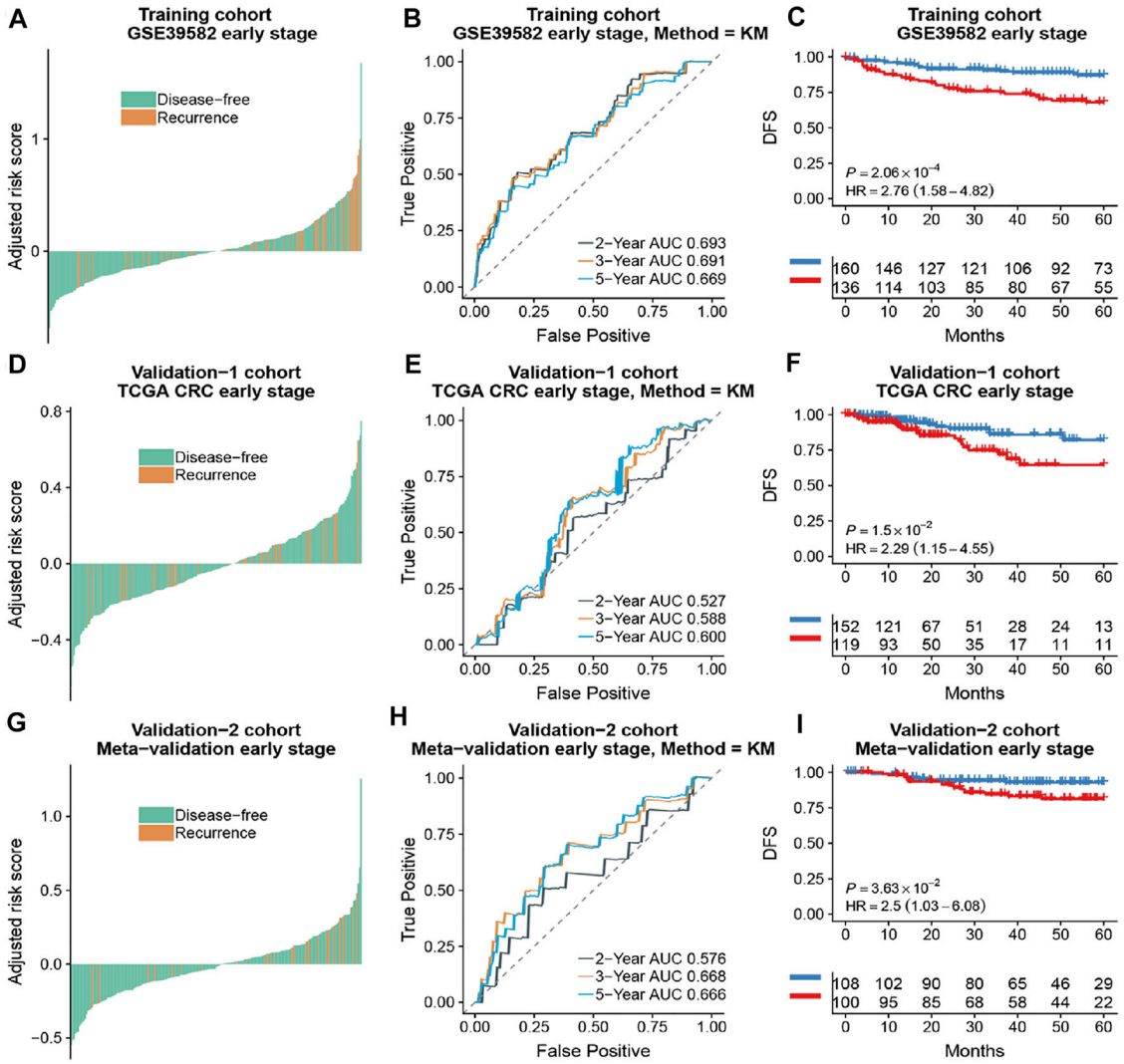

FIGURE 4 | The association of the ATG signature with DFS in early-stage (stage I and II) patients with CRC. Patients with CRC of the early stage were ranked by autophagy risk scores in the training cohort **(A)**, the TCGA cohort **(D)**, and the meta-validation cohort **(G)**. Time-dependent ROC curves for DFS in early stage (stage I and II) patients achieved with different durations in the training cohort **(B)**, the TCGA cohort **(E)**, and the meta-validation cohort **(H)**. Kaplan–Meier curves showed DFS of early-stage patients in low and high autophagy groups in the training cohort **(C)**, the TCGA cohort **(F)**, and meta-validation cohort **(I)**, respectively. *p* values comparing risk groups were calculated with the log-rank test. Hazard ratios (HRs) and 95% CIs are for low vs. high autophagy risk. CRC, colorectal cancer; ATG, autophagy-related gene; DFS, disease-free survival.

DISCUSSION

Thanks to the improved awareness of cancer screening, CRC is now detected at an early stage, resulting in a better rate of survival. Surgery without chemotherapy, which was deemed as the curative treatment, was carried out on the majority of patients with stage I/II colon cancer and in some cases of stage I/II rectal cancer (Miller et al., 2019). Indeed, it enabled prevention from unnecessary side effects of chemotherapy. Nevertheless, more than 20% of patients with stage I/II CRC who underwent surgical resection still suffered recurrence (Markle et al., 2010). Quite a few multigene prognostic signatures have been developed for CRC, but none of them graduated to the widespread application due to the uncertainty of prognostic accuracy. Accordingly, an effective prognostic model composed of multiple biomarkers to distinguish early-stage patients with a high risk of recurrence is crucial and necessary for elective adjuvant chemotherapy or other targeted treatments.

Emerging studies have revealed that autophagy functions diversely in the development, maintenance, and progression of tumors. While autophagy may prevent cellular cancerous transformation in normal tissue, it acts as a survival mechanism in established tumors, especially under stress conditions and in response to chemotherapy (Mokarram et al., 2017). Several autophagy inhibitors and activators have been brought up as improved chemotherapeutic options for cancer

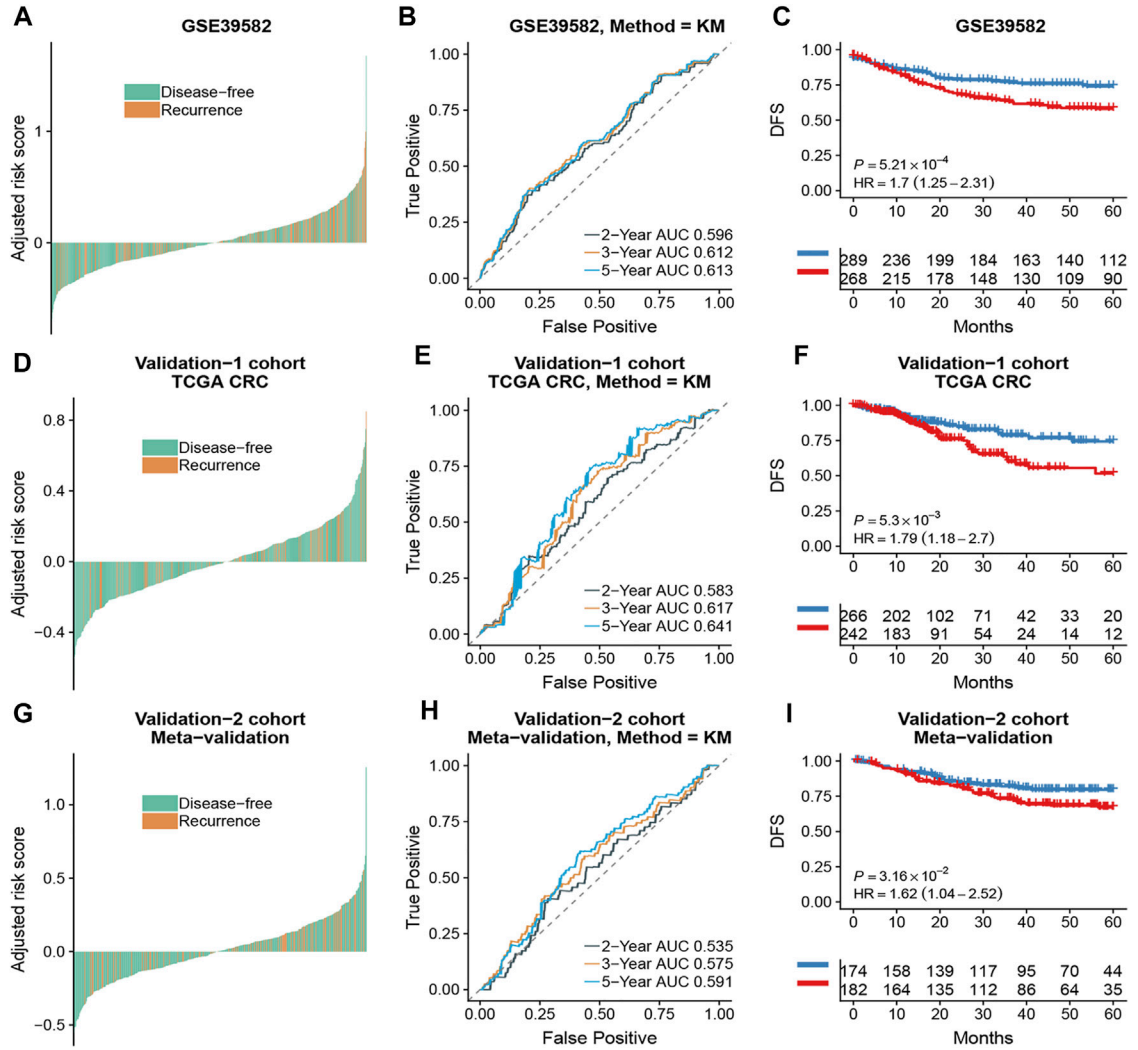

FIGURE 5 | The association of the ATG signature with DFS in CRC patients with all stages. Patients with CRC of the early stage were ranked by autophagy risk scores in the GSE39582 cohort **(A)**, the TCGA cohort **(D)**, and the meta-validation cohort **(G)**. Time-dependent ROC curves for DFS in CRC patients were achieved with different durations in the GSE39582 cohort **(B)**, the TCGA cohort **(E)**, and the meta-validation cohort **(H)**. Kaplan–Meier curves showed DFS of CRC patients in low and high autophagy groups in the GSE39582 cohort **(C)**, TCGA cohort **(F)**, and meta-validation cohort **(I)**, respectively. *p* values comparing risk groups were calculated with the log-rank test. Hazard ratios (HRs) and 95% CIs are for low vs. high autophagy risk. CRC, colorectal cancer; ATG, autophagy-related gene; DFS, disease-free survival.

TABLE 3 | C-index for autophagy risk compared with Oncotype DX colon in three cohorts.

Cohorts	Autophagy risk		Oncotype DX colon	
	C-index	95% CI	C-index	95% CI
GSE39582	0.74	0.63–0.85	0.65	0.53–0.77
TCGA	0.70	0.54–0.85	0.61	0.44–0.77
Meta-validation	0.70	0.51–0.89	0.62	0.43–0.82

TCGA, The Cancer Genome Atlas.

treatment (Koustas et al., 2019), but without sufficient clinically significant results, especially in CRC. Accordingly, further investigation on the biological mechanism of autophagy in the tumor microenvironment deserves attention, and more targets associated with autophagy await to be found.

In our study, we developed a prognostic model comprised of 10 ATGs for stage I/II CRC. This ATG signature, which classified patients into high and low autophagy risk groups, demonstrated a significant difference in 5-year DFS. The C-index of the ATG signature exhibited a good clinical predicting fitness superior to the Oncotype DX colon. Validation results suggested that the ATG signature could successfully predict DFS for stage I/II CRC patients after treatment. This novel model enabled us to identify CRC patients with high autophagy risk which stood for increased risk of tumor recurrence. Surprisingly, despite the original intention for DFS prediction of early-stage CRC patients, the ATG signature also showed a significant effect in the prediction

TABLE 4 | Univariate and multivariate analyses of ATG signature and clinical and pathologic factors in validation cohorts.

Characteristic	GSE39582				TCGA				Meta-validation			
	Univariate		Multivariate		Univariate		Multivariate		Univariate		Multivariate	
	HR (95%CI)	*p*	HR (95%CI)	*p*	HR (95%CI)	*p*	HR (95%CI)	*p*	HR (95%CI)	*p*	HR (95%CI)	*p*
ATG signature	2.76 (1.58–4.82)	0.00021	2.66 (1.52–4.65)	0.0006	2.27 (1.14–4.51)	0.016	2.27 (1.14–4.51)	0.016	2.50 (1.03–6.08)	0.036	2.50 (1.03–6.08)	0.036
Gender	1.51 (0.87–2.61)	0.14			1.29 (0.65–2.53)	0.46			0.86 (0.38–1.96)	0.73		
Age	1.01 (0.99–1.03)	0.4			1.01 (0.98–1.04)	0.46			0.97 (0.94–1.00)	0.08		
Tumor location	1.09 (0.63–1.88)	0.75			0.94 (0.48–1.83)	0.85			1.28 (0.55–2.96)	0.56		
TNM stage	7.61 (1.07–53.86)	0.016	7.19 (1.01–51.32)	0.049	1.74 (0.72–4.20)	0.22			2.67 (0.63–11.38)	0.17		
MMR status	1.60 (0.68–3.75)	0.28			0.78 (0.40–1.54)	0.47						
CIMP positive	0.99 (0.46–2.11)	0.97										
CIN positive	1.60 (0.71–3.62)	0.26										
TP53 mutation	1.40 (0.77–2.54)	0.26										
KRAS mutation	1.26 (0.74–2.13)	0.4			0.82 (0.17–4.07)	0.81						

ATG, autophagy-related gene; MMR, mismatch repair; CIMP, CpG island methylator phenotype; CIN, chromosomal instability; HR, hazard ratio; CI, confident interval.

for all stages. Therefore, this model could be applied to predict tumor recurrence in all CRC patients regardless of tumor stage.

As we looked over the genes within the ATG signature, some of them have been found correlated with CRC but mostly bear context-dependent biological functions in cancers, similar to autophagy. For example, the cytosolic histone deacetylase 6 (HDAC6) served as a tumor suppressor in hepatocellular carcinogenesis (Yang et al., 2019), while another study revealed that the HDAC6 inhibitor significantly suppressed the proliferation and viability and induced apoptosis in CRC cells, where autophagy activation was observed (Chen et al., 2019). Elevated Sirtuin 2 (SIRT2) was found to be associated with poor prognosis in CRC patients *via* its participation in tumor angiogenesis (Hu et al., 2019). Meanwhile, in a separate study SIRT2 was found to be downregulated in colon cancer biopsies compared to adjacent noncancerous tissues, and overexpression of SIRT2 inhibited the proliferation and metastatic progression of SW480 cells (Zhang et al., 2017). In terms of autophagy-related functions, a previous investigation reported that in response to oxidative stress or serum starvation, SIRT2 dissociated as acetylated FOXO1, which later bound to autophagy protein 7 (ATG7) and induced autophagy in tumors (Zhao et al., 2010). As these inconsistencies make it difficult to clarify the role of autophagy in CRC, we further investigated the biological functions of the ATG signature, expecting to find some clues in the autophagy-related functions in tumors.

GSEA revealed that the ATG signature included genes that were robustly involved in multiple immune/inflammatory pathways including IL6/JAK/STAT3, IL2/STAT5, IFN-α, IFN-γ, and TNF-α/NF-κB, and the inflammatory response presented a particular relation to CRC proliferation or prognosis as previous studies revealed (Nichols et al., 1994; Eguchi et al., 2003; De Simone et al., 2015; Park et al., 2017; Giordano et al., 2019). As our findings suggested that a high autophagy risk score correlated with the downregulation of these immune/inflammatory pathways, we speculated that autophagy might play a role in CRC tumorigenesis and tumor proliferation *via* an anti-immune/anti-inflammatory response. Moreover, increased infiltration of Tregs and decreased infiltration of M1 macrophages observed in the high autophagy risk group during immune infiltration analysis seemingly catered to the anti-immune/anti-inflammatory response. Tregs are known to suppress both antibody-mediated and cell-mediated immune responses (Wing and Sakaguchi, 2010). The pro-inflammatory M1 macrophages, a phenotype of tumor-associated macrophages (TAMs), correlated with a better prognosis in CRC patients for its tumor-suppressing function (Edin et al., 2012; Shapouri-Moghaddam et al., 2018). By triggering an anti-immune or anti-inflammatory response, autophagy might promote polarization or re-polarization toward the M2 phenotype spontaneously and thus lead to the decrease of M1 infiltration observed. Previous studies have described the link between autophagy and macrophage polarization in the tumor microenvironment. For example, mTOR which functions as a conserved kinase protein in the regulation of autophagy also participates in the polarization of monocytes into TAMs (Chen et al., 2014). More new strategies targeting TAM polarization as well as autophagy await exploration, and further studies are needed to clarify the above speculations.

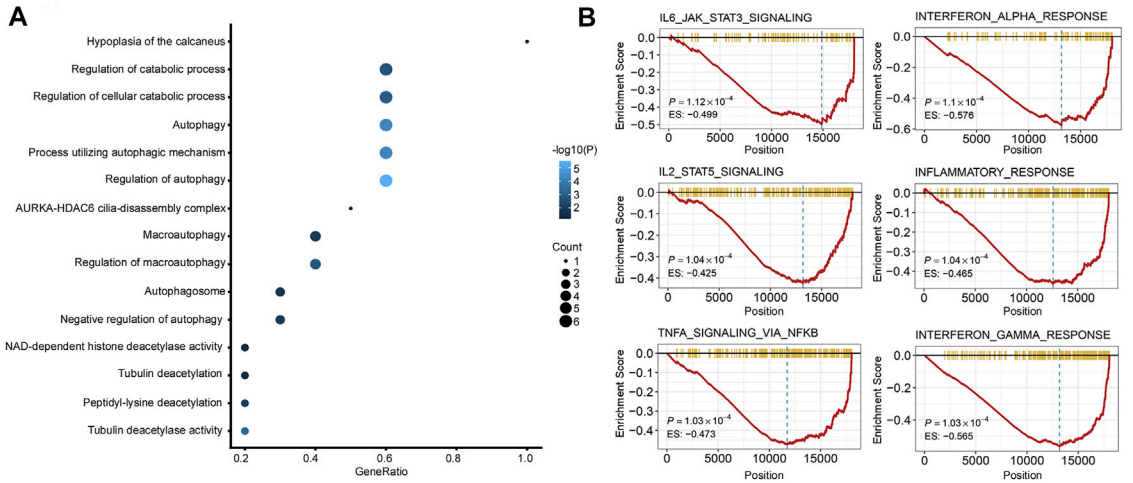

FIGURE 6 | Functional analysis of the 10 autophagy signature genes. **(A)**. Gene ontology (GO) of the 10 autophagy signature genes. **(B)**. GSEA showed some immune/inflammatory pathways downregulated in high autophagy risk patients.

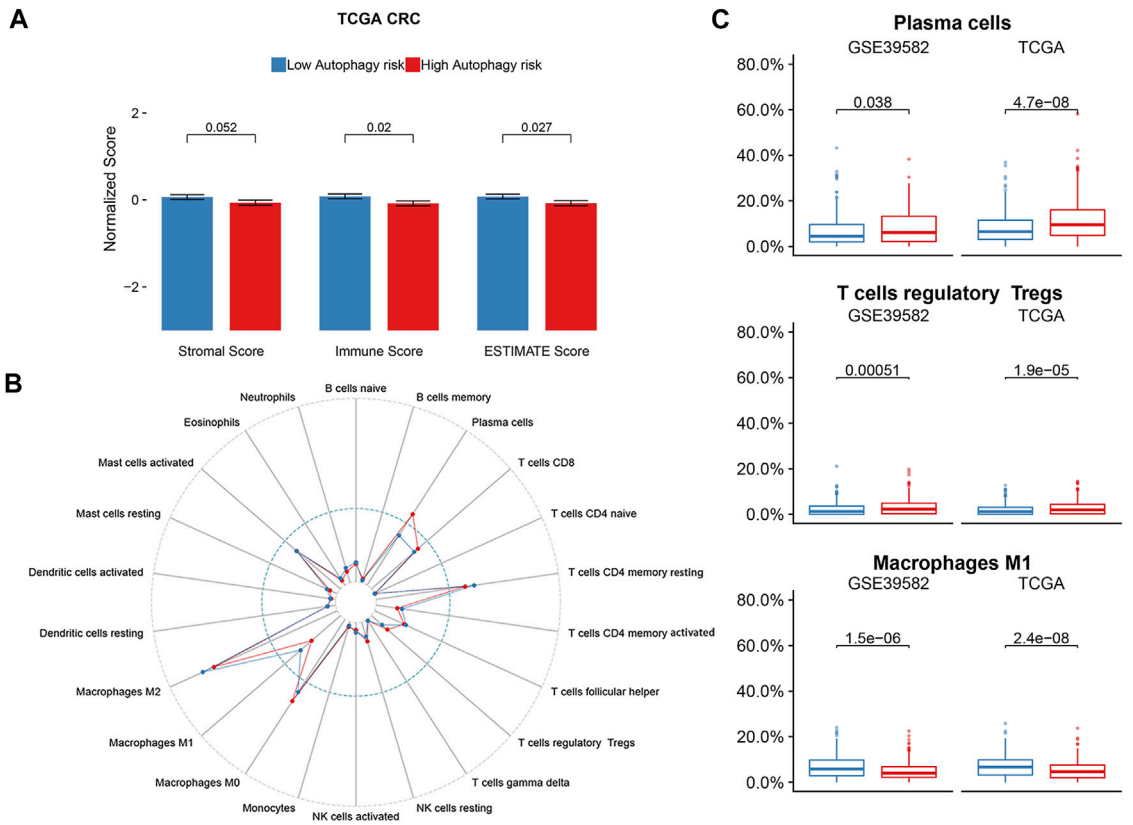

FIGURE 7 | Immune infiltration status of ATG signature risk groups. **(A)** ESTIMATE algorithm in the TCGA cohort. **(B)** Twenty-two immune cells abundance for different immune risk groups. **(C)** Plasma cells and T cells regulatory were enriched in the high-risk group. The macrophage M1 was enriched in the low-risk group. *p*-values are based on the Wilcoxon test. Hazard ratios (HRs) and 95% CIs are for low vs. high autophagy risk. CRC, colorectal cancer; DFS, disease-free survival.

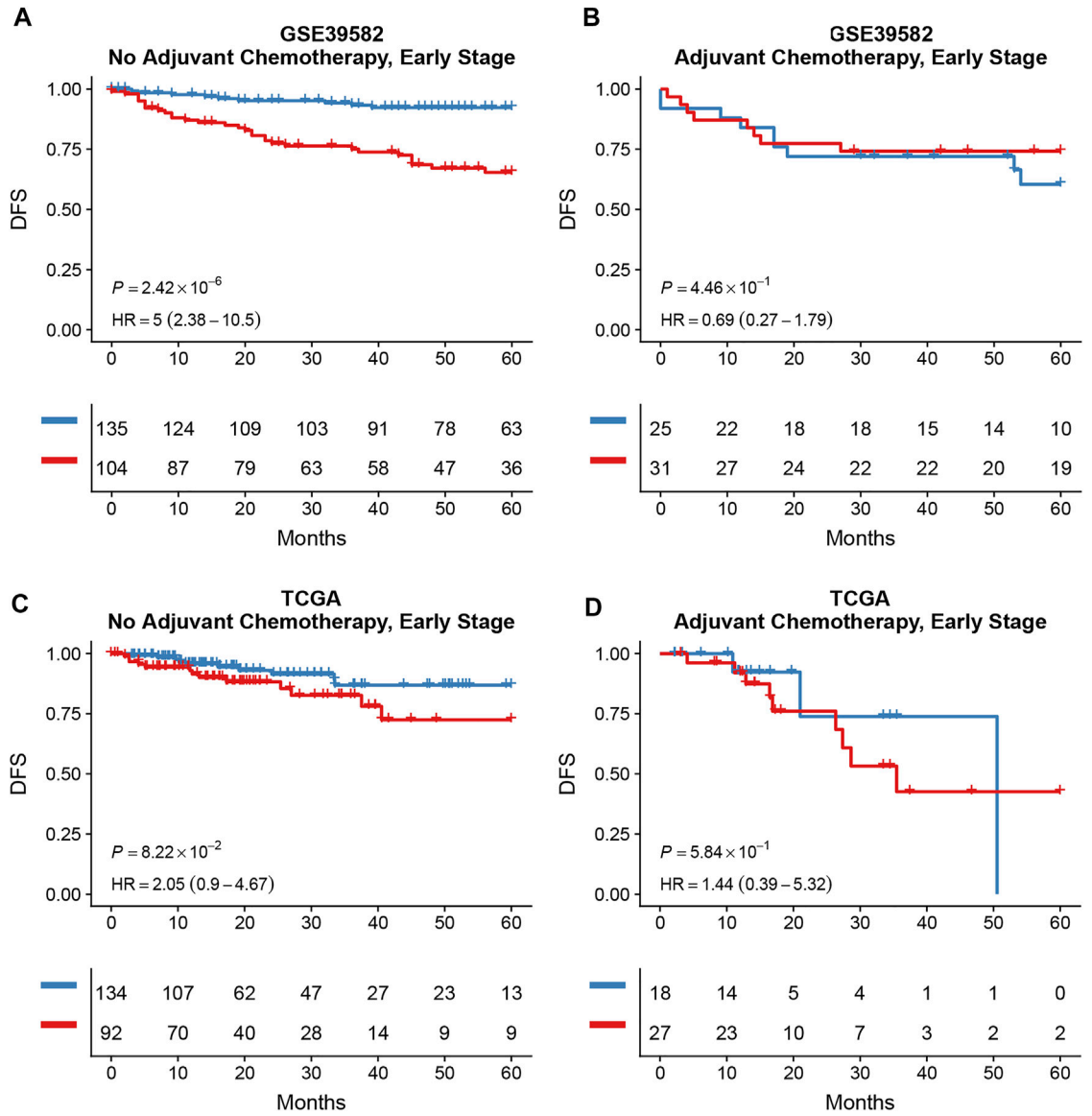

FIGURE 8 | Effects of adjuvant chemotherapy on DFS between different autophagy risk groups in early stage (stage I and II) patients with CRC. Kaplan–Meier curves showed DFS of early-stage patients between low and high autophagy groups with and without adjuvant chemotherapy respectively in the GSE39582 **(A,B)** cohort and the TCGA cohort **(C,D)**. *p* values comparing risk groups were calculated with the log-rank test.

Upon the prognostic prediction, clinicians could thus make an informed decision regarding supplementary treatment regimens. For example, we can selectively apply more aggressive chemotherapy strategies to early CRC patients of high autophagy risk groups. As our statistics suggest, DFS showed no significant difference between high and low autophagy risk groups for early CRC patients with adjuvant chemotherapy. However, the predicted results of the IC-50 of the colorectal cancer-associated cell lines showed a higher sensitivity to chemotherapy in the low autophagy risk group. This may be due to the limitations of retrospective studies, in which clinicians choose chemotherapy based on the patient's condition rather than random assignment. Cell experiments can reduce heterogeneity. It is hard to state whether those antitumor agents of chemotherapy are inhibiting autophagy or not. However, combined with our findings that a high autophagy risk score correlated with the downregulation of these immune/inflammatory pathways, chemotherapy may prevent tumor proliferation or recurrence by triggering specific immune/inflammatory responses or modifying the tumor microenvironment through TAMs. Besides, it is possible that chemotherapy just reduces the impact of autophagy on tumor relapse through massive, indiscriminate cell killing, and immunosuppression. Possible mechanisms will be explored in the further basic experiment and prospective clinical studies.

However, our prognostic ATG signature relies on the gene expression profiles from microarray platforms, which makes it too

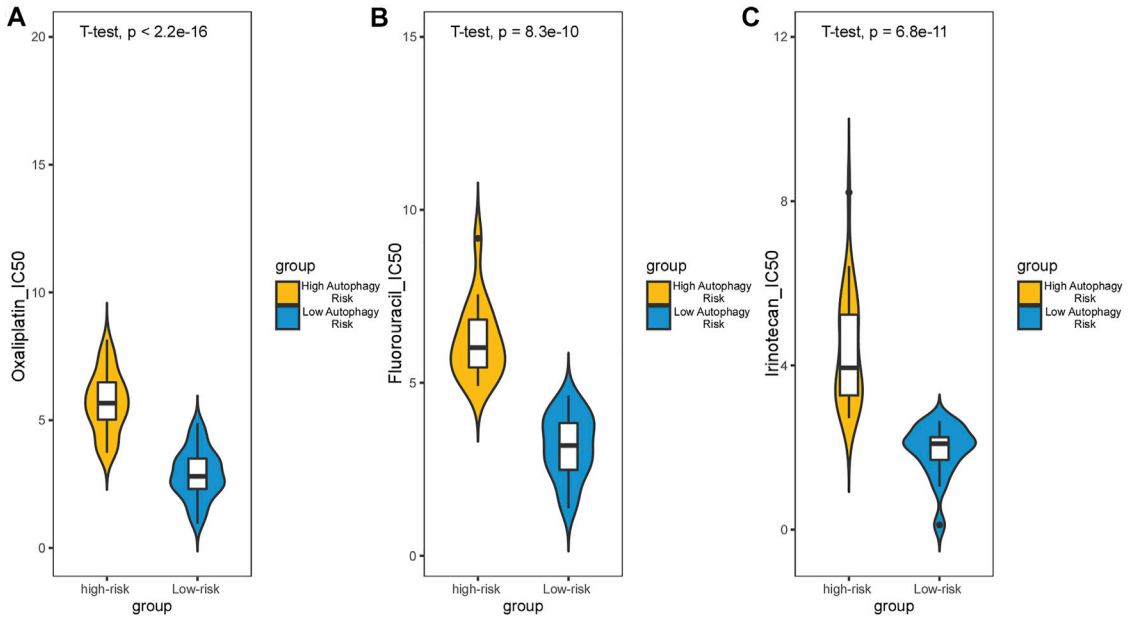

FIGURE 9 | Effect of chemotherapy drugs on colorectal cancer cell lines from different autophagy risk groups. IC-50 of oxaliplatin **(A)**, fluorouracil **(B)**, and irinotecan **(C)** were lower in the low autophagy risk group. *p* values were calculated by T-test.

expensive and time-consuming to popularize in clinical application. In addition to the dataset limitations from retrospective studies, further prospective clinical tests are recommended to validate our results. Despite the limitations, our research proposes a novel perspective to predicting the prognosis of early CRC patients and offers valuable insights into the relationship between autophagy, immune/inflammatory response, and tumorigenesis.

In conclusion, our study established a prognostic ATG signature that can effectively predict DFS for early-stage CRC patients. Meanwhile, our study also revealed the possibility that CRC patients in the high autophagy risk group might suffer tumor relapse *via* anti-immune/anti-inflammatory response. Moreover, higher sensitivity to chemotherapy in the low autophagy risk group was discovered in colorectal cancer-associated cell lines.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding authors.

ETHICS STATEMENT

Ethical review and approval was not required for the study on human participants in accordance with the local legislation and institutional requirements. Written informed consent for participation was not required for this study in accordance with the national legislation and the institutional requirements.

AUTHOR CONTRIBUTIONS

FG, Y-mR, and X-tL contributed conception and design of the study; X-tL, SQ, and D-jF collected the datasets; X-tL, SQ, M-yL, and XC performed the statistical analysis; X-tL, Q-nW, D-jF, M-yL, J-wC, J-rW, and Y-fZ participated in the data interpretation; Q-nW wrote the first draft of the manuscript; SQ and D-jF wrote sections of the manuscript. All authors contributed to manuscript revision, read and approved the submitted version.

FUNDING

This work was supported by National Key Clinical Discipline; and "5010 Clinical Research Program" of Sun Yat-sen University (grant number 2010012); and Natural Science Foundation of Guangdong Province, China (grant number 2020A1515010428); and Medical Science Research Grant from the Health Department of Guangdong Province (grant number A2018007).

ACKNOWLEDGMENTS

The authors would like to thank Yidu Cloud Technology Co., Ltd., for the assistance in the data processing.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2021.755789/full#supplementary-material

REFERENCES

Bray, F., Ferlay, J., Soerjomataram, I., Siegel, R. L., Torre, L. A., and Jemal, A. (2018). Global Cancer Statistics 2018: Globocan Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries. *CA: A Cancer J. Clinicians* 68 (6), 394–424. doi:10.3322/caac.21492

Carroll, R. G., and Martin, S. J. (2013). Autophagy in Multiple Myeloma: what Makes You Stronger Can Also Kill You. *Cancer Cell* 23 (4), 425–426. doi:10.1016/j.ccr.2013.04.001

Chen, M., Lin, Y., Liao, Y., Liou, J., and Chen, C. (2019). MPT0G612, a Novel HDAC6 Inhibitor, Induces Apoptosis and Suppresses IFN-γ-Induced Programmed Death-Ligand 1 in Human Colorectal Carcinoma Cells. *CANCERS* 11 (161710), 1617. doi:10.3390/cancers11101617

Chen, P., Cescon, M., and Bonaldo, P. (2014). Autophagy-mediated Regulation of Macrophages and its Applications for Cancer. *Autophagy* 10 (2), 192–200. [Journal Article; Research Support, Non-U.S. Gov't; Review]. doi:10.4161/auto.26927

Cheng, L., Han, T., Zhang, Z., Yi, P., Zhang, C., Zhang, S., et al. (2021). Identification and Validation of Six Autophagy-Related Long Non-coding RNAs as Prognostic Signature in Colorectal Cancer. *Int. J. Med. Sci.* 18 (1), 88–98. doi:10.7150/ijms.49449

De Simone, V., Franzè, E., Ronchetti, G., Colantoni, A., Fantini, M. C., Di Fusco, D., et al. (2015). Th17-type Cytokines, IL-6 and TNF-α Synergistically Activate STAT3 and NF-kB to Promote Colorectal Cancer Cell Growth. *Oncogene* 34 (27), 3493–3503. doi:10.1038/onc.2014.286

Dekker, E., Tanis, P. J., Vleugels, J. L. A., Kasi, P. M., and Wallace, M. B. (2019). Colorectal Cancer. *The Lancet* 394 (10207), 1467–1480. doi:10.1016/S0140-6736(19)32319-0

Edin, S., Wikberg, M. L., Dahlin, A. M., Rutegård, J., Öberg, Å., Oldenborg, P.-A., et al. (2012). The Distribution of Macrophages with a M1 or M2 Phenotype in Relation to Prognosis and the Molecular Characteristics of Colorectal Cancer. *PLoS One* 7 (10), e47045. doi:10.1371/journal.pone.0047045

Eguchi, J.-i., Hiroishi, K., Ishii, S., and Mitamura, K. (2003). Interferon-α and Interleukin-12 Gene Therapy of Cancer: Interferon-α Induces Tumor-specific Immune Responses while Interleukin-12 Stimulates Non-specific Killing. *Cancer Immunol. Immunother.* 52 (6), 378–386. doi:10.1007/s00262-002-0367-2

Giordano, G., Parcesepe, P., D'Andrea, M. R., Coppola, L., Di Raimo, T., Remo, A., et al. (2019). Jak/stat5-mediated Subtype-specific Lymphocyte Antigen 6 Complex, Locus G6d (Ly6g6d) Expression Drives Mismatch Repair Proficient Colorectal Cancer. *J. Exp. Clin. Cancer Res.* 38 (1). doi:10.1186/s13046-018-1019-5

Guinney, J., Dienstmann, R., Wang, X., de Reyniès, A., Schlicker, A., and Soneson, C. (2015). The Consensus Molecular Subtypes of Colorectal Cancer. *Nat. Med.* 21 (11), 1350–1356. [Journal Article; Research Support, N.I.H., Extramural; Research Support, Non-U.S. Gov't]. doi:10.1038/nm.3967

Hu, F., Sun, X., Li, G., Wu, Q., Chen, Y., Yang, X., et al. (2019). Inhibition of Sirt2 Limits Tumour Angiogenesis via Inactivation of the Stat3/vegfa Signalling Pathway. *Cel Death Dis.* 10 (1), 9. doi:10.1038/s41419-018-1260-z

Huang, Z., Liu, J., Luo, L., Sheng, P., Wang, B., Zhang, J., et al. (2019). Genome-wide Identification of a Novel Autophagy-Related Signature for Colorectal Cancer. [Journal Article]. *Dose Response* 17 (4), 711624285. doi:10.1177/1559325819894179

Kannarkatt, J., Joseph, J., Kurniali, P. C., Al-Janadi, A., and Hrinczenko, B. (2017). Adjuvant Chemotherapy for Stage Ii colon Cancer: a Clinical Dilemma. [Journal Article; Review]. *J. Oncol. Pract.* 13 (4), 233–241. doi:10.1200/JOP.2016.017210

Kim, S. K., Kim, S. Y., Kim, C. W., Roh, S. A., Ha, Y. J., Lee, J. L., et al. (2019). A Prognostic index Based on an Eleven Gene Signature to Predict Systemic Recurrences in Colorectal Cancer. [Journal Article; Research Support, Non-U.S. Gov't]. *Exp. Mol. Med.* 51 (10), 1–12. doi:10.1038/s12276-019-0319-y

Kongara, S., and Karantza, V. (2012). The Interplay between Autophagy and Ros in Tumorigenesis. *[Journal Article]Frontiers Oncol.* 2, 171. doi:10.3389/fonc.2012.00171

Koustas, E., Sarantis, P., Kyriakopoulou, G., Papavassiliou, A. G., and Karamouzis, M. V. (2019). The Interplay of Autophagy and Tumor Microenvironment in Colorectal Cancer—Ways of Enhancing Immunotherapy Action. *Cancers* 11 (4), 533. doi:10.3390/cancers11040533

Liberzon, A., Birger, C., Thorvaldsdóttir, H., Ghandi, M., Mesirov, J., and Tamayo, P. (2015). The Molecular Signatures Database Hallmark Gene Set Collection. *Cel Syst.* 1 (6), 417–425. doi:10.1016/j.cels.2015.12.004

Liberzon, A., Subramanian, A., Pinchback, R., Thorvaldsdottir, H., Tamayo, P., and Mesirov, J. P. (2011). Molecular Signatures Database (Msigdb) 3.0. *Bioinformatics* 27 (12), 1739–1740. doi:10.1093/bioinformatics/btr260

Markle, B., May, E. J., and Majumdar, A. P. N. (2010). Do nutraceutics Play a Role in the Prevention and Treatment of Colorectal Cancer? *Cancer Metast. Rev.* 29 (3), 395–404. doi:10.1007/s10555-010-9234-3

Miller, K. D., Nogueira, L., Mariotto, A. B., Rowland, J. H., Yabroff, K. R., Alfano, C. M., et al. (2019). Cancer Treatment and Survivorship Statistics, 2019. *CAA Cancer J. Clinicians* 69 (5), 363–385. doi:10.3322/caac.21565

Mokarram, P., Albokashy, M., Zarghooni, M., Moosavi, M. A., Sepehri, Z., Chen, Q. M., et al. (2017). New Frontiers in the Treatment of Colorectal Cancer: Autophagy and the Unfolded Protein Response as Promising Targets. *Autophagy* 13 (5), 781–819. doi:10.1080/15548627.2017.1290751

Newman, A. M., Liu, C. L., Green, M. R., Gentles, A. J., Feng, W., Xu, Y., et al. (2015). Robust Enumeration of Cell Subsets from Tissue Expression Profiles. *Nat. Methods* 12 (5), 453–457. doi:10.1038/nmeth.3337

Nichols, P. H., Ward, U., Ramsden, C. W., and Primrose, J. N. (1994). The Effect of 5-fluorouracil and Alpha Interferon and 5-fluorouracil and Leucovorin on Cellular Anti-tumour Immune Responses in Patients with Advanced Colorectal Cancer. *Br. J. Cancer* 70 (5), 946–949. doi:10.1038/bjc.1994.426

O'Connor, E. S., Greenblatt, D. Y., LoConte, N. K., Gangnon, R. E., Liou, J. I., Heise, C. P., et al. (2011). Adjuvant Chemotherapy for Stage Ii colon Cancer with Poor Prognostic Features. *J. Clin. Oncol.* 29 (25), 3381–3388. Journal Article; Research Support, N.I.H., Extramural; Research Support, Non-U.S. Gov't]. doi:10.1200/JCO.2010.34.3426

Park, J. H., Van Wyk, H., McMillan, D. C., Quinn, J., Clark, J., Roxburgh, C. S. D., et al. (2017). Signal Transduction and Activator of Transcription-3 (Stat3) in Patients with Colorectal Cancer: Associations with the Phenotypic Features of the Tumor and Host. *Clin. Cancer Res.* 23 (7), 1698–1709. doi:10.1158/1078-0432.CCR-16-1416

Rosich, L., Colomer, D., and Roué, G. (2013). Autophagy Controls Everolimus (Rad001) Activity in Mantle Cell Lymphoma. *Autophagy* 9 (1), 115–117. doi:10.4161/auto.22483

Shapouri-Moghaddam, A., Mohammadian, S., Vazini, H., Taghadosi, M., Esmaeili, S. A., Mardani, F., et al. (2018). Macrophage Plasticity, Polarization, and Function in Health and Disease. [Journal Article; Review]. *J. Cel. Physiol.* 233 (9), 6425–6440. doi:10.1002/jcp.26429

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., et al. (2005). Gene Set Enrichment Analysis: a Knowledge-Based Approach for Interpreting Genome-wide Expression Profiles. *P. Natl. Acad. Sci. Usa.* 102 (43), 15545–15550. doi:10.1073/pnas.0506580102

Van Der Stok, E. P., Spaander, M. C. W., Grünhagen, D. J., Verhoef, C., and Kuipers, E. J. (2017). Surveillance after Curative Treatment for Colorectal Cancer. *Nat. Rev. Clin. Oncol.* 14 (5), 297–315. doi:10.1038/nrclinonc.2016.199

White, E., Mehnert, J. M., and Chan, C. S. (2015). Autophagy, Metabolism, and Cancer. *Clin. Cancer Res.* 21 (22), 5037–5046. [Journal Article; Research Support, N.I.H., Extramural; Research Support, Non-U.S. Gov't; Review]. doi:10.1158/1078-0432.CCR-15-0490

Wing, K., and Sakaguchi, S. (2010). Regulatory T Cells Exert Checks and Balances on Self Tolerance and Autoimmunity. *Nat. Immunol.* 11 (1), 7–13. doi:10.1038/ni.1818

Yang, H. D., Kim, H. S., Kim, S. Y., Na, M. J., Yang, G., Eun, J. W., et al. (2019). Hdac6 Suppresses Let-7i-5p to Elicit Tsp1/cd47-Mediated Anti-tumorigenesis and Phagocytosis of Hepatocellular Carcinoma. *Hepatology* 70 (4), 1262–1279. doi:10.1002/hep.30657

Zhang, L. L., Zhan, L., Jin, Y. D., Min, Z. L., Wei, C., Wang, Q., et al. (2017). Sirt2 Mediated Antitumor Effects of Shikonin on Metastatic Colorectal Cancer. *Eur. J. Pharmacol.* 797, 1–8. doi:10.1016/j.ejphar.2017.01.008

Zhao, H., Huang, C., Luo, Y., Yao, X., Hu, Y., Wang, M., et al. (2021). A Correlation Study of Prognostic Risk Prediction for Colorectal Cancer Based on Autophagy Signature Genes. [Journal Article]. *Front. Oncol.* 11, 595099. doi:10.3389/fonc.2021.595099

Zhao, Y., Yang, J., Liao, W., Liu, X., Zhang, H., Wang, S., et al. (2010). Cytosolic Foxo1 Is Essential for the Induction of Autophagy and Tumour Suppressor Activity. *Nat. Cel Biol.* 12 (7), 665–675. doi:10.1038/ncb2069

Zhi, X., Feng, W., Rong, Y., and Liu, R. (2018). Anatomy of Autophagy: from the Beginning to the End. *Cell. Mol. Life Sci.* 75 (5), 815–831. [Journal Article; Research Support, Non-U.S. Gov't; Review]. doi:10.1007/s00018-017-2657-z

Zhou, Z., Mo, S., Dai, W., Ying, Z., Zhang, L., Xiang, W., et al. (2019). Development and Validation of an Autophagy Score Signature for the Prediction of post-operative Survival in Colorectal Cancer. [Journal Article]. *Front. Oncol.* 9, 878. doi:10.3389/fonc.2019.00878

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers in Genetics

ORIGINAL RESEARCH
published: 08 December 2021
doi: 10.3389/fgene.2021.790426

Key Regulatory Differentially Expressed Genes in the Blood of Atrial Septal Defect Children Treated With Occlusion Devices

Bo-Ning Li[1†], Quan-Dong Tang[2†], Yan-Lian Tan[3†], Liang Yan[3], Ling Sun[4], Wei-Bing Guo[4,5], Ming-Yang Qian[4], Allen Chen[6], Ying-Jun Luo[6], Zhou-Xia Zheng[6], Zhi-Wei Zhang[4], Hong-Ling Jia[3]* and Cong Liu[1]**

[1]The Department of Cardiology, Shenzhen Children's Hospital, Shenzhen, China, [2]Department of Pathophysiology, The Key Immunopathology Laboratory of Guangdong Province, Shantou University Medical College, Shantou, China, [3]Department of Medical Biochemistry and Molecular Biology, School of Medicine, Jinan University, Guangzhou, China, [4]Guangdong Cardiovascular Institute, Guangdong Provincial People's Hospital, Guangdong Academy of Medical Sciences, Guangzhou, China, [5]The Department of Cardiology, Zhong Shan Affiliated Hospital of Xiamen University, Xiamen, China, [6]Guangzhou Mendel Genomics and Medical Technology Co., Guangzhou, China

Atrial septal defects (ASDs) are the most common types of cardiac septal defects in congenital heart defects. In addition to traditional therapy, interventional closure has become the main treatment method. However, the molecular events and mechanisms underlying the repair progress by occlusion device remain unknown. In this study, we aimed to characterize differentially expressed genes (DEGs) in the blood of patients treated with occlusion devices (metal or poly-L-lactic acid devices) using RNA-sequencing, and further validated them by qRT-PCR analysis to finally determine the expression of key mediating genes after closure of ASD treatment. The result showed that total 1,045 genes and 1,523 genes were expressed differently with significance in metal and poly-L-lactic acid devices treatment, respectively. The 115 overlap genes from the different sub-analyses are illustrated. The similarities and differences in gene expression reflect that the body response process involved after interventional therapy for ASDs has both different parts that do not overlap and the same part that crosses. The same portion of body response regulatory genes are key regulatory genes expressed in the blood of patients with ASDs treated with closure devices. The gene ontology enrichment analysis showed that biological processes affected in metal device therapy are immune response with CXCR4 genes and poly-L-lactic acid device treatment, and the key pathways are nuclear-transcribed mRNA catabolic process and proteins targeting endoplasmic reticulum process with ribosomal proteins (such as RPS26). We confirmed that CXCR4, TOB1, and DDIT4 gene expression are significantly downregulated toward the pre-therapy level after the post-treatment in both therapy groups by qRT-PCR. Our study suggests that the potential role of CXCR4, DDIT4, and TOB1 may be key regulatory genes in the process of endothelialization in the repair progress of ASDs, providing molecular insights into this progress for future studies.

Keywords: atrial septal defects, interventional closure, differentiated expressed genes, RNA-sequencing analysis, congenital heart defects

OPEN ACCESS

Edited by:
Chuan-Le Xiao,
Sun Yat-sen University, China

Reviewed by:
Fu Lijun,
Shanghai Children's Medical Center, China
Yu Wang,
Emory University, United States
Hua Chen,
Beijing Institute of Genomics (CAS), China

***Correspondence:**
Zhi-Wei Zhang
drzhangzw@sohu.com
Hong-Ling Jia
jiahongling@aliyun.com
Cong Liu
szliucong@163.com

[†]These authors have contributed equally to this work

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

Received: *06 October 2021*
Accepted: *10 November 2021*
Published: *08 December 2021*

Citation:
Li B-N, Tang Q-D, Tan Y-L, Yan L, Sun L, Guo W-B, Qian M-Y, Chen A, Luo Y-J, Zheng Z-X, Zhang Z-W, Jia H-L and Liu C (2021) Key Regulatory Differentially Expressed Genes in the Blood of Atrial Septal Defect Children Treated With Occlusion Devices. Front. Genet. 12:790426. doi: 10.3389/fgene.2021.790426

INTRODUCTION

Atrial septal defects (ASDs) are the most common types of congenital heart defects (CHDs) and typically present with left to right shunts, which account for up to 10% and 40% of all CHDs, respectively (Penny and Vick, 2011; Rao and Harris, 2017). The patients with ASDs may exhibit poor growth and development, decreased activity tolerance, repeated respiratory infections, and hyperhidrosis, and they are accompanied by heart enlargement, increased pulmonary circulation pressure and resistance, heart failure, and atrial arrhythmia (Huang et al., 2013; Leppert et al., 2016; Karunanithi et al., 2017; Wu et al., 2018; Pillai et al., 2019). Surgery as the traditional method to treat ASDs has several disadvantages including large trauma, long recovery time and permanent scar left. It is more serious that residual septal defects are frequently associated with surgery-related side-effect complications, such as reoperation, infection, sternotomy scarring, and even death (Gaynor et al., 2001; Oses et al., 2010). To avoid those side effects of surgery, interventional closure has been developed to the main treatment to septal defects (Huang et al., 2013; Shimpo et al., 2013; Morray, 2019).

During the past four decades, several nondegradable types of occluders based on shape memory alloys have been used in clinical settings. Compared with surgery, interventional therapy has become the first choice for ASDs with the advantages of less trauma, less pain, no scar, short hospital stays, less complications, and no need for blood transfusion and extracorporeal circulation. With the recent development of occlusion devices and improved implantation techniques, the use of transcatheter closure of ASDs has increased over the years, considering that the permanent existence of foreign non-degradable materials *in vivo* can cause many potential complications in the long term (Luermans et al., 2010; Abaci et al., 2013). On the other hand, the use of biodegradable materials in the construction of occluders may overcome the drawbacks of metal devices (Shi et al., 2019). So, the research and development of biodegradable occluders has emerged as a crucial field for interventional treatment of ASDs. However, the main biological phenomena triggered after treatment with either degradable occluders or metal occluders are the same, including cardiac remodeling phenomena triggered by hemodynamic changes and biological responses induced by occluders. Because metal and biodegradable occluders are derived from different materials and have different structures, the phenomenon of cardiac remodeling after treatment behaves differently. The same point lies in the biological response of the body induced by the occluder. In general, the occlusion device is used to provide a temporary scaffold for tissue endothelialization (Tang et al., 2018). Some studies revealed that endothelialization is related to cell proliferation, cell migration, and cell junction (Bazzoni and Dejana, 2004; Dejana 2004). The previous studies showed that normal expression of genes encoding transcription factors, cell signaling molecules, and structural proteins are important for heart development (Williams et al., 2019). It was also reported that both metal and biodegradable occluders are beneficial to endothelial cell coverage by histological and electron microscopic examinations (Li et al., 2019). However, whether the occlusion device affects ASD repair by regulating the expression of key genes remains unclear.

RNA-sequencing (RNA-Seq) is a useful method to explore the molecular events in many different samples, including blood, cells, and tissues. In this study, we performed differentially expressed genes (DEGs) analysis on RNA-sequencing data in blood samples of patients with the occlusion device (metal or biodegradable device) post-treatment group compared to the pre-therapy group. Combining with the expression level validation of DEGs by quantitative real-time PCR, our study aims to discover DEGs and overlap genes in ASDs, to illustrate the potential role of specific overlap genes and its function on biological processes.

MATERIALS AND METHODS

Preparation of Samples

Between January 2019 and December 2019, pediatric patients undergoing closure of secundum ASD with either metal occluder or PLLA occluder in our hospital were included in this study. Indications for ASD closure were as follows: an ASD ≥5 mm and ≤30 mm in diameter, with sufficient rims of atrial tissue (superior to the coronary sinus, superior/inferior vena cava, and pulmonary vein by 5 mm and superior to the mitral valve by 7 mm), signs of right ventricular volume overload, and/or evidence of significant left-to-right shunting (Qp:Qs ≥ 1.5:1). Patients with other congenital or significant cardiac defects, history of ASD repair, metal implant, or PLLA implant were excluded from the study. The study was approved by the Committee on the Ethics of Shenzhen Children's Hospital (202000903), and written informed consent was obtained from all guardians. The samples used for RNA-sequencing analysis were collected from Shenzhen Children's Hospital, including two patient samples before and after metal device therapy (Cera™, Lifetech Scientific, Shenzhen, China), three patient samples before and after poly-L-lactic acid (PLLA) device therapy (Absnow™, Lifetech Scientific, Shenzhen, China), two patient samples before metal device therapy and one patient sample after PLLA device therapy, and four samples from healthy volunteers; the basic clinical characteristics of these children are listed in **Table 1A**. For qRT-PCR, the blood sample of treatment groups was randomly selected, including 11 samples before and after PLLA device therapy, 10 samples before and after metal device therapy, and 8 healthy people as the control group. The basic characteristics of the children for qRT-PCR analysis are listed in **Table 1B**. All patients with ASDs underwent interventional therapy and samples were collected that day before and 30 days after the intervention.

RNA Extraction

The RNA was extracted from whole blood sample following the Trizol reagent manual (Invitrogen Life Technologies, Carlsbad, CA). In brief, 5 ml of Trizol reagent was added to 1 ml of whole blood sample for 10 min on ice, and then RNA was precipitated in 1:1 isopropanol/Trizol (v/v) and 1 μl of glycogen at −20°C overnight followed by use for mRNA-sequencing.

Library Preparation

cDNA library preparation: total RNA (200 ng) was used to prepare cDNA libraries using the NEBNext Ultra RNA library prep kit for Illumina (New England Biolabs) following the

TABLE 1A | Basic characteristics of the patient samples of atrial septal defects used for RNA-sequencing analysis.

Samples source	Cases	Male/ Female	Before/ After therapy	Before/ After therapy	Age (months)	Weight (kg)	Sptal defect (mm)	Qp/ Qs	Occluder (mm)
PLLA device therapy patients	1	F	B	A	56	17.4	9	2.1	12
	2	F	B	A	68	14.2	9	1.9	14
	3	F	B	A	52	17.3	8	2.0	16
	4	M	—	A	13	10.3	6	1.8	12
Metal device therapy patients	5	F	B	A	10	9.3	7	1.9	12
	6	M	B	A	19	11.5	10	2.3	14
	7	F	B	—	29	13	8	1.7	12
	8	M	B	—	56	19	6	1.6	10
Healthy volunteers	9	M	—	—	11	8	—	—	—
	10	F	—	—	13	11	—	—	—
	11	F	—	—	19	12.2	—	—	—
	12	F	—	—	16	10.2	—	—	—

PLLA: poly-L-lactic acid; Qp/Qs: Pulmonary-to-Systemic-Blood-Flow Ratio.

TABLE 1B | Basic characteristics of the patient samples of atrial septal defects used for qRT-PCR.

Samples source	Cases	Male/Female	Age (months)	Weight (kg)	Sptal defect (mm)	Qp/Qs	Occluder (mm)
PLLA device therapy patients	*N* = 11	4/7	34.5 ± 20.2	12.8 ± 4.6	13.5 ± 5.0	2.1 ± 0.4	18.0 ± 5.4
Metal device therapy patients	*N* = 10	4/6	29.1 ± 21.1	12.9 ± 3.9	12.1 ± 0.8	1.9 ± 0.3	16.0 ± 1.5
Healthy volunteers	*N* = 8	2/6	24.1 ± 15.1	12.0 ± 3.0	—	—	—

PLLA: poly-L-lactic acid; Qp/Qs: Pulmonary-to-Systemic-Blood-Flow Ratio.

manufacturer's protocol. Quality and integrity of the tagged libraries were initially assessed with the HT DNA HiSens Reagent kit (Perkin Elmer) using a LabChip GX bioanalyzer (Caliper Life Sciences/Perkin Elmer). Tagged libraries were then sized and quantitated in duplicate (Agilent TapeStation system) using D1000 ScreenTape and reagents (Agilent). Sequencing was performed as PE150 on an Illumina NovaSeq 6000 sequencer. The high-quality reads that passed the Illumina filter were subjected to the subsequent bioinformatics analysis.

Transcriptome Profiling

Adapters and low-quality bases with the sequencing reads for each sample were preprocessed by fastp (https://academic.oup.com/bioinformatics/article/34/17/i884/5093234) with a default setting. Filtered reads were mapped to the latest version of human genome (*Homo sapiens*, GRCh38) by STAR (https://doi.org/10.1093/bioinformatics/bts635) aligner with parameters: --outSAMtype BAM SortedByCoordinate, and the mapping results were summarized into a gene expression matrix using featureCounts v1.6 (https://doi.org/10.1093/bioinformatics/btt656). Output data were then processed with customized R scripts.

Differential Expression Genes (DEGs) Analysis

Gene-level differential expression was analyzed using DESeq (https://doi.org/10.1089/omi.2011.0118) for the metal or poly-L-lactic acid device sample group, respectively. The ASDs pre-therapy or post-treatment were specified as the experimental design. Benjamini and Hochberg *p*-value adjustment methods were used for multiple comparisons. Parameter alpha (significance cutoff) was set to 0.1 and lfcThreadshold (log2 fold change threshold) was set to 0 following the best practice of DESeq pipeline. Genes with an absolute fold change (FC) greater two and a *p*-value less than 0.05 were selected for the downstream analysis.

Gene Ontology Enrichment Analysis

DEGs were annotated by pre-defined terminologies such as GO analysis, and over-representation analysis (ORA) was performed by clusterProfiler (https://doi.org/10.1089/omi.2011.0118).

Weighted Gene Co-Expression Network Analysis

Weighted Gene Co-Expression Network Analysis (WGCNA) was carried out to evaluate the correlation between genes and to classify highly correlated genes into the same module. The data submitted to the WGCNA R package (https://doi.org/10.1186/1471-2105-9-559) was firstly processed by differential expression analysis to filter out irrelevant information. The data submitted to WGCNA R package was firstly processed by Variance Stabilizing Transformation (VST) algorithm. The topological overlap measure (TOM) was employed to identify modules of highly co-expressed genes, and genes with high absolute correlations were clustered into the same modules by cutting the dendrogram into branches. The only number of genes that exceed 30 will be defined as a module. Then, pairwise correlations between gene modules and clinical datasets were calculated. Modules with

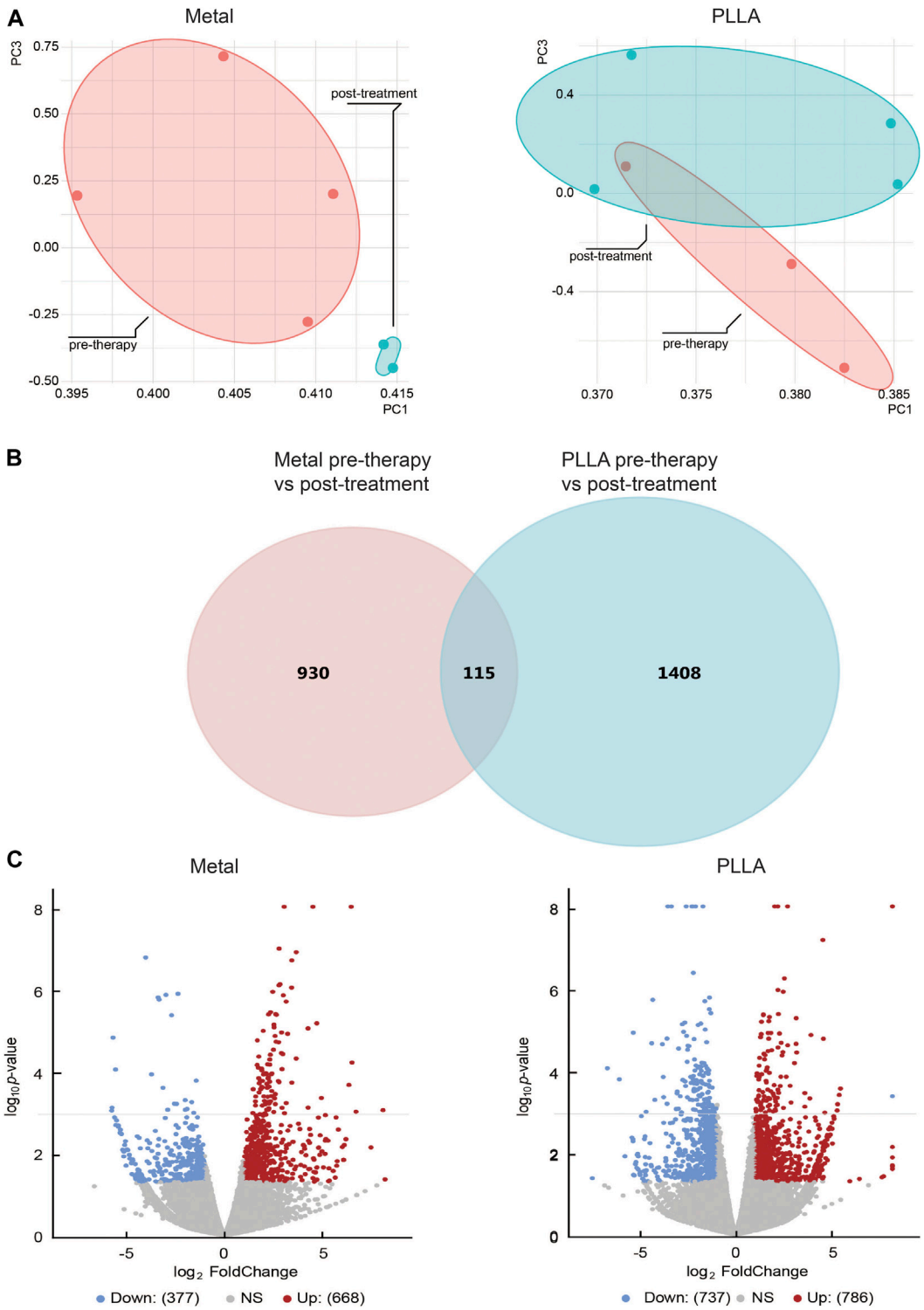

FIGURE 1 | The differential expression genes (DEGs) analysis of atrial septal defects patients before and after metal and PLLA device therapy. **(A)** Samples can be distinguished by principal component analysis (PCA). **(B)** Venn diagram showing the genes identified in PLLA and metal device after and before therapy. **(C)** Significantly changed genes were discovered from differentially expression analysis. Gene with a *p*-value less than 0.05 and an absolute fold change greater than 2 is considered as a significantly changed gene. In each panel, the blue dots represented downregulated genes and the red dots represented upregulated genes.

higher correlation will be merged ($r < 0.25$); each module was assigned to different colors for visualization.

Protein–Protein Interactions Analysis

PPIs are physical contacts of high specificity established between two or more protein molecules as a result of biochemical events steered by interactions that include electrostatic forces, hydrogen bonding, and the hydrophobic effect. PPI with known disease genes have been used to find new disease genes by identifying key core genes. We derived core genes by network connection scores to describe the module elements, including core and ring components. Functions of core genes were highly correlated with those of essential genes in the same modules.

Quantitative Reverse Transcriptase-Polymerase Chain Reaction Analysis

Total RNA was extracted with Trizol according to the manufacturer's instructions. Unique genomic DNA remover is combined with EasyScript® First-Strand cDNA Synthesis SuperMix to achieve simultaneous genomic DNA removal and cDNA synthesis. The cDNA levels were measured by SYBR Green in real-time PCR using the LightCycler. The housekeeping gene GAPDH was used as normalized in each individual sample and the $2^{-\Delta\Delta Ct}$ method was used to quantify relative expression changes. The sequences of specific primers used for qRT-PCR assays in this study are listed in **Supplementary Table S1**.

Statistical Analysis

Statistical significance was performed using Student's *t*-test and $p < 0.05$ was considered statistically significant.

RESULTS

The Expression Profile Diverse Before and After the Occlusion Device Therapy

Previous studies showed that the occlusion device is used to provide a temporary scaffold for tissue endothelialization,15 but whether it plays a role in biological processes is unclear. Therefore, we explore the effects of the occlusion device on biological processes by RNA-Seq. The DEGs analysis was performed for ASDs cases and healthy control. Principal component analysis and inspection of the first two principal components illustrate the presence of four groups of samples (**Figure 1A**). DEGs were tested utilizing two strategies. Firstly, the occlusion device post-treatment pools were compared against the occlusion device pre-therapy pools. Secondly, the overlap genes of DEGs between PLLA and metal device post-treatment and pre-therapy pools were analyzed. From the sub-analyses, we obtained 1,523 genes and 1,045 genes that were statistically significantly differently expressed between the occlusion device post- and pre-treatment (**Figure 1B**). The overlap genes in the results from the different sub-analyses are illustrated, with 115 genes differently expressed (**Figure 1B**). The distribution of DEGs between the metal device post-treatment and pre-therapy or the PLLA device post-treatment and pre-therapy is shown in a volcano plot, respectively (**Figure 1C**). Among these genes, 337 genes were downregulated (blue dots) and 668 genes were upregulated (red dots) in metal device post-treatment vs. pre-therapy. Compared to the PLLA device pre-therapy, 737 genes were downregulated (blue dots) and 786 genes were upregulated (red dots) in PLLA device post-treatment (**Figure 1C**). Overlap of the top 50 DEGs in metal device groups and the top 50 PLLA device groups is shown in **Figures 2A–C**. In addition, **Supplementary Tables S2–4** show the DEGs of metal device groups, PLLA device groups, and overlap genes by the occlusion device post-treatment vs. pre-therapy, respectively.

Pathway and Functional Enrichment Analysis

DEGs that the occlusion device treatment induced were analyzed in the above results. We subsequently compiled a list of the most frequently altered linked genes (including upregulated and downregulated genes), prior to analyzing this gene list using the GO tools in clusterProfiler (https://doi.org/10.1089/omi.2011.0118). **Figure 3** summarizes the most significantly overrepresented GO terms in the biological process category and also PPI core gene analysis in metal and PLLA device therapy, respectively. We found that the following processes were affected by the occlusion device treatment: DEGs in the metal device group were enriched in immune response-regulating signaling pathway, immune response-regulating cell surface receptor, leukocyte migration, immune response-activating signal transduction, and immune response-activating cell surface receptor (**Figure 3A**). DEGs in the PLLA device group were most highly enriched for the GO terms establishment of proteins localization to membrane, nuclear-transcribed mRNA catabolic process, proteins targeting the membrane, proteins targeting the endoplasmic reticulum process, and establishment of protein localization to endoplasmic reticulum (**Figure 3B**). The biological processes identified in this analysis are likely to contribute to the pathobiology of the occlusion device treatment. These results suggest that mechanisms of development and remodeling of ASDs might be different in metal or PLLA device treatment. PPI analysis shows the core and ring genes in the immune response pathway and the CXCR4 are the core genes identified in the metal device group (**Figure 3C**), and the key pathways are nuclear-transcribed mRNA catabolic process and proteins targeting the endoplasmic reticulum process with ribosomal proteins (such as RPS26) in the PLLA device group (**Figure 3D**).

Weighted Gene Co-Expression Network Analysis of Differentially Expressed Genes

To investigate the important role of gene interactions in ASDs, the weighted gene co-expression network analysis was used to construct an interaction network with genes, in which the nodes represent the genes and the edges depict their associations, the genes having expression commonality are in the same gene network, and the co-expression relationship between genes is

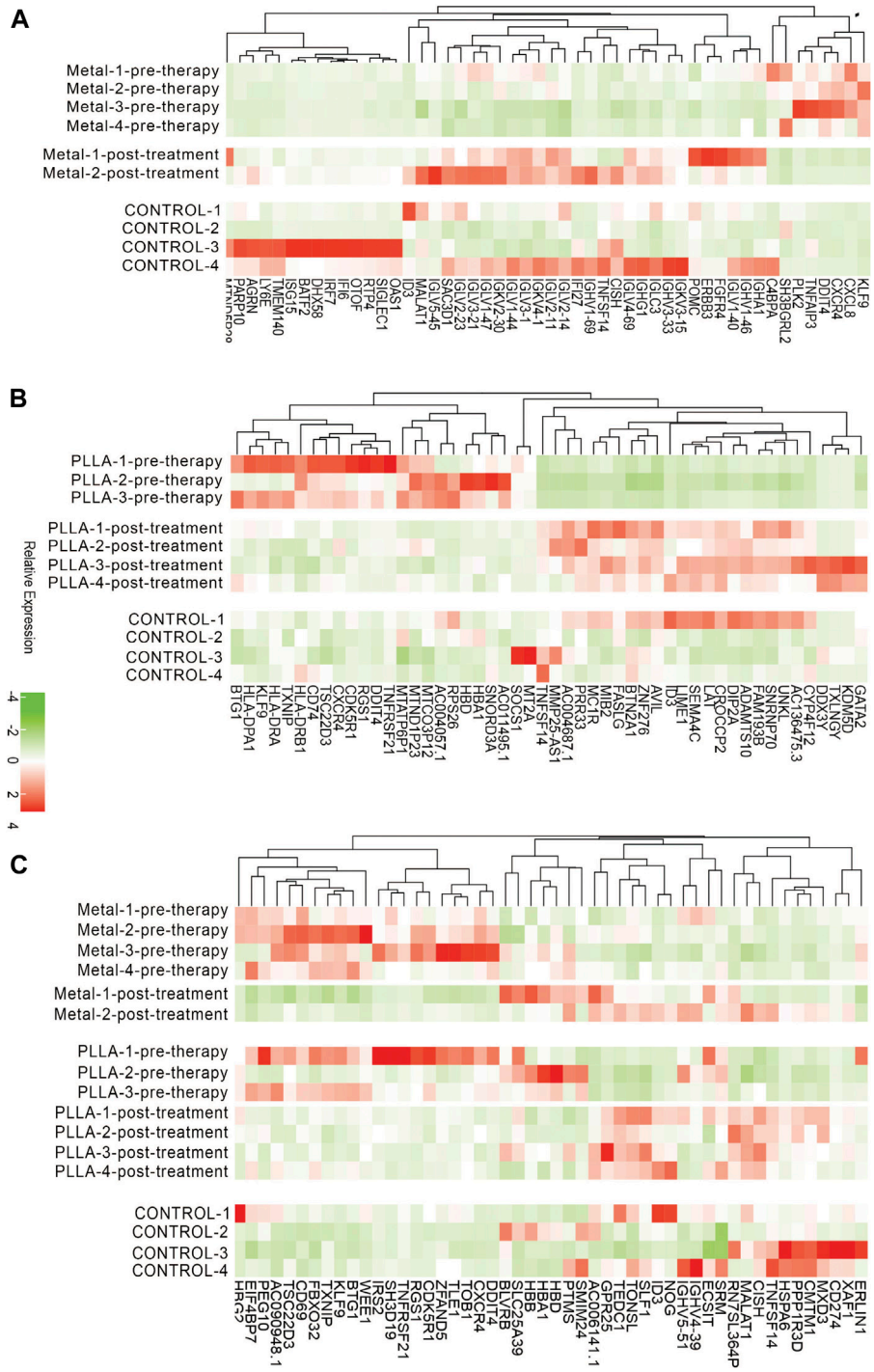

FIGURE 2 | Expression levels of DEGs from metal and PLLA device groups. **(A)** Color key of the heatmap shows the relative expression level of DEGs in metal device pre-therapy group vs. post-treatment group. **(B)** Heatmap showing DEGs in PLLA device pre-therapy group vs. post-treatment group. **(C)** Heatmap of overlap DEGs between metal and PLLA device groups.

generally measured by the expression correlation coefficient between them. By setting soft-thresholding power as 18 (scale free $R^2 = 0.85$) and cut height as 0.25, we eventually identified 18 modules (**Figures 4A–D**; non-clustering DEGs shown in gray). From the heatmap of module–trait correlations, we identified that the M2 was the most highly correlated with therapy of septal defects. In addition, Module annotation by KEGG pathway is shown in **Supplementary Table S5**.

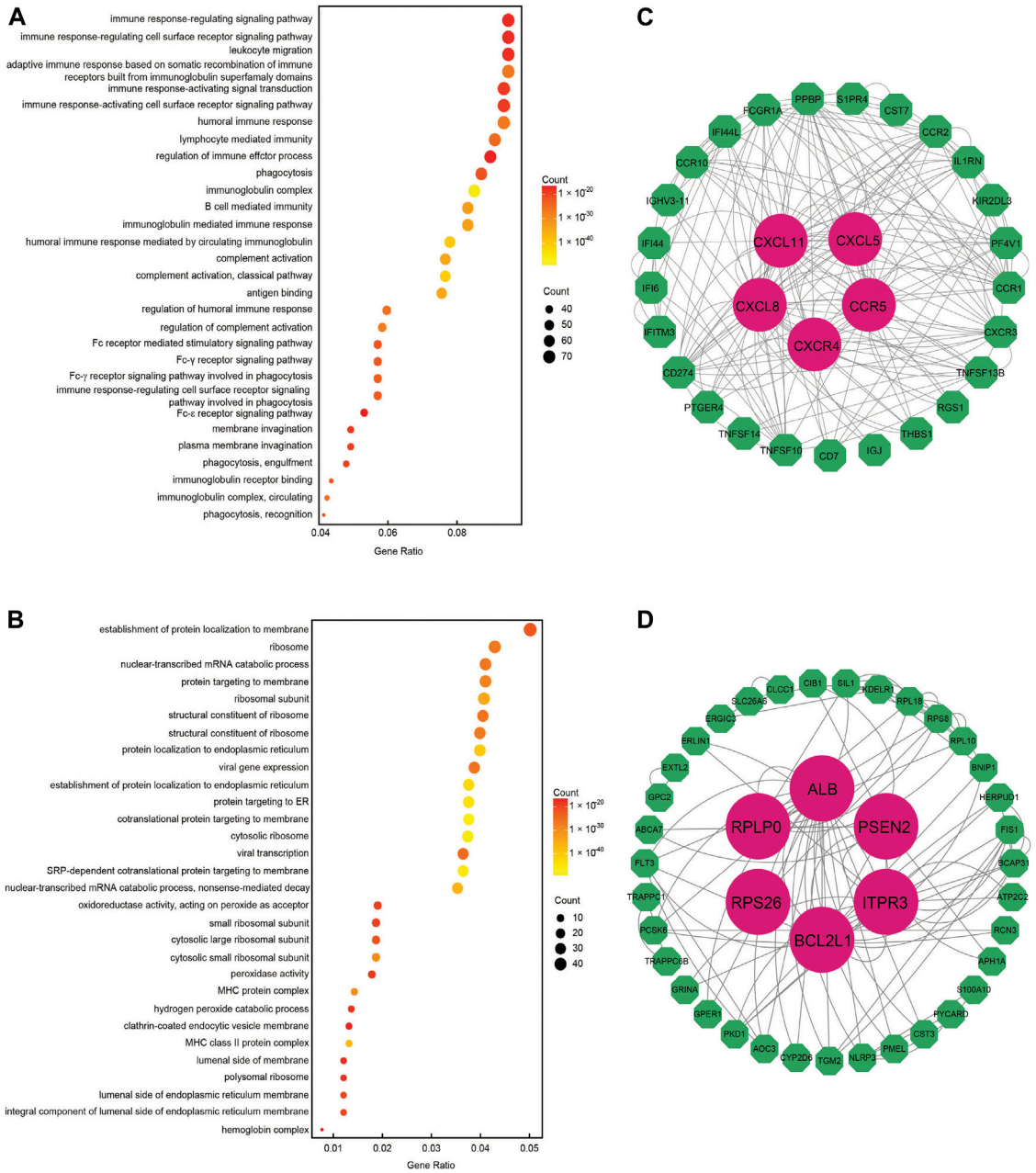

FIGURE 3 | DEGs are significantly enriched in multiple functional groups. The position of each dot represents the ratio of gene number for each GO term. The absolute number of gene and significant level are annotated by size and color, respectively. **(A)** Significant GO terms for DEGs between the metal device post-treatment group and pre-therapy. **(B)** Significant GO terms for DEGs between the PLLA device post-treatment group and pre-therapy. **(C)** Protein–protein interactions (PPIs) for the metal device therapy group. The CXCR4 gene is the core gene for the immunity response pathway. **(D)** PPI analysis for the PLLA device therapy group. The RPS26 gene is the core gene for the proteins targeting the endoplasmic reticulum process pathway.

Differentially Expressed Genes Were Validated by Quantitative Reverse Transcriptase-Polymerase Chain Reaction

Of the total 1,045 genes and 1,523 genes induced by metal and PLLA device treatment, respectively, which were differently expressed in the RNA-seq, down- and upregulated genes with relevance to occlusion device treatment differentially expressed below the $p < 0.01$ and with a logFC >2 were selected for further validation by quantitative RT-PCR. Downregulated genes in both metal and poly-L-lactic acid device treatment (*DDIT4*, *BTG1*, *CXCR4*, *IRS2*, *RGS1*, *PEG10*, and *TOB1*), upregulated genes in poly-L-lactic acid device treatment (*LY6E* and *ERBB3*), and downregulated genes in

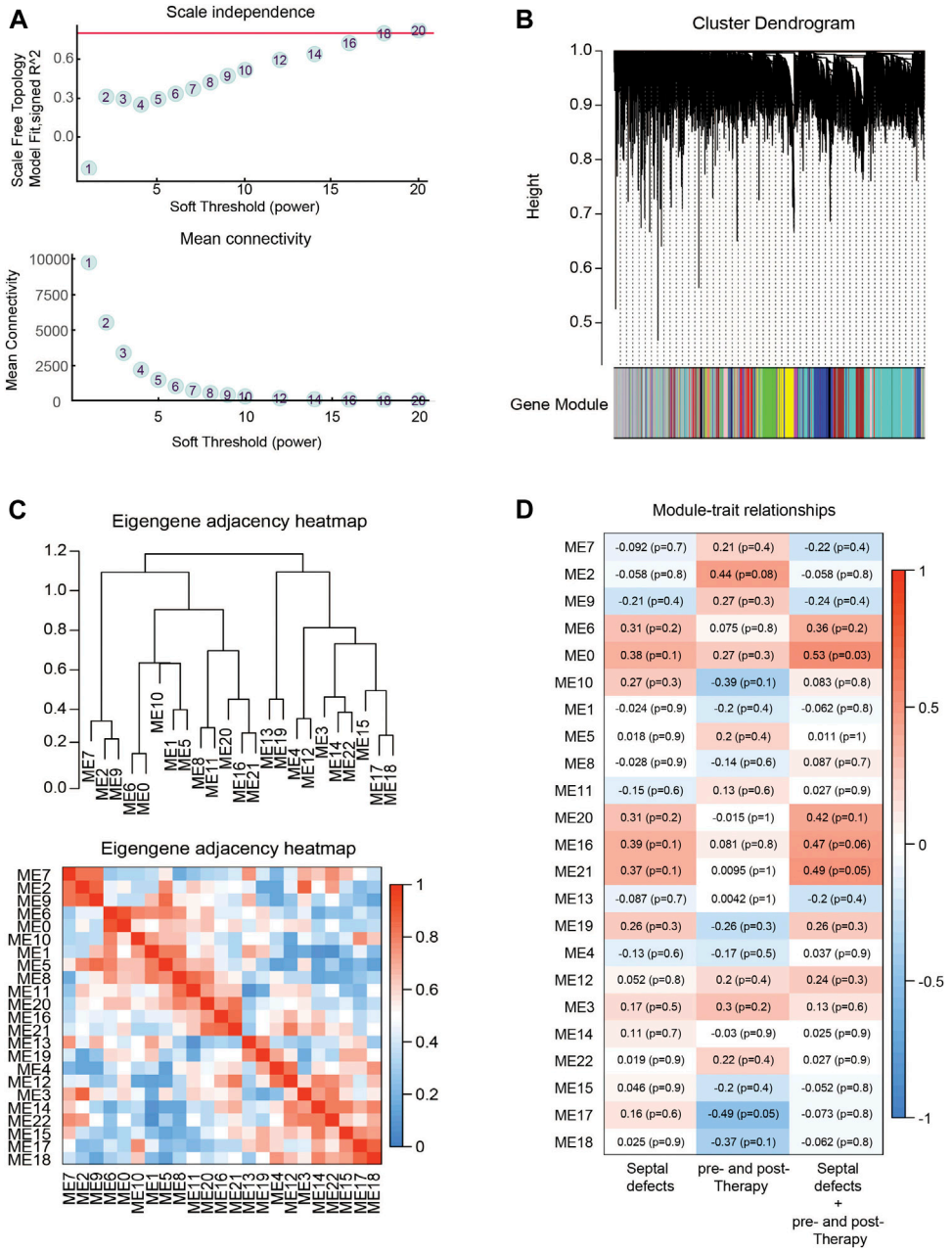

FIGURE 4 | Identification of key modules through WGCNA. **(A)** Analysis of the scale-free fit index **(up)** and the mean connectivity **(down)** for various soft-thresholding powers. **(B)** Eigengenes adjacency heatmap. **(C)** Dendrogram of all DEGs clustered based on a dissimilarity measure (1-TOM). **(D)** Heatmap of the correlation between module eigengenes (ME) and traits of septal defects or therapy. Each grid of the heatmap contains the correlation coefficient and *p*-value.

metal device treatment (*CDK5R1* and *TXNIP*). In addition, the upregulated gene (*ID3*) in both PLLA and metal device treatment was also selected. The results of qRT-PCR are shown in **Figure 5**. As expected, the expression levels of *DDIT4*, *IRS2*, *TOB1*, *BTG1*, *PEG10*, *CXCR4*, and *RGS1* in both poly-L-lactic acid and metal device treatment were significantly downregulated, which was consistent with a significant decrease in the expression of these genes in the DESeq differential expression analysis. The upregulated *ID3* gene in the DESeq2 differential expression analysis was significantly upregulated by qRT-PCR validated in PLLA and metal device treatment. *LY6E* and *ERBB3* in PLLA device treatment showed upregulation by qRT-PCR and *CDK5R1* in metal device treatment showed downregulation by qRT-PCR validation. However, there were some exceptions to some gene expression; *TXNIP* in metal device treatment was not significantly changed by qRT-PCR validation (**Figures 5A,B**). The inconsistency between qRT-PCR validation and DESeq2 differential expression analysis may be accounted from varying mRNA levels of the gene in different patient samples.

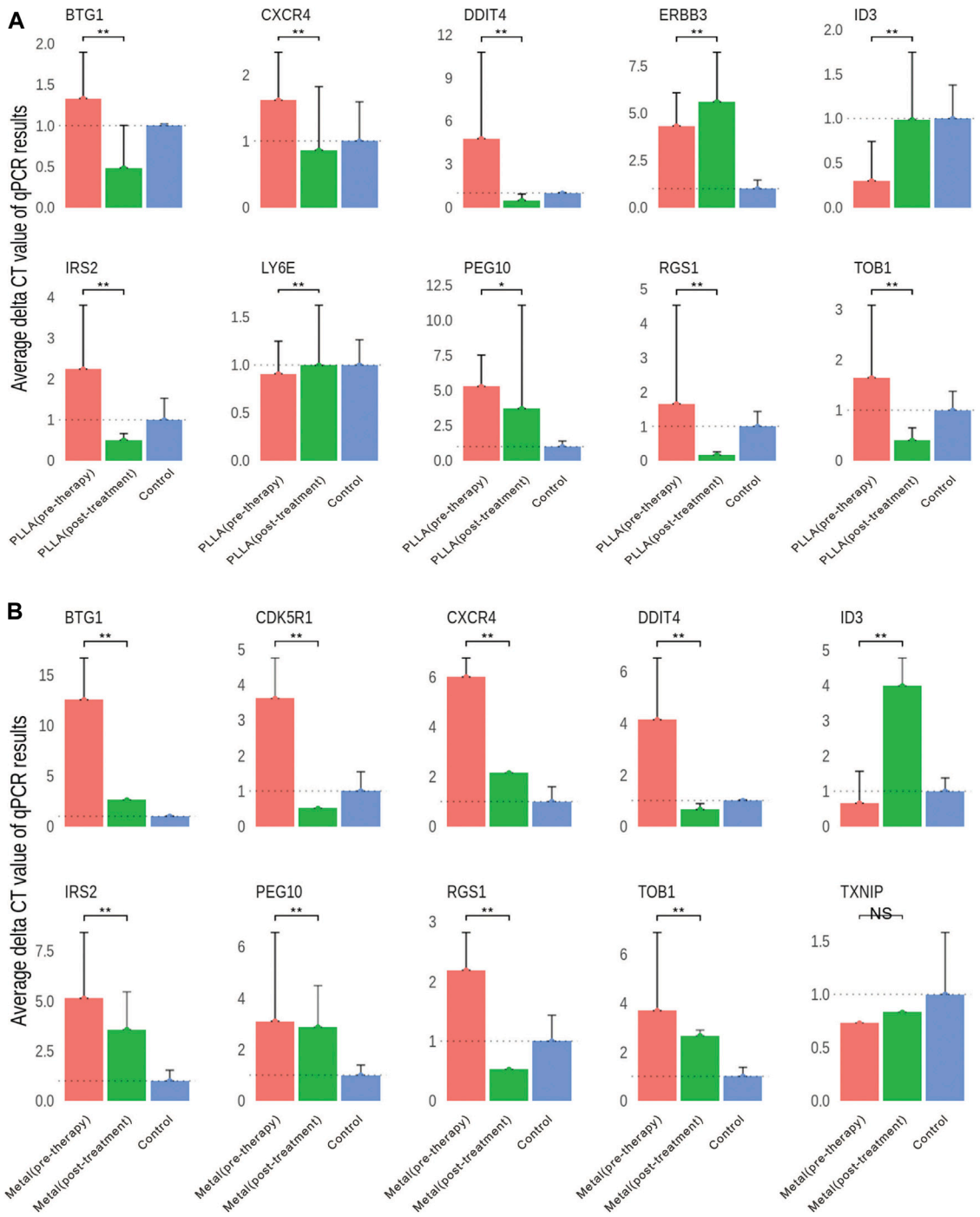

FIGURE 5 | Validation of DEGs by qRT-PCR. **(A)** Gene expression differences in the PLLA device pre-therapy group and post-treatment group. **(B)** Gene expression differences in the metal device pre-therapy group and post-treatment group. NS, no significant; $^{*}p < 0.05$, $^{**}p < 0.01$.

DISCUSSION

With the advancement of interventional therapy for congenital heart disease and the progress of device research and development, more and more patients with ASD receive interventional therapy, in which metal and degradable occluders are the two kinds of the most used closure devices in these days (Morray, 2019; Shi et al., 2019; O'Byrne and Levi, 2019; Alnasser et al., 2018). One of the most important indicators for evaluating the histocompatibility of the occluder is the endothelialization induced by the occluders. Endothelialization is crucial and of major clinical importance and impaired

endothelialization may lead to prolonged anticoagulant therapy and even serious complications such as residual shunt, device-related thrombosis, endocarditis, and occluder displacement (Chessa et al., 2004; Nguyen et al., 2016; Kalayc and Kalayc, 2017; Chen et al., 2018; Li et al., 2020). Therefore, the observation of endothelialization after occluder implantation is particularly important. However, due to the characteristics that the occluder cannot be removed after implantation *in vivo*, whether the degradable occluders are comparable with the metal occluder in endothelialization is critical. The evaluation of endothelialization is mostly based on the data obtained from animal experiments or a few cases of surgery and autopsy. The observation methods are also limited to electron microscopy, histopathology, and immunohistochemistry (Kuhn et al., 1996; Zahn et al., 2001; Foth et al., 2009; Morray, 2019; Shi et al., 2019).

Few reports have evaluated the process of occluder endothelialization in human by observing the differential expression of genes by RNA-sequencing technology, as well as studies on the mechanism of gene regulation of this process. Therefore, in this study, we carried out RNA-sequencing technology combined with qRT-PCR validation to determine the DEGs and its function on biological processes in the occlusion device (metal or PLLA device) treatment. Transcriptome profile revealed that a total of 1,045 and 1,523 confidently detected genes, respectively, are differentially expressed (FDR < 0.05), of which 337 genes were downregulated and 668 genes were upregulated in metal device post-treatment, and 737 genes were downregulated and 786 genes were upregulated in PLLA device post-treatment. GO analysis revealed the enrichment of these DEGs on the biological process. Then, the differential expression of RNA-Seq data was verified by qRT-PCR, and this differential expression finding confirmed that occluder implantation produced a series of molecular biological changes at the level of gene regulation in the human body, which was finally manifested as endothelialization on the device surface. Theoretically, by observing the differential expression of RNA-Seq data at different time points in the same individual after occluder implantation, it can reflect the degree of endothelialization on the surface of the occluder, making it possible to monitor the endothelialization induced by occluder *in vivo* by RNA-sequencing technology, and also providing a basis for further study of the specific mechanism of gene-level regulation of endothelialization after occluder implantation in patients.

Since the closure of different materials at the same site may involve many similar gene regulatory mechanisms, there are too many overlapping genes, and it is difficult to highlight the genes that play the most critical regulatory role. Therefore, in this study, different material occluders were selected to occlude ASDs, that is, biodegradable or metal materials to occlude ASDs, hoping to select genes that play a key regulatory role among the overlapping expressed genes by RNA-sequencing technology. According to many previous studies observing the process of occluder endothelial coverage, it has been confirmed that the process of endothelialization is similar to wound healing and is a complex biological process of tissue repair (Lock et al., 1989; Sideris et al., 1990; Das et al., 1993; Kuhn et al., 1996; Sharafuddin et al., 1997; Thomsen et al., 1998; Zahn et al., 2001). These include fibroblasts embedded in loose collagen extracellular matrix, newly formed blood vessels, and inflammatory cells (Reinke and Sorg, 2012; Sinno and Prakash, 2013). Degradable occluders differ from metal occluders in structure, require different endothelialization time, but have similar pathophysiological changes, and neo-endothelialization, angiogenesis, and extracellular matrix accumulation are the key events to control the process. Therefore, it is reasonable to believe that in the overlapping part of gene expression between degradable and metal occluders, genes that play a role in regulating cytokines related to neo-endothelialization, angiogenesis, or extracellular matrix accumulation are key regulatory genes.

Our results showed that *CXCR4*, *DDIT4*, and *TOB1* were the highest before occluder treatment and downregulated after treatment with both PLLA and metal device. Previous studies demonstrated that *DDIT4* regulates cell growth, proliferation, and survival by inhibiting the activity of mammalian mTORC1 targets (Wang et al., 2015), while *TOB1*, as an anti-proliferative gene, can regulate cell growth and differentiation and has a migratory role (Liu et al., 2015; Guan et al., 2017; Shangguan et al., 2019). Finally, we also confirmed that CXCR4 is a candidate gene responsible for cardiac congenital pathologies in human as previously suggested in mouse studies (Escot et al., 2013; Wang et al., 2014; Zhong and Rajagopalan, 2015; Page et al., 2018).

Immune response genes and pathways (**Supplementary Table S2**, **Figure 3C**) were also identified in our study. Similar clinical studies in device closures of ASDs in children also found that systemic inflammatory reactions occurred after device closure of ASDs in pediatric patients. However, these inflammatory reactions were more significant in patients who underwent a transthoracic approach than in patients who underwent a transcatheter approach (Hong et al., 2020).

Several studies showed that ribosomal protein mutations are associated with patients in Diamond–Blackfan anemia patients with septal defects (Gazda et al., 2008; Chae et al., 2014). Our GO and PPI analysis also provided support for these findings (**Figure 3D**).

Therefore, it can be preliminarily speculated that *CXCR4*, *DDIT4*, and *TOB1* may be key regulatory genes in the process of endothelialization, and the process of endothelialization may be promoted by downregulation of *CXCR4*, *DDIT4*, and *TOB1* expression after occluder implantation. The differential changes of *CXCR4*, *DDIT4*, and *TOB1* before and after closure also provide a direction for further establishment of knockout model studies to verify the key regulatory genes of endothelialization after implantation.

CONCLUSION

In this study, we analyzed RNA-Seq data from the PLLA device therapy group, metal device treatment group, and healthy volunteer group. We found potential genes and pathways that may be involved in endothelialization and remodeling in the progress of atrial septal defect repair, making it possible to

monitor the endothelialization of occluders *in vivo* by RNA-seq and RT-PCR methods. At the same time, the changes in gene expression levels and their involvement in different pathways showed that *CXCR4, DDIT4*, and *TOB1* may be key regulatory genes for endothelialization induced by occluder implantation *in vivo*. Our study provides a basis for further research on the underlying mechanisms of regulation endothelialization progression at the transcriptional level after occluder implantation in human.

DATA AVAILABILITY STATEMENT

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found in the article/ **Supplementary Material**.

ETHICS STATEMENT

The studies involving human participants were reviewed and approved by the Committee on the Ethics of Shenzhen children's hospital. Written informed consent to participate in this study was provided by the participants' legal guardian/ next of kin.

AUTHOR CONTRIBUTIONS

H-LJ, Z-WZ, and CL designed the study and take responsibility for the integrity of the data and the accuracy of the data analysis. B-NL, LS, W-BG, and M-YQ participated in sample diagnosis and collection. Q-DT contributed to data process and analysis. Y-LT and LY contributed to data interpretation. AC and Z-XZ performed the bioinformatics analysis and PPI core gene analysis. Y-JL revised the manuscript, especially the English and the biology lab procedure. H-LJ contributed to manuscript preparation and revision. All authors reviewed and approved the final version.

FUNDING

This work was supported by Sanming Project of Medicine in Shenzhen (No. SZSM201612057) and Shenzhen Fund for Guangdong Provincial High-level Clinical Key Specialties (No. SZGSP012).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2021.790426/full#supplementary-material

REFERENCES

Abaci, A., Unlu, S., Alsancak, Y., Kaya, U., and Sezenoz, B. (2013). Short and Long Term Complications of Device Closure of Atrial Septal Defect and Patent Foramen Ovale: Meta-Analysis of 28,142 Patients from 203 Studies. *Cathet. Cardiovasc. Intervent* 82, 1123–1138. doi:10.1002/ccd.24875

Alnasser, S., Lee, D., Austin, P. C., Labos, C., Osten, M., Lightfoot, D. T., et al. (2018). Long Term Outcomes Among Adults post Transcatheter Atrial Septal Defect Closure: Systematic Review and Meta-Analysis. *Int. J. Cardiol.* 270, 126–132. doi:10.1016/j.ijcard.2018.06.076

Bazzoni, G., and Dejana, E. (2004). Endothelial Cell-To-Cell Junctions: Molecular Organization and Role in Vascular Homeostasis. *Physiol. Rev.* 84, 869–901. doi:10.1152/physrev.00035.2003

Chae, H., Park, J., Lee, S., Kim, M., KimKim, Y., Lee, J.-W., et al. (2014). Ribosomal Protein Mutations in Korean Patients with Diamond-Blackfan Anemia. *Exp. Mol. Med.* 46, e88. doi:10.1038/emm.2013.159

Chen, R., Luo, J., Deng, X., and Huang, P. (2018). Displacement of Occluder as a Rare Complication of Transcatheter Closure of Ventricular Septal Defect. *Medicine* 97, e11327. doi:10.1097/MD.0000000000011327

Chessa, M., Butera, G., Frigiola, A., and Carminati, M. (2004). Endothelialization of ASD Devices for Transcatheter Closure: Possibility or Reality?. *Int. J. Cardiol.* 97, 563–564. doi:10.1016/j.ijcard.2003.09.009

Das, G. S., Voss, G., Jarvis, G., Wyche, K., Gunther, R., and Wilson, R. F. (1993). Experimental Atrial Septal Defect Closure with a New, Transcatheter, Self-Centering Device. *Circulation* 88, 1754–1764. doi:10.1161/01.cir.88.4.1754

Dejana, E. (2004). Endothelial Cell-Cell Junctions: Happy Together. *Nat. Rev. Mol. Cel. Biol.* 5, 261–270. doi:10.1038/nrm1357

Escot, S., Blavet, C., Härtle, S., Duband, J.-L., and Fournier-Thibault, C. (2013). Misregulation of SDF1-CXCR4 Signaling Impairs Early Cardiac Neural Crest Cell Migration Leading to Conotruncal Defects. *Circ. Res.* 113, 505–516. doi:10.1161/CIRCRESAHA.113.301333

Foth, R., Quentin, T., Michel-Behnke, I., Vogt, M., Kriebel, T., Kreischer, A., et al. (2009). Immunohistochemical Characterization of Neotissues and Tissue Reactions to Septal Defect-Occlusion Devices. *Circ. Cardiovasc. Interv.* 2, 90–96. doi:10.1161/CIRCINTERVENTIONS.108.810507

Gaynor, J. W., O'Brien, J. E., Rychik, J., Sanchez, G. R., DeCampli, W. M., and Spray, T. L. (2001). Outcome Following Tricuspid Valve Detachment for Ventricular Septal Defects Closure. *Eur. J. Cardiothorac. Surg.* 19, 279–282. doi:10.1016/s1010-7940(01)00577-2

Gazda, H. T., Sheen, M. R., Vlachos, A., Choesmel, V., O'Donohue, M.-F., Schneider, H., et al. (2008). Ribosomal Protein L5 and L11 Mutations Are Associated with Cleft Palate and Abnormal Thumbs in Diamond-Blackfan Anemia Patients. *Am. J. Hum. Genet.* 83, 769–780. doi:10.1016/j.ajhg.2008.11.004

Guan, R., Peng, L., Wang, D., He, H., Wang, D., Zhang, R., et al. (2017). Decreased TOB1 Expression and Increased Phosphorylation of Nuclear TOB1 Promotes Gastric Cancer. *Oncotarget* 8, 75243–75253. doi:10.18632/oncotarget.20749

Hong, Z.-N., Huang, J.-S., Sun, K.-P., Luo, Z.-R., and Chen, Q. (2020). Comparison of Postoperative Changes in Inflammatory Marker Levels between Transthoracic and Transcatheter Device Closures of Atrial Septal Defects in Children. *Braz. J. Cardiovasc. Surg.* 35, 498–503. doi:10.21470/1678-9741-2019-0207

Huang, X.-m., Zhu, Y.-f., Cao, J., Hu, J.-q., Bai, Y., Jiang, H.-b., et al. (2013). Development and Preclinical Evaluation of a Biodegradable Ventricular Septal Defect Occluder. *Cathet. Cardiovasc. Intervent.* 81, 324–330. doi:10.1002/ccd.24580

Kalaycı, B., and Kalayc, S. (2017). Right Atrial ball Thrombus Associated with Atrial Septal Occluder Device: A Late Complication of Transcatheter Atrial Septal Defect Closure. *Anatol. J. Cardiol.* 18, E9. doi:10.14744/AnatolJCardiol.2017.8012

Karunanithi, Z., Nyboe, C., and Hjortdal, V. E. (2017). Long-term Risk of Atrial Fibrillation and Stroke in Patients with Atrial Septal Defect Diagnosed in Childhood. *Am. J. Cardiol.* 119, 461–465. doi:10.1016/j.amjcard.2016.10.015

Kuhn, M. A., Latson, L. A., Cheatham, J. P., McManus, B., Anderson, J. M., Kilzer, K. L., et al. (1996). Biological Response to Bard Clamshell Septal Occluders in the Canine Heart. *Circulation* 93, 1459–1463. doi:10.1161/01.cir.93.7.1459

Leppert, M., Poisson, S. N., and Carroll, J. D. (2016). Atrial Septal Defects and Cardioembolic Strokes. *Cardiol. Clin.* 34, 225–230. doi:10.1016/j.ccl.2015.12.004

Li, B. N., Xie, Y. M., Xie, Z. F., Chen, X. M., Zhang, G., Zhang, D. Y., et al. (2019). Study of Biodegradable Occluder of Atrial Septal Defect in a Porcine Model. *Catheter. Cardiovasc. Interv.* 93, E38–E45. doi:10.1002/ccd.27852

Li, Y. F., Xie, Y. M., Chen, J., Li, B. N., Xie, Z. F., Wang, S. S., et al. (2020). Initial Experiences with a Novel Biodegradable Device for Percutaneous Closure of Atrial Septal Defects: From Preclinical Study to First-in-human Experience. *Catheter. Cardiovasc. Interv.* 95, 282–293. doi:10.1002/ccd.28529

Liu, C., Tao, T., Xu, B., Lu, K., Zhang, L., Jiang, L., et al. (2015). BTG1 Potentiates Apoptosis and Suppresses Proliferation in Renal Cell Carcinoma by Interacting with PRMT1. *Oncol. Lett.* 10, 619–624. doi:10.3892/ol.2015.3293

Lock, J. E., Rome, J. J., Davis, R., Van Praagh, S., Perry, S. B., Van Praagh, R., et al. (1989). Transcatheter Closure of Atrial Septal Defects. Experimental studiesExperimental Studies. *Circulation* 79, 1091–1099. doi:10.1161/01.cir.79.5.1091

Luermans, J. G. L. M., Post, M. C., and Yilmaz, A. (2010). Late Device Thrombosis after Atrial Septal Defect Closure. *Eur. Heart J.* 31, 142. doi:10.1093/eurheartj/ehp512

Morray, B. H. (2019). Ventricular Septal Defect Closure Devices, Techniques, and Outcomes. *Interv. Cardiol. Clin.* 8, 1–10. doi:10.1016/j.iccl.2018.08.002

Nguyen, A. K., Palafox, B. A., Starr, J. P., Gates, R. N., and Berdjis, F. (2016). Endocarditis and Incomplete Endothelialization 12 Years after Amplatzer Septal Occluder Deployment. *Tex. Heart Inst. J.* 43, 227–231. doi:10.14503/THIJ-14-4949

O'Byrne, M. L., and Levi, D. S. (2019). State-of-the-Art Atrial Septal Defect Closure Devices for Congenital Heart. *Interv. Cardiol. Clin.* 8, 11–21. doi:10.1016/j.iccl.2018.08.008

Oses, P., Hugues, N., Dahdah, N., Vobecky, S. J., Miro, J., Pellerin, M., et al. (2010). Treatment of Isolated Ventricular Septal Defects in Children: Amplatzer versus Surgical Closure. *Ann. Thorac. Surg.* 90, 1593–1598. doi:10.1016/j.athoracsur.2010.06.088

Page, M., Ridge, L., Gold Diaz, D., Tsogbayar, T., Scambler, P. J., and Ivins, S. (2018). Loss of CXCL12/CXCR4 Signalling Impacts Several Aspects of Cardiovascular Development but Does Not Exacerbate Tbx1 Haploinsufficiency. *PLoS One* 13, e0207251. doi:10.1371/journal.pone.0207251

Penny, D. J., and Vick, G. W., III. (2011). Ventricular Septal Defect. *The Lancet* 377, 1103–1112. doi:10.1016/S0140-6736(10)61339-6

Pillai, A. A., Rangasamy, S., and Balasubramonian, V. R. (2019). Transcatheter Closure of Moderate to Large Perimembranous Ventricular Septal Defects in Children Weighing 10 Kilograms or Less. *World J. Pediatr. Congenit. Heart Surg.* 10, 278–285. doi:10.1177/2150135119825562

Rao, P. S., and Harris, A. D. (2017). Recent Advances in Managing Septal Defects: Atrial Septal Defects. *F1000Res* 6, 2042. doi:10.12688/f1000research.11844.1

Reinke, J. M., and Sorg, H. (2012). Wound Repair and Regeneration. *Eur. Surg. Res.* 49, 35–43. doi:10.1159/000339613

Shangguan, W. J., Liu, H.-T., Que, Z. J., Qian, F. F., Liu, L. S., and Tian, J. H. (2019). TOB1-AS1 S-uppresses N-on-small C-ell L-ung C-ancer C-ell M-igration and I-nvasion through a ceRNA N-etwork. *Exp. Ther. Med.* 18, 4249–4258. doi:10.3892/etm.2019.8103

Sharafuddin, M. J. A., Gu, X., Titus, J. L., Urness, M., Cervera-Ceballos, J. J., and Amplatz, K. (1997). Transvenous Closure of Secundum Atrial Septal Defects. *Circulation* 95, 2162–2168. doi:10.1161/01.cir.95.8.2162

Shi, D., Kang, Y., Zhang, G., Gao, C., Lu, W., Zou, H., et al. (2019). Biodegradable Atrial Septal Defect Occluders: A Current Review. *Acta Biomater.* 96, 68–80. doi:10.1016/j.actbio.2019.05.073

Shimpo, H., Hojo, R., Ryo, M., Konuma, T., and Tempaku, H. (2013). Transcatheter Closure of Secundum Atrial Septal Defect. *Gen. Thorac. Cardiovasc. Surg.* 61, 614–618. doi:10.1007/s11748-013-0268-7

Sideris, E. B., Sideris, S. E., Fowlkes, J. P., Ehly, R. L., Smith, J. E., and Gulde, R. E. (1990). Transvenous Atrial Septal Defect Occlusion in Piglets with a "buttoned" Double-Disk Device. *Circulation* 81, 312–318. doi:10.1161/01.cir.81.1.312

Sinno, H., and Prakash, S. (2013). Complements and the Wound Healing cascade: an Updated Review. *Plast. Surg. Int.* 2013, 1–7. doi:10.1155/2013/146764

Tang, B., Su, F., Sun, X., Wu, Q., Xing, Q., and Li, S. (2018). Recent Development of Transcatheter Closure of Atrial Septal Defect and Patent Foramen Ovale with Occluders. *J. Biomed. Mater. Res.* 106, 433–443. doi:10.1002/jbm.b.33831

Thomsen, A. B., Schneider, M., Baandrup, U., Stenbog, E. V., Hasenkam, J. M., Bagger, J. P., et al. (1998). Animal Experimental Implantation of an Atrial Septal Defect Occluder System. *Heart* 80, 606–611. doi:10.1136/hrt.80.6.606

Wang, E. R., Jarrah, A. A., Benard, L., Chen, J., Schwarzkopf, M., Hadri, L., et al. (2014). Deletion of CXCR4 in Cardiomyocytes Exacerbates Cardiac Dysfunction Following Isoproterenol Administration. *Gene Ther.* 21, 496–506. doi:10.1038/gt.2014.23

Wang, Y., Han, E., Xing, Q., Yan, J., Arrington, A., Wang, C., et al. (2015). Baicalein Upregulates DDIT4 Expression Which Mediates mTOR Inhibition and Growth Inhibition in Cancer Cells. *Cancer Lett.* 358, 170–179. doi:10.1016/j.canlet.2014.12.033

Williams, K., Carson, J., and Lo, C. (2019). Genetics of Congenital Heart Disease. *Biomolecules* 9, 879. doi:10.3390/biom9120879

Wu, R.-H., Li, D.-F., Tang, W.-T., Qiu, K.-Y., Li, Y., Liao, X.-Y., et al. (2018). Atrial Septal Defect in a Patient with Congenital Disorder of Glycosylation Type 1a: a Case Report. *J. Med. Case Rep.* 12, 17. doi:10.1186/s13256-017-1528-4

Zahn, E. M., Wilson, N., Cutright, W., and Latson, L. A. (2001). Development and Testing of the Helex Septal Occluder, a New Expanded Polytetrafluoroethylene Atrial Septal Defect Occlusion System. *Circulation* 104, 711–716. doi:10.1161/hc3301.092792

Zhong, J., and Rajagopalan, S. (2015). Dipeptidyl Peptidase-4 Regulation of SDF-1/CXCR4 Axis: Implications for Cardiovascular Disease. *Front. Immunol.* 6, 477. doi:10.3389/fimmu.2015.00477

Conflict of Interest: AC, Y-JL and Z-XZ were employed by the company Guangzhou Mendel Genomics and Medical Technology Inc.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Genetics

ORIGINAL RESEARCH
published: 15 December 2021
doi: 10.3389/fgene.2021.802894

smplot: An R Package for Easy and Elegant Data Visualization

Seung Hyun Min and Jiawei Zhou**

School of Ophthalmology and Optometry, Affiliated Eye Hospital, State Key Laboratory of Ophthalmology, Optometry and Vision Science, Wenzhou Medical University, Wenzhou, China

R, a programming language, is an attractive tool for data visualization because it is free and open source. However, learning R can be intimidating and cumbersome for many. In this report, we introduce an R package called "smplot" for easy and elegant data visualization. The R package "smplot" generates graphs with defaults that are visually pleasing and informative. Although it requires basic knowledge of R and ggplot2, it significantly simplifies the process of plotting a bar graph, a violin plot, a correlation plot, a slope chart, a Bland-Altman plot and a raincloud plot. The aesthetics of the plots generated from the package are elegant, highly customisable and adhere to important practices of data visualization. The functions from smplot can be used in a modular fashion, thereby allowing the user to further customise the aesthetics. The *smplot* package is open source under the MIT license and available on Github (https://github.com/smin95/smplot), where updates will be posted. All the example figures in this report are reproducible and the codes and data are provided for the reader in a separate online guide (https://smin95.github.io/dataviz/).

Keywords: smplot, data visualisation, R software, data analysis, ggplot2

OPEN ACCESS

Edited by:
Guangchuang Yu,
Southern Medical University, China

Reviewed by:
Zongcheng Li,
Fifth Medical Center of the PLA General Hospital, China
Binbin Wang,
National Research Institute for Family Planning (CAS), China

****Correspondence:***
Seung Hyun Min
seung.min@mail.mcgill.ca
Jiawei Zhou
zhoujw@mail.eye.ac.cn

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

Received: *27 October 2021*
Accepted: *29 November 2021*
Published: *15 December 2021*

Citation:
Min SH and Zhou J (2021) smplot: An R Package for Easy and Elegant Data Visualization. Front. Genet. 12:802894. doi: 10.3389/fgene.2021.802894

INTRODUCTION

Data visualization is an important skill in scientific writing. The reader may agree that most memorable aspects of a scientific paper are its figures rather than texts. There are various programs for plotting data. However, some require subscription fees, such as Matlab. On the other hand, others such as matplotlib in Python (Hunter, 2007) and *ggplot2* in R (Wickham, 2016) are free and open source but can overwhelm incoming research trainees because the students are often required to overcome a steep learning curve. Moreover, the learning curves can enforce students to spend a long time to change typesetting or making such minute changes, forcing them to use vector graphics editor such as Adobe Illustrator to polish the figures instead of modifying the codes that generate the original plot. This practice of creating a figure using multiple programs, however, can be time-consuming in the long run. For instance, when the trainee is asked to make changes in the figure, one must make changes in all programs that one has used sequentially, which can be tedious and laborious. In this report, we hope to convince the reader that a polished, satisfying figure can be created using only one software environment by introducing a new, free, and easy-to-use tool for data visualization.

Biomedical research increasingly incorporates the usage of complex, computational tools for data analysis. For this reason, we introduce an R package "smplot" that is an intuitive and quick tool for performing elegant data visualization for research trainees. Since the use of smplot requires a basic knowledge of R and ggplot2, an online tutorial about R that incorporates smplot has been posted on a separate webpage entitled *Data Visualization in R Using smplot* (https://smin95.github.io/dataviz).

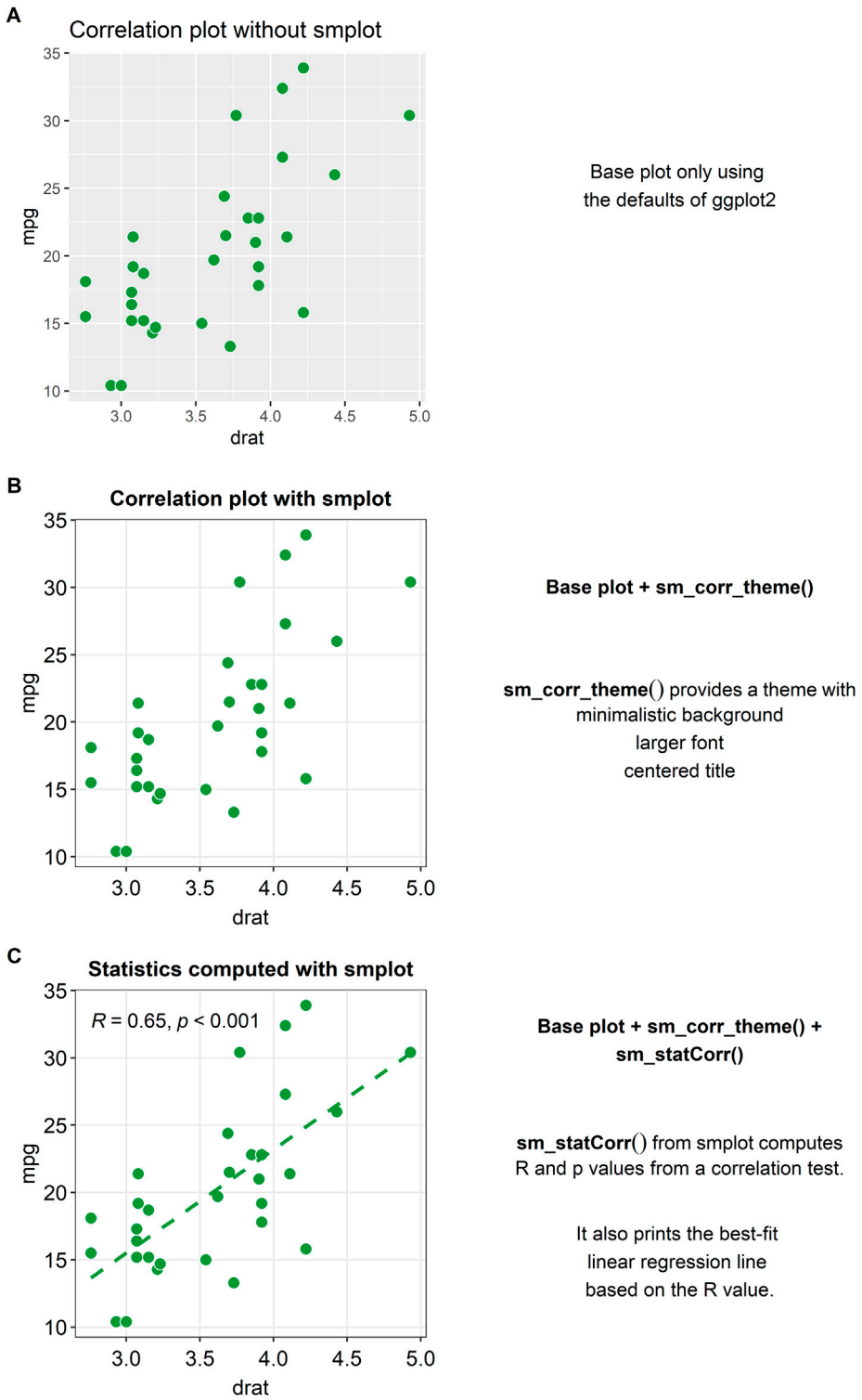

FIGURE 1 | Correlation plots with and without *smplot*. **(A)** A correlation plot without using *smplot*. **(B)** A correlation plot with a default theme of *smplot*. The theme can added in a modular fashion by adding "*sm_corr_theme()*" to the base plot. This function provides a theme with a minimalistic background, a larger font and a centered title. **(C)** A correlation plot with the default theme of *smplot*, printed statistical information (*R* = correlation coefficient, p = statistical significance) and a best-fit regression line. The *R* and *p* values as well as the regression line can be printed by adding the "*sm_statCorr()*" to the given plot in a modular fashion.

Why R?

In R, one can plot data without necessarily using programming concepts such as the *for loop*. This is because the *ggplot2* package in R can automatically plot all data points if necessary. However, this is not the case with Python (matplotlib) and Matlab. All the codes and data for the figures in this report can be found in Chapter 6 of the online guide entitled *Data Visualization in R Using smplot* (https://smin95.github.io/dataviz).

METHODS

Installation of the smplot Package

At the time of writing the paper, the smplot package is only available on Github. Therefore, if the reader is interested in installing the package, the reader must open RStudio and directly download the package by typing these commands:

install.packages ('devtools).

devtools:install_github ("smin95/smplot")

To load the smplot package into the local environment (and therefore use it), the reader must type this code below:

library (smplot).

A complete tutorial on *smplot* is available in Chapter 4 of the online guide (https://smin95.github.io/dataviz/). If the reader is not familiar with *R*, then please consider reading the online guide from Chapter 1 (https://smin95.github.io/dataviz/download-rstudio-basics-of-r.html). If the reader is familiar with ggplot2 and only interested in recreating the figures in this report, please read Chapter 6 (https://smin95.github.io/dataviz/recreating-the-manuscript-figures.html). The package is scheduled to be submitted to the CRAN (The Comprehensive R Archive Network) in near future. All updates will be posted on Github and the online guide.

RESULTS AND DISCUSSION

Correlation Plot

A correlation refers to a relationship between two variables. The smplot package provides some functions for plotting a correlation.

Figure 1A shows a correlation plot with defaults of ggplot2 and without smplot. The example is cluttered with distracting features, such as the grey background, and major and minor vertical and horizontal grid lines. Also, the title is not centered. These issues can be resolved by modularly adding a single line of code provided by *smplot*, as shown in **Figure 1B**. The example in **Figure 1B** uses the default theme of *smplot* [with the function "*sm_corr_theme()*"]. The minor grid lines have been removed and the title has been centered. Also, the font is generally larger and consistent. The aesthetics can be modified by adding the ggplot2 functions to the base plot. However, *smplot* provides a wrapper function for a clean default theme that can be added in a modular fashion to the base plot. This modularity can allow the user to customise further with ease.

When plotting a correlation, one is often recommended to report statistics and print a best-fit regression line. A function called "sm_statCorr()" can be added modularly to print the correlation coefficient (R, not R^2) and the *p*-value for statistical significance of the relationship between two variables (see **Figure 1C**). There are several arguments that are used in this function. The regression is set to be linear by default but can also be set to be non-linear by specifying the argument "*lm_method*" (ex., "*lm_method = lm*" for linear regression, "*lm_method = loess*" for non-linear local regression). Also, the type of the correlation test can be specified into either Pearson, Spearman or Kendall using the argument "*corr_method*" (ex., "*corr_method = pearson*" is the default). When the user adds "sm_statCorr()" modularly to the base plot without specifying these arguments, the function uses the defaults for the two arguments.

Bar Plot

Plotting a bar graph in *ggplot2* can appear to be not straightforward because the functions that plot the bar graph depend on the structure of the data file that is uploaded in RStudio. For instance, although both "*geom_bar()*" and "*stat_summary()*," which has multiple usages, can both plot the bar graph in a *ggplot2* setting, "*geom_bar()*" requires that the loaded data contain summarised data (ex., mean, standard error of the sample), whereas "*stat_summary()*" requires that the loaded data contain individual data so that function can directly summarise the data as the mean and the standard deviation. This subtle difference between the functions can be confusing. Also, the arguments for the function "*stat_summary()*" are not always clear.

In **Figure 2A**, a bar graph that uses the default theme of *ggplot2* is shown. Individual data points and error bars are missing. Major and minor vertical and horizontal grids overly crowd the graph. In a bar graph, since explanatory variables (levels in the x-axis) are often categorical, vertical grids are often not necessary. Also, the bar graph alone does not represent the distribution of data accurately, so plotting individual points and the error bar (ex., standard error, standard deviation or 95% confidence interval) are often recommended when the bar graph is plotted. These issues in **Figure 2A** can be resolved by modularly adding the function "*sm_bar()*" to the base plot, as shown in **Figure 2B**. This function enlarges the font, plots individual data points, automatically removes unnecessary grids, centers the title, narrows the bar width for aesthetics, and plots the error bar (in this example: standard error). These aesthetic features, such as the transparency, color and shape of the points, can be customised by using the specifying the arguments of "*sm_bar()*," such as "*bar_alpha*," "*point_alpha*" and "*point_size*." If the reader is interested in learning more about the function, please visit Chapter 4 of the online guide (smin95.github.io/dataviz/).

Boxplot

A preferred method of illustration to a bar graph when reporting data across different groups/time is a boxplot. It reports the median, 25 and 75% quartiles, spread of the data, distribution and outliers, all of which the bar graph does not show. For instance, the minimum and maximum data points are depicted with the *whiskers* that extend to the top and the bottom of the box in the

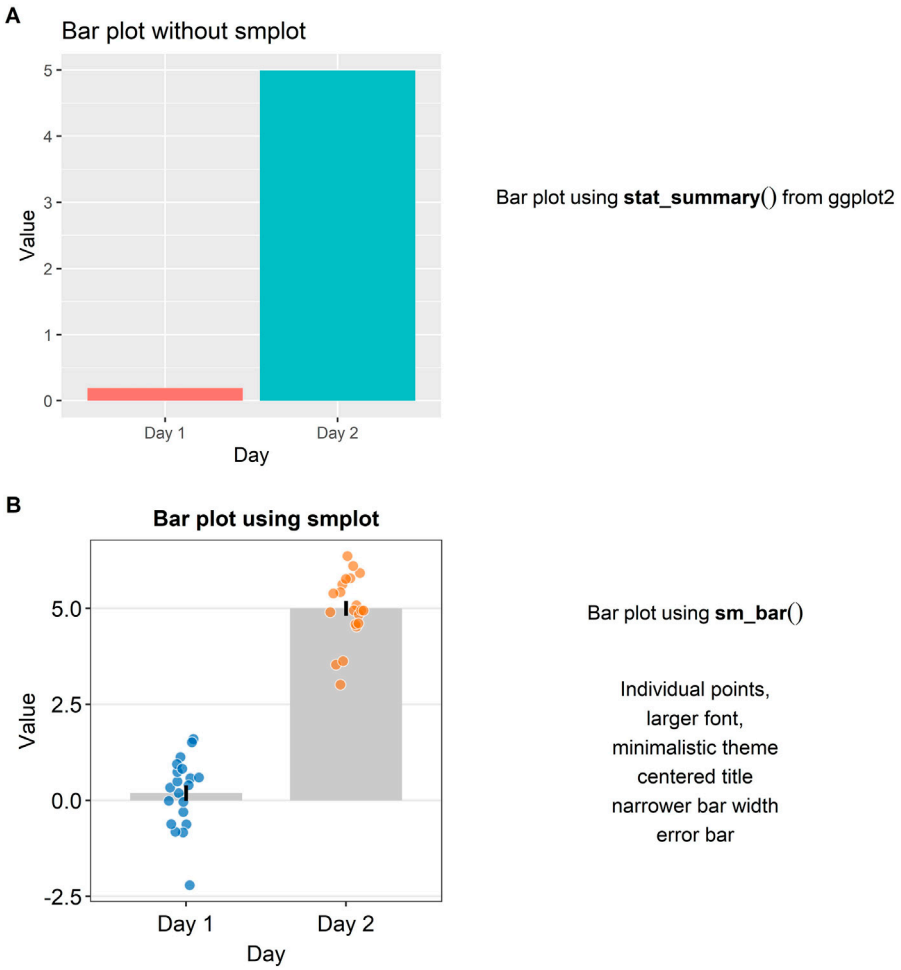

FIGURE 2 | Bar plots with and without *smplot*. **(A)** A bar plot drawn with "*stat_summary()*", which is a function of ggplot2. **(B)** A bar plot drawn with "*sm_bar()*", which is a function of smplot. This function automatically provides several features, such as individual data points, larger font, minimalistic theme, centered title, narrower bar width and error bars, such as standard error, standard deviation or 95% confidence interval.

center. The horizontal line within the box represents the *sample median*. Points that are residing above or below the whiskers represent *outliers*.

In **Figure 3A**, a boxplot using the default themes of *ggplot2* is shown. On its own, it is not very informative because the individual points are not displayed. Aesthetically, there are some distracting features such as the major and minor vertical and horizontal grids and unnecessarily wide boxes. If the reader adds "*sm_boxplot()*" to the given plot in **Figure 3A**, she will be able to resolve these aesthetic issues (see **Figure 3B**).

Violin Plot

Another alternative to a bar graph is a violin plot. A violin plot is sometimes preferred to a boxplot because it shows the full distribution of the data while the boxplot fails to do so. The "violin" of the violin plot represents the data distribution (see **Figure 4A**). The region with the largest width denotes the highest density of the data. The upper- and lowermost tips of the "violin" represent the maximum and minimum values of the data.

In **Figure 4A**, a violin plot using the default theme of ggplot2 (and without *smplot*) is shown. It lacks individual points and error bars, such as standard deviation. The aesthetics also need some improvement. Instead, when "*sm_violin()*" is modularly added to the plot in **Figure 4A**, the violin plot gets improved visually (see **Figure 4B**).

Slope Chart

A slope chart is often used to directly compare paired data at different timepoints or instances (see **Figures 5A,B**). With a slope chart, one can track changes over time for each data point (i.e., before and after experimental manipulation). If one is interested in performing a statistical test that accounts for repeated measures (ex., repeated measures one-way analysis of variance), a slope chart can be a good choice for plotting data.

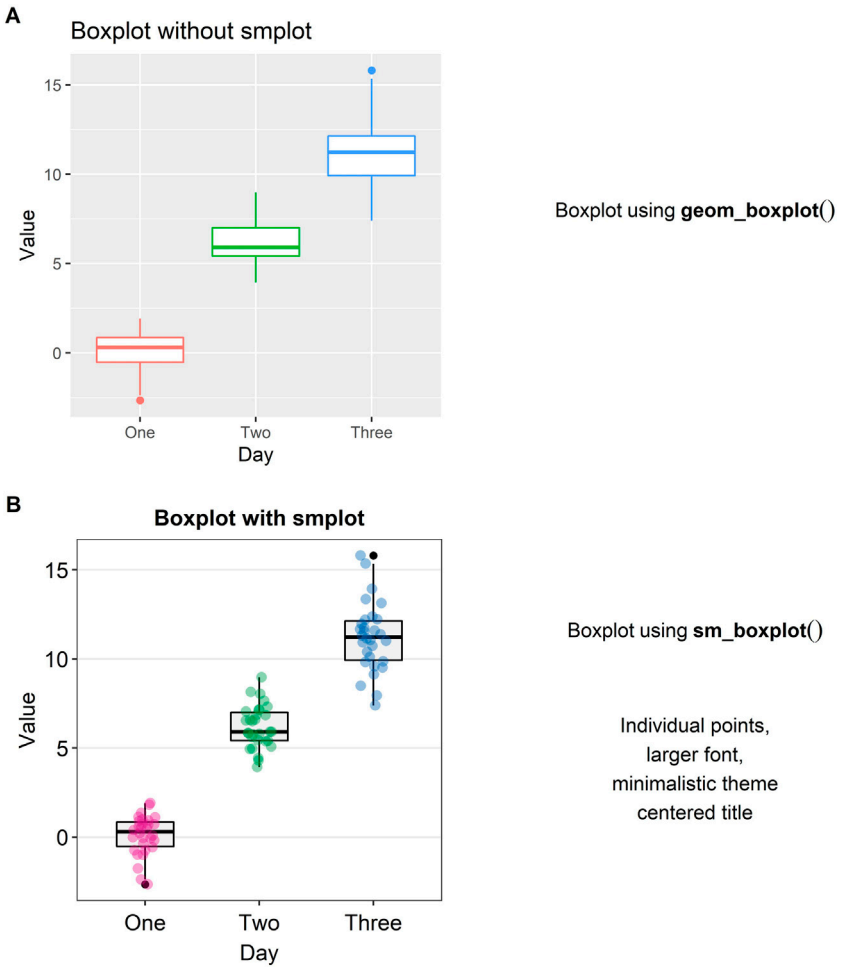

FIGURE 3 | Boxplots with and without *smplot*. **(A)** A boxplot drawn with "*geom_boxplot()*", which is a function of *ggplot2*. **(B)** A boxplot drawn with "*sm_boxplot()*". This function automatically provides several features, such as individual data points, larger font, minimalistic theme, and centered title.

The *ggplot2* package does not offer a single function that plots a slope chart. To use *ggplot2*, one might need to code multiple line of code to strip away the default *ggplot2* theme and construct an appropriate slope chart, a task that can be tedious and repetitive. For this reason, "*sm_slope()*" has been created.

Figure 5A shows a slope chart that has two levels in the x-axis, whereas **Figure 5B** shows a slope chart that has four levels in the x-axis. "*sm_slope()*" plots these slope charts with the same command code by automatically detecting the number of discrete x-levels, provided that the loaded data has a proper data frame structure (see example: https://github.com/smin95/dataviz/blob/master/data.csv).

Raincloud Plot = Violin Plot + Boxplot + Individual Data

A raincloud plot is a combination of a violin plot (halved), a boxplot and jittered individual data (Allen et al., 2021). Plotting a raincloud plot might be challenging for newcomers in R. Although there exists an R package (the *raincloudplots* package) that plots a raincloud plot (Allen et al., 2021), the function "*sm_raincloud()*" has been created to allow for more visual customisation.

Figure 6A shows a raincloud plot that has two discrete levels in the x-axis (Day 1 and Day 2). These levels are denoted by the distinct colors pink and blue. In this example, the jittered points, boxplot and violin plot overlap with each other because the separation level is set to 0 ("*sep_level = 0*"). "*sep_level*" is an argument for the function "*sm_raincloud()*." The separation level ranges from 0 to 4, so one can increase the separation amongst the plots by setting "*sep_level = 2*" within the "*sm_raincloud()*" function as shown in **Figure 5B**. When "*sep_level = 2*," the violin plot and the boxplot overlap each other but not the individual data points are located apart.

Another argument for "*sm_raincloud()*" is "*which_side.*" The reader may notice that the direction at which the pink violin plot is facing is to the left rather than the right (see **Figure 6A**). If the

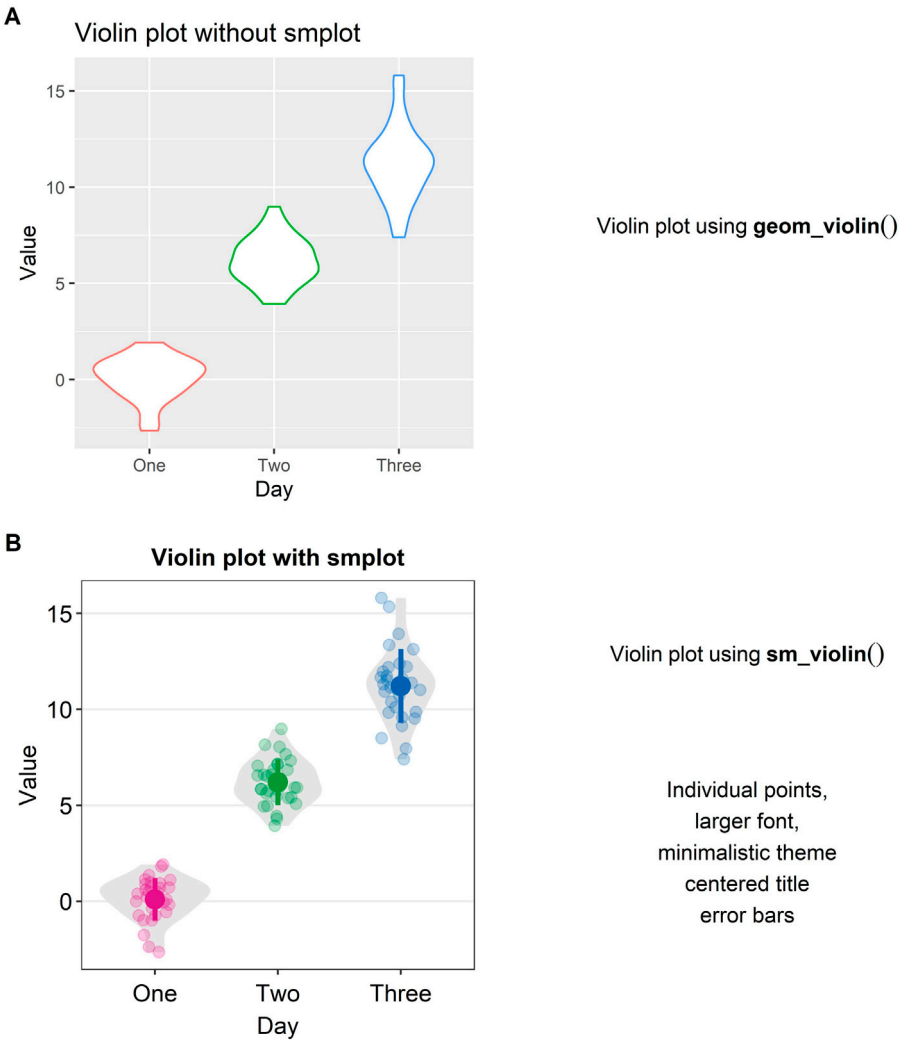

FIGURE 4 | Violin plots with and without *smplot*. **(A)** A violin plot drawn with "*geom_violin()*", which is a function of *ggplot2*. **(B)** A violin plot drawn with "*sm_violin()*". This function automatically provides several features, such as individual data points, larger font, minimalistic theme, centered title, narrower bar width and error bars, such as standard error, standard deviation and 95% confidence interval.

argument "*which_side*" is set to right, all the violin plots face to the right (see **Figure 6A**). However, if "*which_side = mixed*," then the directions of the violin plots become asymmetric so that the jittered individual points at each of the two x-level are closest to one another (**Figure 6B**). Also, "*which_side = mixed*" is only allowed when there are two discrete levels of x-axis, and the function "*sm_raincloud()*" throws an error when the condition is not met.

In **Figure 6C**, separation level has been specified to 4, i.e., "*sep_level = 4.*" This allows the features of the raincloud to be separated from one another more. Also, the violin plots at each x level are facing to the left, i.e., "*which_side = left.*"

The function "*sm_raincloud()*" also plots a raincloud plot when the x-level exceeds 2 (see **Figure 6D**). This is a novel feature that is not included in the original R package (the *raincloudplots* package) that draws a raincloud plot. It also automatically counts the number of discrete x levels if the loaded data has a proper data frame structure (see example: https://github.com/smin95/dataviz/blob/master/data.csv).

Case Study Using *smplot*: Test-Retest Reliability of a Novel Method

When one is interested in introducing a new measurement method, one must examine whether the new method (i.e., Method 2) shows agreeable results to those obtained from the standard method (i.e., Method 1). In this section, we present a case study where smplot might be useful. All data and codes are uploaded in Chapter 6 of the online guide (https://smin95.github.io/dataviz/).

If the data across two different instances/methods are paired, one can draw a slope chart, rather than a bar plot, to see if the data between these two instances show a small variability. For instance, each dot in **Figure 7A** can represent an individual

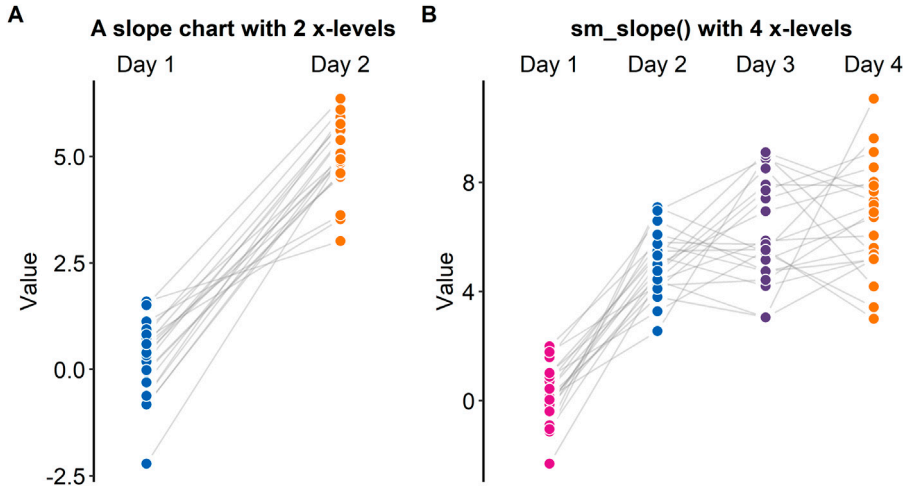

FIGURE 5 | Slope charts drawn with "sm_slope *()*", which is a function of *smplot*. **(A)** A slope chart with two discrete x-levels. **(B)** A slope chart with four discrete x-levels.

sample (ex. Patient 1 out of 20) from which the gene expression level is measured. In this example, if a new method is consistent with the standard method, the individual data from each specimen will have a flat grey line. As we have previously mentioned, this can be achieved by using the "*sm_slope()*" function.

Another popular method to demonstrate that a new technique is reliable is to compute whether there is a high correlation coefficient (see **Figure 7B**; Pearson's correlation test is provided in this example). Unfortunately, a high correlation does not indicate a good replicability. In **Figure 7B**, we see that the correlation is robust (R = 0.64, p = 0.014). However, the correlation seems to be heavily dependent on one single point in the top-right corner of **Figure 7B** (Method 1 = 2.5, Method 2 = 2.5). If the correlation coefficient is computed without the extreme point, it might be more representative of whether the new method is truly correlated with the standard method. In this example, the correlation without the extreme point turns out to be weak, R = 0.24, p = 0.32. If the reader encounters a similar situation to this case study, we suggest that the reader compute the correlation with and without the outlier, and then determine which of the two correlation coefficients is more representative.

An appropriate approach to report test-retest variability is to show a Bland-Altman plot (see **Figure 7C**), which is also known as a MA plot (M = minus, A = average) in the field of genomics (Bland and Altman, 1986; Giavarina, 2015). The y-axis of the Bland-Altman is the difference between data from the two methods, whereas the x-axis denotes the mean of the data from the two methods. This plot aims to describe agreements between data from two instances. Bland and Altman have stated that 95% of the scatter points in a Bland-Altman plot should reside within the limits of agreement (dashed line in **Figure 6C**), which represent ±1.96 standard deviations from the mean difference between data from two sessions (Bland and Altman, 1986). Whether the mean difference between two instances is too large or not can be determined by calculating the mean difference of all paired individual data. If the mean difference of the data is not significantly different from 0 (i.e., one-sample t-test), then it is acceptable to surmise from the given data that there is a good agreement between the two methods. This is also the case in **Figure 7C**. A Bland-Altman plot can be drawn using these two functions "*sm_statBlandAlt()*" and "*sm_bland_altman()*."

Who Is smplot for?

The *smplot* package is for those who is interested in plotting elegant graphs with minimal codes in a modular fashion. It aims to simplify the process of data visualization for incoming research trainees in fields such as biomedical sciences. That being said, it is not necessary to produce high-quality graphs. We have encountered numerous medical students who use multiple software environments to create and polish figures, a process that is often laborious and tedious. For instance, if students have already created a figure and decide to collect additional data, they will find themselves to change their figures across multiple software platforms, such as Matlab and Adobe Illustrator. We hope to have convinced the reader that *smplot* can be used to create a polished, satisfying figure within one software environment with minimal coding.

If the reader is interested in learning more about R, please consider reading *R for Data Science* by Hadley Wickham (Wickham and Grolemund, 2016). If the reader is interested in developing her own color palette, please visit the online guide of Seaborn (https://seaborn.pydata.org/tutorial/color_palettes.html), which is a data visualization library in Python (Waskom, 2021). If the reader is interested in learning important practices of data visualization, please consider reading *Fundamentals of Data Visualization* by Claus Wilke (Wilke, 2019b); he is the author of the *cowplot* package (Wilke, 2019a).

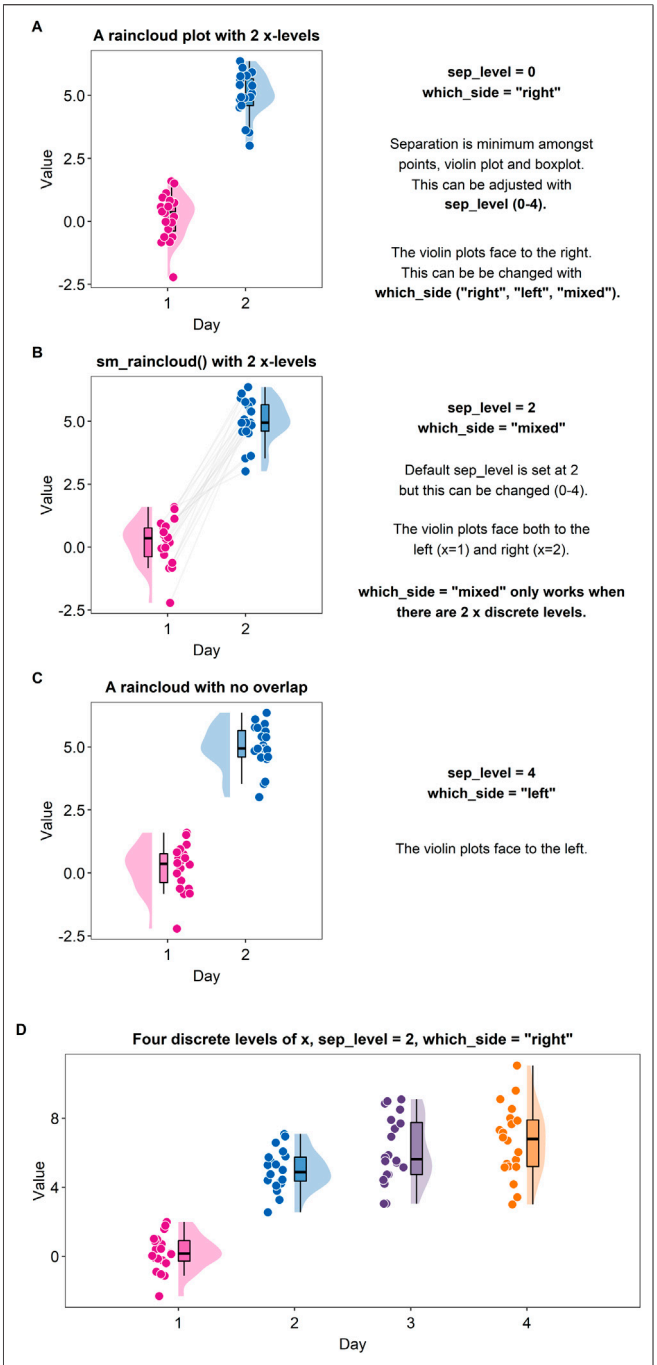

FIGURE 6 | Raincloud plots drawn with "*sm_raincloud ()*". **(A)** A raincloud plot with two discrete x-levels. In this example, the halved violin plots face to the right because the violins have been specified to face to the right, i.e., "*which_side = "right"*". Also, the violin plots, boxplots and jittered individual data points all overlap among each other because the separation level has been specified to 0, i.e., "*sep_level = 0*". **(B)** A raincloud plot with two discrete x-levels. In this example, the halved violin plots both face to the left and right, i.e., "*which_side = "mixed"*". This mixed configuration allows the individual points to be paired visually using the grey lines. Also, although the halved violin plots and boxplots overlap, the jittered individual data are located separately because the separation level has been specified to 2, i.e., "*sep_level = 2*". **(C)** A raincloud plot with two discrete x-levels. In this example, the halved violin plots face to the left because the argument

(*Continued*)

FIGURE 6 | "*which_side = "left"*". Also, the halved violin plots, boxplots and the jittered individual data do not overlap. There is more separating distance than the plot in panels A and B because the separation level has been specified to 4, i.e., "*sep_level = 4*". **(D)** A raincloud plot with four discrete x-levels.

Contributions of the Package

The smplot package provides numerous functions that quicken the process of data visualization. Most functions are wrapper functions around ggplot2 that aim to change the default of the aesthetics. We also provide new functions, such as "*sm_bland_altman()*" and "*sm_raincloud()*" that do more than changing the default theme of ggplot2. "*sm_bland_altman()*" plots a Bland-Altman plot (along with the mean, upper and lower limits) and a grey shaded region that represents the 95% confidence interval; these labels are all necessary but the R package (ex. the *BlandAltmanLeh* package) for Bland-Altman plots do not necessarily plot all of these features by default and do not use the ggplot2 interface. Moreover, "*sm_raincloud()*" draws a raincloud plot that is more customisable than the original package that draws a raincloud plot (the *raincloudplots* package). smplot does not impose the limits of the number of discrete x-levels unlike the original package (the *raincloudplots* package). For example, the *raincloudplots* package is not capable of plotting **Figure 6D** because the graph requires 4 discrete x-levels. In addition, the configuration of the violin plots in the raincloud plot as well as the aesthetics can also be more customised than before. Lastly, unlike the *raincloudplots* package, the data structure can have the same format as the one required for ggplot2; this consistency of the data structure between raincloud plots and other ggplot2 figures can allow the user to draw multiple graphs without modifying the data structure.

The *ggpubr* package is a well-known R package for data visualization. However, many plotting functions of the *ggpubr* package are one-liner, rather than modular, functions that plot a complete graph. For this reason, there are numerous stored defaults that might not be accessible for the user to modify. If a modular function is added to a plot that is created with *ggpubr* to change default aesthetics of *ggpubr*, warnings may appear. For this reason, the *smplot* package provides functions that can be added modularly (ex. "*sm_hgrid*" and "*sm_statCorr*") to the given plot built with ggplot2 or be added to ("*sm_bland_altman*" and "*sm_raincloud*") by other modular functions.

The smplot package provides multiple themes with an interesting feature. First, as is the case of the themes of the *cowplot* package, they can be added in a modular fashion to a given ggplot2 plot (ex. base plot "+ *sm_hgrid()*"). Also, the theme functions of smplot provide a separate argument for the border and the legend (ex. "*sm_hgrid(legends = FALSE, borders = TRUE)*." If "*legends = FALSE,*" the legend will be hidden; if "*borders = TRUE,*" there will be a border around the panel. When these settings are flipped ("*legends = TRUE*" and "*borders =*

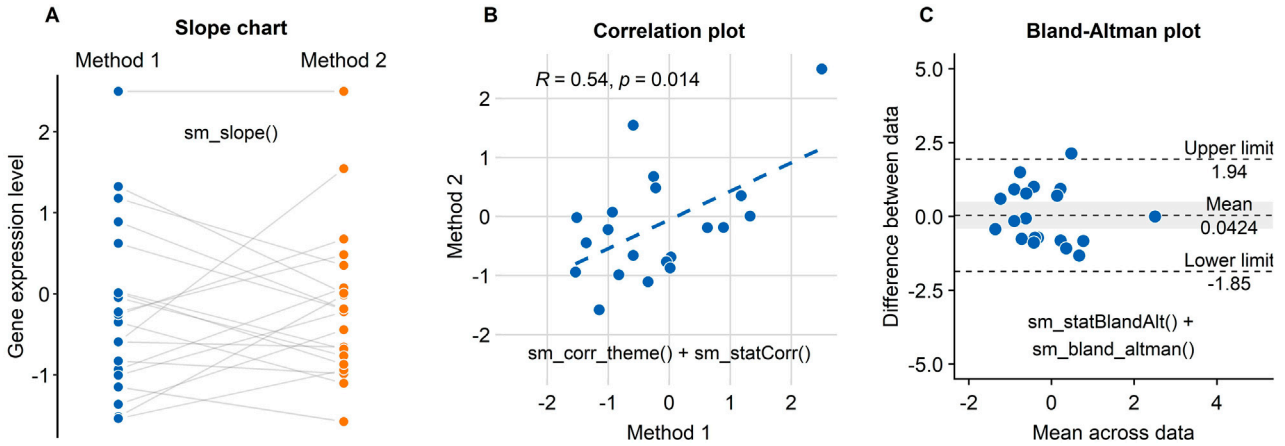

FIGURE 7 | Three figures that examine the test-retest reliability of a new method/technology to measure gene expression have been created using *smplot*. **(A)** A slope chart drawn using "*sm_slope()*". The blue points represent data from the standard method, whereas the orange points show data from the new method. Each dot represents an individual sample. The grey lines indicate the pairing of the points. If the grey lines uniformly have a positive or negative slope, then one can infer that the new method consistently show data of higher or lower values, respectively. **(B)** A correlation plot drawn using "*sm_corr_theme()*" and "*sm_statCorr()*". Unfortunately, there is an extreme point at the top-right of the panel; it heavily skews the correlation to be robust. Without it, the correlation is much weaker, i.e., $R = 0.24$, $p = 0.32$. **(C)** A Bland-Altman plot drawn using "*sm_statBlandAlt()*" and "*sm_bland_altman()*". The difference in data between the two methods are plotted as a function of the mean across the data from the two methods. The upper and lower dashed lines denote 95% limits of agreement; the wider the range encapsulated by the limits of agreement, the more measurement variability there is between the methods. The dashed line in the middle indicates the mean of the difference between the data from the two methods. The grey area represents 95% confidence interval of the difference in data between the methods estimated from a t-distribution. If the middle-dashed line (mean difference) does not overlap with the grey area (95% confidence interval), then the mean difference is significantly different from 0 based on one-sample t-test ($p < 0.05$), thereby indicating that the new method shows a very poor agreement with the standard method. In this panel, however, we see an overlap.

TRUE"), the relative proportion of the figure as well as the perceived size of the text have been set to appear the same. These features have been added for convenience because the user is otherwise forced to use "*theme()*," which can be tedious and confusing to use. The themes provided by the *cowplot* package do not offer these features.

DATA AVAILABILITY STATEMENT

The *smplot* R package is free and open source. All sample data and codes of the figures can be accessed in the online guide (https://smin95.github.io/dataviz/). The source codes of smplot are available in Github (https://github.com/smin95/smplot). *smplot* requires *ggplot2* (Wickham, 2016) and *cowplot* (Wilke, 2019a) packages, both of which are automatically downloaded when *smplot* is installed via RStudio. Please cite this article when smplot is used.

AUTHOR CONTRIBUTIONS

SM and JZ wrote the manuscript. SM created the *smplot* package. All authors approved the final submission.

FUNDING

This study was supported by the Project of State Key Laboratory of Ophthalmology, Optometry and Vision Science, Wenzhou Medical University (J02-20210203).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2021.802894/full#supplementary-material

REFERENCES

Allen, M., Poggiali, D., Whitaker, K., Marshall, T. R., Kievit, R. A., and Kievit, R. (2021). Raincloud Plots: A Multi-Platform Tool for Robust Data Visualization. *Wellcome Open Res.* 4, 63. doi:10.12688/wellcomeopenres.15191.1

Bland, J. M., and Altman, D. (1986). Statistical Methods for Assessing Agreement between Two Methods of Clinical Measurement. *The Lancet* 327 (8476), 307–310. doi:10.1016/s0140-6736(86)90837-8

Giavarina, D. (2015). Understanding Bland Altman Analysis. *Biochem. Med.* 25 (2), 141–151. doi:10.11613/bm.2015.015

Hunter, J. D. (2007). Matplotlib: A 2D Graphics EnvironmentComputing in Science & Engineering. *Comput. Sci. Eng.* 9 (3), 90–95. doi:10.1109/mcse.2007.55

Waskom, M. (2021). Seaborn: Statistical Data Visualization. *J. Open Source Softw.* 6 (60), 3021. doi:10.21105/joss.03021

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer.

Wickham, H., and Grolemund, G. (2016). *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. O'Reilly Media, Inc.

Wilke, C. O. (2019a). Cowplot: Streamlined Plot Theme and Plot Annotations for 'ggplot2. R package version 0.9 4.
Wilke, C. O. (2019b). *Fundamentals of Data Visualization: A Primer on Making Informative and Compelling Figures*. O'Reilly Media.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers in Genetics

ORIGINAL RESEARCH
published: 20 December 2021
doi: 10.3389/fgene.2021.799799

Network Pharmacology and Inflammatory Microenvironment Strategy Approach to Finding the Potential Target of *Siraitia grosvenorii* (Luo Han Guo) for Glioblastoma

Juan Li[1†], *De Bi*[2†], *Xin Zhang*[1], *Yunpeng Cao*[3]*, *Kun Lv*[1,4]* and *Lan Jiang*[1,4]*

[1]Key Laboratory of Non-coding RNA Transformation Research of Anhui Higher Education Institution, Yijishan Hospital of Wannan Medical College, Wuhu, China, [2]Suzhou Polytechnic Institute of Agriculture, Suzhou, China, [3]Wuhan Botanical Garden, Chinese Academy of Sciences, Wuhan, China, [4]Central Laboratory, Yijishan Hospital of Wannan Medical College, Wuhu, China

OPEN ACCESS

Edited by:
Meng Zhou,
Wenzhou Medical University, China

Reviewed by:
Fuhai Li,
Washington University in St. Louis, United States
Shuanglong Yi,
ShanghaiTech University, China
Wei Li,
Central South University, China

***Correspondence:**
Lan Jiang
jianglanhi@163.com
Kun Lv
lvkun@yjsyy.com
Yunpeng Cao
xfcypeng@126.com

†These authors share first authorship

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

Received: 22 October 2021
Accepted: 15 November 2021
Published: 20 December 2021

Citation:
Li J, Bi D, Zhang X, Cao Y, Lv K and Jiang L (2021) Network Pharmacology and Inflammatory Microenvironment Strategy Approach to Finding the Potential Target of Siraitia grosvenorii (Luo Han Guo) for Glioblastoma. Front. Genet. 12:799799. doi: 10.3389/fgene.2021.799799

Background: Glioblastoma (GBM) is the most common and aggressive primary intracranial tumor of the central nervous system, and the prognosis of GBM remains a challenge using the standard methods of treatment—TMZ, radiation, and surgical resection. Traditional Chinese medicine (TCM) is a helpful complementary and alternative medicine. However, there are relatively few studies on TCM for GBM.

Purpose: We aimed to find the connection between TCM and anti-GBM.

Study design: Network pharmacology and inflammatory microenvironment strategy were used to predict *Siraitia grosvenorii* (Luo Han Guo) target for treating glioblastoma.

Methods: We mainly used network pharmacology and bioinformatics.

Results: CCL5 was significantly highly expressed in GBM with poor prognostics. Uni-cox and randomForest were used to determine that CCL5 was especially a biomarker in GBM. CCL5 was also the target for SG and TMZ. The active ingredient of Luo Han Guo — squalene and CCL5 —showed high binding efficiency. CCL5, a chemotactic ligand, was enriched and positively correlated in eosinophils. CCL5 was also the target of Luo Han Guo, and its effective active integrate compound —— squalene — might act on CCL5.

Conclusion: SG might be a new complementary therapy of the same medicine and food, working on the target CCL5 and playing an anti-GBM effect. CCL5 might affect the immune microenvironment of GBM.

Keywords: ***Siraitia grosvenorii*, CCL5, glioblastoma, in silico, network pharmacology**

INTRODUCTION

Glioblastoma (GBM) is the most common and aggressive primary intracranial tumor of the central nervous system (Barthel et al., 2019; Miller et al., 2019). Most of them are induced by genetic mutations of high penetrance genes related to rare syndromes, mainly manifested as increased intracranial pressure, neurocognitive dysfunction, and seizures, resulting in central nervous system damage and endangering the lives of patients (Zanders et al., 2019). The standard treatment for GBM

is surgery, drug therapy, and radiation therapy, and the median survival time of patients is only 15 months (Kumar et al., 2019). With the changes in eating habits, living environment, and work pressure, the incidence of GBM is increasing and getting younger. Surgical resection combined with postoperative radiotherapy, chemotherapy, and immunotherapy will inevitably damage the body's normal function and cause adverse reactions. Multi-drug resistance, especially temozolomide (TMZ), leads to frequent GBM recurrences, which is a challenge in treating GBM, and its underlying molecular mechanism is still unclear (Yin et al., 2019). Since the blood-brain barrier (BBB) can prevent the accumulation of charged or macromolecules in the tumor microenvironment at a physically relevant concentration, thereby exerting an oncolytic effect, the content of TMZ in the brain is only 40% percent of the content in the blood, and new component pharmacological methods must be developed to enhance the curative effect of the current treatment, prolong the median survival time of the patient to exceed the median survival time of 15 months (Kumar et al., 2019).

The plants of traditional Chinese medicine (TCM) were used for the treatment of various cancers (Dai et al., 2016), such as GBM (Wang et al., 2019). The use of TCM to promote health and adjuvant therapy is becoming increasingly popular worldwide (Khan and Tania, 2020). The active components of *Salvia miltiorrhiza* can inhibit the proliferation of U87 cells, induce apoptosis, and enhance the efficacy of TMZ (Wang et al., 2019). *Lycium chinense* can up-regulate CD3+T, CD8+T, and TNF-α, inhibit the proliferation of mouse C6 cells, and up-regulate $CD4^{+}CD5+T$ cells to prolong survival and regulate the BBB (Wang et al., 2019). Magnolol inhibits the migration and proliferation of GBM cells through the JAK-STAT3 signal pathway, mainly by inhibiting the production of GBM stem cell-like cells (Fan et al., 2019). However, the clinical application value of TCM in the treatment of GBM has not been promoted, and more molecular mechanism studies are needed to verify it. Therefore, our research aims to provide new potential for the treatment of GBM with a medicinal plant.

The TCM *Siraitia grosvenori* (SG) is a perennial herbaceous plant of the Cucurbitaceae family with huge resource reserves and native to southern China, also known as monk fruit and Luo Han Guo, which is a medicinal food homologous species granted by the China Food and Drug Administration with significant clinical effects (Xia et al., 2018). Mogroside has an excellent biological development, which can inhibit the excessive activation of Signal Transducer and Activator of Transcription 3 (STAT3) and promote tumor cell apoptosis (Liu et al., 2018), and targeting STAT3 can improve tumor progression and anticancer immunity response (Lee et al., 2019); reversing emergency medical technician (EMT) and destroying the cytoskeleton to inhibit hyperglycemia-induced lung cancer cell metastasis (Guan et al., 2019). Mogroside IIV and IIIV activate AMP-activated protein kinase (AMPK) and produce anti-hyperglycemic and anti-lipid properties in the body (Abdel-Hamid et al., 2020); mogroside V can cross the BBB and affect schizophrenia-like behavior (Ju et al., 2020) and can also exert neuroprotective activity (Xia et al., 2013); mogroside IVe may be potentially used as a bioactive phytochemical supplement for the treatment of colorectal cancer and laryngeal cancer (Liu et al., 2016). Monk fruit also has other pharmacological effects, such as up-regulating Sirtuin 1 (SIRT1) to reduce oxidative stress and alleviate the decline in oocyte quality during *in vitro* aging (Nie et al., 2019).

Network pharmacology is based on the high-throughput multi-omics data analysis to clarify the mechanism of multi-component/multi-target/multiple action pathways in medicinal plants (Hopkins, 2008). The newly network pharmacology analysis was employed to integrate active compounds, targets and pathways prediction, and network analysis which may provide novel insights into the therapeutic effects and molecular mechanisms of SG in the treatment for GBM (Abdel-Hamid et al., 2020). Then, we offered a new flowchart to explain the potential target of *Siraitia grosvenorii* (Luo Han Guo) for GBM (**Figure 1**).

MATERIALS AND METHODS

The Integration of SG-TMZ-GBM (*Siraitia grosvenori* - Temozolomide - Glioblastoma) Targets

Through PubMed database (https://pubmed.ncbi.nlm.nih.gov) text mining, we selected the most effective clinical drug in the treatment of GBM. Based on the TCMSP (Ru et al., 2014) database (blood-brain barrier (BBB) ≥ 0.3, drug-like (DL) ≥ 0. 18, oral bioavailability (OB) ≥ 30%), and TCMID (Huang et al., 2018), we collected the active ingredients and targets in monk fruit. Then, we used the chemical components to obtain the structure files by the PubChem Compound database (Kim et al., 2019) and uploaded the structure files to predict the targets across the PharmMapper (Wang et al., 2017) and Swiss Target Prediction (Gfeller et al., 2014). A Venn diagram (http://bioinformatics.psb.ugent.be/webtools/Venn/) was drawn for visualizing the SG-TMZ interacting targets. Gliomas-related targets were predicted by OMIM (Amberger et al., 2015), DrugBank (Wishart et al., 2018), and PubMed. Then, taking the intersection with the prediction targets of SG-TMZ, which is named Siraitia grosvenori - temozolomide – gene (SG-TMZ-G).

We downloaded GBM's transcriptomic and clinical data and normal brain tissues from XENA TCGA and GTEx (https://xena.ucsc.edu/public/). Differentially expressed GBMs (DE-GBMs) were computed by limma (Smyth, 2005) with $|\log FoldChange\ (logFC)| > 2$ and q-value < 0.05 as previously reported (Jiang et al., 2020a; Jiang et al., 2020b). Subsequently, common GBM-related targets were integrated between SG-TMZ-G. A volcano plot was used to show the distribution of SG-TMZ-GBM (SG-TMZ-glioblastoma).

Functional Analysis and Network Construction of SG-TMZ-GBM

STRING v11.5 was used to construct a protein-protein interaction (PPI) network, scores >0.70 were considered to have high confidence (Szklarczyk et al., 2021). Functional analyses of the gene ontology (GO), the Kyoto Encyclopedia of Genes and Genomes (KEGG) were performed by ClueGO

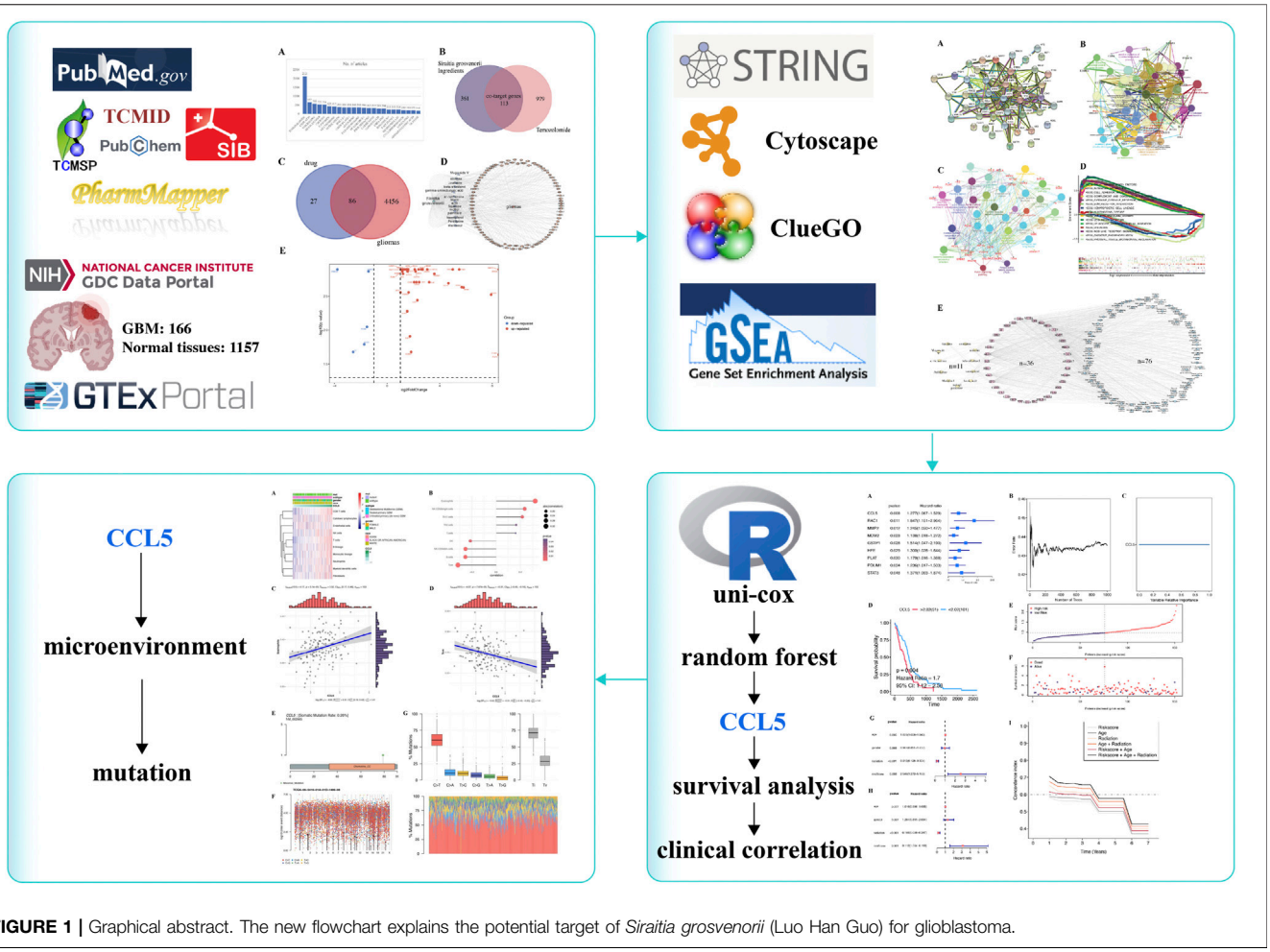

FIGURE 1 | Graphical abstract. The new flowchart explains the potential target of *Siraitia grosvenorii* (Luo Han Guo) for glioblastoma.

plug-in (Bindea et al., 2009) in Cytoscape v3.8.2 (Reimand et al., 2019) with q-value ≤ 0.001. The compound-target-pathway network was built by Cytoscape (Reimand et al., 2019).

The Determination of the Key SG-TMZ-GBMs

Hazard ratios (HR) were applied using univariable Cox (uni-cox) regression analysis (p-value < 0.05). We then detected the key SG-TMZ-GBMs by "survival" and "survminer" package (Jiang et al., 2020b). Random forest was calculated by randomSurvivalForest to rank the importance of survival-related SG-TMZ-GBMs, with a relative importance >0.7 as the final feature (Liu et al., 2021). Survival analysis was built with the best cutoff value (Liu et al., 2021), the Kaplan-Meier method was used to draw survival curves, and the log-rank test was used to evaluate differences. A scatter plot of C-C Motif Chemokine Ligand 5 (CCL5) expression and survival time in GBM patients were drawn by ggrisk (Jiang et al., 2020b). The forest plot was used for performing uni-cox and multiple cox (multi-cox) regression analysis (Jiang et al., 2020b). We also used the receiver operating characteristic curve (ROC), concordance index (c-index) to evaluate the multi-clinical prognostic performance (Longato et al., 2020).

INFLAMMATORY MICROENVIRONMENT AND MUTATION ANALYSIS

The microenvironment cell population-counter method was chosen to evaluate the association between CCL5 and immune cell populations (Petitprez et al., 2020). We used immune cells markers and GBM transcriptome data to validate the strong correlation between CCL5 and 24 immune cells markers (Bindea et al., 2013). Gene mutations of GBM expression by "maftools" package (Mayakonda et al., 2018). CCL5 protein expression was detected by immunohistochemistry from the HPA database (https://www.proteinatlas.org/ENSG00000271503-CCL5/pathology/glioma#).

RESULTS

SG-TMZ-GBM Detection

PubMed text mining showed 2121 literature reports on the treatment of GBM with TMZ (**Figure 2A**). Through the text data mining of the Therapeutic Target Database (TTD) database and PubMed published articles, we identified 1092 target genes for treating GBM with TMZ.

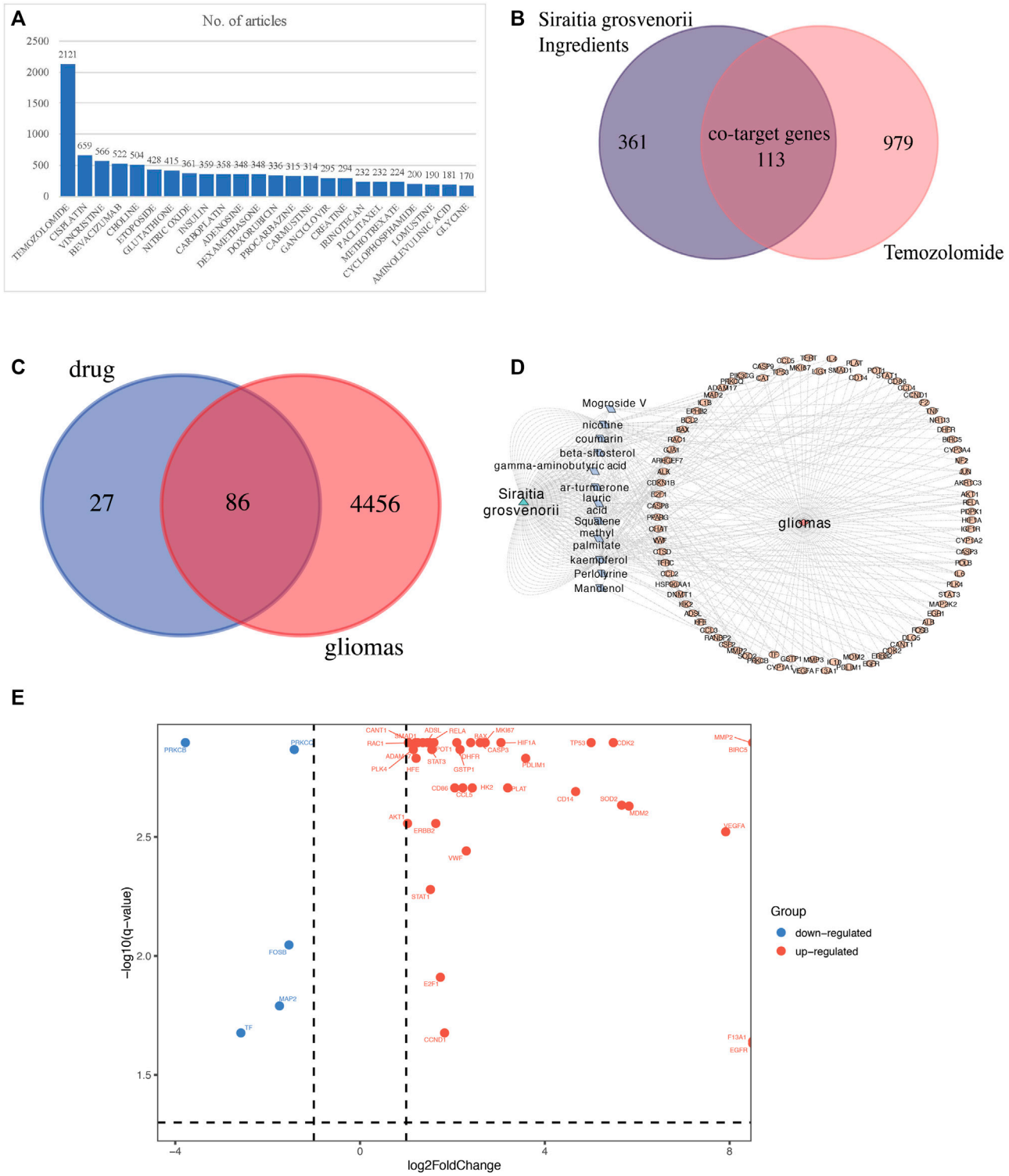

FIGURE 2 | SG-TMZ-GBM detection. **(A)** The analysis result on the clinical drug in treating gliomas in PubMed database (updated by 2021-8-10); **(B)** co-target genes predicted between the ingredients of *Siraitia grosvenorii* and temozolomide; **(C)** co-target genes predicted between the ingredients of *Siraitia grosvenorii* -temozolomide and gliomas; **(D)** co-target genes predicted network between the ingredients of *Siraitia grosvenorii* and gliomas, a triangle means *Siraitia grosvenorii*, a diamond means the 12 ingredients of *Siraitia grosvenorii*, an oval means co-target genes in gliomas; **(E)** volcano plot for differentially expressed target genes.

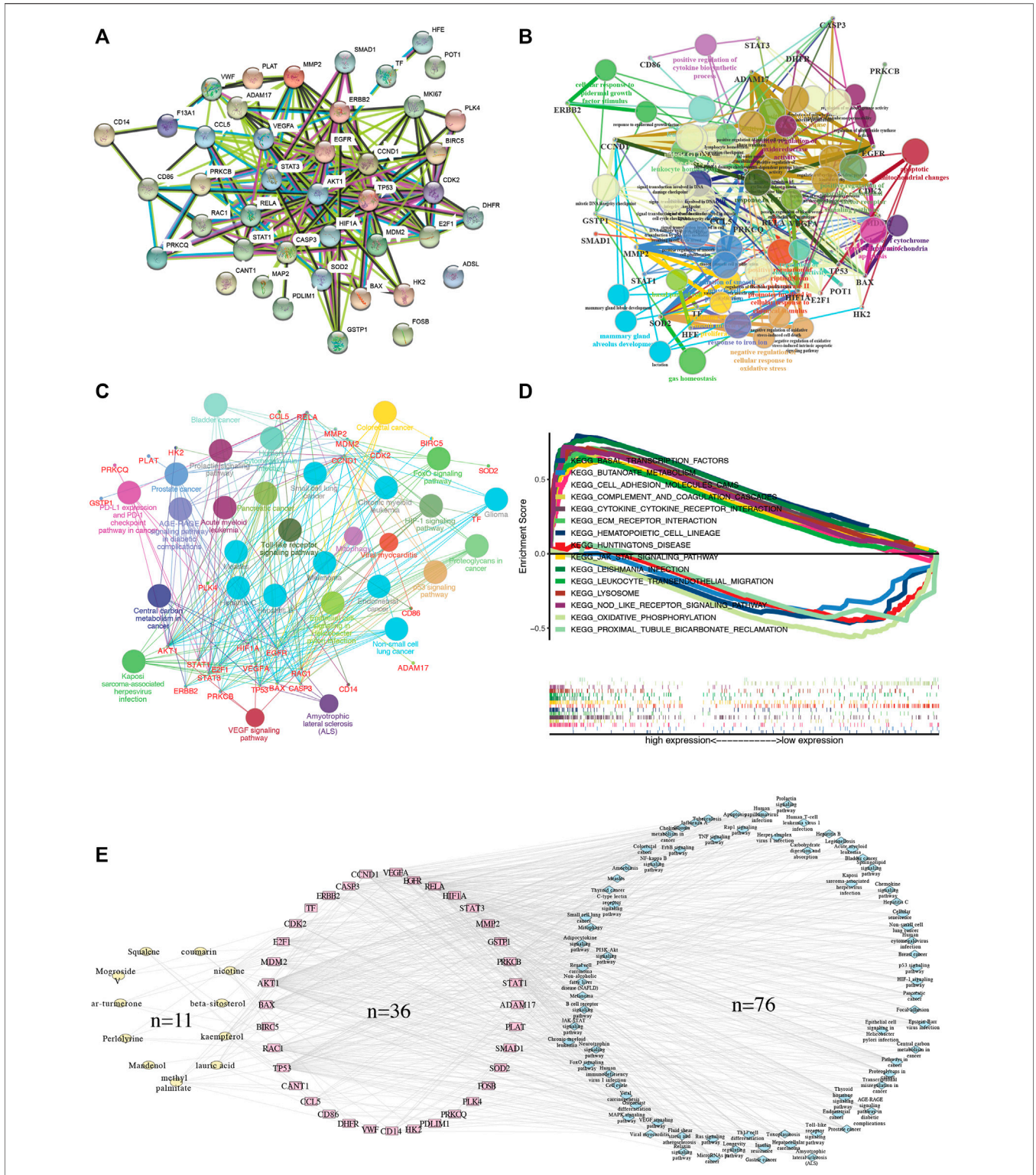

FIGURE 3 | Targeted gene functional analysis. **(A)** PPI network analysis; **(B)** GO enrichment analysis; **(C)** KEGG pathway enrichment analysis; **(D)** the study of GSEA KEGG enrichment plot. **(E)** Compound-target-KEGG pathway network. The network was generated by Cytoscape 3.8.2. Yellow circles represent 11 ingredient targets from *Siraitia grosvenorii*. Red circles represent 36 common targets between ingredient targets from *Siraitia grosvenorii* and GBM significant targets. Blue circles represent 76 KEGG pathways.

We obtained 12 chemical compositions in SG by TCMSP and TCMID, and gained the compound structure by PubChem, and predicted the 474 target genes of SG by the Swiss Target Prediction and PharmMapper. A total of 113 SG-TMZ targets were found by taking the intersection (**Figure 2B**). We further discovered 4542 target genes related to gliomas through PALM-IST, filtered 86 target genes as SG-TMZ-G (**Figure 2C**), and drew a network diagram (**Figure 2D**). For example, IL6 was a co-target gene in the gamma-aminobutyric acid, lauric acid, and methyl palmitate of SG and GBM; CCL5 was a co-target gene in squalene. According to the cutoff log2FoldChange> 2 and q-value <0.05, we screened the interaction of differentially expressed genes (DEGs) in GBM-normal brain tissues and SG-TMZ-G, Volcano plot for 42 SG-TMZ-GBM targets were detected for the following research (**Figure 2E**).

Luo Han Guo Compound-Target-Disease Interaction Network and Functional Enrichment Analysis

We imported 42 SG-TMZ-GBMs into the STRING database to construct a protein-protein interaction (PPI) network, the primary connection in the network which might have pharmacological effects in GBM. In addition, the four targets, including Protection of Telomeres 1 (POT1), Adenylosuccinate Lyase (ADSL), FosB Proto-Oncogene, AP-1 Transcription Factor Subunit (FOSB), and Calcium Activated Nucleotidase 1 (CANT1), did not interact with other targets (**Figure 3A**). Tumor Protein P53 (TP53) and MDM2 Proto-Oncogene (MDM2), Cyclin Dependent kinase 2 (CDK2) and Cyclin D1 (CCND1) (scores >0.70) were considered to have high confidence. We further explore the correlation between 42 SG-TMZ-GBMs and glioblastoma by GO (**Figure 3B**), KEGG (**Figure 3C**), and GSEA (**Figure 3D**) enrichment analyses. We discovered that 54 significant GO enrichment results, such as "lactation," "response to iron ion," "apoptotic mitochondrial changes," CCL5 was enriched in "human cytomegalovirus infection," "toll-like receptor signaling pathway" and "epithelial cell signaling in *helicobacter pylori* infection," Vascular Endothelial Growth Factor A (VEGFA), Rac Family Small GTPase 1 (RAC1), Protein kinase C Beta (PRKCB), and AKT Serine/Threonine kinase 1 (AKT1) were enriched in Vascular Endothelial Growth Factor (VEGF) signaling pathway (**Figure 3B**); pathway analysis revealed that SG-TMZ-GBMs were associated with cancer-related pathway, including glioma, non-small cell lung cancer, pancreatic cancer, and thyroid cancer, AKT1, BCL2 Associated X, Apoptosis Regulator (BAX), CCND1, E2F Transcription Factor 1 (E2F1), MDM2, PRKCB, and TP53 were enriched in "glioma" pathway, suggesting Luo Han Guo may play a role in cancer treatment; PRKCB, RELA Proto-Oncogene, NF-KB Subunit (RELA), STAT3, and VEGFA were enriched in "AGE-RAGE signaling pathway" and "HIF-1 signaling pathway" might be related in the inflammation-related diseases (**Figure 3C**). The GSEA KEGG enrichment analysis is shown in **Figure 3D**, and we found the top three significantly activated KEGG pathways were "KEGG hematopoietic cell lineage", "KEGG leishmania infection", and "KEGG nod like receptor signaling pathway". A compound-target-pathway network was established based on the target recognition and pathway analysis, with nodes mapping compounds, targets, or pathways, and indicated interactions by Cytoscape (**Figure 3E**).

The Determination of the Key SG-TMZ-GBMs

Uni-cox analysis revealed that 9 SG-TMZ-GBMs were determined as the significant survival-related risk genes. CCL5 was the most significant gene (p-value = 0.008) (**Figure 4A**). We found CCL5 was the key SG-TMZ-GBM (importance = 1) by random forest calculation (**Figures 4B,C**) and the survival analysis with p-value = 0.004 (**Figure 4D**). To further explore the effect of CCL5 on tshe GBM prognosis, a scatter plot of CCL5 expression and survival time in GBM patients was created (**Figures 4E,F**). Uni-cox and multi-cox regression analysis revealed that radiation (p-value < 0.001) and risk score (p-value = 0.008) were independent risk factors for overall survival analysis (**Figures 4G,H**). ROC c-index analysis illustrated that risk score + age + radiation, age + radiation, and radiation were the top three (**Figure 4I**).

INFLAMMATORY MICROENVIRONMENT AND MUTATION ANALYSIS

The microenvironment cell population-counter method evaluated the association between CCL5 and 10 immune cell populations from transcriptomic data. A strong correlation between CCL5 and CD8 T cells, T cells, B lineage, and fibroblasts were seen (**Figure 5A**). Then we further found the significant correlation (p-value < 0.05) between CCL5 and 9 of 24 immune cells markers (**Figure 5B**), such as the positive correlation in eosinophils (**Figure 5C**) and the negative correlation in Tcm (**Figure 5D**). In addition, exploring somatic mutations is helpful to understand the occurrence and development of GBM. The lollipop map shows the mutation distribution and protein domain of CCL5 with somatic mutation (**Figure 5E**). The distribution of the mutation spectrum of GBM samples can also be identified by a rainfall map (**Figure 5F**). The transition plot classified single nuclear variants into six categories (**Figure 5G**). Among them, the C > T mutation accounted for more than 50% of the total mutations. Furthermore, CCL5 protein expression can be detected by immunohistochemistry from the HPA database (**Figure 5H**).

DISCUSSION

GBM is the most frequent and the least treatable type of brain tumor, and the prognosis of GBM remains a challenge using the standard methods of treatment—TMZ, radiation, and surgical resection (Jiang et al., 2020b). TMZ is a novel methylating agent that demonstrated activity against recurrent GBM and is ineffective due to drug resistance (Wu et al., 2021). TCMs were considered anti-GBM auxiliary drugs, such as *Solanum nigrum L.* (Li et al., 2021), *Panax ginseng*, licorice, *Lycium*

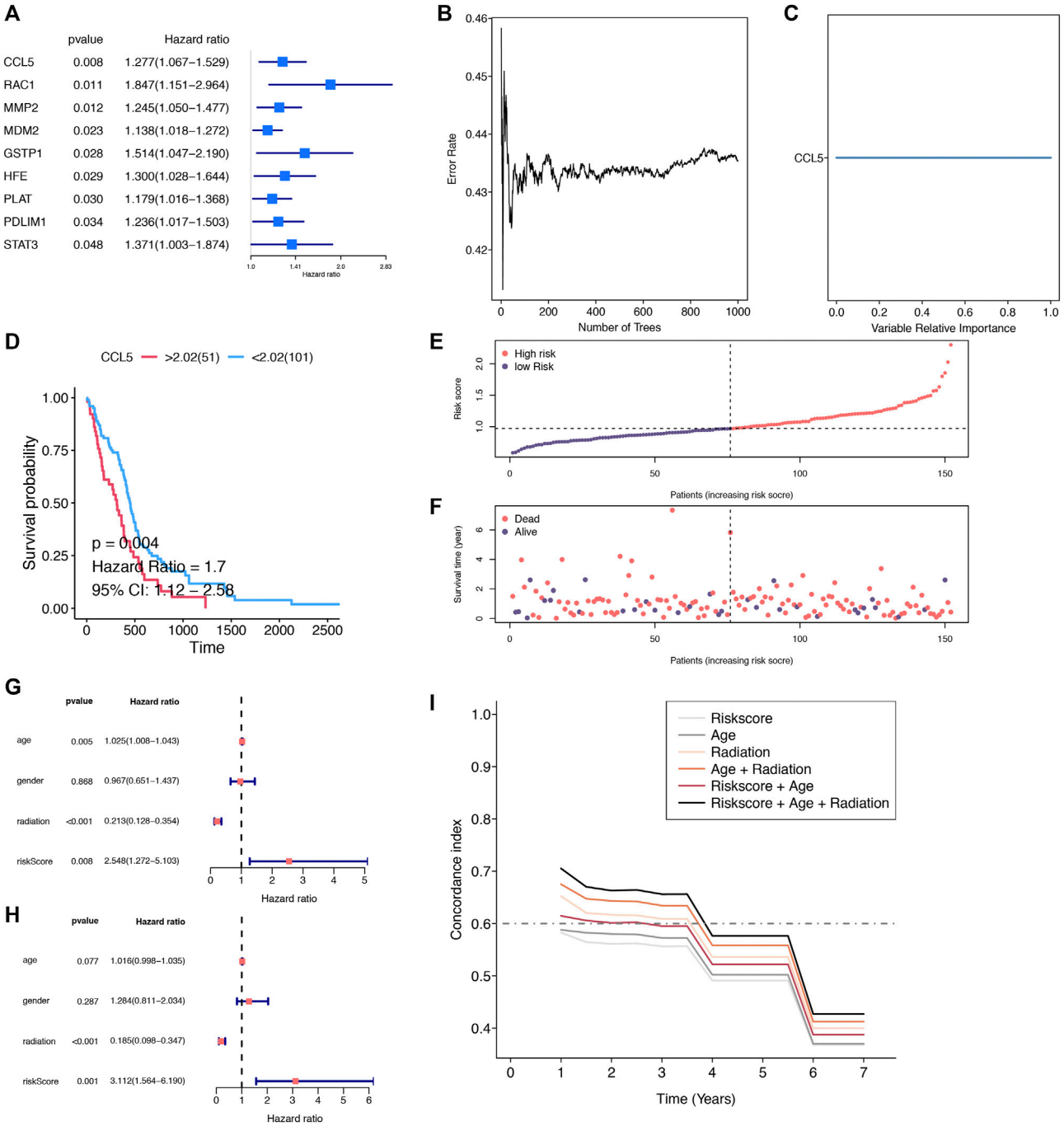

FIGURE 4 | The determination of the key SG-TMZ-GBMs. **(A)** Uni-cox analysis; **(B)** error tree of randomForest; **(C)** variable relative importance of randomForest; **(D)** survival analysis; **(E)** the curve of risk score; **(F)** survival status by ggrisk; **(G)** uni-cox with clinical information; **(H)** multi-cox with clinical information; and **(I)** ROC-concordance index.

barbarum, *Salvia miltiorrhiza bunge*, *Coptis rhizoma*, and *Sophora flavescens* (Wang et al., 2019). TCM is a helpful complementary and alternative medicine, however, there are few studies on TCM for GBM (Wang et al., 2019). The anti-GBM effects of TCM extract provided the new medium for the treatment of GBM (Li et al., 2021).

We tried to find a new TCM complementary method to treat GBM and hope that through combining Chinese and Western medicine, TMZ resistance could be reversed and anti-tumor therapeutic effects could be achieved. Luo Han Guo is a TCM with the same medicine and food. The multiple compounds in Luo Han Guo not only act on the same target protein, but a single

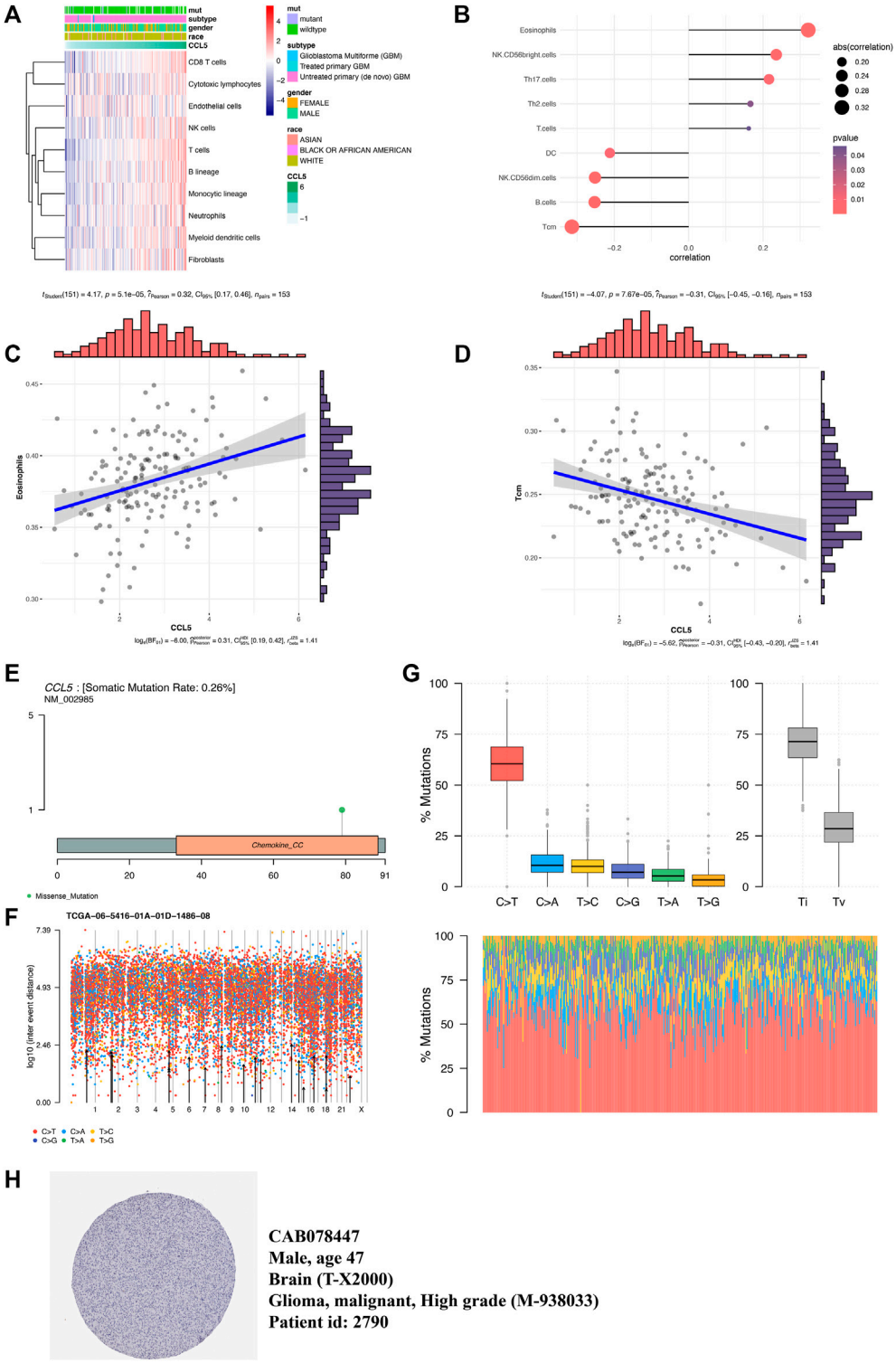

FIGURE 5 | Inflammatory microenvironment and mutation analysis. **(A)** Association between CCL5 expression and 10 immune cell populations in GBM. **(B)** The association between CCL5 expression and 24 immune cell markers in GBM. **(C)** A scatter plot of the positive correlation between CCL5 expression and eosinophils. **(D)** A scatter plot of the negative correlation between CCL5 expression and Tcm. **(E)** The lollipop map shows the mutation distribution and protein domain of CCL5 with somatic mutation. **(F)** The rainfall map of TCGA-AC-A23H-01A-11D-A159-09 in the GBM sample. **(G)** The transition and crosscut graphs show the distribution of SNV in GBM with six transition and crosscut events. The stacked bar graph **(bottom)** shows the mutation spectrum distribution of each sample. **(H)** CCL5 protein expression can be detected by immunohistochemistry from the HPA database.

compound also acts on various target proteins and multiple pathways, which reflects the "multiple components, multiple targets, and multiple pathways" of Luo Han Guo's synergistic effect. Luo Han Guo may work with ar-turmerone, methyl palmitate, lauric acid, beta-sitosterol, gamma-aminobutyric acid, coumarin, mogroside V, and squalene. GO functional enrichment analyses reflected that most of the active ingredients in SG might target nerve cells.

Through network pharmacology and bioinformatics analysis, we found that the CCL5 molecule is a potential target of SG, TMZ, and GBM, maybe the key to the clinical development of TMZ resistance (**Figure 1**). CCL5-CCR5 paracrine signaling could be an effective therapeutic strategy to improve chemotherapeutic efficacy against GBM (Zhang et al., 2021). CCL5 of glioma-associated microglia/macrophages regulates glioma migration and invasion via calcium-dependent matrix metalloproteinase 2 (Yu-Ju Wu et al., 2020). Knockdown or pharmacological inhibition of CCL5 increased the sensitivity of GBM cells treated with pericyte conditioned media to TMZ (Sprowls and Lathia, 2021). CCL5 was significantly highly expressed in GBM with poor prognostic. Uni-cox and randomForest were used to determine that CCL5 was a significantly important biomarker in GBM. CCL5 was also the target for SG and TMZ. The active ingredient of Luo Han Guo — squalene and CCL5 —show high binding efficiency. SG may be used as a new complementary therapy of the same medicine and food, acting on the target CCL5 and playing an anti-glioblastoma effect. Increasing the effective content of squalene in SG also needs further research. The radiation-related factors were the most critical in ROC c-index analysis. CCL5 plays a vital role in maintaining chemotherapy and radiation resistance.

Compared to genetically distinct syngeneic GBM models, the difference in mouse GBM models was eosinophils, reported in GBM (Khalsa and Shah, 2021). Eosinophils were associated with prognostic risk in the GBM microenvironment (Liang et al., 2020). We found that the SG-TMZ-GBM target, CCL5, a chemotactic ligand, is enriched and positively correlated in eosinophils. CCL5 is also the target of Luo Han Guo, and its effective active integrate compound – squalene—might act on CCL5, thereby affecting the immune microenvironment of GBM.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/Supplementary Material. Further inquiries can be directed to the corresponding author.

AUTHOR CONTRIBUTIONS

Data curation, L.J., J.L., X.Z., and Y.C.; formal analysis, L.J., J.L., D.B., and Y.C.; funding acquisition, L.J.; validation, L.J.; writing—original draft, L.J.; writing—review and editing, L.J., Y.C., and K.L.

FUNDING

This project was supported by the Talent Scientific Research Start-up Foundation of Yijishan Hospital, Wannan Medical College (grant no. YR202001); the Opening Foundation of Key Laboratory of Non-coding RNA Transformation Research of Anhui Higher Education Institution (grant no. RNA202004); the Key Projects of Natural Science Research of Universities in Anhui Province (grant no. KJ 2020A0622).

REFERENCES

Abdel-Hamid, M., Romeih, E., Huang, Z., Enomoto, T., Huang, L., and Li, L. (2020). Bioactive Properties of Probiotic Set-Yogurt Supplemented with Siraitia Grosvenorii Fruit Extract. *Food Chem.* 303, 125400. doi:10.1016/j.foodchem.2019.125400

Amberger, J. S., Bocchini, C. A., Schiettecatte, F., Scott, A. F., and Hamosh, A. (2015). OMIM.org: Online Mendelian Inheritance in Man (OMIM), an Online Catalog of Human Genes and Genetic Disorders. *Nucleic Acids Res.* 43, D789–D798. doi:10.1093/nar/gku1205

Barthel, F. P., Johnson, K. C., Johnson, K. C., Varn, F. S., Moskalik, A. D., Tanner, G., et al. (2019). Longitudinal Molecular Trajectories of Diffuse Glioma in Adults. *Nature* 576, 112–120. doi:10.1038/s41586-019-1775-1

Bindea, G., Mlecnik, B., Hackl, H., Charoentong, P., Tosolini, M., Kirilovsky, A., et al. (2009). ClueGO: a Cytoscape Plug-In to Decipher Functionally Grouped Gene Ontology and Pathway Annotation Networks. *Bioinformatics* 25, 1091–1093. doi:10.1093/bioinformatics/btp101

Bindea, G., Mlecnik, B., Tosolini, M., Kirilovsky, A., Waldner, M., Obenauf, A. C., et al. (2013). Spatiotemporal Dynamics of Intratumoral Immune Cells Reveal the Immune Landscape in Human Cancer. *Immunity* 39, 782–795. doi:10.1016/j.immuni.2013.10.003

Dai, S. X., Li, W. X., Han, F. F., Guo, Y. C., Zheng, J. J., Liu, J. Q., et al. (2016). In Silico identification of Anti-cancer Compounds and Plants from Traditional Chinese Medicine Database. *Sci. Rep.* 6, 25462–25511. doi:10.1038/srep25462

Fan, Y., Xue, W., Schachner, M., and Zhao, W. (2019). Honokiol Eliminates Glioma/glioblastoma Stem Cell-like Cells via JAK-STAT3 Signaling and Inhibits Tumor Progression by Targeting Epidermal Growth Factor Receptor. *Cancers (Basel)* 11, 22. doi:10.3390/cancers11010022

Gfeller, D., Grosdidier, A., Wirth, M., Daina, A., Michielin, O., and Zoete, V. (2014). SwissTargetPrediction: a Web Server for Target Prediction of Bioactive Small Molecules. *Nucleic Acids Res.* 42, W32–W38. doi:10.1093/nar/gku293

Guan, J., Lin, L., and Ouyang, M. (2019). Zhengyuan Capsule Alleviates Chemotherapy-Related Fatigue in Nude Mice with Human Lung Adenocarcinoma A549 Xenografts. *Cancer* 1, 1–7.

Hopkins, A. L. (2008). Network Pharmacology: the Next Paradigm in Drug Discovery. *Nat. Chem. Biol.* 4, 682–690. doi:10.1038/nchembio.118

Huang, L., Xie, D., Yu, Y., Liu, H., Shi, Y., Shi, T., et al. (2018). TCMID 2.0: a Comprehensive Resource for TCM. *Nucleic Acids Res.* 46, D1117–D1120. doi:10.1093/nar/gkx1028

Jiang, L., Zhong, M., Chen, T., Zhu, X., Yang, H., and Lv, K. (2020). Gene Regulation Network Analysis Reveals Core Genes Associated with Survival in Glioblastoma Multiforme. *J. Cel. Mol. Med.* 24, 10075–10087. doi:10.1111/jcmm.15615

Jiang, L., Zhu, X., Yang, H., Chen, T., and Lv, K. (2020). Bioinformatics Analysis Discovers Microtubular Tubulin Beta 6 Class V (TUBB6) as a Potential Therapeutic Target in Glioblastoma. *Front. Genet.* 11, 566579. doi:10.3389/fgene.2020.566579

Ju, P., Ding, W., Chen, J., Cheng, Y., Yang, B., Huang, L., et al. (2020). The Protective Effects of Mogroside V and its Metabolite 11-Oxo-Mogrol of

Intestinal Microbiota against MK801-Induced Neuronal Damages. *Psychopharmacology* 237, 1011–1026. doi:10.1007/s00213-019-05431-9

Khalsa, J. K., and Shah, K. (2021). Immune Profiling of Syngeneic Murine and Patient GBMs for Effective Translation of Immunotherapies. *Cells* 10, 491. doi:10.3390/cells10030491

Khan, M. A., and Tania, M. (2020). Cordycepin in Anticancer Research: Molecular Mechanism of Therapeutic Effects. *Cmc* 27, 983–996. doi:10.2174/0929867325666181001105749

Kim, S., Chen, J., Cheng, T., Gindulyte, A., He, J., He, S., et al. (2019). PubChem 2019 Update: Improved Access to Chemical Data. *Nucleic Acids Res.* 47, D1102–D1109. doi:10.1093/nar/gky1033

Kumar, V., Radin, D., and Leonardi, D. (2019). Studies Examining the Synergy between Dihydrotanshinone and Temozolomide against MGMT+ Glioblastoma Cells *In Vitro*: Predicting Interactions with the Blood-Brain Barrier. *Biomed. Pharmacother.* 109, 386–390. doi:10.1016/j.biopha.2018.10.069

Lee, H., Jeong, A. J., and Ye, S.-K. (2019). Highlighted STAT3 as a Potential Drug Target for Cancer Therapy. *BMB Rep.* 52, 415–423. doi:10.5483/bmbrep.2019.52.7.152

Li, J.-H., Li, S.-Y., Shen, M.-X., Qiu, R.-Z., Fan, H.-W., and Li, Y.-B. (2021). Anti-tumor Effects of Solanum nigrum L. Extraction on C6 High-Grade Glioma. *J. Ethnopharmacology* 274, 114034. doi:10.1016/j.jep.2021.114034

Liang, P., Chai, Y., Zhao, H., and Wang, G. (2020). Predictive Analyses of Prognostic-Related Immune Genes and Immune Infiltrates for Glioblastoma. *Diagnostics* 10, 177. doi:10.3390/diagnostics10030177

Liu, C., Dai, L., Liu, Y., Dou, D., Sun, Y., and Ma, L. (2018). Pharmacological Activities of Mogrosides. *Future Med. Chem.* 10, 845–850. doi:10.4155/fmc-2017-0255

Liu, C., Dai, L., Liu, Y., Rong, L., Dou, D., Sun, Y., et al. (2016). Antiproliferative Activity of Triterpene Glycoside Nutrient from Monk Fruit in Colorectal Cancer and Throat Cancer. *Nutrients* 8, 360. doi:10.3390/nu8060360

Liu, H., Tang, C., and Yang, Y. (2021). Identification of Nephrogenic Therapeutic Biomarkers of Wilms Tumor Using Machine Learning. *J. Oncol.* 2021. doi:10.1155/2021/6471169

Longato, E., Vettoretti, M., and Di Camillo, B. (2020). A Practical Perspective on the Concordance index for the Evaluation and Selection of Prognostic Time-To-Event Models. *J. Biomed. Inform.* 108, 103496. doi:10.1016/j.jbi.2020.103496

Mayakonda, A., Lin, D.-C., Assenov, Y., Plass, C., and Koeffler, H. P. (2018). Maftools: Efficient and Comprehensive Analysis of Somatic Variants in Cancer. *Genome Res.* 28, 1747–1756. doi:10.1101/gr.239244.118

Miller, A. M., Shah, R. H., Pentsova, E. I., Pourmaleki, M., Briggs, S., Distefano, N., et al. (2019). Tracking Tumour Evolution in Glioma through Liquid Biopsies of Cerebrospinal Fluid. *Nature* 565, 654–658. doi:10.1038/s41586-019-0882-3

Nie, J., Sui, L., Zhang, H., Zhang, H., Yan, K., Yang, X., et al. (2019). Mogroside V Protects Porcine Oocytes from *In Vitro* Ageing by Reducing Oxidative Stress through SIRT1 Upregulation. *Aging* 11, 8362–8373. doi:10.18632/aging.102324

Petitprez, F., Levy, S., Sun, C. M., Meylan, M., Linhard, C., Becht, E., et al. (2020). The Murine Microenvironment Cell Population Counter Method to Estimate Abundance of Tissue-Infiltrating Immune and Stromal Cell Populations in Murine Samples Using Gene Expression. *Genome Med.* 12, 86–15. doi:10.1186/s13073-020-00783-w

Reimand, J., Isserlin, R., Voisin, V., Kucera, M., Tannus-Lopes, C., Rostamianfar, A., et al. (2019). Pathway Enrichment Analysis and Visualization of Omics Data Using g:Profiler, GSEA, Cytoscape and EnrichmentMap. *Nat. Protoc.* 14, 482–517. doi:10.1038/s41596-018-0103-9

Ru, J., Li, P., Wang, J., Zhou, W., Li, B., Huang, C., et al. (2014). TCMSP: a Database of Systems Pharmacology for Drug Discovery from Herbal Medicines. *J. Cheminform.* 6, 13–16. doi:10.1186/1758-2946-6-13

Smyth, G. K. (2005). *Limma: Linear Models for Microarray Data, Bioinformatics and Computational Biology Solutions Using R and Bioconductor*. Springer, 397–420.

Sprowls, S. A., and Lathia, J. D. (2021). Neutralizing Shapeshifting Pericytes Enhances Glioblastoma Therapeutic Efficacy. *Cell Res*, 1–2. doi:10.1038/s41422-021-00538-1

Szklarczyk, D., Gable, A. L., Nastou, K. C., Lyon, D., Kirsch, R., Pyysalo, S., et al. (2021). The STRING Database in 2021: Customizable Protein-Protein Networks, and Functional Characterization of User-Uploaded Gene/measurement Sets. *Nucleic Acids Res.* 49, D605–D612. doi:10.1093/nar/gkaa1074

Wang, J., Qi, F., Wang, Z., Zhang, Z., Pan, N., Huai, L., et al. (2019). A Review of Traditional Chinese Medicine for Treatment of Glioblastoma. *Bst* 13, 476–487. doi:10.5582/bst.2019.01323

Wang, X., Shen, Y., Wang, S., Li, S., Zhang, W., Liu, X., et al. (2017). PharmMapper 2017 Update: a Web Server for Potential Drug Target Identification with a Comprehensive Target Pharmacophore Database. *Nucleic Acids Res.* 45, W356–W360. doi:10.1093/nar/gkx374

Wishart, D. S., Feunang, Y. D., Guo, A. C., Lo, E. J., Marcu, A., Grant, J. R., et al. (2018). DrugBank 5.0: a Major Update to the DrugBank Database for 2018. *Nucleic Acids Res.* 46, D1074–D1082. doi:10.1093/nar/gkx1037

Wu, W., Klockow, J. L., Zhang, M., Lafortune, F., Chang, E., Jin, L., et al. (2021). Glioblastoma Multiforme (GBM): An Overview of Current Therapies and Mechanisms of Resistance. *Pharmacol. Res.* 171, 105780. doi:10.1016/j.phrs.2021.105780

Xia, M., Han, X., He, H., Yu, R., Zhen, G., Jia, X., et al. (2018). Improved De Novo Genome Assembly and Analysis of the Chinese Cucurbit Siraitia Grosvenorii, Also Known as Monk Fruit or Luo-han-guo. *GigaScience* 7, giy067. doi:10.1093/gigascience/giy067

Xia, X., Zhong, Z., Xiao, Y., and Liang, L. (2013). Protective Effect of Mogroside against H_2O_2 Induced Apoptosis in PC12 Cell. *Chin. J. Hosp. Pharm.* 33, 786–789.

Yin, J., Zeng, A., Zhang, Z., Shi, Z., Yan, W., and You, Y. (2019). Exosomal Transfer of miR-1238 Contributes to Temozolomide-Resistance in Glioblastoma. *EBioMedicine* 42, 238–251. doi:10.1016/j.ebiom.2019.03.016

Yu-Ju Wu, C., Chen, C.-H., Lin, C.-Y., Feng, L.-Y., Lin, Y.-C., Wei, K.-C., et al. (2020). CCL5 of Glioma-Associated Microglia/macrophages Regulates Glioma Migration and Invasion via Calcium-dependent Matrix Metalloproteinase 2. *Neuro Oncol.* 22, 253–266. doi:10.1093/neuonc/noz189

Zanders, E. D., Svensson, F., and Bailey, D. S. (2019). Therapy for Glioblastoma: Is it Working? *Drug Discov. Today* 24, 1193–1201. doi:10.1016/j.drudis.2019.03.008

Zhang, X.-N., Yang, K.-D., Chen, C., He, Z.-C., Wang, Q.-H., Feng, H., et al. (2021). Pericytes Augment Glioblastoma Cell Resistance to Temozolomide through CCL5-CCR5 Paracrine Signaling. *Cel Res*, 1–16. doi:10.1038/s41422-021-00528-3

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

GLOSSARY

ADSL adenylosuccinate lyase
AKT1 AKT serine/threonine kinase 1
AMPK AMP-activated protein kinase;
BAX BCL2 associated X, apoptosis regulator
BBB blood-brain barrier
c-index concordance index
CCL5 C-C motif chemokine ligand 5
CANT1 calcium activated nucleotidase 1
CDK2 CDKCyclin dependent kinase 2
CCND1 cyclin D1
DEGs differentially expressed genes
DL drug-like
DE-GBMs differentially expressed GBMs
EMT emergency medical technician
E2F1 E2F transcription factor 1
FOSB FosB proto-oncogene, AP-1 transcription factor subunit
GBM glioblastoma
GO gene ontology
HR hazard ratios
KEGG kyoto encyclopedia of genes and Genomes
logFC logFoldChange
MDM2 MDM2 proto-oncogene
multi-cox multivariate cox
OB oral bioavailability
PPI protein-protein interaction
POT1 protection of telomeres 1
PRKCB protein kinase C beta
RELA RELA proto-oncogene, NF-KB subunit
RAC1 Rac family small GTPase 1
ROC receiver operating characteristic curve
SG *Siraitia grosvenorii*
SG-TMZ-GBM *Siraitia grosvenorii* - temozolomide – glioblastoma
STAT3 signal transducer and activator of transcription 3
SIRT1 sirtuin 1
TP53 tumor protein P53
TCM traditional chinese medicine
TMZ temozolomide; uni-cox, univariable Cox
VEGFA vascular endothelial growth factor A
VEGF vascular endothelial growth factor

frontiers in Genetics

ORIGINAL RESEARCH
published: 04 January 2022
doi: 10.3389/fgene.2021.803627

EasyMicroPlot: An Efficient and Convenient R Package in Microbiome Downstream Analysis and Visualization for Clinical Study

Bingdong Liu[1,2], *Liujing Huang*[2,3], *Zhihong Liu*[2], *Xiaohan Pan*[4], *Zongbing Cui*[2], *Jiyang Pan*[1]* *and Liwei Xie*[2,3,5]*

[1]The First Affiliated Hospital of Jinan University, Guangzhou, China, [2]State Key Laboratory of Applied Microbiology Southern China, Guangdong Provincial Key Laboratory of Microbial Culture Collection and Application, Guangdong Open Laboratory of Applied Microbiology, Institute of Microbiology, Guangdong Academy of Sciences, Guangzhou, China, [3]Zhujiang Hospital, Southern Medical University, Guangzhou, China, [4]Department of Applied Biology and Chemical Technology, The Hong Kong Polytechnic University, Kowloon, Hong Kong SAR, China, [5]School of Public Health, Xinxiang Medical University, Xinxiang, China

Advances in next-generation sequencing (NGS) have revolutionized microbial studies in many fields, especially in clinical investigation. As the second human genome, microbiota has been recognized as a new approach and perspective to understand the biological and pathologic basis of various diseases. However, massive amounts of sequencing data remain a huge challenge to researchers, especially those who are unfamiliar with microbial data analysis. The mathematic algorithm and approaches introduced from another scientific field will bring a bewildering array of computational tools and acquire higher quality of script experience. Moreover, a large cohort research together with extensive meta-data including age, body mass index (BMI), gender, medical results, and others related to subjects also aggravate this situation. Thus, it is necessary to develop an efficient and convenient software for clinical microbiome data analysis. EasyMicroPlot (EMP) package aims to provide an easy-to-use microbial analysis tool based on R platform that accomplishes the core tasks of metagenomic downstream analysis, specially designed by incorporation of popular microbial analysis and visualization used in clinical microbial studies. To illustrate how EMP works, 694 bio-samples from Guangdong Gut Microbiome Project (GGMP) were selected and analyzed with EMP package. Our analysis demonstrated the influence of dietary style on gut microbiota and proved EMP package's powerful ability and excellent convenience to address problems for this field.

Keywords: 16s rDNA sequencing, next-generation sequencing, microbiota, script, clinical data

OPEN ACCESS

Edited by:
Meng Zhou,
Wenzhou Medical University, China

Reviewed by:
Qixiao Zhai,
Jiangnan University, China
Zhang Wang,
South China Normal University, China

****Correspondence:***
Jiyang Pan
jiypan@vip.163.com
Liwei Xie
xielw@gdim.cn

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

Received: *28 October 2021*
Accepted: *02 December 2021*
Published: *04 January 2022*

Citation:
Liu B, Huang L, Liu Z, Pan X, Cui Z, Pan J and Xie L (2022) EasyMicroPlot: An Efficient and Convenient R Package in Microbiome Downstream Analysis and Visualization for Clinical Study. Front. Genet. 12:803627. doi: 10.3389/fgene.2021.803627

INTRODUCTION

The in-depth understanding of human microbiome has dramatically reshaped our understanding of the relationship between human health and microbiome (Marchesi et al., 2016; Fan and Pedersen, 2021). A tremendous number of studies have demonstrated that microbiomes residing in the human body are key contributors in modulating host physiology and metabolism (Van Treuren and Dodd, 2020). As the second genome of the human being, the microbiomes are thought to be responsible for

the complex pathophysiology nature of various diseases, e.g., neurological, metabolic, and immunity disorders (Oleskin and Shenderov, 2016; Cryan et al., 2019). Undeniably, the revolution in DNA sequencing technologies has enabled us to generate massive amounts of microbial data and accelerate the progression of studies and researches to explore the relationship between microbiomes and human health. Thus, a growing number of hospitals and medical centers endeavored largely to recruit volunteers and collect bio-samples associated with microbiomes (Claesson et al., 2017). For example, the Human Microbiome Project (HMP) in 2007 expanded our understanding of the microbiome across different body sites of a healthy person and its physiological roles in human genetic and metabolic landscapes (Peterson et al., 2009). Furthermore, emerging evidence indicate that microbiomes could be used as a non-invasive approach serving as novel diagnostic biomarkers and therapeutic targets. For example, 30 bacterial taxa identified from a cohort study could distinguish patients with early hepatocellular carcinoma with area under the curve (AUC) of 80.64% (Ren et al., 2018), and *Bacteroides vulgatus* may alter bile acid metabolism to improve the risk of polycystic ovary syndrome (Qi et al., 2019). In this regard, there is an urgent necessity to integrate microbial data into clinical practice for evidence-based medicine.

With the advancement of next-generation sequencing (NGS) and bioinformatics in basic and clinical biomedicine investigation, mathematics and statistical approaches in microbial downstream analysis are able to provide us comprehensive information of the relationship between microbiomes and human health and diseases (Knight et al., 2018). For example, diversity metric was introduced from ecology to access microbiota richness (Faith, 1992), while machine learning technology was popularly used for bacterial biomarkers screening (Vangay et al., 2019). In order to perform such measurements, clinical researchers usually have to take additional bioinformatics courses, which significantly obstruct the progression and frustrate amateurs without computational and coding experience (Knight et al., 2018). Here are three aspects of problems that clinical investigators face if they want to perform microbiome-related studies: First, clinical meta-data generally consist of a wide range of information including but not limited to age, body mass index (BMI), gender, and medical diagnostics, which brings about giant challenges for researchers to estimate and select proper features to determine inclusion criteria (He et al., 2018b). Moreover, in many retrospective studies, due to the complexity of subjects in hospitals, clinicians are not able to clearly determine grouping information based on meta-data, which challenges clinical researchers, especially various missing value in meta-data. Second, a large scale of microbial data always contains various information bias. For example, low abundance and occurrence taxa are often observed in microbial data analysis, which may be due to experimental contamination, sequence alignment error, and other factors. Normally, these taxa are filtered in downstream analysis according to study design and researchers' experience due to the lack of a well-recognized protocol, which may lead to biased and poorly reproducible results. Particularly, due to poor coding abilities, clinical researchers may find unexpected difficulties without enough knowledge in the data filtering step. Third, although many existing software (Caporaso et al., 2010) and R packages (Liu et al., 2021; Zhao et al., 2021) have been developed and integrated multiple methods from various fields, none of them are specially designed for clinical studies and could not address problems such as missing data, data filtering, and sample regrouping easily and efficiently. Moreover, due to large and comprehensive function and workflow, clinical researchers may spend additional time to learn and modify clinical data. The manual step to select the most appropriate parameters is still puzzling and tedious, and inconsistent application of such tools may reduce the reproducibility of the results. Thus, an efficient and convenient tool to meet the fast-developed clinical microbial studies is necessary.

Here, EasyMicroPlot (EMP) incorporates packages used in basic and clinical microbial studies for data analysis and visualization. In this package, regular downstream analysis covering core tasks of metagenomic analysis could be performed efficiently and conveniently in this field.

MATERIALS AND METHODS

Package Description

EMP is developed based on R language 3.6 version and contains three main modules, which include EMP_META, EMP_MICRO, and EMP_COR. Compared to existing microbial analysis software in this filed, EMP extremely simplifies the whole process to the best and focuses on core microbiota and meta-data analysis in clinical studies. Each function in the EMP package is standalone and flexible, which enables users to design their own pipeline and utilize necessary functions without tedious parameterization and scripts. The overall design and workflow of EMP package is illustrated in **Figure 1**.

EMP_META module includes two functional units: the meta_summary and meta_regroup. The meta_summary function could enable users to easily visualize the distribution of missing value in meta-data, summarize basic information, and generate bivariate tables. The other function, meta_regroup, is designed to utilize various cluster analyses and 26 evaluation algorithms to determine the best regroup strategy based on different kinds of clinical information containing categorical and continuous variables.

EMP_MICRO module consists of data_filter, beta_plot, cooc_plot, structure_plot, tax_plot, RFCV, and RFCV_roc functions and mainly aims to provide investigators a fast and simple approach to accomplish the core tasks of data filter, such as α-diversity analysis, β-diversity analysis, co-occurrence network analysis, taxonomic stack bar plot, and random forest models for key taxa screening. The function EMP_MICRO could automatically identify data directly from R workspace and transform these into core microbial data at six levels (phylum, class, order, family, genus, and species). The feature allows users to activate a complete workflow with default parameters and generate results in workspace by applying function EMP_MICRO with only microbial abundance files and mapping file in user's R

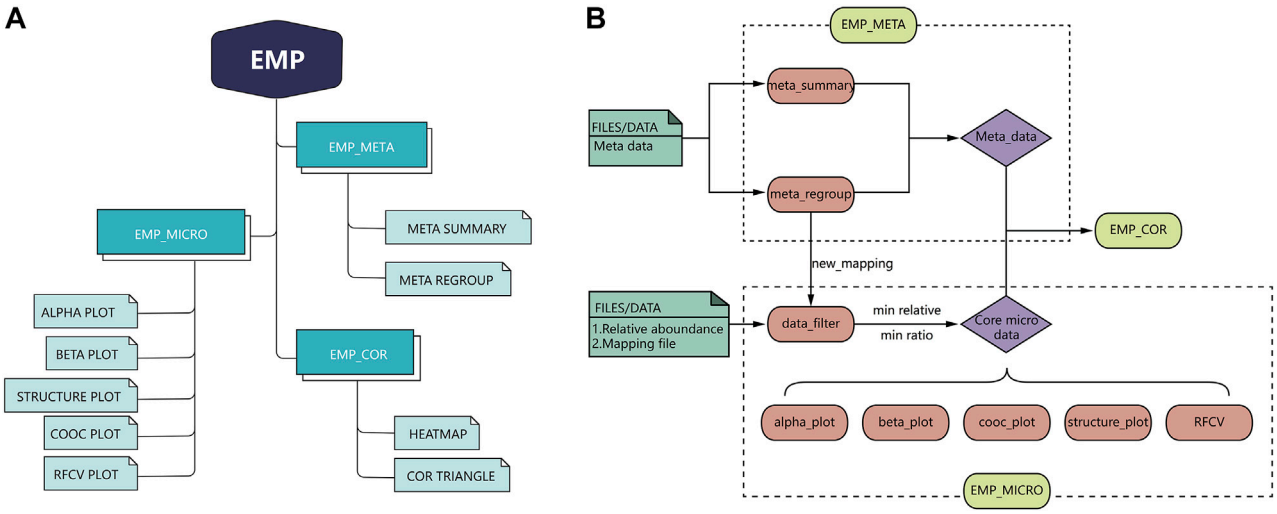

FIGURE 1 | Overall design and workflow of EasyMicroPlot package.

workspace. Most analysis functions including α-diversity, β-diversity, and taxonomy boxplot not only provide Student *t*-test and one-way analysis of variance (ANOVA) comparison methods but also offer an interactive plot in html format, which means users could easily identify outliers and recognize abnormal samples. Moreover, most of the existing microbial analysis software suggest investigators to provide well-matched microbial abundance files and mapping files. In this case, investigators have to modify all the files if they want to perform sub-group analysis or regroup analysis. To avoid such problem, EMP is designed in a way that users only need to edit their mapping file without modifying microbial abundance data.

EMP_COR module is designed to integrate metagenomic and clinical data. Investigators can explore links between meta-data and microbial abundance using Pearson index and Spearman and Kendall index, and we have developed two artistic styles for data visualization in this module.

Data Preparation

To test our package, we selected a part of 16S rDNA sequencing data from Guangdong Gut Microbiome Project (GGMP) (He et al., 2018b). This dataset is composed of samples from a population in Shenzhen, China, of GGMP. A total of 618 16S rDNA sequencing data with meta-data including diets, districts, defecation, and metabolic syndrome (MetS) status in Shenzhen province was included in this analysis. Microbial relative abundance was generated at phylum, class, order, family, genus, and species levels using a standard QIIME 1.91 pipeline. All meta-data and microbial abundance were deposited in the **Supplementary material**.

RESULTS

Subjects Enrollment

After data preparation, the function meta_summary in EMP_META could map the distribution of missing data and generate a general summary of meta-data based on MetS status (**Figure 2A** and **Supplementary material**). There are more than 20 missing information in features of "salt," "plant oil," "soy sauce," and "sugar" intake. A three-line table also showed detailed dietary structure information among groups (**Supplementary Table S1**). Consider that gastrointestinal disorder, antibiotic therapy, and probiotics are closely linked to the dysbiosis of gut microbiota. Finally, only 394 samples were qualified and included into downstream analysis, and those who have experience of diarrhea, astriction, antibiotics, and synbiotics were excluded. In order to explore the microbial difference without bias of dietary pattern, the function of meta_regroup incorporated 26 indexes to estimate the cluster for dietary structure to determine the best regrouping design utilizing "Kmeans" and "Euclidean" parameter (**Figure 2B**). After calculation for continuous and categorical variables, 394 samples were included into downstream analysis and divided into four groups based on dietary structure and MetS status (Control_1: subjects without Mets whose dietary structure belong to type 1; Control_2: subjects without Mets whose dietary structure belong to type 2; Cases_1: subjects with MetS whose dietary structure belong to type 1; Cases_2: subjects with MetS whose dietary structure belong to type 2). Those who have experience of diarrhea, astriction, antibiotics, and synbiotics were excluded.

Diets Are Associated With Significant Structural Changes of Gut Microbiota

To avoid interference of rare taxa, species data whose relative abundance was below 1‰ or prevalence rate was not more than 70% in any group was excluded using function of data_filter. Function structure_plot provided a general composition picture for these core data at species level (**Figure 2F** and **Supplementary**

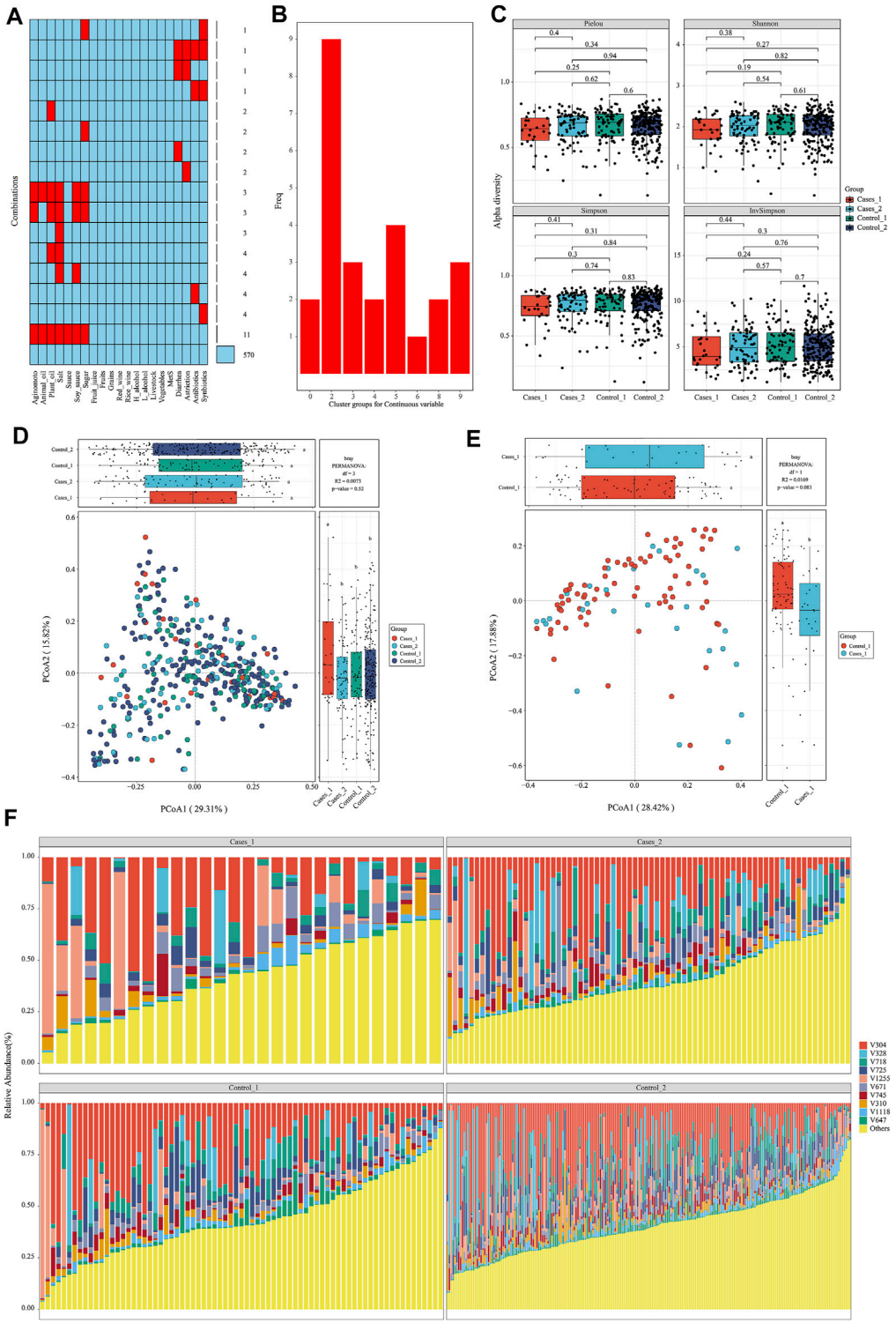

FIGURE 2 | Diets are associated with significantly structural changes of gut microbiota. **(A)** The distribution of missing value in the meta data. **(B)** Twenty-six estimate indices vote for the best cluster number based on dietary structure. **(C)** α-Diversity on Pielou, Shannon, Simpson, and InvSimpson index among different subgroups. **(D, E)** β-Diversity on Bray–Curtis index and permutational MANOVA test among different subgroups with consideration of dietary structure. **(F)** The structure plot for top 10 gut bacterial taxa.

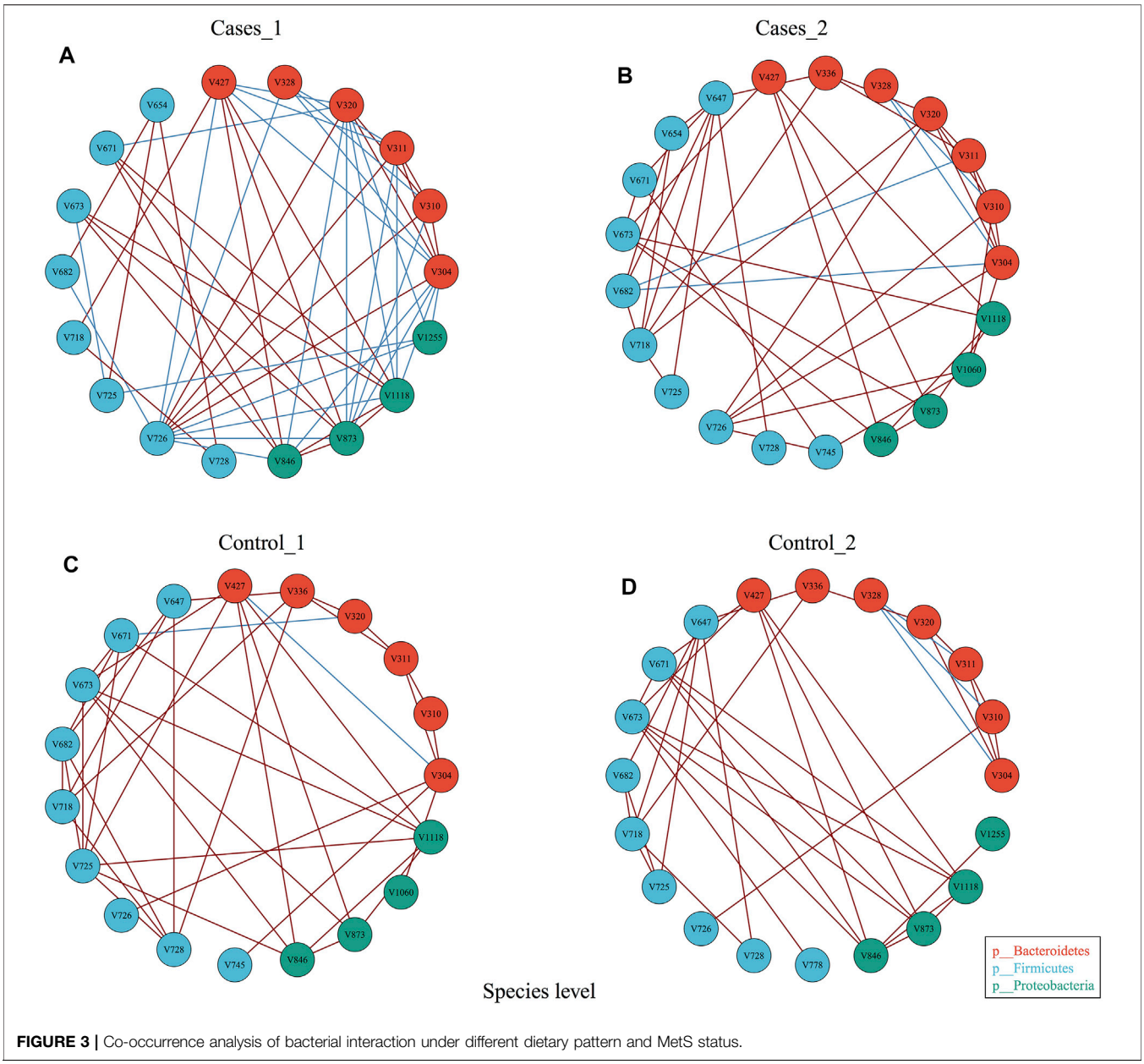

FIGURE 3 | Co-occurrence analysis of bacterial interaction under different dietary pattern and MetS status.

Material). With this core microbiota in hand, rarefaction measurement of Pielou, Shannon, Simpson, and InvSimpson index showed α-diversity difference was not significant with each other ($p > 0.05$) (**Figure 2C**). β-Diversity calculated with Bray–Curtis distance showed samples in Cases_1 group was far away from the other three groups in two-dimensional space (**Figure 2D**), which indicated these microbiota structures for MetS subjects with type A diet were significantly different from others (least significant difference $p < 0.05$). Particularly, when only two groups including Cases_1 and Control_2 were performed in PCoA analysis, permutational multivariate analysis of variance (MANOVA) test was almost statistically significant ($r^2 = 0.01$, $p = 0.083$) (**Figure 2E**). In contrast, we also performed β-diversity with the same parameter and could not observe significant change, which suggested diets indeed disturb the structure of the microbiota (**Supplementary Figure S1**).

Diets Perturb the Ecology and Network of Gut Microbiota

In order to explore whether diet may influence the gut microbiota community network, EMP provides an easy function to perform co-occurrence analysis and generate network plot for each group. Co-occurrence analysis at species level with parameter of Spearman confident index [abs(r) > 0.3, $p < 0.05$] showed each group has almost the same vertices but presented totally different cross-talk among core gut bacterial taxa (**Figure 3**). For example,

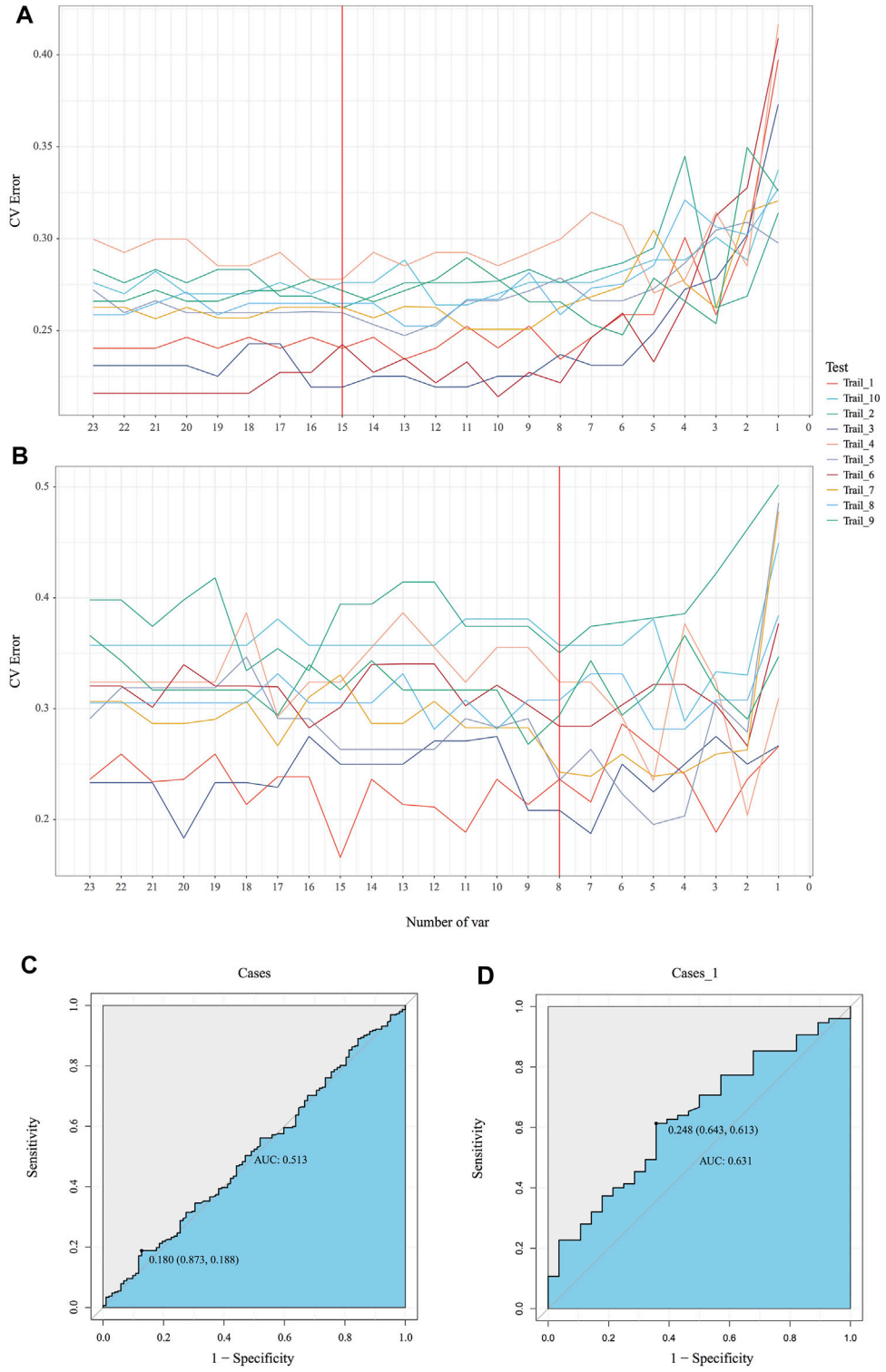

FIGURE 4 | Identification of the signature gut microbiota by random forest. **(A, B)** To explore the signature biomarkers, a fivefold cross validation together with random forest was performed. **(C, D)** Based on key bacterial taxa generated by EMP package, receiver operating characteristic curves (ROC) were performed to test prediction models.

Control_1 group has unique V745 (*Veillonellaceae Phascolarctobacterium*) and V1060 (*Alcaligenaceae Sutterella*) in its network, while Cases_1 has V328 (*Prevotella copri*) and V1225 (*Aeromonadales Succinivibrionaceae*) in its network. Another comparison demonstrated that other than the common species between Control_2 and Cases_2, Control_2 group has V1225 (*Aeromonadales Succinivibrionaceae*), while case_2 includes additional three species, that is, V654 (*Clostridiaceae Clostridium*), V745 (*Veillonellaceae Phascolarctobacterium*), and V1060 (*Alcaligenaceae Sutterella*). Particularly, Cases_1 group with high network complexity (transitivity = 0.6315789, centralization degree = 0.3006536, graph density = 0.3464052) was obviously higher than others (**Supplementary material**), which suggested different dietary structures may change the systemic ecology of gut bacteria.

Diets Significantly Interfere With the Accuracy of Random Forest Prediction for Patients With MetS

Emerging evidence proved microbiota could be characterized as markers for clinical auxiliary approach. As the most popular machine learning, random forest together with cross validation could robustly select key bacteria as biomarkers to build prediction model. In our MicroEasyPlot, RFCV function allows users to utilize relative abundance data to generate random forest prediction model and select potential maker taxa according to mean and standard deviation at a series of random number. With this, we constructed random forest model together with cross validation and explored microbial biomarkers to distinguish individuals with MetS from healthy ones (**Figure 4B**, **Supplementary Figure S2**, and **Supplementary material**). Fifteen bacterial taxa at species level were considered to be the most important biomarkers by a union of 10 random processes, while V647 (*Clostridia Clostridiales*), V671 (*Clostridiales Lachnospiraceae*), V725 (*Faecalibacterium prausnitzii*), V726 (*Ruminococcaceae Oscillospira*), and V718 (*Clostridiales Ruminococcaceae*) changed between groups significantly ($p = 0.0088$, 0.095, 0.018, 0.04, 0.088) (**Supplementary Figure S3**). RFCV_roc function also could be used to test this prediction model, through which we established receiver operating characteristic curve with AUC area 0.63 (**Figure 4D**). As a control, random forest model with the same parameters was performed to test the relative abundance data directly without subgroup analysis; AUC area only achieved 0.51, which indicated dietary style affected gut microbiota composition indeed and should be included into downstream microbial analysis in clinical studies (**Figures 4A, C**).

Identification of the Relationship Between Dietary Structure and Microbial Abundance

To explore the detailed relationship of diets and core microbiota, function cor_plot_heat and cor_plot_detail module provides two kinds of visualization using "pearson," "spearman," and "kendall" measurement. Correlation analysis showed 17 species of 23 core taxa generated from data_fiter function were strongly associated with dietary changes. Especially for several key taxa identified by random forest model below, V647 (*Clostridia Clostridiales*) was positively correlated with red wine intake ($r = 0.121$, $p = 0.017$), V726 (Ruminococcaceae *Oscillospira*) was positively correlated with sugar ($r = 0.136$, $p = 0.007$), V671 (*Clostridiales* Lachnospiraceae) was highly correlated with salt consumption ($r = -0.163$, $p = 0.001$), and V718 (*Clostridiales* Ruminococcaceae) was highly correlated with fruits ($r = -0.107$, $p = 0.034$) and plant oil ($r = -0.113$, $p = 0.025$) (**Figures 5A, B** and **Supplementary material**). On the other hand, vegetarian diet including fruits, vegetables, and fruit juice influenced nine core gut bacterial taxa, which was considered to be the most influencing factor. High- and low-degree alcohol affected three taxa core gut bacteria, while red and rice wine disturbed four taxa. Seasoning including salt, sugar, soy sauce, and plant and animal oil also presented close relationships with various gut bacteria.

DISCUSSION

Due to the advent of bioinformatics and high-throughput sequencing technology, bioinformatics has become a well-qualified tool in establishing auxiliary diagnostic measurement in clinical practice (Kordahi et al., 2021). Notably, microbiome has gained more attention in fields investigating the biological and pathological nature of various diseases (Claesson et al., 2017). However, clinical researchers often encountered several difficulties in data analysis and visualization. In order to fill the gap between clinical researchers and microbiome data mining, we collected 16S rDNA sequencing data set from GGMP and performed data analysis with EMP to present the convenience and professional practice of our tool. With this dataset, we proved dietary pattern is an important contributor to different gut microbiota patterns.

Notably, huge meta-data analysis with detailed participants' information usually brings tremendous problems. Employing inappropriate strategy to estimate the features including continuous and categorical variables may lead to unexpected bias and errors. For example, dietary pattern could dramatically change or in the long term reshape the composition of gut microbiota (Singh et al., 2017). However, in previous studies, researchers may ether ignore the effects of diets in downstream microbial analysis or divide subjects into different groups according to dietary classification such as Western diet, Mediterranean diet, Vegetarian diet, etc. (Bian et al., 2017; Garcia-Mantrana et al., 2018). In the present study, 394 qualified subjects were selected from 618 volunteers in Shenzhen, Guangdong province of China. However, in the process of 26 estimating votes under different algorithms, "Kmeans" successfully divided them into two groups based on "Euclidean" distance, which indicated those 394 subjects from the same districts had two different dietary structures. Among the regrouped subjects only based on MetS status, the present study observed more changes of microbial structure and diversity under different dietary status indeed. Regrouping based on diet also improved the robustness of random forest prediction and

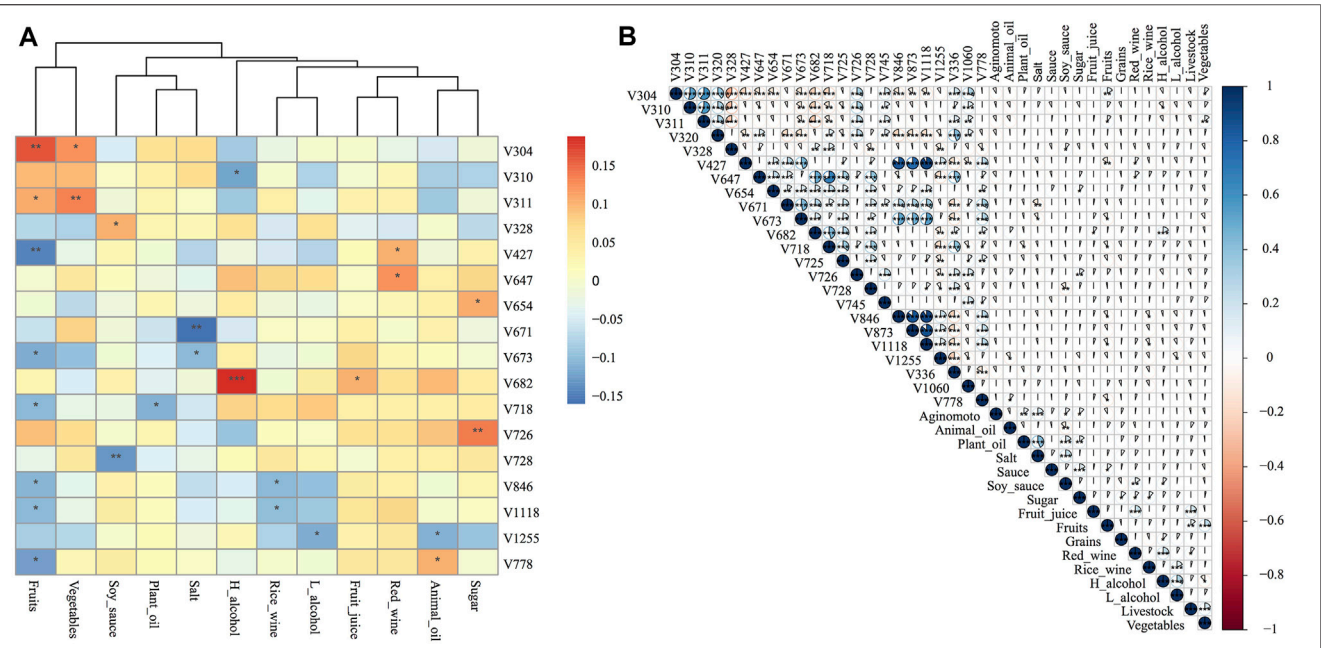

FIGURE 5 | Correlation analysis between relative abundance of core bacterial taxa and meta data in clinical study.

increased AUC area in receiver operating characteristic curves (ROC) model. Additionally, correlation analysis further confirmed dietary components including fruits, vegetables, alcohol, sugar, oil, and salt significantly alter the core bacterial taxa. Animal studies have revealed that additional salt supplement could significantly deplete genera Clostridia, which was consistent with our observation (Wilck et al., 2017). Furthermore, in two large cohort studies (1,879 middle-aged elderly Chinese adults from Guangzhou Nutrition and Health Study and 6,626 subjects from GGMP), dietary fruit and vegetables were also proved to reshape gut microbiota (Jiang et al., 2020). Altogether, these results confirmed the importance of regrouping based on diet, suggesting microbiome-related clinical studies should take the dietary factor into consideration for participants' regrouping. Moreover, factors including smoking, education, life style, and others could also exert a great influence on the structure and diversity of gut microbiota (He et al., 2018a). A big cohort study demonstrated prolonged sedentary lifestyle may increase the prevalence of MetS through modulation of gut microbiota (He et al., 2018a). Other studies also suggested various sports and exercise could reshape human microbiota. Of note, elite athletes harbor several special taxa in the gut, which were proven to be able to catalyze lactate into propionate to extend running time (Scheiman et al., 2019). Thus, multiple factors should be taken into consideration and estimate their influence carefully in microbiota-related studies.

Secondly, low abundance and prevalent bacterial taxa may affect reliability and reproducibility of microbial-related studies and analysis and were thus considered to be contaminants (Claesson et al., 2017). Researchers have reached a consensus that it was hard for bacteria with low abundance to exert significant effects on the host, and taxa with low prevalence were likely to lead to false positive and negative results in classification and prediction models (Knight et al., 2018). For example, in our previous study, random forest model based on species data observed several markers in mathematics to distinguish patients with insomnia from healthy control, while many taxa determined by classification model without decontamination were actually outliers and lack biological significance between groups (Liu et al., 2019). Another cohort study has also set a strict decontamination standard with >0.5% of relative abundance and >30% of prevalence in downstream analysis to avoid potential bias (Biagi et al., 2016). Thus, an appropriate threshold for data filter is extremely necessary. Though the concept of filtering microbial data is well accepted in microbial studies, there is no professional tool in this area. Data analysts always filter data by self-developed script, while others even modified data in excel format manually. EMP package provides a convenient function, data_filter, to address this problem. Bacteria could be excluded by two thresholds including minimum abundance and prevalence, which means users could easily customize the filter according to study design and generate core bacterial taxa in one step. Before the application of data filter function, a total of 1,503 species annotated from 394 feces samples were generated, and a handful of taxa only presented in few samples with low abundance. In terms of biologic aspect, these taxa were believed to originate from contamination and annotation error and may have an adverse effect on downstream computation. Utilizing data filter function in EMP package with 0.001 minimum relative abundance and 0.7 minimum prevalence threshold, only 23 core species were qualified for the following analysis, which dramatically economized the computational resources and reduced the bias and errors. Among these core

species, 17 taxa were proven to be highly correlated with diet, which further confirmed the value of data filter function. For the first time, clinical researchers could easily decontaminate microbial data sets and generate core bacteria for downstream analysis with a well-recognized process.

Third, solid and meaningful results were normally generated under standardized and scientific approaches in data analysis. Although tremendous tools and online platforms were developed in the past decade, clinical researchers without coding experience were not satisfied with the complicated instruction or limited functions. Particularly, certain tutorials in written books or online websites about microbial data analysis only offer sets of scripts containing the usage of many independent software, and following such tutorials is time consuming. Even worse, it is common to see codes shared by publishers containing kinds of errors without peer review, including but not limited to the inappropriate usage of certain software and tools. Additionally, most of the researchers did not provide detailed script pipeline, and editors merely require researchers to upload key codes and scripts in **Supplementary material** or open-source platform due to the complexity of script. Without confirmation of the correctness of the self-written codes, it is hard to realize the unexpected false positive and false negative conclusion, and this makes it impossible to reproduce the computational results. On the other hand, researchers also need an easy way to design their own pipeline to continue attempting and computing results many times. Given a lot of independent software were integrated into code text by hand, collaborators may find it different to read and use, which may largely increase the risk of error and bugs. Thus, after collecting and screening popular analysis strategy, EMP package divides the whole analyzing process into three modules, and each module could be utilized separately, which provides enormous convenience in research work. In the present study, EMP package helped to estimate missing data and classify 394 samples into four groups according to dietary structure. After receiving group information, Microplot module could simply analyze microbial data in one script covering α-diversity, β-diversity, co-occurrence, structure plot, and random forest models. At last, correlation analysis revealed the influence of dietary structure on gut microbiota.

There are three main advantages of the current EMP package: First of all, packages integrated into EMP package are well accepted by users in this field and documented on the Comprehensive R Archive Network (CRAN). All of these packages are widely utilized to perform microbial data analysis and visualization. Moreover, EMP package is an open-source tool, and users are welcome to report any bugs. Second, the existing tools and R packages made great effort to incorporate a wide range of microbial analysis approaches and statistics method, while EMP package focuses on clinical studies, and the whole process is divided into three parts for the core microbial data analysis. Given many retrospective studies cannot determine groups for samples, EMP provides scientific method to help clinical scientists screen and regroup samples. Besides, EMP package does not need well-matched relative abundance files and mapping file and could automatically identify bacterial level and perform data analysis according to mapping file containing samples identifiers and group information in text format or data frame generated from R script without modifying bacterial data, which may significantly reduce mistakes in many attempts. Third, in order to maximally simplify the operating procedure, EMP package allows users to perform the whole workflow with only one step and generate all results in the workspace. Each core analysis in workflow also could be performed by applying one function, which means researchers could design their own pipeline in a few lines of script with modules they are interested in for the study. In this case, EMP simplified clinical users' self-developed codes, allowing peer reviewers and readers to also test and reproduce specific results with few codes. Thus, with EMP package, clinical investigators could explore a huge scale of clinical data together with microbial abundance information and publish their result easily and reliably.

CONCLUSION

EMP package incorporates widely used microbial data analysis and visualization tools deposited in CRAN and provides clinical investigators with a convenient approach to perform downstream data filtering, analysis, and visualization. From the demo data, we demonstrated that researchers could simply utilize different modules to identify missing data, classify patients into different groups, and regroup them based on different parameters. Most importantly, this package could help clinicians robustly select key microbial biomarkers and calculate the correlation index between core microbiota and clinical parameters, such as BMI, age, and height, etc. Overall, EMP package provides an efficient and convenient downstream microbiome analysis pipeline, especially for clinical investigators without additional script experience.

DATA AVAILABILITY STATEMENT

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found below: https://github.com/xielab2017/EasyMicroPlot.

AUTHOR CONTRIBUTIONS

LX and JP designed the experiments and collected the grant support. BL, LH, ZL, and ZC constructed this package and performed the data analysis. XP supported the test work in code review session.

FUNDING

This work was supported by "GDAS" Project of Science and Technology Development (Grant No. 2021GDASYL-

20210102003 and 2018GDASCX-0102), Guangdong Basic and Applied Basic Research Foundation (Grant No. 2020B1515020046), and Natural Science Foundation of China (Grant No. 82072436, 81900797, and 81871036).

ACKNOWLEDGMENTS

We appreciate Zelong Zhao for supportive script and comments on β-diversity module. We thank Xiaochen Wang for picture design and scheme plot. Due to some confits in package build and requirement for streamline, we only utilized part codes from package "agricolae" and appreciate their contribution in this filed.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2021.803627/full#supplementary-material

REFERENCES

Biagi, E., Franceschi, C., Rampelli, S., Severgnini, M., Ostan, R., Turroni, S., et al. (2016). Gut Microbiota and Extreme Longevity. *Curr. Biol.* 26, 1480–1485. doi:10.1016/j.cub.2016.04.016

Bian, G., Gloor, G. B., Gong, A., Jia, C., Zhang, W., Hu, J., et al. (2017). The Gut Microbiota of Healthy Aged Chinese Is Similar to that of the Healthy Young. *mSphere* 2, e00327. doi:10.1128/mSphere.00327-17

Caporaso, J. G., Kuczynski, J., Stombaugh, J., Bittinger, K., Bushman, F. D., Costello, E. K., et al. (2010). QIIME Allows Analysis of High-Throughput Community Sequencing Data. *Nat. Methods* 7, 335–336. doi:10.1038/nmeth.f.303

Claesson, M. J., Clooney, A. G., and O'Toole, P. W. (2017). A Clinician's Guide to Microbiome Analysis. *Nat. Rev. Gastroenterol. Hepatol.* 14, 585–595. doi:10.1038/nrgastro.2017.97

Cryan, J. F., O'Riordan, K. J., Cowan, C. S. M., Sandhu, K. V., Bastiaanssen, T. F. S., Boehme, M., et al. (2019). The Microbiota-Gut-Brain Axis. *Physiol. Rev.* 99, 1877–2013. doi:10.1152/physrev.00018.2018

Faith, D. P. (1992). Conservation Evaluation and Phylogenetic Diversity. *Biol. Conserv.* 61, 1–10. doi:10.1016/0006-3207(92)91201-3

Fan, Y., and Pedersen, O. (2021). Gut Microbiota in Human Metabolic Health and Disease. *Nat. Rev. Microbiol.* 19, 55–71. doi:10.1038/s41579-020-0433-9

Garcia-Mantrana, I., Selma-Royo, M., Alcantara, C., and Collado, M. C. (2018). Shifts on Gut Microbiota Associated to Mediterranean Diet Adherence and Specific Dietary Intakes on General Adult Population. *Front. Microbiol.* 9, 890. doi:10.3389/fmicb.2018.00890

He, Y., Wu, W., Wu, S., Zheng, H.-M., Li, P., Sheng, H.-F., et al. (2018a). Linking Gut Microbiota, Metabolic Syndrome and Economic Status Based on a Population-Level Analysis. *Microbiome* 6. doi:10.1186/s40168-018-0557-6

He, Y., Wu, W., Zheng, H.-M., Li, P., McDonald, D., Sheng, H.-F., et al. (2018b). Regional Variation Limits Applications of Healthy Gut Microbiome Reference Ranges and Disease Models. *Nat. Med.* 24, 1532–1535. doi:10.1038/s41591-018-0164-x

Jiang, Z., Sun, T.-y., He, Y., Gou, W., Zuoshi, L.-s. -y., Fu, Y., et al. (2020). Dietary Fruit and Vegetable Intake, Gut Microbiota, and Type 2 Diabetes: Results from Two Large Human Cohort Studies. *BMC Med.* 18. doi:10.1186/s12916-020-01842-0

Knight, R., Vrbanac, A., Taylor, B. C., Aksenov, A., Callewaert, C., Debelius, J., et al. (2018). Best Practices for Analysing Microbiomes. *Nat. Rev. Microbiol.* 16, 410–422. doi:10.1038/s41579-018-0029-9

Kordahi, M. C., Stanaway, I. B., Avril, M., Chac, D., Blanc, M.-P., Ross, B., et al. (2021). Genomic and Functional Characterization of a Mucosal Symbiont Involved in Early-Stage Colorectal Cancer. *Cell Host & Microbe* 29, 1589–1598. doi:10.1016/J.CHOM.2021.08.013

Liu, B., Lin, W., Chen, S., Xiang, T., Yang, Y., Yin, Y., et al. (2019). Gut Microbiota as an Objective Measurement for Auxiliary Diagnosis of Insomnia Disorder. *Front. Microbiol.* 10, 1770. doi:10.3389/fmicb.2019.01770

Liu, C., Cui, Y., Li, X., and Yao, M. (2021). Microeco: an R Package for Data Mining in Microbial Community Ecology. *FEMS Microbiol. Ecol.* 97. doi:10.1093/femsec/fiaa255

Marchesi, J. R., Adams, D. H., Fava, F., Hermes, G. D. A., Hirschfield, G. M., Hold, G., et al. (2016). The Gut Microbiota and Host Health: A New Clinical Frontier. *Gut* 65, 330–339. doi:10.1136/gutjnl-2015-309990

Oleskin, A. V., and Shenderov, B. A. (2016). Neuromodulatory Effects and Targets of the SCFAs and Gasotransmitters Produced by the Human Symbiotic Microbiota. *Microb. Ecol. Health Dis.* 27, 1–12. doi:10.3402/mehd.v27.30971

Peterson, J., Peterson, J., Garges, S., Giovanni, M., McInnes, P., Wang, L., et al. (2009). The NIH Human Microbiome Project. *Genome Res.* 19, 2317–2323. doi:10.1101/GR.096651.109

Qi, X., Yun, C., Sun, L., Xia, J., Wu, Q., Wang, Y., et al. (2019). Gut Microbiota-Bile Acid-Interleukin-22 axis Orchestrates Polycystic Ovary Syndrome. *Nat. Med.* 25, 1225–1233. doi:10.1038/s41591-019-0509-0

Ren, Z., Li, A., Jiang, J., Zhou, L., Yu, Z., Lu, H., et al. (2018). Gut Microbiome Analysis as a Tool towards Targeted Non-invasive Biomarkers for Early Hepatocellular Carcinoma. *Gut* 68, 1014–1023. doi:10.1136/gutjnl-2017-315084

Scheiman, J., Luber, J. M., Chavkin, T. A., MacDonald, T., Tung, A., Pham, L.-D., et al. (2019). Meta-omics Analysis of Elite Athletes Identifies a Performance-Enhancing Microbe that Functions via Lactate Metabolism. *Nat. Med.* 25, 1104–1109. doi:10.1038/s41591-019-0485-4

Singh, R. K., Chang, H.-W., Yan, D., Lee, K. M., Ucmak, D., Wong, K., et al. (2017). Influence of Diet on the Gut Microbiome and Implications for Human Health. *J. Transl. Med.* 15, 73. doi:10.1186/s12967-017-1175-y

Van Treuren, W., and Dodd, D. (2020). Microbial Contribution to the Human Metabolome: Implications for Health and Disease. *Annu. Rev. Pathol. Mech. Dis.* 15, 345–369. doi:10.1146/annurev-pathol-020117-043559

Vangay, P., Hillmann, B. M., and Knights, D. (2019). Microbiome Learning Repo (ML Repo): A Public Repository of Microbiome Regression and Classification Tasks. *Gigascience* 8. doi:10.1093/gigascience/giz042

Wilck, N., Matus, M. G., Kearney, S. M., Olesen, S. W., Forslund, K., Bartolomaeus, H., et al. (2017). Salt-responsive Gut Commensal Modulates TH17 axis and Disease. *Nature* 551, 585–589. doi:10.1038/nature24628

Zhao, Y., Federico, A., Faits, T., Manimaran, S., Segrè, D., Monti, S., et al. (2021). Animalcules : Interactive Microbiome Analytics and Visualization in R. *Microbiome* 9 (1), 76. doi:10.1186/s40168-021-01013-0

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers in Genetics

TECHNOLOGY AND CODE
published: 05 January 2022
doi: 10.3389/fgene.2021.784531

BioInfograph: An Online Tool to Design and Display Multi-Panel Scientific Figure Interactively

*Kejie Li, Jessica Hurt, Christopher D. Whelan, Ravi Challa, Dongdong Lin and Baohong Zhang**

Translational Biology, Biogen Inc., Cambridge, MA, United States

Many fit-for-purpose bioinformatics tools generate plots to interpret complex biological data and illustrate findings. However, assembling individual plots in different formats from various sources into one high-resolution figure in the desired layout requires mastery of commercial tools or even programming skills. In addition, it is a time-consuming and sometimes frustrating process even for a computationally savvy scientist who frequently takes a trial-and-error iterative approach to get satisfactory results. To address the challenge, we developed bioInfograph, a web-based tool that allows users to interactively arrange high-resolution images in diversified formats, mainly Scalable Vector Graphics (SVG), to produce one multi-panel publication-quality composite figure in both PDF and HTML formats in a user-friendly manner, requiring no programming skills. It solves stylesheet conflicts of coexisting SVG plots, integrates a rich-text editor, and allows creative design by providing advanced functionalities like image transparency, controlled vertical stacking of plots, versatile image formats, and layout templates. To highlight, the sharable interactive HTML output with zoom-in function is a unique feature not seen in any other similar tools. In the end, we make the online tool publicly available at https://baohongz.github.io/bioInfograph while releasing the source code at https://github.com/baohongz/bioInfograph under MIT open-source license.

Keywords: bioinformatics, infographic, high-resolution, Scalable Vector Graphics, multi-panel figure

OPEN ACCESS

Edited by:
Guangchuang Yu,
Southern Medical University, China

Reviewed by:
Marco Brandizi,
Rothamsted Research,
United Kingdom
Veronica A. Segarra,
High Point University, United States

****Correspondence:***
Baohong Zhang
baohong.zhang@biogen.com

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

Received: *28 September 2021*
Accepted: *26 November 2021*
Published: *05 January 2022*

Citation:
Li K, Hurt J, Whelan CD, Challa R, Lin D and Zhang B (2022) BioInfograph: An Online Tool to Design and Display Multi-Panel Scientific Figure Interactively. Front. Genet. 12:784531. doi: 10.3389/fgene.2021.784531

INTRODUCTION

Popular computational biology databases such as Reactome (Jassal et al., 2020), WikiPathways (Martens et al., 2021), and visualization tools such as Coral (Metz et al., 2018) and ComplexHeatmap (Gu et al., 2016) often produce biological images in Scalable Vector Graphics (SVG) format. SVG is an Extensible Markup Language (XML)-based vector image format, scalable to any resolution without blurry pixelization that happens in other popular image formats such as png, gif, and jpg. This format has become one of the most broadly used image outputs adopted by many data analysis tools used by computational biologists, notably R (Venables et al., 2002), ggplot2 (Wickham, 2016), and numerous R and Bioconductor (Gentleman et al., 2004) packages. In addition, SVG is usually set as the default image output by many JavaScript-based plotting libraries like D3 (Bostock et al., 2011). To point out, these SVG images are rendered naturally by modern web browsers including Chrome, Firefox, Safari, and Microsoft Edge.

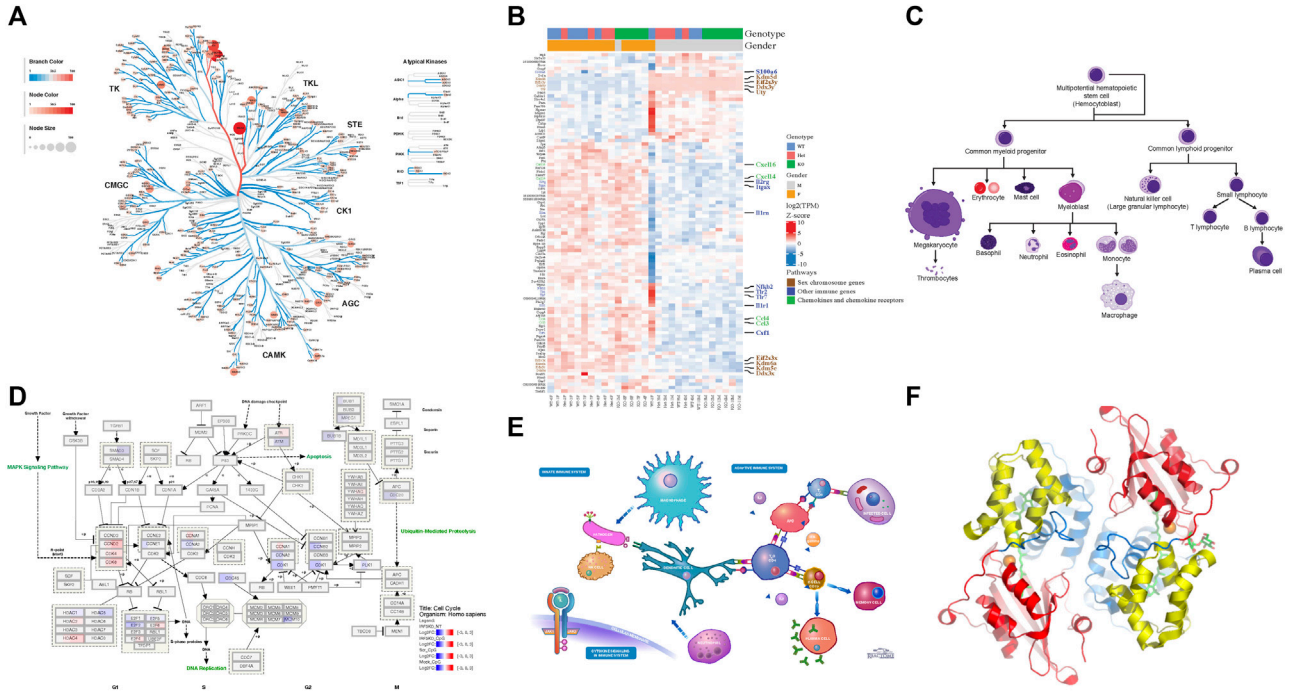

FIGURE 1 | High-resolution plots generated by various tools are arranged by the online tool, bioInfograph, to produce a composite plot. Unless specified, the source plot is in Scalable Vector Graphics (SVG) format. **(A)** Human kinome tree generated by Coral web app. **(B)** Gene expression heatmap by R package ComplexHeatmap. **(C)** Blood cell lineage from Wikimedia Commons (https://bit.ly/2Wjc5aS). **(D)** Human cell cycle pathway diagram from WikiPathways. **(E)** Human immune system illustration from Reactome. **(F)** Protein 3D structure ribbon form in png format by PyMOL (DeLano, 2000). An interactive version of the figure for the enlarged view of individual panels is available at https://bit.ly/39ClQnD.

Composing multi-panel publication-ready figures, such as the one presented in **Figure 1**, usually poses a challenge for biologists with no or modest programming skills after gathering individual plots from various sources in diversified formats, such as png, gif, jpg, tiff, pdf, and svg. Nevertheless, creating graphical abstracts like **Figure 1** to give a high-level comprehensive story becomes a routine task in biological publication. And often, such illustration is required to be in high resolution. Biologists usually turn to user-friendly commercial tools, such as Microsoft PowerPoint, as viable options to arrange such plots. But these tools either cannot deal with complex pathway diagrams in SVG format from WikiPathways, or render this format in low resolution with missing colors, sometimes even in malformed appearance as shown in **Figure 2**.

A previously developed web-based plot designing tool, canvasDesigner (Zhang et al., 2018), attempted to provide a solution but with limited success. It fails to handle stylesheet conflicts caused by SVG files from different tools and lacks flexibility in design where images are required to overlay onto each other. Moreover, singular input image format and rudimentary text support hinder its usability. To address these major shortcomings, we revamped the new version to accept more image formats in bioInfograph beyond only SVG, as acquiring such format might be unfeasible in certain circumstances such as scanned gel images, and we improved usability tremendously by implementing advanced functions outlined in the *Materials and Methods* section.

MATERIALS AND METHODS

Implementation and Usage of BioInfograph

With simplicity and accessibility in mind, it is implemented as a one-page, client-only, web-based application without the server-side component, available online at https://baohongz.github.io/bioInfograph. Written in plain JavaScript language, bioInfograph takes advantage of open-source JavaScript libraries including common ones like jQuery, bootstrap, and lodash. As shown in **Figure 3A**, other special JavaScript libraries are listed under each of three functional modules, "Upload images," "Layout images," and "Save HTML," to show the design of the software. First, dropzone.js makes it easy to upload or drag and drop image files to the tool. The content of uploaded or dropped files will be put on the canvas for layout. The source code in the library is modified to allow emitting "previewReady" status when an image is fully loaded into memory and displayed in the preview box; see https://bit.ly/3Gup4Zp for details. Second, gridstack.js is used to layout draggable, resizable, responsive bootstrap-friendly panels in a grid on the designing canvas. Each panel in the grid holds one image that can be panned or zoomed in and out by attached control provided by svg-pan-zoom.js. Modifications are made in gridstack.js to preserve inline styles, including positions, size, and z-index in order to drag a panel to an accurate location instead of pre-defined stops; see https://bit.ly/3CaOg3T for details. Functions of tinymce.js and svg-inject.js libraries are discussed in the following related sections. Third,

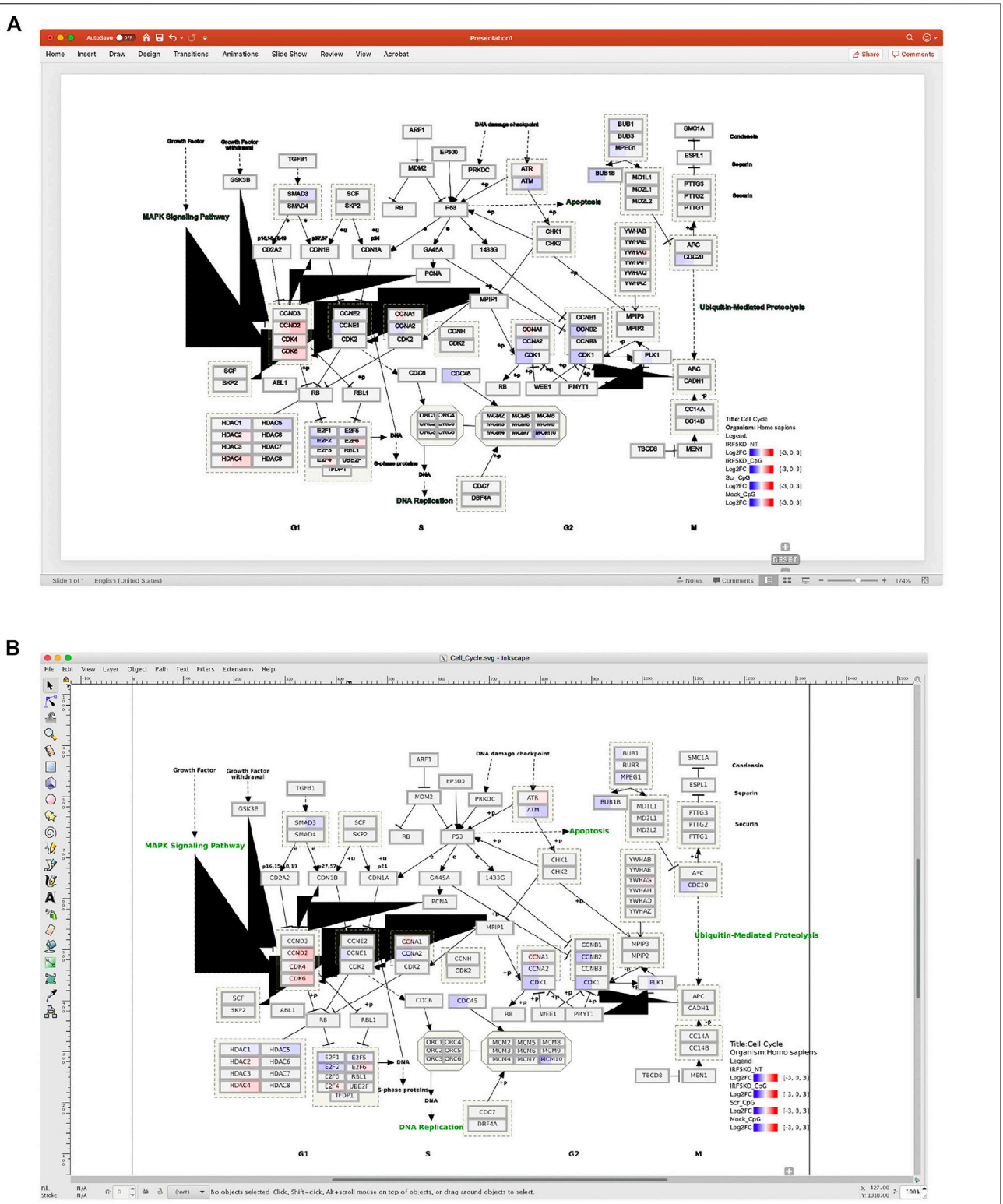

FIGURE 2 | An example of a pathway diagram from WikiPathways is not properly rendered by **(A)** Microsoft PowerPoint or **(B)** Inkscape. Please note the unexpected black triangles generated by both tools and loss of green color in the text (e.g., DNA replication) by PowerPoint, while the same Scalable Vector Graphics (SVG) image is rendered perfectly by bioInfograph as shown in **Figure 1D**.

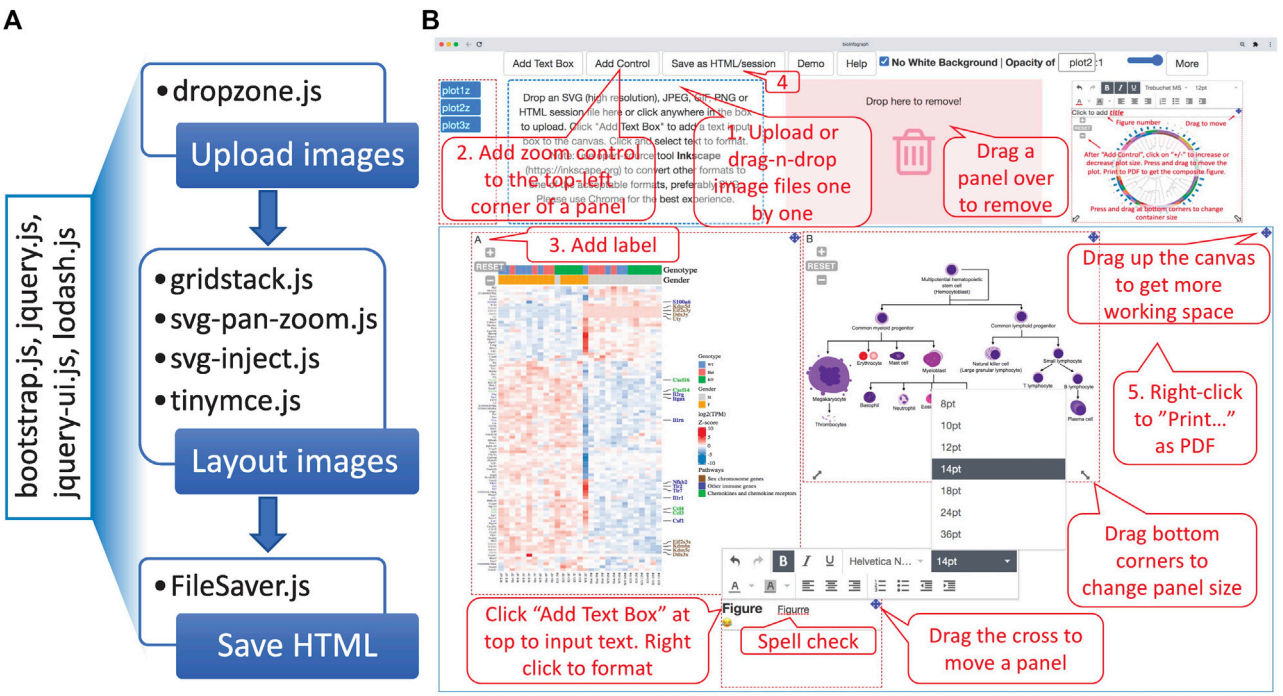

FIGURE 3 | Overview of the software architecture and main features. **(A)** The one-page web app is powered by general JavaScript libraries listed in the rotated box and specific libraries used in each of three functional modules, "Upload images," "Layout images," and "Save HTML." **(B)** BioInfograph allows users to easily arrange multiple plots in Scalable Vector Graphics (SVG), png, jpeg, or gif format exported by other tools. Each plot can be adjusted in size and placed freely on the canvas. A minimal workflow is outlined by numbered callout boxes in red, while important features not required by the minimal workflow are briefed in un-numbered boxes.

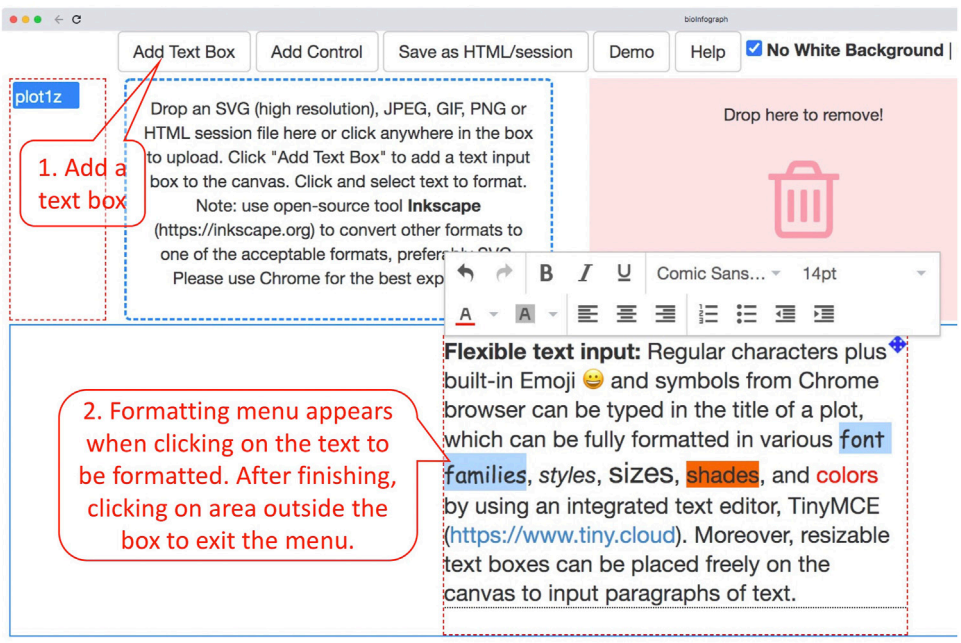

FIGURE 4 | Input of paragraphs of text by clicking on "Add Text Box" and then typing in the resizable box that was added to the canvas a moment ago. Clicking on the text will fire up the formatting menu, while clicking on any area outside the box will exit the menu. Formatting is always applied to selected text.

FileSaver.js is utilized to save image content and associated metadata about size, position, opacity, and zoom scale in an HTML file. When taken together, an intuitive user interface is built and shown in action as illustrated in **Figure 3B**, where control elements are located at the top, functional modules in the middle, and a movable, dynamically resizable canvas at the bottom. A very basic workflow is outlined by numbered callout boxes consisting of five steps: 1) uploading images; 2) adding pan-zoom control to fine-tune image size and position; 3) adding labels; 4) saving the work as an HTML file; and 5) printing as PDF. While not required in the minimal setting, all other un-numbered boxes highlight important features to smooth the design process, such as moving the canvas up to create more working space, changing the size of an individual panel, dropping a panel to a trash bin, and adding text box for typing paragraphs of text with spell checking. Due to the space limitation of the figure, some features are discussed in more detail below.

Besides online access, users can install it as a desktop app by downloading the html page or creating a shortcut of the page on the desktop by following the instruction in GitHub repo, https://bit.ly/3wTxoxk. To be aware, the tool is fully tested in the Chrome browser, which provides the best experience.

Flexible Text Input

Regular characters plus built-in Emoji and symbols from Chrome browser can be typed in the title of a plot, which can be fully formatted in various font families, styles, sizes, shades, and colors by using an integrated text editor, TinyMCE (https://www.tiny.cloud). Moreover, resizable text boxes can be placed freely on the canvas to input paragraphs of text by following the instructions in **Figure 4**. The markdown language has gained popularity in authoring simple documents especially within R and GitHub communities. A very simple markdown processor is enabled by using tinymce's text pattern plugin that matches the following patterns (source code block from index.html) in the text and applies corresponding formats on these patterns; e.g., "*test*" will become "*test*" in the editor.

```
textpattern_patterns: [
    {start: '*', end: '*', format: 'italic'},
    {start: '**', end: '**', format: 'bold'},
    {start: '#', format: 'h1'},
    {start: '##', format: 'h2'},
    {start: '###', format: 'h3'},
    {start: '####', format: 'h4'},
    {start: '#####', format: 'h5'},
    {start: '######', format: 'h6'},
    {start: '1. ', cmd: 'InsertOrderedList'},
    {start: '* ', cmd: 'InsertUnorderedList'},
    {start: '- ', cmd: 'InsertUnorderedList'}
    ],
```

Versatile Image Formats

Besides the SVG format, bioInfograph accepts directly additional popular image formats including png, gif, and jpg as input. For other formats like tiff or pdf, free tools such as Inkscape (https://inkscape.org) (Bah, 2007) or pdf2svg (https://bit.ly/2NVtj6E) can

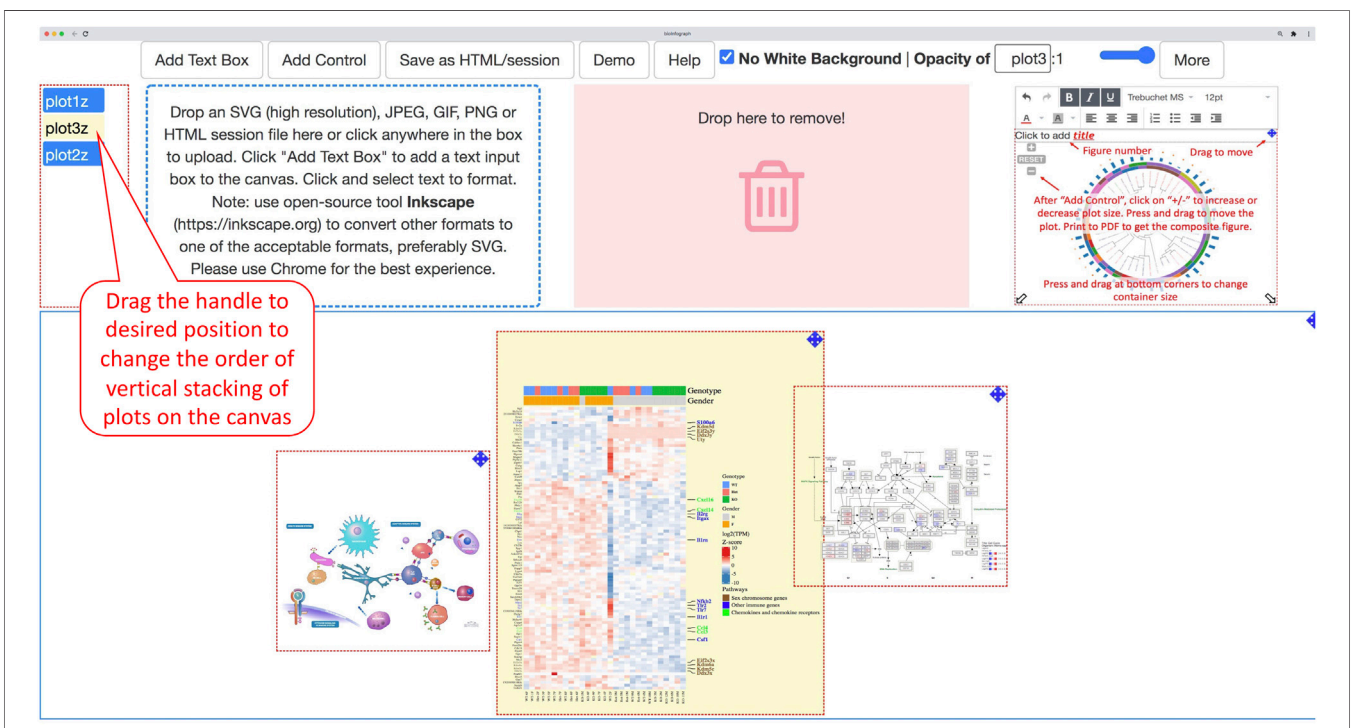

FIGURE 5 | Each plot will get a button handle when it is loaded onto the canvas. A plot and the linked handle will be highlighted in yellow when hovering the mouse over a handle. Dragging the handle and dropping it at the desired position among these buttons will change the relative vertical stacking, also known as z-index of the plot. The bottom position represents the top layer of the stack of plots on the canvas.

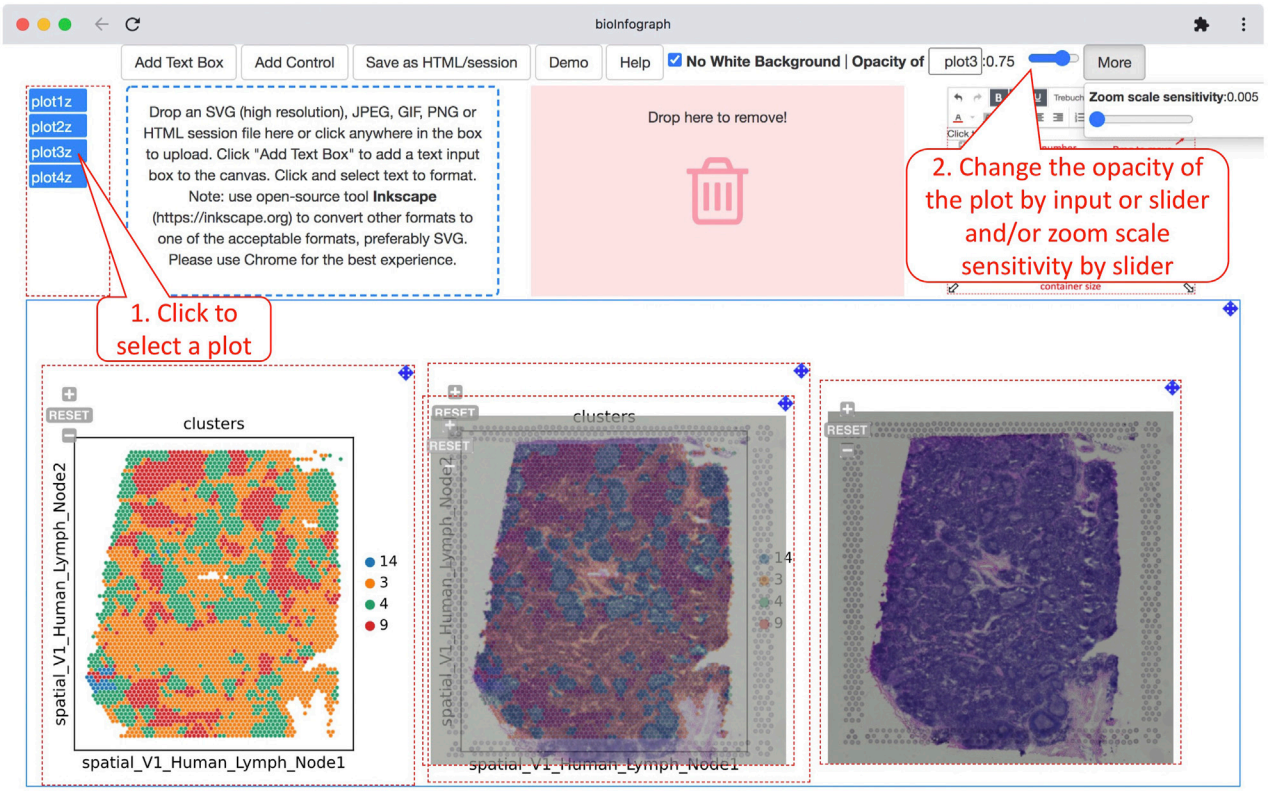

FIGURE 6 | Overlaying colored spatial clustering plot and histological image to illustrate the relationship of histological features and clusters based on 10X Genomics spatial transcriptomics data (https://bit.ly/3F0xKWD). Users can adjust the opacity of a selected plot by clicking a plot handle to select the plot and then typing a number or using the slide to change the value. The zoom scale sensitivity of the plot is tuned to the smallest number for fine alignment of overlaid plots.

be utilized to convert these to one of the acceptable formats, preferably SVG.

Stylesheet Conflict

Since stylesheet definitions in SVG files are always applied globally to style elements, they share the same parse tree when multiple inline SVGs are embedded in a single document. Therefore, style overwriting and component id collisions can occur and upset the rendering in canvasDesigner as shown in **Supplementary Figure S1**. To overcome these shortcomings, bioInfograph automatically converts global definitions into inline styles embedded in each targeting element individually, stores it locally, and then removes these definitions from the global scope to solve the overwriting issue. Then, it utilizes a modified version of svg-inject.js (see https://bit.ly/3Gus3kz for details) to make ids in the document unique by appending original ids with a suffix in the form of "--inject-X", where X is a running number that is incremented with each added SVG image.

Vertical Stacking

Each image is associated with a vertically stacked control button. Desired vertical stacking order (z-index) is attainable by moving these control buttons up or down by mouse as demonstrated in **Figure 5**, which provides an additional dimension for creative design that often requires overlapped images in a certain order.

Image Transparency

The white background in the SVG file is optionally removable to make it transparent so that plots can be overlaid onto each other to create appealing art. Opacities of individual images can be adjusted granularly as well to make a comprehensive effect of overlaid images as showcased in visualizing spatial transcriptomics data, which is displayed in **Figure 6**. In this use case, vertical stacking of gene expression data on top of histopathology images or vice versa with adjustable transparency is a crucial visualization capability to investigate the relationship between the transcriptional signals and disease pathology.

Interactive HTML Output and Saved Session

The finished work can be saved as a self-contained HTML file with necessary JavaScript code embedded for easy sharing by email or hosting at GitHub-like services as exemplified at https://bit.ly/39ClQnD. An individual plot can be enlarged and further zoomed in to view details in high resolution by clicking on the plot and then the button with a plus sign in the popup window. Unique to this HTML presentation, links to detailed information of proteins in UniProt database (The UniProt Consortium, 2021) are active in panel A of the interactive figure as shown in **Figure 7**. Therefore, bioInfograph output can act as an information portal beyond mere pictures by embedding links to dissipated computational biology resources in SVG figures. Meanwhile,

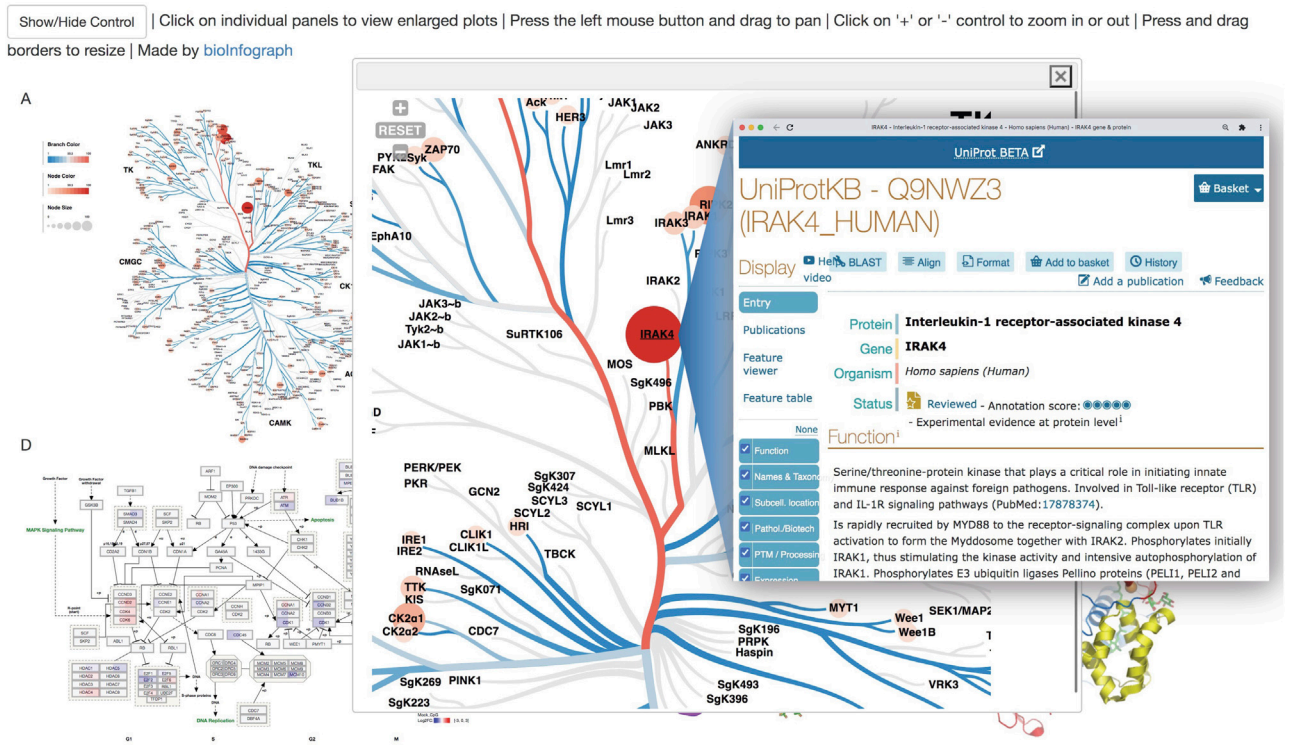

FIGURE 7 | Interactive online HTML presentation of **Figure 1** with zoom-in and link-out features. Clicking on an individual plot will bring up a popup window with the enlarged zoomable version. Links in Scalable Vector Graphics (SVG) are active, so clicking on "IRAK4" on the node in the phylogenetic tree will show detailed information about the protein in UniProt database.

TABLE 1 | Comparison scorecard of figure design tools.

	BioInfograph v1.0	Canvasdesigner v1.0	MS powerpoint v16.3	Adobe acrobat pro DC v2020.006	Patchwork v1.0	Inkscape v0.92
Open source/cost	Yes/free	Yes/free	No/license fee	No/license fee	Yes/free	Yes/free
Multi-image formats	Yes	No	Yes	Yes	No	Yes
Rendering speed	Fast	Fast	Fast	Fast	Fast	Slow
Text input	Yes	No	Yes	Yes	Yes	Yes
Interactive HTML output	Yes	Yes	No	No	No	No
SVG input	Yes	Yes	Yes	No	No	Yes
SVG stylesheet compatibility	Yes	No	Yes	N/A	N/A	No
Image transparency	Yes	No	Yes	No	N/A	Yes
Saving session	Yes	No	Yes	No	No	No
Installation free	Yes	Yes	No	No	No	No

SVG, Scalable Vector Graphics.

the saved HTML file also serves as a session file that can be loaded back into the tool to restore the work for further modification.

RESULTS

We developed bioInfograph, an interactive web-based tool with a focus on computational biology, which arranges high-resolution images in various formats, mainly SVG, to produce one multi-panel publication-quality composite figure in both PDF and interactive HTML formats in a user-friendly manner, requiring no programming skills.

We compared it with several popular tools to illustrate the advanced features of bioInfograph. Among the six tools listed in **Table 1**, except patchwork (Pedersen, 2019), which is a command line based tool, the rest offers an interactive user-

friendly interface. In addition, bioInfograph and canvasDesigner are conveniently accessible web-based tools. Regarding image formats, Adobe Acrobat and patchwork will not take SVG as input natively, while PowerPoint and Inkscape have issues when rendering complex pathway diagrams in SVG format as shown in **Figure 2**. Although canvasDesigner and bioInfograph share many common features, bioInfograph breaks the limitations of canvasDesigner by solving conflicting stylesheet issues, accepting images in various formats, overlaying images in any order vertically, adjusting image transparency, and providing flexible text input. In summary, we outline a comparison scorecard of features among these tools including both open source solutions and popular commercial tools available to the authors in **Table 1**.

CONCLUSION

BioInfograph is an open-source and publicly available web-based tool that can be accessed online or downloaded as a desktop application. It has the most feasible features to improve productivity in the case of creating high-resolution multi-panel figures for scientific publication. Furthermore, the innovative HTML output brings a new way of illustrating high-resolution figures interactively with unlimited zoom-in capability, which could be a nice feature for journals to incorporate in online publishing.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**. Further inquiries can be directed to the corresponding author.

AUTHOR CONTRIBUTIONS

KL and BZ conceived and designed the tool that BZ implemented. KL, JH, CW, RC, DL, and BZ tested the tool, contributed to the writing, and approved the final manuscript.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2021.784531/full#supplementary-material

REFERENCES

Bah, T. (2007). *Inkscape: Guide to a Vector Drawing Program*. Upper Saddle River, NJ: prentice hall press.

Bostock, M., Ogievetsky, V., and Heer, J. (2011). D^3 Data-Driven Documents. *IEEE Trans. Vis. Comput. Graphics* 17, 2301–2309. doi:10.1109/tvcg.2011.185

DeLano, W. L. (2000). *The PyMOL Molecular Graphics System*. 2.0 ed. San Carlos, USA: Schrödinger, LLC.

Gentleman, R. C., Carey, V. J., Bates, D. M., Bolstad, B., Dettling, M., Dudoit, S., et al. (2004). Bioconductor: Open Software Development for Computational Biology and Bioinformatics. *Genome Biol.* 5, R80. doi:10.1186/gb-2004-5-10-r80

Gu, Z., Eils, R., and Schlesner, M. (2016). Complex Heatmaps Reveal Patterns and Correlations in Multidimensional Genomic Data. *Bioinformatics* 32, 2847–2849. doi:10.1093/bioinformatics/btw313

Jassal, B., Matthews, L., Viteri, G., Gong, C., Lorente, P., Fabregat, A., et al. (2020). The Reactome Pathway Knowledgebase. *Nucleic Acids Res.* 48, D498–D503. doi:10.1093/nar/gkz1031

Martens, M., Ammar, A., Riutta, A., Waagmeester, A., Slenter, D. N., Hanspers, K., et al. (2021). WikiPathways: Connecting Communities. *Nucleic Acids Res.* 49, D613–D621. doi:10.1093/nar/gkaa1024

Metz, K. S., Deoudes, E. M., Berginski, M. E., Jimenez-Ruiz, I., Aksoy, B. A., Hammerbacher, J., et al. (2018). Coral: Clear and Customizable Visualization of Human Kinome Data. *Cel Syst.* 7, 347–350. e341. doi:10.1016/j.cels.2018.07.001

Pedersen, T. (2019). *Patchwork: The Composer of Plots*.

The UniProt Consortium (2021). UniProt: the Universal Protein Knowledgebase in 2021. *Nucleic Acids Res.* 49, D480–D489. doi:10.1093/nar/gkaa1100

Venables, W. N., and Smith, D. M.R Development Core Team (2002). *An Introduction to R : Notes on R: A Programming Environment for Data Analysis and Graphics, Version 1.4.1*. Bristol: Network Theory.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Verlag New York: Springer.

Zhang, B., Zhao, S., and Neuhaus, I. (2018). canvasDesigner: a Versatile Interactive High-Resolution Scientific Multi-Panel Visualization Toolkit. *Bioinformatics* 34, 3419–3420. doi:10.1093/bioinformatics/bty377

Conflict of Interest: All authors are current or former employees of Biogen and hold Biogen stocks.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers in Genetics

REVIEW
published: 27 January 2022
doi: 10.3389/fgene.2021.785290

Analysis and Visualization of Spatial Transcriptomic Data

*Boxiang Liu *†, Yanjun Li† and Liang Zhang*

Baidu Research, Sunnyvale, CA, United States

Human and animal tissues consist of heterogeneous cell types that organize and interact in highly structured manners. Bulk and single-cell sequencing technologies remove cells from their original microenvironments, resulting in a loss of spatial information. Spatial transcriptomics is a recent technological innovation that measures transcriptomic information while preserving spatial information. Spatial transcriptomic data can be generated in several ways. RNA molecules are measured by *in situ* sequencing, *in situ* hybridization, or spatial barcoding to recover original spatial coordinates. The inclusion of spatial information expands the range of possibilities for analysis and visualization, and spurred the development of numerous novel methods. In this review, we summarize the core concepts of spatial genomics technology and provide a comprehensive review of current analysis and visualization methods for spatial transcriptomics.

Keywords: spatial transcriptomics, single-cell RNA-seq (scRNA-seq), clustering, cell-type identification, dimensionality reduction, spatial expression pattern, spatial interaction, visualization

OPEN ACCESS

Edited by:
Guangchuang Yu,
Southern Medical University, China

Reviewed by:
Xin Li,
Shanghai Institute of Nutrition and Health (CAS), China
Lu Zhang,
Hong Kong Baptist University, Hong Kong SAR, China

****Correspondence:***
Boxiang Liu
jollier.liu@gmail.com

†These authors have contributed equally to this work

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

Received: *29 September 2021*
Accepted: *24 December 2021*
Published: *27 January 2022*

Citation:
Liu B, Li Y and Zhang L (2022) Analysis and Visualization of Spatial Transcriptomic Data. Front. Genet. 12:785290. doi: 10.3389/fgene.2021.785290

1 INTRODUCTION

Quantification of gene expression has important applications across various aspects of biology. Understanding the spatial distribution of gene expression has helped to answer fundamental questions in developmental biology (Asp et al., 2019; Rödelsperger et al., 2021), pathology (Maniatis et al., 2019; Chen et al., 2020), cancer microenvironment (Berglund et al., 2018; Thrane et al., 2018; Ji et al., 2020; Moncada et al., 2020), and neuroscience (Shah et al., 2016; Moffitt et al., 2018; Close et al., 2021). Two widely used methods for gene expression quantification are fluorescent *in situ* hybridization (FISH) and next-generation sequencing. With FISH, fluorescently-labeled RNA sequences are used as probes to identify its naturally occurring complementary sequence in cells while preserving the spatial location of the target sequences (Schwarzacher and Heslop-Harrison, 2000). Traditionally, the number of target sequences simultaneously identified by *in situ* hybridization is restricted by the number of fluorescent channels, making this method suitable for targeted gene detection. On the other hand, next-generation sequencing methods use a shotgun approach to quantify RNA molecules across the entire transcriptome (Metzker, 2010). To achieve transcriptome-wide quantification, RNA must be first isolated and purified, which removes RNA molecules from their native microenvironment. Even with single-cell sequencing, where the cellular origin of RNA molecules is preserved, spatial information of cells can only be inferred but not directly measured (Shapiro et al., 2013; Gawad et al., 2016).

Various approaches have been made to measure gene expression while preserving spatial information. Tomo-seq applied the principle of tomography to measure spatial transcriptomic information in 3D. In tomo-seq, tissue samples are sliced by cryosection and measured with RNA-seq. Each measurement corresponds to the average gene expression within a slice. Measurements are taken along multiple axes to reconstruct pixel-wise 3D gene expression information. (Junker et al.,

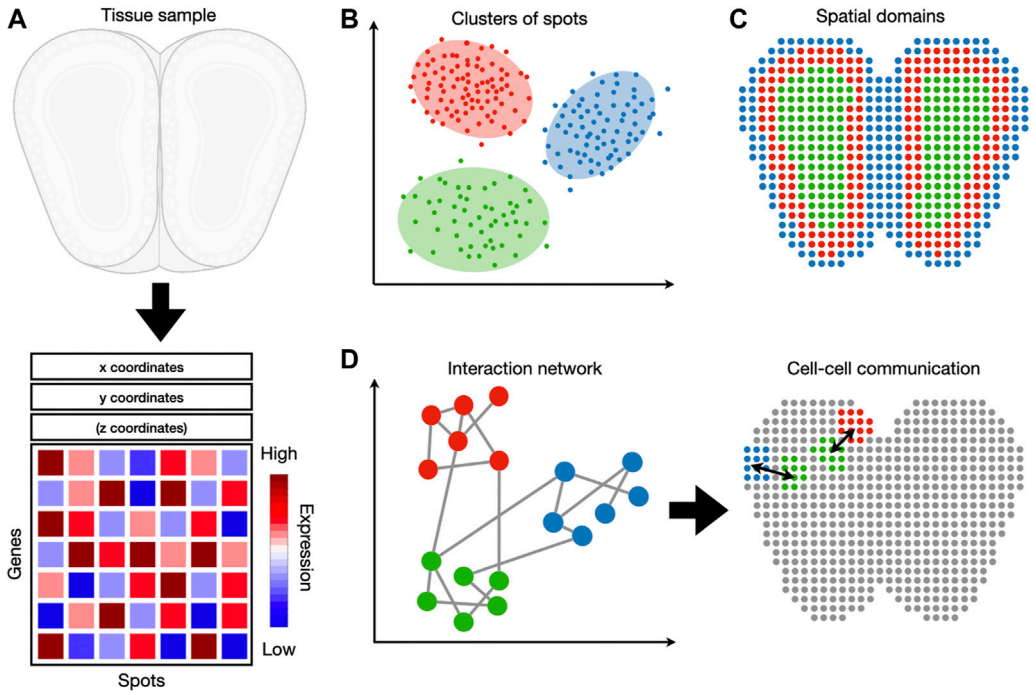

FIGURE 1 | An overview of spatial transcriptomic tasks. **(A)** Spatial transcriptomic datasets map gene expression measurements to their respective locations. **(B)** A spatial transcriptomic dataset can be analyzed in gene expression space, irrespective of spatial locations. Tasks such as clustering and cell-type identification fall into this category. **(C)** Spatial information can be used jointly with gene expression to detect spatial expression patterns and spatial domains. **(D)** These two sources of information can also be used to detect cell-cell and gene-gene interactions.

2014). LCM-seq isolates single cells with laser capture microscopy (LCM) and measures captured cells with single-cell RNA-sequencing. LCM can capture cells of desired types and with specific spatial locations of the tissue specimen (Nichterwitz et al., 2016). While these methods retain the spatial location of RNA-seq measurements, they suffer from high labor costs and incomplete spatial coverage. In this review, we cover recent advances in spatial transcriptomic methods that attempt to address these challenges. In addition, we provide a comprehensive review of analysis and visualization techniques for spatial transcriptomic datasets.

The following sections are organized as follows (**Figure 1**). **Section 2** discusses the latest developments in experimental spatial transcriptomic technologies. **Section 3** discusses preprocessing of spatial transcriptomic data, an essential step prior to any analysis or visualization. **Section 4** dissects methods whose inputs are gene expression without spatial coordinates. This includes dimensionality reduction, clustering, and cell-type identification. **Section 5** describes methods whose inputs are gene expression combined with spatial coordinates. This includes identification of spatially coherent gene expression patterns and identification of spatial domains. **Section 6** describes methods that analyze the interaction between cells or genes. All methods reviewed are listed in **Table 1**. This includes the identification of cell-to-cell communication and gene interaction. We note that other reviews on spatial transcriptomic technology (Dries et al., 2021a) have been published during the peer review of this article.

2 SPATIAL TRANSCRIPTOMIC TECHNOLOGIES

Integration of spatial information with transcriptome-wide quantification has given rise to the emerging field of spatial transcriptomics. Currently, spatial transcriptome quantification falls into three broad categories (**Table 2**). First, spatial barcoding methods ligate oligonucleotide barcodes with known spatial locations to RNA molecules prior to sequencing (Ståhl et al., 2016; Rodriques et al., 2019; Vickovic et al., 2019; Liu et al., 2020; Chen et al., 2021; Cho et al., 2021; Stickels et al., 2021). Both barcodes and RNA molecules are jointly sequenced, and spatial information of sequenced RNA molecules can be recovered from associated barcodes. Second, *in situ* hybridization methods coupled with combinatorial indexing can vastly increase the number of RNA species identified (Lubeck et al., 2014; Chen et al., 2015; Moffitt et al., 2016; Eng et al., 2019). The latest *in situ* hybridization methods can detect around 10,000 RNA species from a given sample (Eng et al., 2019). Third, *in situ* sequencing method uses fluorescent-based direct sequencing to read out base pair information from RNA molecules in their original spatial location (Lee et al., 2014; Wang et al., 2018).

TABLE 1 | Current analysis and visualization tools for spatial transcriptomic datasets (accession date: 12/22/2021).

Task	Tool	Inputs	Description	Language	Availability
Preprocessing	Space Ranger	Microscope images and FASTQ files	Space Ranger is an analysis pipeline for alignment, tissue and fiducial detection, barcode/UMI counting, and feature-spot matrix generation.	Bash and GUI	https://support.10xgenomics.com/spatial-gene-expression/software/pipelines/latest/what-is-space-ranger
	Scran (2016); Lun et al. (2016)	Gene expression	Scran uses pool-based and deconvoluted cell-based size factors for single-cell gene expression normalization.	R	http://bioconductor.org/packages/release/bioc/html/scran.html
	SCNorm (2017); Bacher et al. (2017)	Gene expression	SCNorm uses double quantile regression-based model for gene-group normalization.	R	https://www.bioconductor.org/packages/release/bioc/html/SCnorm.html
Clustering	K-means	Gene expression	K-means iteratively assigns observations to the cluster with the nearest left.	R and Python	R: https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/kmeans; Python: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
	Gaussian mixture model	Gene expression	GMM is similar to K-means but softly assigns observations to clusters based on the Gaussian distribution.	R and Python	R: https://cran.r-project.org/web/packages/ClusterR/vignettes/the_clusterR_package.html; Python: https://scikit-learn.org/stable/modules/mixture.html
	hierarchical clustering	Gene expression	Hierarchical clustering iteratively merges closest observations.	R and Python	R: https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/hclust; Python: https://scikit-learn.org/stable/modules/clustering.html#hierarchical-clustering
	Louvain (2008); Blondel et al. (2008)	Gene expression	Louvain performs community detection within networks by iterative optimization of modularity.	R and Python	R: https://igraph.org/r/doc/cluster_louvain.html; Python: https://github.com/vtraag/louvain-igraph
	Leiden (2019); Traag et al. (2019)	Gene expression	Leiden is a variant of the Louvain algorithm that guarantees well-connected communities.	R and Python	R: https://cran.r-project.org/web/packages/leiden/; Python: https://github.com/vtraag/leidenalg
	SC3 (2017); Kiselev et al. (2017)	Gene expression	SC3 performs consensus clustering of single-cell RNA-seq data.	R	http://bioconductor.org/packages/release/bioc/html/SC3.html
	SIMLR (2017); Wang et al. (2017)	Gene expression	SIMLR is a multi-kernel learning approach for single-cell RNA-seq clustering.	R and MATLAB	MATLAB: https://github.com/BatzoglouLabSU/SIMLR; R: https://www.bioconductor.org/packages/release/bioc/html/SIMLR.html
Cell-specific marker genes	scran (2016); Lun et al. (2016)	Gene expression	Scran identifies consistently up-regulated genes through pairwise comparisons between clusters.	R	https://bioconductor.org/packages/devel/bioc/vignettes/scran/inst/doc/scran.html#6_Identifying_marker_genes
	scGeneFit (2021); Dumitrascu et al. (2021)	Gene expression	ScGeneFit is a label-aware compressive classification method to select informative marker genes.	Python	https://github.com/solevillar/scGeneFit-python
Cell-type identification	scmap (2018); Kiselev et al. (2018)	Gene expression	Scmap projects single-cell to References data sets with an approximate k-nearest-neighbor search.	R	http://bioconductor.org/packages/release/bioc/html/scmap.html; Web version: https://www.sanger.ac.uk/tool/scmap/
	SingleR (2019); Aran et al., 2019	Gene expression	SingleR iteratively calculates pairwise correlation across single cells and remove lowly correlated cell type for noise control.	R	https://github.com/dviraran/SingleR
	Cell-ID (2021); Cortal et al. (2021)	Gene expression of References and target single-cell datasets	Cell-ID performs multiple correspondence analysis (MCA) based gene signature extraction and cell identification	R	https://bioconductor.org/packages/devel/bioc/html/CelliD.html
	JSTA (2021); Littman et al. (2021)	*in situ* hybridization dataset	JSTA is a deep-learning-based cell segmentation and type annotation method by iteratively adjusting the assignment of boundary pixels based on the cell type probabilities for each pixel.	Python	https://github.com/wollmanlab/JSTA; https://github.com/wollmanlab/PySpots

(Continued on following page)

TABLE 1 | (*Continued*) Current analysis and visualization tools for spatial transcriptomic datasets (accession date: 12/22/2021).

Task	Tool	Inputs	Description	Language	Availability
Dimensionality reduction	Principal component analysis	Gene expression	PCA identifies orthogonal vectors that maximize the variance of projections from data points.	R and Python	R: https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/prcomp; Python: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
	t-SNE (2008); Van der Maaten and Hinton. (2008)	Gene expression	T-SNE iteratively refines projections in the low dimensional space to match pairwise distances in the high dimension space.	R and Python	R: https://cran.r-project.org/web/packages/Rtsne/; Python: https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html
	UMAP (2018); McInnes et al. (2018)	Gene expression	UMAP is similar to t-SNE but faster and better preserves high dimensional structure.	R and Python	R: https://cran.r-project.org/web/packages/umap/index.html; Python: https://umap-learn.readthedocs.io/en/latest/
Spatially coherent genes	SpatialDE (2018); Svensson et al. (2018)	Gene expression + spatial coordinates	SpatialDE uses gaussian process regression to decompose variability into spatial and non-spatial components.	Python	https://github.com/Teichlab/SpatialDE
	Trendsceek (2018); Edsgärd et al. (2018)	Gene expression + spatial coordinates	Trendsceek uses marked point processes to identify spatial expression patterns.	R	https://github.com/edsgard/trendsceek
	Spark (2018); Sun et al. (2020)	Gene expression + spatial coordinates	Spark is a generalized linear spatial model to identify spatial expression patterns.	R	https://xzhoulab.github.io/SPARK/
Spatial domains	Zhu *et al.* (2018); Zhu et al. (2018)	Gene expression + spatial coordinates	Zhu *et al.* uses a hidden Markov random field to compare gene expression of neighboring cells to identify coherent expression patterns.	R and Python	R: https://bitbucket.org/qzhudfci/smfishhmrf-r/src/master/; Python: https://bitbucket.org/qzhudfci/smfishhmrf-py/src/master/
	SpaGCN (2021); Hu et al. (2021b)	Gene expression + spatial coordinates + histology image	SpaGCN is a graph-convolutional-network-based method to jointly identify spatial domains and spatially variable genes.	Python	https://github.com/jianhuupenn/SpaGCN
Spot deconvolution	DSTG (2021); Song and Su (2021)	Gene expression + spatial coordinates	DSTG builds a graph consisting of real and pseudo spatial transcriptomic data and apply graph convolutional network to predict real data's cell type composition with help from pseudo data's label.	Python	https://github.com/Su-informatics-lab/DSTG
Super-resolution	BayesSpace (2021); Zhao et al. (2021)	Gene expression + spatial coordinates	BayesSpace is a Bayesian model to leverage neighborhood information to enhance resolution.	R	http://www.bioconductor.org/packages/release/bioc/html/BayesSpace.html
Cell-cell interaction	SpaOTsc (2020); Cang and Nie (2020)	Gene expression + spatial coordinates	SpaOTsc uses structured optimal transport between distribution of sender and receiver cells to identify cell-cell communication.	Python	https://github.com/zcang/SpaOTsc
Receptor-ligand interaction	GCNG (2020); Yuan and Bar-Joseph, (2020)	Gene expression + spatial coordinates	GCNG is a graph convolutional neural network to encode the spatial information as a graph and to predict whether a gene pair will interact.	Python	https://github.com/xiaoyeye/GCNG
Integrative	Seurat (2018); Waltman and Van Eck. (2013)	Gene expression + spatial coordinates	Seurat is an R package for integrative single-cell transcriptomic analysis.	R	https://cran.r-project.org/web/packages/Seurat/index.html
	Giotto (2021); Dries et al. (2021b)	Gene expression + spatial coordinates	Giotto is an R package for integrative spatial transcriptomic analysis.	R	https://rubd.github.io/Giotto_site/
	Scanpy (2018); Wolf et al. (2018)	Gene expression + spatial coordinates	Scanpy is a Python package for integrative single-cell transcriptomic analysis.	Python	https://scanpy.readthedocs.io/en/latest/
	Squidpy (2021); Palla et al. (2021)	Gene expression + spatial coordinates	Squidpy is a Python package for integrative spatial transcriptomic analysis.	Python	https://squidpy.readthedocs.io/en/stable/

Several metrics need to be considered when selecting a method for a specific application (**Table 2**). Methods employing *in situ* hybridization provide subcellular resolution. Leveraging super-resolution microscopy, *in situ* hybridization methods can achieve a resolution of ~10nm, sufficient to distinguish single RNA molecules (Schermelleh et al., 2019). In addition, *in situ* methods require no PCR amplification of cDNA, thus avoiding amplification bias. However, the number of RNA

TABLE 2 | Current experimental methods for spatial transcriptomic profiling.

Method	Type	Resolution	Genes	References
Visium	Spatial barcoding	55 μm	Whole transcriptome	Ståhl et al. (2016)
Slide-seq	Spatial barcoding	10 μm	Whole transcriptome	Rodriques et al. (2019), Stickels et al. (2021)
HDST	Spatial barcoding	2 μm	Whole transcriptome	Vickovic et al. (2019)
DBiT-Seq	Spatial barcoding	10 μm	Whole transcriptome	Liu et al. (2020)
Seq-scope	Spatial barcoding	0.5-0.8 μm	Whole transcriptome	Cho et al. (2021)
Stereo-seq	Spatial barcoding	0.5 or 0.715 μm	Whole transcriptome	Chen et al. (2021)
SeqFISH	*in situ* hybridization	single-molecule	>10,000	Lubeck et al. (2014), Eng et al. (2019)
MerFISH	*in situ* hybridization	single-molecule	100–1,000	Chen et al. (2015), Moffitt et al. (2016)
STARmap	*in situ* sequencing	single-cell	160–1,020	Wang et al. (2018)
FISSEQ	*in situ* sequencing	subcellular	~8,000	Lee et al. (2014)

species detected by *in situ* methods is limited by the indexing scheme. The current detection limit is ~10,000 genes but will likely improve in the future. Furthermore, the area examined by *in situ* methods is limited by the field-of-view of the microscope objective lens. In contrast, spatial barcoding followed by shotgun sequencing can in principle sample the whole transcriptome. This is ideal if the target molecules are unknown *a priori*. Spatial barcoding can also examine larger tissue areas, making it ideal for larger samples such as tissue slices from the brain. However, the density of measurement spots limits the spatial resolution of current spatial barcoding methods, ranging from multicellular to subcellular. In addition, shotgun sequencing inevitably suffers from PCR amplification bias (Aird et al., 2011), as well as "dropout" when sequencing read depth is insufficient (Kim et al., 2020). Thus far, we have provided an overall picture of different spatial transcriptomic methods and their characteristics. Because this review focuses on analysis and visualization of spatial transcriptomics, readers who wish to understand the experimental details can refer to comprehensive reviews elsewhere (Crosetto et al., 2015).

3 PREPROCESSING

Spatial transcriptomic datasets add a new dimension to transcriptomic analyses. Spatial coordinates of cells enable novel analyses such as spatial differential expression (Svensson et al., 2018) and cell-cell interaction (Cang and Nie, 2020). Similar to single-cell RNA-seq datasets, a spatial transcriptomic dataset can be represented by a gene-by-cell count matrix. A second matrix of coordinates is attached to the cell dimension of the count matrix to represent spatial information. Comprehensive toolkits such as Space Ranger can process raw sequence reads into count matrices. Taking a microscope image and FASTQ files as input, Space Ranger can perform alignment, tissue and fiducial detection, barcode/UMI counting, and feature-spot matrices generation.

Various preprocessing steps may be performed prior to any analysis. First, genes and cells may be filtered based on a threshold specific to the dataset. For example, a cell may be removed if it has 1) less than 1,000 expressed genes or 2) a high proportion of mitochondria RNA. A gene may be removed if it is detected in less than ten cells (Wolf et al., 2018; Lun et al., 1000). Transformation of count data may be performed according to downstream modeling assumptions. Methods modeling raw counts do not require any transformation (Sun et al., 2020). Otherwise, gene expression per cell may be normalized to have the same total library size such that expression levels are comparable across cells. The gene expression matrix may then be log-transformed and be regressed against confounders such as batch effect, percentage of mitochondria genes, and other technical variations. Although preprocessing steps mentioned above are widely adopted, the exact configuration should follow input data modality and modeling assumptions, and there is no one-size-fits-all strategy.

3.1 Gene Expression Normalization

Current spatial transcriptomic techniques introduce unwanted technical artifacts. Raw data commonly exhibit spot-to-spot variation and high dropout rates, which may impact downstream analyses. Several normalization strategies have been created to address these challenges. Due to the similarity between spatial transcriptomics and scRNA-seq, many normalization methods for spatial transcriptomics data are inspired by scRNA-seq studies.

A widely-used normalization tool is scran, a method based on the summation of expression values and deconvolution of pooled size factors (Lun et al., 2016). In the first step, expression values of all cells in the data set are averaged to serve as a reference. The cells are then partitioned into different pools, where the summation of expression values in each pool is normalized against the reference to generate a pool-based size factor. A linear system can be constructed by repeating the above normalization over multiple pools. Finally, the normalized cell-based counterparts can be calculated by solving the linear system with standard least-squares methods, i.e., deconvolving the pool-based size factor to individual cells. By representing the individual cells with multiple pools of cells, scran is capable of avoiding estimation inaccuracy in the presence of stochastic zeroes and is robust to differentially expressed genes. Similar to scran, a number of methods adopt the global scale factor strategy, where one normalization factor is applied to each cell, and all genes in this cell share the same factor. When the relationship between transcript-specific expression and sequencing depth is not shared across genes, such strategy will likely lead to overcorrection for weakly and moderately expressed genes. To address the problem, Bacher et al. proposed SCnorm, a quantile-regression based method that can estimate the dependence of expression on sequencing depth for each gene (Bacher et al., 2017). Then genes are grouped based on the

similarity of dependence, and a second quantile regression is used to estimate a shared scale factor within each group.

Lytal et al. conducted an empirical survey to evaluate the effectiveness of seven single-cell normalization methods. Based on the experimental results over several real and simulated data sets, the study concludes that there is no "one-size-fits-all" normalization technique for every data set (Lytal et al., 2020). Further, Saiselet et al. investigated whether normalization is warranted for spatial transcriptomic datasets. They discovered that variation of total read counts is related to morphology and local cell density. Therefore, total counts per spot are biologically informative and do not necessarily need to be normalized out (Saiselet et al., 2020).

4 ANALYSIS AND VISUALIZATION IN THE EXPRESSION DOMAIN

A first step in the spatial transcriptomic analysis is to identify the cell type (for datasets of single-cell resolution) or cell mixture (for datasets of multicellular resolution) of each spatial unit or spot. Cell type identification usually starts with the dimensionality reduction technique to reduce time and space complexity for downstream analysis. The reduced representations are used to cluster cells based on the assumption that cells of the same type fall into the same cluster.

4.1 Clustering

The selection of clustering techniques is critical for obtaining good clustering results. Certain methods with assumptions about cluster shapes may not be suitable for spatial genomic data. For example, K-means clustering assumes that the shapes of clusters are spherical and that clusters are of similar size (Kanungo et al., 2002), and Gaussian mixture models assume that points with each cluster follow a Gaussian distribution (Reynolds, 2009). These assumptions are rarely satisfied by spatial transcriptomic data.

Agglomerative clustering methods are a class of methods that iteratively aggregate data points into clusters. These methods do not carry assumptions about the shape and size of clusters. At each iteration, data points are aggregated to optimize a pre-defined metric. Popular agglomerative clustering methods include hierarchical agglomerative clustering (Johnson, 1967) and community detection methods such as Louvain (Blondel et al., 2008) and Leiden (Traag et al., 2019) algorithms. Hierarchical agglomerative clustering is initialized by treating each point as its own cluster. Each iteration aggregates two clusters with the closest distance to form a new cluster until no clusters can be merged. Community detection methods, i.e., Louvain (Blondel et al., 2008) and Leiden (Traag et al., 2019) algorithms, have seen wide adoption in the single-cell and the spatial transcriptomics community. Both algorithms try to iteratively maximize the modularity, which can be understood as the difference between the number of observed and expected edges. Intuitively, a tightly connected community or cluster should have a large number of observed edges relative to the expected number of edges. The Louvain algorithm is initialized by assigning each node to its own community. At each iteration, each node moves from its own community to all neighboring communities, and changes in modularity are calculated. The node is moved to the community, which results in the largest increase in H. At the end of each iteration, a new network is built by aggregating all nodes within the same community, and a new iteration begins. The procedure will terminate when the increase in H can no longer be achieved.

These general-purpose methods can be combined into more sophisticated pipelines tailored towards single-cell clustering. SC3 is an ensemble clustering method in which multiple clustering outcomes are merged into a consensus. SC3 first calculates distance matrices using the Euclidean distance, as well as Pearson and Spearman correlations. Spectral clustering is performed on these distance matrices with a varying number of eigenvectors. These results were combined to assign a consensus cluster membership to each point (Kiselev et al., 2017). Seurat uses a smart local moving (SLM) algorithm (Waltman and Van Eck, 2013) to perform modularity-based clustering. Seurat first constructs a distance matrix based on canonical correlation vectors and a shared nearest neighbor (SNN) graph based on the distance matrix. The SNN graph is used as an input to the SLM algorithm to find clusters (Butler et al., 2018). SIMLR calculates a distance matrix as a weighted sum of multiple distance kernels and solves for a similarity matrix to minimize the product between the distance and similarity matrices. To ensure a fixed number of connected components, SIMLR uses constrained optimization to encourage a block diagonal structure in the similarity matrix (Wang et al., 2017).

4.2 Identification of Cell Types

Identification of cell types starts by defining cell-type specific genes or marker genes. A straightforward approach is to perform differential expression analysis (McCarthy et al., 2012; Love et al., 2014) between all pairs of clusters. Genes that are consistently over-expressed in one cluster are considered the cluster's marker genes. This is the approach implemented in scran (Lun et al., 1000) and Mast (Finak et al., 2015).

Another method, scGeneFit, uses a label-aware compression method to find marker genes (Dumitrascu et al., 2021). Given cell-by-gene expression matrix and corresponding cell labels inferred from clustering results, scGeneFit finds a projection onto a lower-dimensional space, in which cells with the same labels are closer in the lower-dimensional space than cells with different labels. The projection is constrained such that the axes in the lower-dimensional space align with a single gene. Therefore, the marker genes will be the set of axes in the lower-dimensional space that best conserves label structures. The marker genes can then be matched with an expert-curated list of cell-type specific genes to infer cell types (Kim and Volsky, 2005; Subramanian et al., 2005). Other methods directly map unknown cell types onto a reference dataset, bypassing the target gene identification step. Scmap projects the query cells onto the reference cell types from other experiments and datasets (Kiselev et al., 2018). The known reference cluster is represented by its centroid, and the

projection is carried out by a fast approximate k-nearest-neighbor (KNN) search by cluster using product quantization (Jégou et al., 2010), where a similarity matrix between the query cell and reference clusters is used as the distance in KNN search. Another reference-based method is SingleR (Aran et al., 2019). The method proceeds by first identifying variable genes among cell types in the reference set. Next, SingleR calculates the Spearman correlation between each single cell and the reference variable genes. Multiple correlation coefficients within each cell type are aggregated to form one correlation per reference cell type per single cell. Only the top 80% of correlation values are selected to remove random noise. In the fine-tuning step, the correlation analysis is iteratively re-run but only for the top cell types from the previous step, and the lowly correlated cell types are removed. Eventually, the cell type with the top correlation is assigned to the query single-cell. Using SingleR, the authors identified a novel disease-associated macrophage subgroup between monocyte-derived and alveolar macrophages. Cortal et al. proposed a clustering-free multivariate statistical method named Cell-ID for gene signature extraction and cell identification (Cortal et al., 2021). Cell-ID first performs a dimensionality reduction on the cell-by-gene expression matrix using the multiple correspondence analysis (MCA). Both cells and genes are simultaneously projected in a common low-dimensional space, where the distance between a gene and a cell represents the specific degree between them. According to the distance, Cell-ID can build up a gene-rank for each cell, and the top-ranked genes are defined as the cell's gene signature. With the gene signature of the query cell, Cell-ID can perform automatic cell type and functional annotation via the hypergeometric tests against reference marker gene lists and/or gene signatures of reference single-cell datasets. The authors demonstrated the consistently reproducible gene signatures across diverse benchmarks, which helps to improve biological interpretation at the individual cell level. Unlike the above approaches, JSTA uses deep learning for cell-type identification and incorporates three distinct and interactive components: a segmentation map and two deep neural network-based cell type classifiers for pixel-level and cell-level classification (Littman et al., 2021). JSTA first trains a taxonomy-based cell-level classifier with the external data from the Neocortical Cell Type Taxonomy (NCTT) set (Yuste et al., 2020). Then the segmentation map and pixel-level classifier are iteratively refined with an expectation-maximization (EM) algorithm. Specifically, the segmentation map is initialized by a classical image segmentation algorithm watershed (Roerdink and Meijster, 2000) and paired with the trained cell-level type classifier to predict the current cell (sub)types. Given the local mRNA density at each pixel as the input, the pixel-level classifier is optimized to closely match each pixel's current cell type assignment. Next, the updated pixel-level classifier reclassifies the cell types of all border pixels, and the resulting segmentation map requires an update of the cell-level classification, which further triggers an update of pixel-level classifier training. This learning process is repeated until convergence. The eventual segmentation map tends to maximize consistency between local RNA density and cell-type expression priors. Abdellaal et al. benchmarked 22 broadly used cell identification methods on 27 publicly available single-cell RNA data. Interested readers are referred to (Abdelaal et al., 2019).

4.3 Visualization of Gene Expression in Low Dimensions

The identified clusters can be visualized to ensure cells assigned to the same cluster are close in expression space. Dimensionality reduction techniques are necessary to project the high dimensional data into 2D or 3D. Principal component analysis (PCA) is widely adopted in the single-cell and spatial transcriptomic literature (Wold et al., 1987). This method identifies linear combinations of the original dimensions, or principal components (PC), that maximize the projection variance from data points onto the principal components (**Figure 2A**). The principal components can be computed in an iterative way: the first PC can point in any direction to maximize the variance of projections, and each subsequent PC is orthogonal to previous PCs (Tsuyuzaki et al., 2020).

In contrast to PCA, manifold learning is a class of non-linear dimensionality reduction techniques that aims to project the data to a lower dimension while maintaining the distance relations in the original high-dimension space; points close to each other in the original space will be close in the low-dimensional space (**Figure 2B**). Uniform manifold approximate and projection (UMAP) and t-distributed stochastic neighbor embedding (t-SNE) are two manifold learning methods widely adopted in single-cell and spatial transcriptomic literature (Van der Maaten and Hinton, 2008; McInnes et al., 2018). Both methods follow a two-step procedure. In the first step, a similarity matrix is computed based on a pre-defined distance metric. In the second step, all data points are placed in a low-dimensional Euclidean space such that the structure of the similarity matrix is preserved. This step is initialized by randomly placing data points in the low-dimensional space. At each iteration, data points are moved according to the similarity matrix from the high-dimensional space; points with high similarity in the high-dimensional space will attract, and those with low similarity will repel. Because optimization is done iteratively, UMAP and t-SNE results are stochastic and vary between runs. Random seeds are needed for reproducibility. The two methods differ in their construction of similarity matrix. In t-SNE, a distance matrix is calculated according to probability density functions (PDF) of the Gaussian distribution in the high-dimension space and PDFs of the t-distribution in the low-dimension embedding. In UMAP, an adjacency matrix is constructed by extending a sphere whose radius depends on the local density of nearby points; two points are connected if their spheres overlap. In practice, UMAP is faster than t-SNE and tends to preserve the high-dimensional structure better.

5 ANALYSIS AND VISUALIZATION IN THE SPATIAL DOMAIN

An important question in spatial transcriptomic data analysis is to identify genes whose expression follow coherent spatial

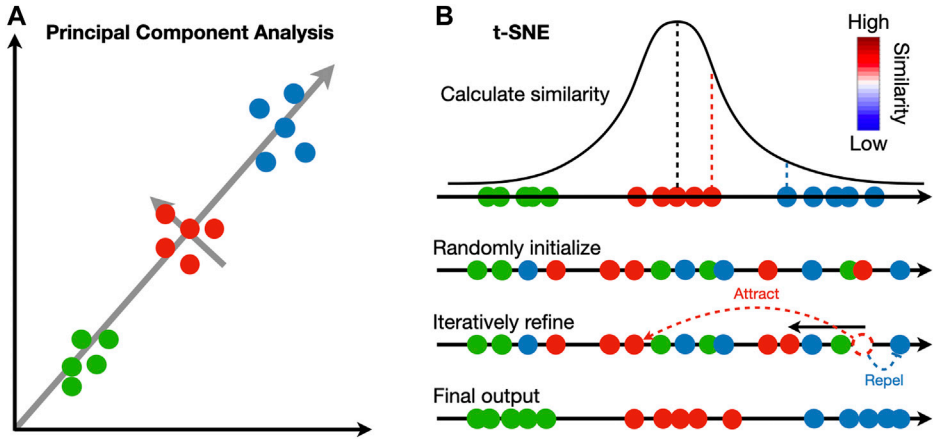

FIGURE 2 | Comparison between principal component analysis and t-SNE. **(A)** Principal component analysis iteratively identifies vectors that minimize the sum of squared distances to the direction of the vector. Each vector is orthogonal to all previously selected vectors. **(B)** t-SNE calculates a pairwise similarity based on the probability density function of the Gaussian distribution in the original high dimension space. The points are randomly projected to a low dimensional space and iteratively refined so that the similarity in low dimension matches that in high dimension. At each iteration, similar pairs attract, and dissimilar pairs repel each other.

patterns. Genes with spatial expression patterns are critical determinants of polarity and anatomical structures. For example, the gene *wingless* is a member of the *wnt* family that plays a central role in anterior-posterior pattern generation during the embryonic development of *Drosophila melanogaster*. It is expressed in alternating stripes across the entire embryo (van den Heuvel et al., 1989). Another example is the neocortex of mammalian brains, which contain six distinct layers. Each layer consists of different types of neurons and glial cells that express cell-type specific marker genes (Lui et al., 2011). Spatial transcriptomic data enables unbiased transcriptome-wide identification of spatially expressed genes, but it is excessively labor-intensive to visually examine all genes. This prompted the development methods including SpatialDE (Svensson et al., 2018), trendsceek (Edsgärd et al., 2018), and Spark (Sun et al., 2020).

5.1 Identification of Genes with Spatial Expression Patterns

SpatialDE (Svensson et al., 2018) uses a Gaussian process to model gene expression levels. Intuitively, a Gaussian process model treats all data points as observations from a random variable that follows a multivariate Gaussian (MVN) distribution (Wang, 2020). To test whether expression levels follow a spatial pattern, the authors specify a null model, in which the covariance matrix is diagonal, and an alternative model, in which the covariance matrix follows a radial basis function kernel:

$$K\left(x_i, x_j\right) = \exp\left(-\gamma \|x_i - x_j\|^2\right) \tag{1}$$

where $K(x_i, x_j)$ is the covariance between ith and jth measurement; x_i and x_j represent the spatial coordinates of the ith and the jth measurement; γ is a scale factor. Intuitively, the Gaussian kernel describes a spatial relationship in which nearby points have similar expression values. This kernel assumes that cells of similar origins tend to neighbor each other in space. A likelihood ratio test can be done by comparing the likelihood of the null and the alternative model. Because SpatialDE is a Gaussian process model, the expression values must be log-transformed which decreases power.

Trendsceek (Edsgärd et al., 2018) uses a marked point process model in which each point of measurement, or a spot, is treated as a point process, and each point is marked with a gene expression value. To decide whether a gene whose expression follows a spatial pattern, trendsceek test whether the probability of finding two marks given the distance between two points deviates from what would be expected if the marks were randomly distributed over points. To calculate the null distribution given no spatial pattern, trendsceek implements a sampling procedure in which marks are permuted with the location of points fixed. In practice, such sampling procedure is computationally expensive and makes trendsceek only suitable for small datasets.

Spark (Sun et al., 2020) uses a generalized linear spatial model (GLSM) to directly model count data (McCullagh and Nelder, 1983; Gotway and Stroup, 1997), which results in better power than SpatialDE. A simplified model is presented below:

$$y(s) \sim Poisson(\lambda(s)) \tag{2}$$

$$\log(\lambda(s)) = x(s)^T\beta + b(s) + \epsilon \tag{3}$$

$$b(s) \sim MVN(0,\ \tau K(s)) \tag{4}$$

Where $y(s)$ is the gene expression of sample s. λ is a Poisson rate parameter, which is modeled as a linear combination of three terms. The first term $x(s)$ represents covariates such as batch effect and library size for sample s. The second term $b(s)$ is the spatial correlation pattern modeled as a Gaussian process. The last term ϵ is random noise. To determine whether a gene follows a spatial pattern, Spark tests whether $\tau = 0$. Parameter estimation is difficult due to the random effects. *Monte Carlo* methods are the gold standard for parameter estimation for GLSM but are

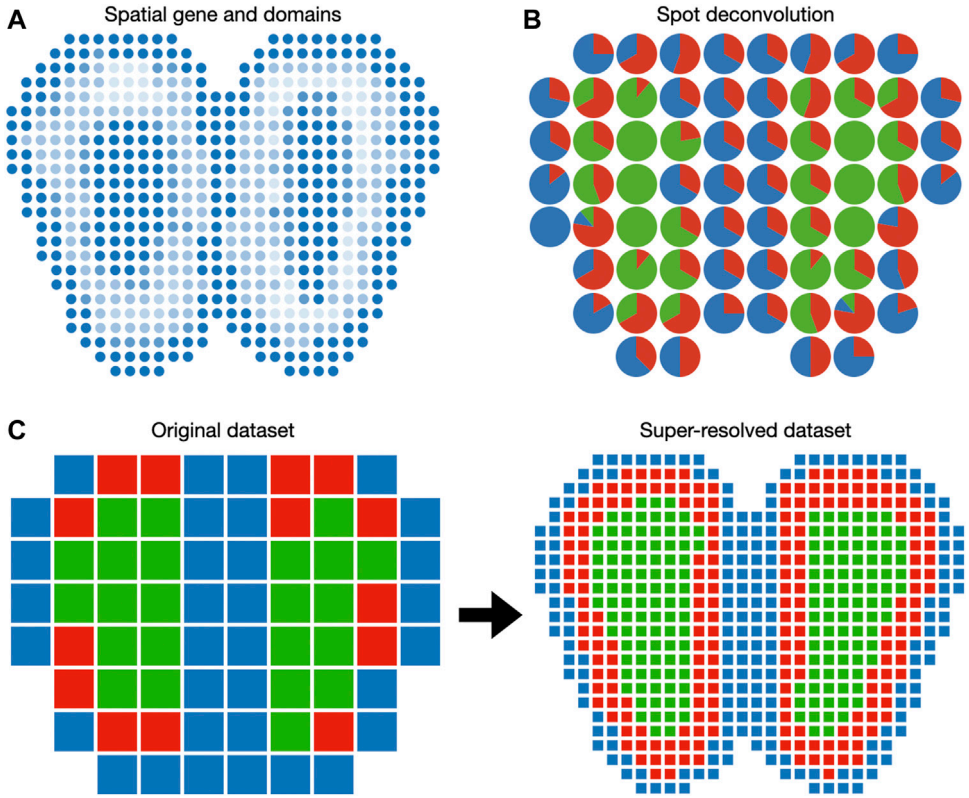

FIGURE 3 | Visualization of gene expression in the Euclidean space. **(A)** Spatially coherent genes and spatial domains can be visualized as 2D images. **(B)** Spot deconvolution methods estimate the proportion of each cell type within each spot. Pie charts are routinely used to represent cell type proportions within each spot. **(C)** Spot super-resolution methods estimate the cell type of sub-pixels based on correlation with neighbor spots. In this case, each spot of the original dataset is divided into nine spots in the super-resolved dataset.

computationally expensive. Instead, the authors developed a penalized quasi-likelihood (PQL) estimation procedure to make computation tractable for large datasets (Wedderburn, 1974; McCullagh and Nelder, 1983; Breslow and Clayton, 1993). Spark produces well-calibrated *p*-values and is more powerful than trendsceek and SpatialDE through a series of simulation experiments.

5.2 Identification of Spatial Domains

Spatially coherent domains often underly important anatomical regions (**Figure 3A**). A motivating example is the histological staining of cancer tissue slides. Cancer regions and normal tissues can be visually distinguished due to differential affinities to staining agents. This enables pathologists to grade and stage individual cancer tissue slides according to the location and size of the cancer regions (Fletcher, 2007). Spatial transcriptomics enables histology-like identification of spatial domains. Regular histology slides can be visualized conveniently with RGB pixels. In contrast, spatial transcriptomic data cannot be directly visualized because each spot (i.e., pixel) in spatial transcriptomic data has a dimension equal to the number of genes. This prompts the development of methods to detect spatial domains, including BayesSpace (Zhao et al., 2021), SpatialDE (Svensson et al., 2018), and a hidden Markov random field (HMRF) method (Zhu et al., 2018).

The three methods share a common assumption that hidden spatial domains can be described by latent variables, which are not directly observed but can be inferred from observed gene expression values. However, these methods use different modeling assumptions to infer latent variables. Zhu et al. (Zhu et al., 2018) developed an HMRF-based method, a widely adopted model in the image processing community to identify patterns in 2D images (Li, 2000; Blake et al., 2011), to identify spatial domains. An HMRF has two components: it uses a Markov random field to describe the joint distribution of latent variables and a set of observed examples that depends on them. The latent variables are assumed to satisfy the Markov property, in which any node in the network is conditionally independent of other nodes given its neighbors. Following this assumption, a Markov random field of latent variables can be decomposed into a set of subgraphs, called cliques, which gives rise to the observed gene expression. The parameters of the model by Zhu et al. are estimated with an EM algorithm (Dempster et al., 1977; Moon, 1996).

Both SpatialDE (Svensson et al., 2018) and BayesSpace (Zhao et al., 2021) model observed gene expression values as a mixture of Gaussian random variables. The means of the Gaussian random variables are determined by the spatial domain membership. In SpatialDE, the mean expression value of each spatial domain is described by a Gaussian process, whose

covariance follows a radial basis function kernel. The observed expression follows a Gaussian distribution centered around the mean expression value of a given spatial domain. The posterior distribution of parameters and the latent spatial domain membership is estimated by variational inference. Different from SpatialDE, BayesSpace uses a diagonal matrix to model the covariance of the mean expression of each spatial domain. The observed gene expression is modeled as a Gaussian random variable centered around the mean expression and has a diagonal covariance matrix modeled as a Wishart random variable. BayesSpace uses a Markov chain Monte Carlo (MCMC) method to estimate model parameters (Geyer, 1992).

While the above methods consider spatial genes and spatial domain as two separate tasks, SpaGCN proposed a graph convolutional network-based (GCN) approach to address these two tasks jointly (Hu et al., 2021b). With the integration of gene expression, spatial location, and histology information, SpaGCN models spatial dependency of gene expression for clustering analysis of spatial domains and identification of domain enriched spatial variable genes (SVG) or meta genes. SpaGCN first converts the spatial transcriptomics data into an undirected weighted graph of spots, and the graph structure represents the spatial dependency of the data. Next, a GCN (Kipf and Welling, 2016) is utilized to aggregate gene expression information from the neighboring spots and update every spot's representation. Then, SpaGCN adopts an unsupervised clustering algorithm (Xie et al., 2016) to cluster the spots iteratively, and each identified cluster will be considered as a spatial domain. The resulting domains guide the differential expression analysis to detect the SVG or meta genes with enriched expression patterns in the identified domains.

5.3 Spot Deconvolution and Super-resolution

Because spots in the spatial transcriptomic dataset may not correspond to cell boundaries, several additional features can be included when plotting on the spatial domain. When the spatial transcriptomic measurement technology has a multicellular resolution, spots can be decomposed into constituent cell types. A 2D array of pie charts can be used to represent the cell types' percentages of spots, as demonstrated in DSTG (**Figure 3B**). To enable the investigation of cellular architecture at higher resolution, DSTG uses a GCN to uncover the cellular compositions within each spot (Song and Su, 2021). DSTG first leverages single-cell RNA-seq data to construct pseudo spatial transcriptomic (pseudo-ST) data by selecting two to eight single cells from the same tissue and combining their transcriptomic profiles. This pseudo-ST data is designed to mimic the cell mixture in the real spatial transcriptomic data and provide the basis for model training. Via canonical correlation analysis, DSTG identifies a link graph of spots with the integration of the pseudo-ST data and the real spatial transcriptomic data. A GCN (Kipf and Welling, 2016) iteratively updates the representation of each spot by aggregating its neighborhoods' information. The GCN model is trained in a semi-supervised manner, where the known cell compositions of the pseudo-ST nodes are served as the labeled data, and the real spatial transcriptomic nodes are the prediction targets. The resulting cell type proportions can be displayed as a pie chart at each spot (**Figure 3B**).

While cell type deconvolution provides an estimation of cellular constituents, it does not directly increase the resolution of the dataset. BayesSpace uses a Bayesian model to increase the resolution to the subspot level, which approaches single-cell resolution with the Visium platform (**Figure 3C**). The model specification is similar to the spatial domain detection model described above, except that unit of analysis is the subspot rather than the spot. Since gene expression is not observed at the subspot level, BayesSpace models it as another latent variable and estimates it using MCMC. The increase in resolution is different across measurement technology. For square spots (Ståhl et al., 2016), BayesSpace by default divides each spot into nine subspots. For hexagonal spots like Visium, they are divided into six subspots by default. The subspots can be visualized in Euclidean space similar to regular spots.

5.4 Visualization in Euclidean Space

After obtaining spatial genes and domains, visualization in the Euclidean space is relatively straightforward. Spatial genes can be visualized by plotting their log-transformed expression values. Spatial domains can be colored by mean expression values or by their identities. Several packages such as Giotto (Dries et al., 2021b), Scanpy (Wolf et al., 2018), Seurat (Hao et al., 2021), and Squidpy (Palla et al., 2021) provide functionalities to plot spatial transcriptomic data in Euclidean space.

6 ANALYSIS AND VISUALIZATION IN THE INTERACTION DOMAIN

Cell signaling describes the process in which cells send, receive, process, and transmit signals within the environment and with themselves. Based on the signaling distance and the sender-receiver identities, cell signaling can be classified into autocrine, paracrine, endocrine, intracrine, and juxtacrine (Bradshaw and Dennis, 2009). It serves critical functions in development (Wei et al., 2004), immunity (Dustin and Chan, 2000), and homeostasis (Taguchi and White, 2008) across all organisms. For example, the Hedgehog signaling pathway is involved in tissue patterning and orientation, and aberrant activations of hedgehog signaling lead to several types of cancers (Taipale and Beachy, 2001). Single-cell datasets enable correlation analysis to unravel cell-to-cell interaction (Krishnaswamy et al., 2014; Friedman et al., 2018; Wirka et al., 2019). Due to the lack of spatial information, single-cell analysis cannot distinguish short-distance (juxtacrine and paracrine) and long-distance (endocrine) signaling. Spatial transcriptomic datasets provide the spatial coordinate of each cell or spot and enable spatial dissection of cell signaling.

6.1 Cell-to-Cell Interaction

Cell signaling frequently occurs between cells in spatial proximity. Giotto takes spatial proximity into consideration to

identify cell-to-cell interaction. It first constructs a spatial neighborhood network to identify cell types that occur in spatial proximity. Each node of the network represents a cell, and pair of neighboring cells are connected through an edge. The neighbors of each cell can be determined by extending a circle of a predefined radius, selecting the k-nearest neighbors, or constructing a Delaunay network (Chen and Xu, 2004). Cell types connected in the network more than expected are considered interacting. Giotto permutes the cell type labels without changing the topology of the network and calculates the expected frequencies between every pair of cell types. *p*-values are derived based on where the observed frequency falls on the distribution of expected frequencies.

Another method, SpaOTsc, leverages both single-cell and spatial transcriptomic data for a comprehensive profile of spatial interaction (Cang and Nie, 2020). It uses an optimal transport algorithm to map single-cell to spatial transcriptomic data. An optimal transport is a function that maps a source distribution to a target distribution while minimizing the amount of effort with respect to a predefined cost function (Villani, 2009). SpaOTsc generates a cost function based on the expression profile dissimilarity of shared genes across the single-cell and the spatial transcriptomic datasets. The optimal transport plan maps single cells onto spatial locations. SpaOTsc then formulates cell-to-cell communication as a second optimal transport problem between sender and receiver cells. The expression of ligand and receptor genes are used to estimate sender and receiver cells, and the spatial distance is the cost function. The resultant optimal transport plan represents the likelihood of cell-to-cell communication.

6.2 Ligand-Receptor Pairing

Another aspect of cell signaling is the pairings between ligands and receptors. Giotto identifies ligand-receptor pairs whose mean expression is higher than expected. To obtain the observed expression of ligand-receptor pairs for a pair of cell types, Giotto averages the expression of ligand in all sender cells and the expression of receptors in all receiver cells in proximity of the sender cells. Giotto then permutes the location of cells to obtain an expected expression of the ligand-receptor pair. A *p*-value can be obtained by mapping the observed expression onto the distribution of expected expression. Different from Giotto, SpaOTsc uses a partial information decomposition (PID) approach to determine gene-to-gene interaction. Intuitively, PID decomposes the mutual information between multiple source variables and a target variable into unique information contributed by each source variable, redundant information shared by many source variables, and synergistic information contributed by the combination of source variables (Kunert-Graf et al., 2020). SpaOTsc estimates the unique information from a source gene to a target gene that is within a predefined spatial distance, taking into consideration all other genes. Yuan *et al.* proposed a method called GCNG (Yuan and Bar-Joseph, 2020) to infer the extracellular gene relationship using Graph convolutional neural networks (GCN). Single-cell spatial expression data is represented as a graph of cells. Cell locations are encoded as a binary cell adjacency matrix with a selected distance threshold, and expression of gene pairs within each cell is encoded as corresponding node features. A GCN is used to combine the graph structure and node information as input and predict whether the studied gene pair can interact. The deep learning model is trained in a supervised manner, where positive samples are built from known ligand-receptor pairs, and negative samples are randomly selected from non-interacting genes. In addition to the methods specifically designed to utilize the spatial expression information for cell-to-cell interaction, many other tools developed for expression data without spatial information can also be applied to the spatial transcriptomic data. Interested readers can refer to a recent review of these methods (Armingol et al., 2021).

6.3 Visualization of Interactions Between Cells and Genes

Cell-to-cell and gene-to-gene interactions are naturally represented as networks and correlation matrices (**Figure 4**). Integrative packages such as Giotto (Dries et al., 2021b) provide functions to visualize cell-to-cell and gene-to-gene interactions as heatmaps, dot plots, or networks. A heatmap is a visual depiction of a matrix whose values are represented as colored boxes on a grid. With heatmaps, large blocks of highly connected cells or genes can be visually identified. A dot plot is similar to a heatmap, except that the boxes are replaced by dots of varying sizes. A dot plot can use both the size and the color of each dot to represent values in each interaction. Different from heatmaps and dot plots, networks use nodes to represent cells or genes and edges to represent their interactions. The widths and colors of edges can be used to describe the strength of interactions. Besides Giotto, the igraph package is widely adopted for network visualization and provides programming interfaces in R, python, C/C++, and Mathematica (Csardi and Nepusz, 2006). Cytoscape is another widely used package to visualize complex network interaction. Its graphical user interface makes it easy to manipulate and examine nodes and edges in the network (Shannon et al., 2003).

7 DISCUSSIONS

Spatial transcriptomic technologies have made tremendous progress in recent years. Although earlier technologies are restricted by the number of profiled genes (Chen et al., 2015; Moffitt et al., 2016; Shah et al., 2016) or the spatial resolution (Ståhl et al., 2016), current methods can profile the whole transcriptome at single-cell or subcellular resolution (Liu et al., 2020; Chen et al., 2021; Cho et al., 2021). While available commercialized methods (Visium) cannot achieve cellular resolution, we believe newer technologies will soon be production-ready. As commercial platforms become more affordable, we believe the speed at which spatial transcriptomic datasets become publicly available will only accelerate. For example, phase two of the Brain Initiative Cell Census Network (BICCN) will map the spatial organization of more than 300,000 cells from the mouse's primary motor cortex (Marx, 2021). Large-scale projects to comprehensively profile

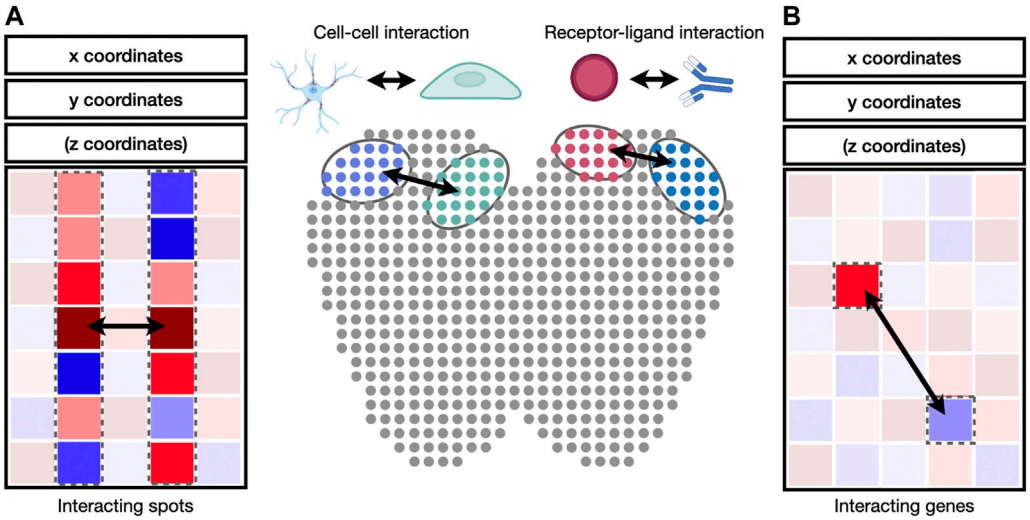

FIGURE 4 | Identification of cell-cell and receptor-ligand interaction. **(A)** Cell-to-cell interaction can be identified by the correlation of gene expression values between cell pairs. **(B)** Receptor-ligand interaction can be identified by the correlation between genes in interacting cell types.

spatial gene expression are currently limited, but we envision that these projects will expand in three directions. First, more model organisms will be profiled, enabling comparative analysis of cell types and their spatial organizations across evolution. Second, more organ and tissue types will be profiled for a comprehensive understanding of spatial expression architecture. Third, cell states (e.g., stimulated vs resting) and disease states (cancer vs normal) will be profiled to understand cellular activation and disease pathology.

As spatial transcriptomic datasets become more abundant, meta-analysis across published datasets will become commonplace. Methods to remove batch effects are needed to account for technical confounders across datasets. Unlike bulk and single-cell sequencing, batch effects in spatial transcriptomic data must account for correlation across space. Further, the batch effect may also occur on companion histology images, and methods to jointly analyze histology image and spatial transcriptomic data are required. Although several methods have been developed for batch effect removal in bulk (Leek et al., 2012; Stegle et al., 2012) and single-cell (Korsunsky et al., 2019; Li et al., 2020) sequencing, it is still an under-explored area for spatial transcriptomics.

Histopathology is widely adopted across various domains of medicine and is considered the gold standard for certain diagnoses such as cancer staging (Edge et al., 2010). However, histology is limited by the type and number of cellular features delineated by staining agents. Spatial transcriptomics extends histology to test for both imaging and molecular features and may enable testing for oncogenic driver mutations critical for determining cancer subtypes. A recent method named SpaCell integrates both histology and spatial transcriptomic information to predict cancer staging (Tan et al., 2020). In this method, histological images are tiled into patches, where each patch corresponds to a spatial transcriptomic spot in a tissue. A convolutional neural network is used to extract image features from each patch, and combine the features with the spot gene count. A subsequent deep network is applied to predict the disease stages. We envision that spatial transcriptomics will become a diagnostic routine as it becomes more affordable and the clinical interpretation becomes more streamlined.

In this review, we surveyed state-of-the-art methods for spatial transcriptomic data analysis and visualization, and categorized them into three main categories according to the way their output is visualized. It is unlikely that we covered all available methods for spatial transcriptomics, but we hope this review will serve as a stepping stone and attract more researchers to this field.

AUTHOR CONTRIBUTIONS

BL conceived the project. BL and YL performed literature review. BL and YL wrote the paper. LZ made the illustrations.

REFERENCES

Abdelaal, T., Michielsen, L., Cats, D., Hoogduin, D., Mei, H., Reinders, M. J. T., et al. (2019). A Comparison of Automatic Cell Identification Methods for Single-Cell RNA Sequencing Data. *Genome Biol.* 20 (1), 194–219. doi:10.1186/s13059-019-1795-z

Aird, D., Ross, M. G., Chen, W.-S., Danielsson, M., Fennell, T., Russ, C., et al. (2011). Analyzing and Minimizing PCR Amplification Bias in Illumina Sequencing Libraries. *Genome Biol.* 12 (2), R18. doi:10.1186/gb-2011-12-2-r18

Aran, D., Looney, A. P., Liu, L., Wu, E., Fong, V., Hsu, A., et al. (2019). Reference-based Analysis of Lung Single-Cell Sequencing Reveals a Transitional Profibrotic Macrophage. *Nat. Immunol.* 20 (2), 163–172. doi:10.1038/s41590-018-0276-y

Armingol, E., Officer, A., Harismendy, O., and Lewis, N. E. (2021). Deciphering Cell-Cell Interactions and Communication from Gene Expression. *Nat. Rev. Genet.* 22 (2), 71–88. doi:10.1038/s41576-020-00292-x

Asp, M., Giacomello, S., Larsson, L., Wu, C., Fürth, D., Qian, X., et al. (2019). A Spatiotemporal Organ-wide Gene Expression and Cell Atlas of the Developing Human Heart. *Cell* 179 (7), 1647–1660. doi:10.1016/j.cell.2019.11.025

Bacher, R., Chu, L.-F., Leng, N., Gasch, A. P., Thomson, J. A., Stewart, R. M., et al. (2017). SCnorm: Robust Normalization of Single-Cell RNA-Seq Data. *Nat. Methods* 14 (6), 584–586. doi:10.1038/nmeth.4263

Berglund, E., Maaskola, J., Schultz, N., Friedrich, S., Marklund, M., Bergenstråhle, J., et al. (2018). Spatial Maps of Prostate Cancer Transcriptomes Reveal an Unexplored Landscape of Heterogeneity. *Nat. Commun.* 9 (1), 2419. doi:10.1038/s41467-018-04724-5

Blake, A., Kohli, P., and Rother, C. (2011). *Markov Random fields for Vision and Image Processing*. MIT press.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast Unfolding of Communities in Large Networks. *J. Stat. Mech.* 2008 (10), P10008. doi:10.1088/1742-5468/2008/10/p10008

Bradshaw, R. A., and Dennis, E. A. (2009). *Handbook of Cell Signaling*. Academic Press.

Breslow, N. E., and Clayton, D. G. (1993). Approximate Inference in Generalized Linear Mixed Models. *J. Am. Stat. Assoc.* 88 (421), 9–25. doi:10.1080/01621459.1993.10594284

Butler, A., Hoffman, P., Smibert, P., Papalexi, E., and Satija, R. (2018). Integrating Single-Cell Transcriptomic Data across Different Conditions, Technologies, and Species. *Nat. Biotechnol.* 36 (5), 411–420. doi:10.1038/nbt.4096

Cang, Z., and Nie, Q. (2020). Inferring Spatial and Signaling Relationships between Cells from Single Cell Transcriptomic Data. *Nat. Commun.* 11 (1), 2084–2097. doi:10.1038/s41467-020-15968-5

Chen, A., Liao, S., Cheng, M., Ma, K., Wu, L., Lai, Y., et al. (2021). Large Field of View-Spatially Resolved Transcriptomics at Nanoscale Resolution. *bioRxiv*, 2021. doi:10.1101/2021.01.17.427004

Chen, K. H., Boettiger, A. N., Moffitt, J. R., Wang, S., and Zhuang, X. (2015). RNA Imaging. Spatially Resolved, Highly Multiplexed RNA Profiling in Single Cells. *Science* 348 (6233), aaa6090. doi:10.1126/science.aaa6090

Chen, L., and Xu, J-c. (2004). Optimal Delaunay Triangulations. *J. Comput. Maths.*, 299–308.

Chen, W.-T., Lu, A., Craessaerts, K., Pavie, B., Sala Frigerio, C., Corthout, N., et al. (2020). Spatial Transcriptomics and *In Situ* Sequencing to Study Alzheimer's Disease. *Cell* 182 (4), 976–991. doi:10.1016/j.cell.2020.06.038

Cho, C.-S., Xi, J., Si, Y., Park, S.-R., Hsu, J.-E., Kim, M., et al. (2021). Microscopic Examination of Spatial Transcriptome Using Seq-Scope. *Cell* 184 (13), 3559–3572. doi:10.1016/j.cell.2021.05.010

Close, J. L., Long, B. R., and Zeng, H. (2021). Spatially Resolved Transcriptomics in Neuroscience. *Nat. Methods* 18 (1), 23–25. doi:10.1038/s41592-020-01040-z

Cortal, A., Martignetti, L., Six, E., and Rausell, A. (2021). Gene Signature Extraction and Cell Identity Recognition at the Single-Cell Level with Cell-ID. *Nat. Biotechnol.* 39 (9), 1095–1124. doi:10.1038/s41587-021-00896-6

Crosetto, N., Bienko, M., and van Oudenaarden, A. (2015). Spatially Resolved Transcriptomics and beyond. *Nat. Rev. Genet.* 16 (1), 57–66. doi:10.1038/nrg3832

Csardi, G., and Nepusz, T. (2006). The Igraph Software Package for Complex Network Research. *InterJournal, complex Syst.* 1695 (5), 1–9.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via theEMAlgorithm. *J. R. Stat. Soc. Ser. B (Methodological)* 39 (1), 1–22. doi:10.1111/j.2517-6161.1977.tb01600.x

Dries, R., Zhu, Q., Dong, R., Eng, C-H. L., Li, H., Liu, K., et al. (2021). Giotto: a Toolbox for Integrative Analysis and Visualization of Spatial Expression Data. *Genome Biol.* 22 (1), 1–31. doi:10.1186/s13059-021-02286-2

Dries, R., Chen, J., Del Rossi, N., Khan, M. M., Sistig, A., and Yuan, G.-C. (2021). Advances in Spatial Transcriptomic Data Analysis. *Genome Res.* 31 (10), 1706–1718. doi:10.1101/gr.275224.121

Dumitrascu, B., Villar, S., Mixon, D. G., and Engelhardt, B. E. (2021). Optimal Marker Gene Selection for Cell Type Discrimination in Single Cell Analyses. *Nat. Commun.* 12 (1), 1186. doi:10.1038/s41467-021-21453-4

Dustin, M. L., and Chan, A. C. (2000). Signaling Takes Shape in the Immune System. *Cell* 103 (2), 283–294. doi:10.1016/s0092-8674(00)00120-3

Edge, S. B., Byrd, D. R., Carducci, M. A., Compton, C. C., Fritz, A., and Greene, F. (2010). *AJCC Cancer Staging Manual*. New York: Springer.

Edsgärd, D., Johnsson, P., and Sandberg, R. (2018). Identification of Spatial Expression Trends in Single-Cell Gene Expression Data. *Nat. Methods* 15 (5), 339–342. doi:10.1038/nmeth.4634

Eng, C.-H. L., Lawson, M., Zhu, Q., Dries, R., Koulena, N., Takei, Y., et al. (2019). Transcriptome-scale Super-resolved Imaging in Tissues by RNA seqFISH+. *Nature* 568 (7751), 235–239. doi:10.1038/s41586-019-1049-y

Finak, G., McDavid, A., Yajima, M., Deng, J., Gersuk, V., Shalek, A. K., et al. (2015). MAST: a Flexible Statistical Framework for Assessing Transcriptional Changes and Characterizing Heterogeneity in Single-Cell RNA Sequencing Data. *Genome Biol.* 16 (1), 278–291. doi:10.1186/s13059-015-0844-5

Fletcher, C. D. (2007). *Diagnostic Histopathology of Tumors: 2-volume Set with CD-ROMs*. Elsevier Health Sciences.

Friedman, C. E., Nguyen, Q., Lukowski, S. W., Helfer, A., Chiu, H. S., Miklas, J., et al. (2018). Single-Cell Transcriptomic Analysis of Cardiac Differentiation from Human PSCs Reveals HOPX-dependent Cardiomyocyte Maturation. *Cell Stem Cell* 23 (4), 586–598. doi:10.1016/j.stem.2018.09.009

Gawad, C., Koh, W., and Quake, S. R. (2016). Single-cell Genome Sequencing: Current State of the Science. *Nat. Rev. Genet.* 17 (3), 175–188. doi:10.1038/nrg.2015.16

Geyer, C. J. (1992). Practical Markov Chain Monte Carlo. *Stat. Sci.*, 473–483. doi:10.1214/ss/1177011137

Gotway, C. A., and Stroup, W. W. (1997). A Generalized Linear Model Approach to Spatial Data Analysis and Prediction. *J. Agric. Biol. Environ. Stat.* 2, 157–178. doi:10.2307/1400401

Hao, Y., Hao, S., Andersen-Nissen, E., Mauck, W. M., III, Zheng, S., Butler, A., et al. (2021). Integrated Analysis of Multimodal Single-Cell Data. *Cell*. doi:10.1016/j.cell.2021.04.048

Hu, J., Li, X., Coleman, K., Schroeder, A., Ma, N., Irwin, D. J., et al. (2021). SpaGCN: Integrating Gene Expression, Spatial Location and Histology to Identify Spatial Domains and Spatially Variable Genes by Graph Convolutional Network. *Nat. Methods* 18 (11), 1342–1351. doi:10.1038/s41592-021-01255-8

Hu, J., Schroeder, A., Coleman, K., Chen, C., Auerbach, B. J., and Li, M. (2021). Statistical and Machine Learning Methods for Spatially Resolved Transcriptomics with Histology. *Comput. Struct. Biotechnol. J.* 19, 3829–3841. doi:10.1016/j.csbj.2021.06.052

J. Xie, R. Girshick, and A. Farhadi (Editors) (2016). "Unsupervised Deep Embedding for Clustering Analysis," *International Conference on Machine Learning* (PMLR).

Jégou, H., Douze, M., and Schmid, C. (2010). Improving Bag-Of-Features for Large Scale Image Search. *Int. J. Comput. Vis.* 87 (3), 316–336.

Ji, A. L., Rubin, A. J., Thrane, K., Jiang, S., Reynolds, D. L., Meyers, R. M., et al. (2020). Multimodal Analysis of Composition and Spatial Architecture in Human Squamous Cell Carcinoma. *Cell* 182 (6), 1661–1662. doi:10.1016/j.cell.2020.08.043

Johnson, S. C. (1967). Hierarchical Clustering Schemes. *Psychometrika* 32 (3), 241–254. doi:10.1007/bf02289588

Junker, J. P., Noël, E. S., Guryev, V., Peterson, K. A., Shah, G., Huisken, J., et al. (2014). Genome-wide RNA Tomography in the Zebrafish Embryo. *Cell* 159 (3), 662–675. doi:10.1016/j.cell.2014.09.038

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., and Wu, A. Y. (2002). An Efficient K-Means Clustering Algorithm: Analysis and Implementation. *IEEE Trans. Pattern Anal. Machine Intell.* 24 (7), 881–892. doi:10.1109/tpami.2002.1017616

Kim, S. Y., and Volsky, D. J. (2005). PAGE: Parametric Analysis of Gene Set Enrichment. *BMC bioinformatics* 6 (1), 144–156. doi:10.1186/1471-2105-6-144

Kim, T. H., Zhou, X., and Chen, M. (2020). Demystifying "Drop-Outs" in Single-Cell UMI Data. *Genome Biol.* 21 (1), 196. doi:10.1186/s13059-020-02096-y

Kipf, T. N., and Welling, M. (2016). *Semi-supervised Classification with Graph Convolutional Networks* arXiv preprint arXiv:160902907.

Kiselev, V. Y., Kirschner, K., Schaub, M. T., Andrews, T., Yiu, A., Chandra, T., et al. (2017). SC3: Consensus Clustering of Single-Cell RNA-Seq Data. *Nat. Methods* 14 (5), 483–486. doi:10.1038/nmeth.4236

Kiselev, V. Y., Yiu, A., and Hemberg, M. (2018). Scmap: Projection of Single-Cell RNA-Seq Data across Data Sets. *Nat. Methods* 15 (5), 359–362. doi:10.1038/nmeth.4644

Korsunsky, I., Millard, N., Fan, J., Slowikowski, K., Zhang, F., Wei, K., et al. (2019). Fast, Sensitive and Accurate Integration of Single-Cell Data with Harmony. *Nat. Methods* 16 (12), 1289–1296. doi:10.1038/s41592-019-0619-0

Krishnaswamy, S., Spitzer, M. H., Mingueneau, M., Bendall, S. C., Litvin, O., Stone, E., et al. (2014). Conditional Density-Based Analysis of T Cell Signaling in Single-Cell Data. *Science* 346 (6213), 1250689. doi:10.1126/science.1250689

Kunert-Graf, J., Sakhanenko, N., and Galas, D. (2020). Partial Information Decomposition and the Information Delta: A Geometric Unification Disentangling Non-pairwise Information. *Entropy* 22 (12), 1333. doi:10.3390/e22121333

Lee, J. H., Daugharthy, E. R., Scheiman, J., Kalhor, R., Yang, J. L., Ferrante, T. C., et al. (2014). Highly Multiplexed Subcellular RNA Sequencing *In Situ*. *Science* 343 (6177), 1360–1363. doi:10.1126/science.1250212

Leek, J. T., Johnson, W. E., Parker, H. S., Jaffe, A. E., and Storey, J. D. (2012). The Sva Package for Removing Batch Effects and Other Unwanted Variation in High-Throughput Experiments. *Bioinformatics* 28 (6), 882–883. doi:10.1093/bioinformatics/bts034

Li, S. Z. (2000). *Modeling Image Analysis Problems Using Markov Random fields*.

Li, X., Wang, K., Lyu, Y., Pan, H., Zhang, J., Stambolian, D., et al. (2020). Deep Learning Enables Accurate Clustering with Batch Effect Removal in Single-Cell RNA-Seq Analysis. *Nat. Commun.* 11 (1), 2338–2352. doi:10.1038/s41467-020-15851-3

Littman, R., Hemminger, Z., Foreman, R., Arneson, D., Zhang, G., Gómez-Pinilla, F., et al. (2021). Joint Cell Segmentation and Cell Type Annotation for Spatial Transcriptomics. *Mol. Syst. Biol.* 17 (6), e10108. doi:10.15252/msb.202010108

Liu, Y., Yang, M., Deng, Y., Su, G., Enninful, A., Guo, C. C., et al. (2020). High-Spatial-Resolution Multi-Omics Sequencing via Deterministic Barcoding in Tissue. *Cell* 183 (6), 1665–1681. doi:10.1016/j.cell.2020.10.026

Love, M. I., Huber, W., and Anders, S. (2014). Moderated Estimation of Fold Change and Dispersion for RNA-Seq Data with DESeq2. *Genome Biol.* 15 (12), 550. doi:10.1186/s13059-014-0550-8

Lubeck, E., Coskun, A. F., Zhiyentayev, T., Ahmad, M., and Cai, L. (2014). Single-cell *In Situ* RNA Profiling by Sequential Hybridization. *Nat. Methods* 11 (4), 360–361. doi:10.1038/nmeth.2892

Lui, J. H., Hansen, D. V., and Kriegstein, A. R. (2011). Development and Evolution of the Human Neocortex. *Cell* 146 (1), 18–36. doi:10.1016/j.cell.2011.06.030

Lun, A. T., Bach, K., and Marioni, J. C. (2016). Pooling across Cells to Normalize Single-Cell RNA Sequencing Data with many Zero Counts. *Genome Biol.* 17 (1), 75–14. doi:10.1186/s13059-016-0947-7

Lun, A. T., McCarthy, D. J., and Marioni, J. C. (10002016). A Step-by-step Workflow for Low-Level Analysis of Single-Cell RNA-Seq Data with Bioconductor. *F1000Res* 5, 2122. doi:10.12688/f1000research.9501.2

Lytal, N., Ran, D., and An, L. (2020). Normalization Methods on Single-Cell RNA-Seq Data: an Empirical Survey. *Front. Genet.* 11, 41. doi:10.3389/fgene.2020.00041

Maniatis, S., Äijö, T., Vickovic, S., Braine, C., Kang, K., Mollbrink, A., et al. (2019). Spatiotemporal Dynamics of Molecular Pathology in Amyotrophic Lateral Sclerosis. *Science* 364 (6435), 89–93. doi:10.1126/science.aav9776

Marx, V. (2021). Method of the Year: Spatially Resolved Transcriptomics. *Nat. Methods* 18 (1), 9–14. doi:10.1038/s41592-020-01033-y

McCarthy, D. J., Chen, Y., and Smyth, G. K. (2012). Differential Expression Analysis of Multifactor RNA-Seq Experiments with Respect to Biological Variation. *Nucleic Acids Res.* 40 (10), 4288–4297. doi:10.1093/nar/gks042

McCullagh, P., and Nelder, J. A. (1983). *Generalized Linear Models*. Thames, Oxfordshire, England: Routledge.

McInnes, L., Healy, J., and Melville, J. (2018). *Umap: Uniform Manifold Approximation and Projection for Dimension Reduction*. arXiv preprint arXiv:180203426.

Metzker, M. L. (2010). Sequencing Technologies - the Next Generation. *Nat. Rev. Genet.* 11 (1), 31–46. doi:10.1038/nrg2626

Moffitt, J. R., Bambah-Mukku, D., Eichhorn, S. W., Vaughn, E., Shekhar, K., Perez, J. D., et al. (2018). Molecular, Spatial, and Functional Single-Cell Profiling of the Hypothalamic Preoptic Region. *Science* 362 (6416). doi:10.1126/science.aau5324

Moffitt, J. R., Hao, J., Wang, G., Chen, K. H., Babcock, H. P., and Zhuang, X. (2016). High-throughput Single-Cell Gene-Expression Profiling with Multiplexed Error-Robust Fluorescence *In Situ* Hybridization. *Proc. Natl. Acad. Sci. USA* 113 (39), 11046–11051. doi:10.1073/pnas.1612826113

Moncada, R., Barkley, D., Wagner, F., Chiodin, M., Devlin, J. C., Baron, M., et al. (2020). Integrating Microarray-Based Spatial Transcriptomics and Single-Cell RNA-Seq Reveals Tissue Architecture in Pancreatic Ductal Adenocarcinomas. *Nat. Biotechnol.* 38 (3), 333–342. doi:10.1038/s41587-019-0392-8

Moon, T. K. (1996). The Expectation-Maximization Algorithm. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=& arnumber=543975.

Nichterwitz, S., Chen, G., Aguila Benitez, J., Yilmaz, M., Storvall, H., Cao, M., et al. (2016). Laser Capture Microscopy Coupled with Smart-Seq2 for Precise Spatial Transcriptomic Profiling. *Nat. Commun.* 7 (1), 12139–12150. doi:10.1038/ncomms12139

Palla, G., Spitzer, H., Klein, M., Fischer, D., Schaar, A. C., Kuemmerle, L. B., et al. (2021). *Squidpy: A Scalable Framework for Spatial Single Cell Analysis*. Laurel Hollow, New York: bioRxiv.

Rao, A., Barkley, D., França, G. S., and Yanai, I. (2021). Exploring Tissue Architecture Using Spatial Transcriptomics. *Nature* 596 (7871), 211–220. doi:10.1038/s41586-021-03634-9

Reynolds, D. (2009). Gaussian Mixture Models. *Encyclopedia of biometrics* 741, 659–663. doi:10.1007/978-0-387-73003-5_196

Rödelsperger, C., Ebbing, A., Sharma, D. R., Okumura, M., Sommer, R. J., and Korswagen, H. C. (2021). Spatial Transcriptomics of Nematodes Identifies Sperm Cells as a Source of Genomic novelty and Rapid Evolution. *Mol. Biol. Evol.* 38 (1), 229–243. doi:10.1093/molbev/msaa207

Rodriques, S. G., Stickels, R. R., Goeva, A., Martin, C. A., Murray, E., Vanderburg, C. R., et al. (2019). Slide-seq: A Scalable Technology for Measuring Genome-wide Expression at High Spatial Resolution. *Science* 363 (6434), 1463–1467. doi:10.1126/science.aaw1219

Roerdink, J. B., and Meijster, A. (2000). The Watershed Transform: Definitions, Algorithms and Parallelization Strategies. *Fundamenta informaticae* 41 (12), 187–228. doi:10.3233/fi-2000-411207

Saiselet, M., Rodrigues-Vitória, J., Tourneur, A., Craciun, L., Spinette, A., Larsimont, D., et al. (2020). Transcriptional Output, Cell-type Densities, and Normalization in Spatial Transcriptomics. *J. Mol. Cel. Biol.* 12 (11), 906–908. doi:10.1093/jmcb/mjaa028

Schermelleh, L., Ferrand, A., Huser, T., Eggeling, C., Sauer, M., Biehlmaier, O., et al. (2019). Super-resolution Microscopy Demystified. *Nat. Cel Biol* 21 (1), 72–84. doi:10.1038/s41556-018-0251-8

Schwarzacher, T., and Heslop-Harrison, P. (2000). *Practical* in Situ *Hybridization*. BIOS Scientific Publishers Ltd.

Shah, S., Lubeck, E., Zhou, W., and Cai, L. (2016). *In Situ* Transcription Profiling of Single Cells Reveals Spatial Organization of Cells in the Mouse Hippocampus. *Neuron* 92 (2), 342–357. doi:10.1016/j.neuron.2016.10.001

Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., et al. (2003). Cytoscape: a Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Res.* 13 (11), 2498–2504. doi:10.1101/gr.1239303

Shapiro, E., Biezuner, T., and Linnarsson, S. (2013). Single-cell Sequencing-Based Technologies Will Revolutionize Whole-Organism Science. *Nat. Rev. Genet.* 14 (9), 618–630. doi:10.1038/nrg3542

Song, Q., and Su, J. (2021). DSTG: Deconvoluting Spatial Transcriptomics Data through Graph-Based Artificial Intelligence. *Brief. Bioinform.* 22 (5), bbaa414. doi:10.1093/bib/bbaa414

Ståhl, P. L., Salmén, F., Vickovic, S., Lundmark, A., Navarro, J. F., Magnusson, J., et al. (2016). Visualization and Analysis of Gene Expression in Tissue Sections by Spatial Transcriptomics. *Science* 353 (6294), 78–82. doi:10.1126/science.aaf2403

Stegle, O., Parts, L., Piipari, M., Winn, J., and Durbin, R. (2012). Using Probabilistic Estimation of Expression Residuals (PEER) to Obtain Increased Power and Interpretability of Gene Expression Analyses. *Nat. Protoc.* 7 (3), 500–507. doi:10.1038/nprot.2011.457

Stickels, R. R., Murray, E., Kumar, P., Li, J., Marshall, J. L., Di Bella, D. J., et al. (2021). Highly Sensitive Spatial Transcriptomics at Near-Cellular Resolution with Slide-seqV2. *Nat. Biotechnol.* 39 (3), 313–319. doi:10.1038/s41587-020-0739-1

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., et al. (2005). Gene Set Enrichment Analysis: A Knowledge-Based Approach for Interpreting Genome-wide Expression Profiles. *Proc. Natl. Acad. Sci.* 102 (43), 15545–15550. doi:10.1073/pnas.0506580102

Sun, S., Zhu, J., and Zhou, X. (2020). Statistical Analysis of Spatial Expression Patterns for Spatially Resolved Transcriptomic Studies. *Nat. Methods* 17 (2), 193–200. doi:10.1038/s41592-019-0701-7

Svensson, V., Teichmann, S. A., and Stegle, O. (2018). SpatialDE: Identification of Spatially Variable Genes. *Nat. Methods* 15 (5), 343–346. doi:10.1038/nmeth.4636

Taguchi, A., and White, M. F. (2008). Insulin-like Signaling, Nutrient Homeostasis, and Life Span. *Annu. Rev. Physiol.* 70, 191–212. doi:10.1146/annurev.physiol.70.113006.100533

Taipale, J., and Beachy, P. A. (2001). The Hedgehog and Wnt Signalling Pathways in Cancer. *Nature* 411 (6835), 349–354. doi:10.1038/35077219

Tan, X., Su, A., Tran, M., and Nguyen, Q. (2020). SpaCell: Integrating Tissue Morphology and Spatial Gene Expression to Predict Disease Cells. *Bioinformatics* 36 (7), 2293–2294. doi:10.1093/bioinformatics/btz914

Thrane, K., Eriksson, H., Maaskola, J., Hansson, J., and Lundeberg, J. (2018). Spatially Resolved Transcriptomics Enables Dissection of Genetic Heterogeneity in Stage III Cutaneous Malignant Melanoma. *Cancer Res.* 78 (20), 5970–5979. doi:10.1158/0008-5472.CAN-18-0747

Traag, V. A., Waltman, L., and van Eck, N. J. (2019). From Louvain to Leiden: Guaranteeing Well-Connected Communities. *Sci. Rep.* 9 (1), 5233. doi:10.1038/s41598-019-41695-z

Tsuyuzaki, K., Sato, H., Sato, K., and Nikaido, I. (2020). Benchmarking Principal Component Analysis for Large-Scale Single-Cell RNA-Sequencing. *Genome Biol.* 21 (1), 9. doi:10.1186/s13059-019-1900-3

van den Heuvel, M., Nusse, R., Johnston, P., and Lawrence, P. A. (1989). Distribution of the Wingless Gene Product in Drosophila Embryos: a Protein Involved in Cell-Cell Communication. *Cell* 59 (4), 739–749. doi:10.1016/0092-8674(89)90020-2

Van der Maaten, L., and Hinton, G. (2008). Visualizing Data Using T-SNE. *J. machine Learn. Res.* 9 (11).

Vickovic, S., Eraslan, G., Salmén, F., Klughammer, J., Stenbeck, L., Schapiro, D., et al. (2019). High-definition Spatial Transcriptomics for *In Situ* Tissue Profiling. *Nat. Methods* 16 (10), 987–990. doi:10.1038/s41592-019-0548-y

Villani, C. (2009). *Optimal Transport: Old and New*. Springer.

Waltman, L., and Van Eck, N. J. (2013). A Smart Local Moving Algorithm for Large-Scale Modularity-Based Community Detection. *The Eur. Phys. J. B* 86 (11), 1–14. doi:10.1140/epjb/e2013-40829-0

Wang, B., Zhu, J., Pierson, E., Ramazzotti, D., and Batzoglou, S. (2017). Visualization and Analysis of Single-Cell RNA-Seq Data by Kernel-Based Similarity Learning. *Nat. Methods* 14 (4), 414–416. doi:10.1038/nmeth.4207

Wang, J. (2020). *An Intuitive Tutorial to Gaussian Processes Regression*. arXiv preprint arXiv:200910862.

Wang, X., Allen, W. E., Wright, M. A., Sylwestrak, E. L., Samusik, N., Vesuna, S., et al. (2018). Three-dimensional Intact-Tissue Sequencing of Single-Cell Transcriptional States. *Science* 361 (6400). doi:10.1126/science.aat5691

Wedderburn, R. W. M. (1974). Quasi-Likelihood Functions, Generalized Linear Models, and the Gauss-Newton Method. *Biometrika* 61 (3), 439–447. doi:10.2307/2334725

Wei, C.-J., Xu, X., and Lo, C. W. (2004). Connexins and Cell Signaling in Development and Disease. *Annu. Rev. Cel Dev. Biol.* 20, 811–838. doi:10.1146/annurev.cellbio.19.111301.144309

Wirka, R. C., Wagh, D., Paik, D. T., Pjanic, M., Nguyen, T., Miller, C. L., et al. (2019). Atheroprotective Roles of Smooth Muscle Cell Phenotypic Modulation and the TCF21 Disease Gene as Revealed by Single-Cell Analysis. *Nat. Med.* 25 (8), 1280–1289. doi:10.1038/s41591-019-0512-5

Wold, S., Esbensen, K., and Geladi, P. (1987). Principal Component Analysis. *Chemometrics Intell. Lab. Syst.* 2 (1-3), 37–52. doi:10.1016/0169-7439(87)80084-9

Wolf, F. A., Angerer, P., and Theis, F. J. (2018). SCANPY: Large-Scale Single-Cell Gene Expression Data Analysis. *Genome Biol.* 19 (1), 15. doi:10.1186/s13059-017-1382-0

Yuan, Y., and Bar-Joseph, Z. (2020). GCNG: Graph Convolutional Networks for Inferring Gene Interaction from Spatial Transcriptomics Data. *Genome Biol.* 21 (1), 300–316. doi:10.1186/s13059-020-02214-w

Yuste, R., Hawrylycz, M., Aalling, N., Aguilar-Valles, A., Arendt, D., Armañanzas, R., et al. (2020). A Community-Based Transcriptomics Classification and Nomenclature of Neocortical Cell Types. *Nat. Neurosci.* 23 (12), 1456–1468. doi:10.1038/s41593-020-0685-8

Zhao, E., Stone, M. R., Ren, X., Guenthoer, J., Smythe, K. S., Pulliam, T., et al. (2021). Spatial Transcriptomics at Subspot Resolution with BayesSpace. *Nat. Biotechnol.* 39 (11), 1375–1384. doi:10.1038/s41587-021-00935-2

Zhu, Q., Shah, S., Dries, R., Cai, L., and Yuan, G. C. (2018). Identification of Spatially Associated Subpopulations by Combining scRNAseq and Sequential Fluorescence *In Situ* Hybridization Data. *Nat. Biotechnol.* 29. doi:10.1038/nbt.4260

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers in Genetics

BRIEF RESEARCH REPORT
published: 28 February 2022
doi: 10.3389/fgene.2022.785877

SSAM-lite: A Light-Weight Web App for Rapid Analysis of Spatially Resolved Transcriptomics Data

Sebastian Tiesmeyer[1], Shashwat Sahay[1], Niklas Müller-Bötticher[1], Roland Eils[1,2], Sebastian D. Mackowiak[1] and Naveed Ishaque[1]**

[1]*Digital Health Center, Berlin Institute of Health at Charité, Universitätsmedizin Berlin, Berlin, Germany,* [2]*Health Data Science Unit, Heidelberg University Hospital, Heidelberg, Germany*

The combination of a cell's transcriptional profile and location defines its function in a spatial context. Spatially resolved transcriptomics (SRT) has emerged as the assay of choice for characterizing cells *in situ*. SRT methods can resolve gene expression up to single-molecule resolution. A particular computational problem with single-molecule SRT methods is the correct aggregation of mRNA molecules into cells. Traditionally, aggregating mRNA molecules into cell-based features begins with the identification of cells via segmentation of the nucleus or the cell membrane. However, recently a number of cell-segmentation-free approaches have emerged. While these methods have been demonstrated to be more performant than segmentation-based approaches, they are still not easily accessible since they require specialized knowledge of programming languages and access to large computational resources. Here we present SSAM-lite, a tool that provides an easy-to-use graphical interface to perform rapid and segmentation-free cell-typing of SRT data in a web browser. SSAM-lite runs locally and does not require computational experts or specialized hardware. Analysis of a tissue slice of the mouse somatosensory cortex took less than a minute on a laptop with modest hardware. Parameters can interactively be optimized on small portions of the data before the entire tissue image is analyzed. A server version of SSAM-lite can be run completely offline using local infrastructure. Overall, SSAM-lite is portable, lightweight, and easy to use, thus enabling a broad audience to investigate and analyze single-molecule SRT data.

Keywords: spatial transcriptomics, web application, cell typing, *in situ* sequencing, *in situ* hybridization, spatially resolved transcriptomics

OPEN ACCESS

Edited by:
Meng Zhou,
Wenzhou Medical University, China

Reviewed by:
Paweł P. Łabaj,
Jagiellonian University, Poland
Valentine Svensson,
Independent researcher, Cambridge, MA, United States

****Correspondence:***
Sebastian Tiesmeyer
sebastian.tiesmeyer@bih-charite.de
Naveed Ishaque
naveed.ishaque@bih-charite.de

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

Received: *29 September 2021*
Accepted: *25 January 2022*
Published: *28 February 2022*

Citation:
Tiesmeyer S, Sahay S, Müller-Bötticher N, Eils R, Mackowiak SD and Ishaque N (2022) SSAM-lite: A Light-Weight Web App for Rapid Analysis of Spatially Resolved Transcriptomics Data. Front. Genet. 13:785877. doi: 10.3389/fgene.2022.785877

1 INTRODUCTION

The biological function of a cell is governed not only by its expression profile but also by its location (Lee, 2017). A cell's spatial embedding defines its cellular neighborhood and determines how intercellular signaling operates to achieve higher-order tissue function. Spatially resolved transcriptomics (SRT) has emerged as the assay of choice for characterizing cells in a tissue context (Burgess, 2019; Marx, 2021). There are a number of SRT methods, with each being able to resolve gene expression to various spatial resolutions, from anatomical features up to sub-cellular resolution of identifying single mRNA molecules (Asp et al., 2020). Single-molecule SRT methods usually require the assignment of each decoded mRNA spot to a cell, which first requires the cell to be identified *via* segmentation. Cell segmentation is usually performed by identifying cell landmark

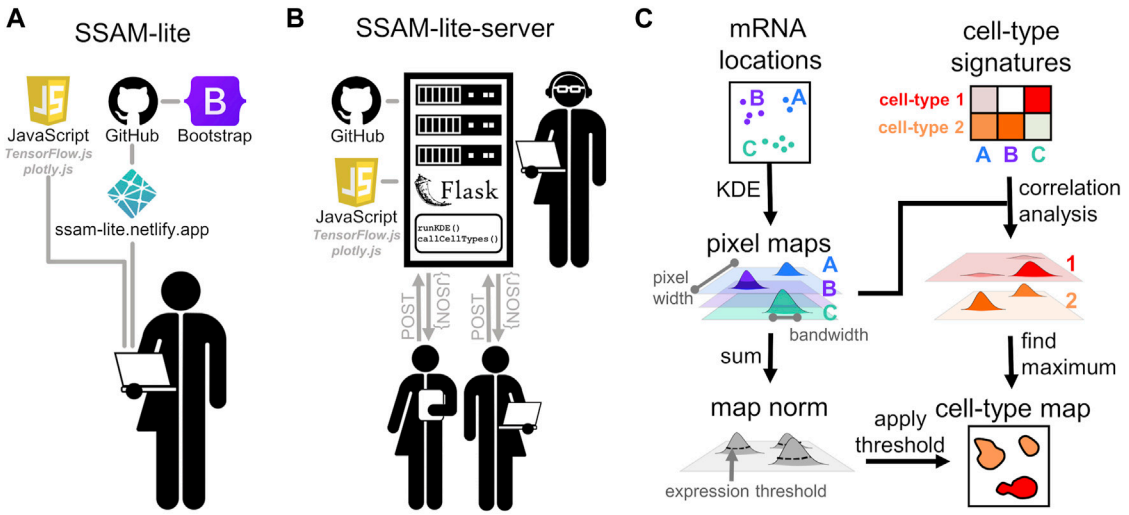

FIGURE 1 | Schematic of SSAM-lite. **(A)** Schematic diagram of SSAM-lite, accessible as a web browser application, and **(B)** a locally installed SSAM-lite-server. **(C)** Schematic of the underlying data processing algorithm proposed by SSAM.

features such as the cell nucleus or protoplasm via DAPI or total mRNA density (Najman and Schmitt, 1994; Chen et al., 2015; Eng et al., 2019). However, accurate cell segmentation remains difficult due to many factors such as staining not covering all features of a cell, imaging artifacts, and overlapping cells (Thomas and John, 2017). Inaccurate cell segmentation can lead to misassignment of mRNA molecules to cells, leading to errors in downstream analysis such as misclassifying cell types. To overcome this issue, a number of computational tools have been developed to improve the assignment of mRNA molecules to cells (Qian et al., 2020; Prabhakaran et al., 2021), incorporate cell typing as part of the segmentation process (Littman et al., 2021), and perform cell-segmentation free analysis (Petukhov et al., 2020; He et al., 2021; Park et al., 2021). While these tools improve cell typing, they all share the problem of being specialized tools that require access to Linux command line terminals, programming expertise, and high-performance hardware. This renders them less accessible to a large proportion of the biomedical research community.

Our prior work (Park et al., 2021) demonstrated improved accuracy and sensitivity of spatial cell typing over traditional segmentation-based approaches by applying the SSAM algorithm to the mouse somatosensory cortex dataset profiled by osmFISH. In particular, our segmentation-free approach identified many more astrocyte cell types that were missed due to low signal. Furthermore, we could reconstruct the ventricle region that was missed due to high occlusion in the segmentation-based approach used in the original study of the data.

Here we present SSAM-lite which is an easy-to-use and lightweight browser-based web application on top of the segmentation-free SRT algorithm SSAM (Park et al., 2021) to make spatial cell typing accessible to biomedical researchers. SSAM-lite runs on modest hardware in any modern browser with JavaScript support and internet access, thus lowering the barrier to analyzing high-dimensional SRT data. To ensure privacy and security, data does not leave the user's machine. Furthermore, our tool has an easy-to-use graphical user interface that provides intuitive visualizations of SRT data. SSAM-lite can be used on mobile devices to analyze smaller datasets. Departments or institutes with access restricted to local networks due to security reasons or which deal with extremely large datasets can make use of SSAM-lite-server. This is a server-side implementation of SSAM-lite that can be installed with minimal effort, providing offline access to SSAM-lite functionality and without limitations of client-side resources.

2 METHODS

2.1 SSAM-lite

SSAM-lite builds on top of the guided mode of the SSAM algorithm (Park et al., 2021) (**Figure 1**). In brief, the algorithm uses Kernel Density Estimation (KDE) to transform the spatial mRNA coordinates into gene expression probability densities that are subsequently cell typed and then projected into the final image of the cell-type map. SSAM-lite is an integrated pipeline aimed at simplifying exploratory data analyses of SRT data with only a few clicks in a web browser. The pipeline workflow combines state-of-the-art web programming libraries such as *Bootstrap*, *plotly.js*, and *TensorFlow.js* (**Figure 1A**). The modern web interface with convenient interactive elements was generated using the *Bootstrap* library, which provides a large body of CSS functions for creating a state-of-the-art and user-friendly layout. In particular, the layout scripts for SSAM-lite make use of *Bootstrap*'s sophisticated scalable grid layout that optimizes user experience on a range of devices from handhelds to desktop machines. The data preparation and presentation routines were implemented using *plotly.js*, and *TensorFlow.js* was chosen to implement a machine learning backend.

A typical SSAM-lite workflow can be summarized in three steps: data upload, parameter selection and optimization, and the

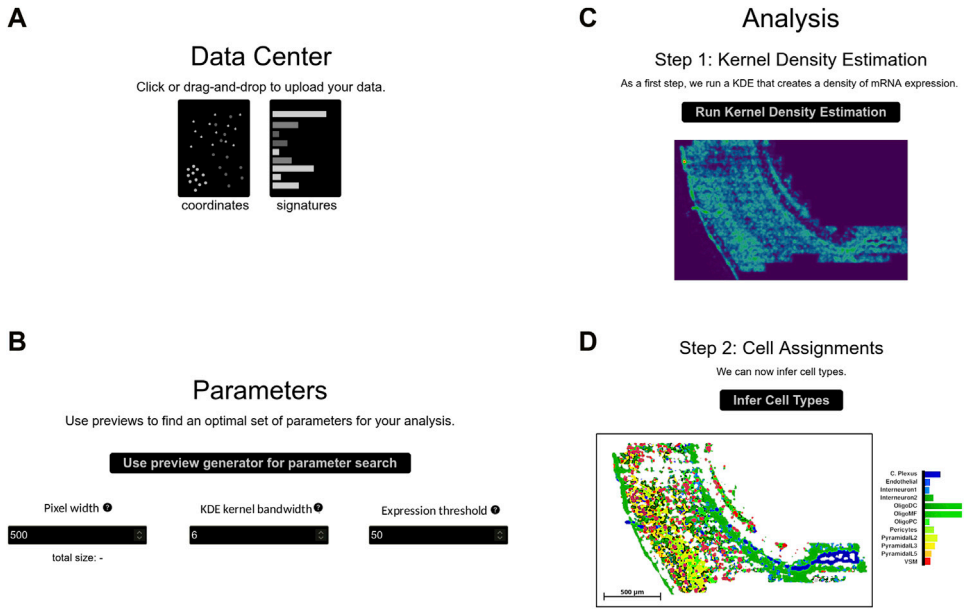

FIGURE 2 | The SSAM-lite interface. The panels display the sections of the SSAM-lite web page demonstrated on osmFISH data of the mouse SSp (Codeluppi et al., 2018): **(A)** the data center for uploading data; **(B)** the parameter selection and optimization section; **(C)** the first analysis section for displaying the results of the KDE analysis; and **(D)** the second analysis section for displaying the final cell-type map image.

final analysis phase. Each step has a dedicated area in the web interface (**Figure 2**).

2.1.1 Data Upload

Data upload is performed in the Data Center section by either using drag-and-drop or an interactive file selection window (**Figure 2A**). The user needs to provide a file with mRNA coordinates from an SRT experiment alongside a so-called signature file that contains gene expression signatures for the cell types of the tissue of interest. Both input files are plain text csv files. The mRNA coordinate file contains gene names and the x- and y-coordinates of all molecules in the analyzed image, consistent with the *DecodedSpot* format defined by the Starfish pipeline (http://github.com/spacetx/starfish). The signature file contains a gene expression matrix with cell types as rows and gene names as columns. The values can either be binary or be normalized gene expression.

After loading, the mRNA molecule coordinate data is displayed in an interactive scatter plot using *plotly.js*'s *scattergl* layout, which is designed explicitly to handle large data sets. The plot is designed to be interactive, so the user can zoom in to investigate local mRNA expression or hide parts of the data to reveal the expression patterns of individual genes. The expression signature matrix is also displayed in an interactive plot after loading using *plotly.js*'s *heatmap* layout, which provides an overview of the data through color coding and by displaying hovering information on each gene-cell type expression indicator.

2.1.2 Parameter Selection and Optimization

In this section, the user can interactively tune the input parameters for the SSAM spatial modeling algorithm (**Figure 2B**; **Supplementary Figure S1**). The three most important parameters of the SSAM algorithm are the bandwidth of the Gaussian KDE function, the pixel width of the output cell-type map, and the total expression threshold value. The bandwidth parameter is necessary to accurately model the local spatial molecular dynamics. To model expression in a sparse dataset (e.g., 3-5 mRNA molecules per cell) a larger bandwidth would need to be employed, and in a dense dataset (e.g., 20–30 mRNA per cell) a smaller bandwidth should be sufficient. As a guideline, we suggest using values between 2 and 25 µm based on analysis of dense and sparse datasets (**Figure 3**). The pixel width of the cell-type map determines the memory footprint and the accuracy of the internal spatial gene expression model. The expression threshold parameter defines the gene expression signal threshold for the foreground (i.e., parts of the image with high gene expression, likely originating from cells) and background (i.e., parts of the image with low gene expression), hence discerning actual spatial expression patterns from background noise. A high number of extracellular, diffused mRNA spots requires a higher expression threshold, where the optimal value differs greatly across data sets.

These parameters can be set in numerical input fields and the analysis of the full data set can be started. However, the user can also try to optimize the parameters on a small section of the image before starting the complete image analysis. This will launch an initial small-scale analysis with instant output to the screen and will show three figure panels that allow for direct evaluation of the chosen parameters.

Of these panels, the left figure panel is an interactive *plotly.js scattergl* plot of the entire mRNA location data set, which can be

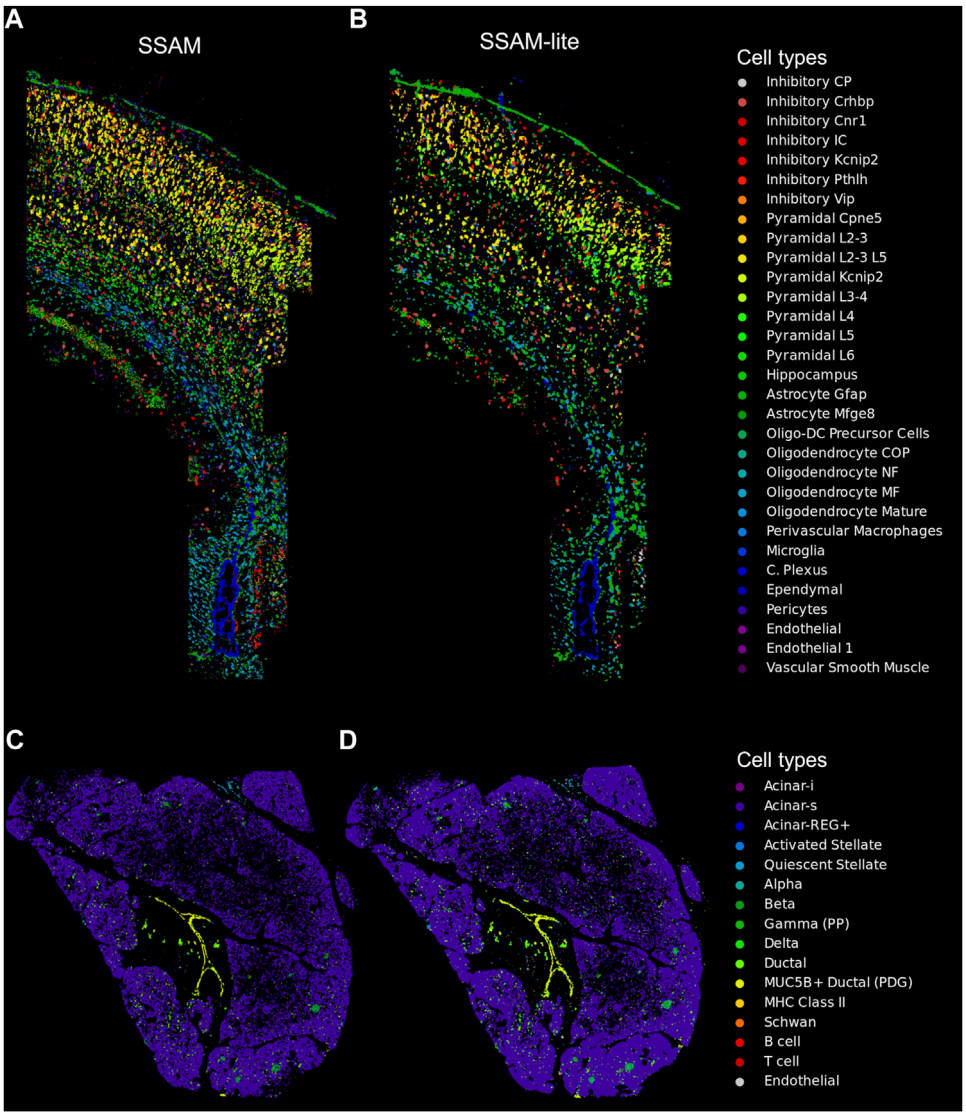

FIGURE 3 | SSAM-lite generates accurate cell-type maps. Demonstrative cell-type maps for osmFISH data of the mouse SSp generated by **(A)** SSAM and **(B)** SSAM-lite, and ISS data of human pancreas generated by **(C)** SSAM and **(D)** SSAM-lite. Resultant cell-type maps generated by SSAM are similar to previous publications (Park et al., 2021; Tosti et al., 2021). Cell-type colors of the original SSAM figures were modified to match the SSAM-lite figure.

used to define the local sub-section of the overall data set that is used for optimization. A rectangle displays the currently chosen sub-section, and the location of the subsection can be changed interactively by clicking onto the desired new central spot in the scatterplot.

The middle figure panel shows an intermediate output of the KDE for the chosen sub-section from the first subfigure using *plotly.js*'s *heatmap* layout. The heatmap is a spatial representation of SSAM-lite's internal model of integrated local signal strength, with the heatmap value indicating the probability for the presence of a cell at a particular location. The value of the modeled signal for each pixel is color-coded and shows up when hovering over it with the mouse pointer. The heatmap is especially useful for choosing an appropriate expression threshold parameter from the signal strength landscape. The KDE figure panel also provides a visual impression of the amount of smoothing produced by the KDE, which helps the user to set the *bandwidth* parameter. The bandwidth parameter should be large enough to smooth out noise and integrate mRNA signals belonging to the same spatial structure, but low enough to keep individual spatial structures separate and retain their shape. The heatmap plot gets updated in real-time whenever the subsample location or KDE parameters change, and in practice, the parameters can be set reasonably after 2–3 trials.

The rightmost figure panel shows the final output cell-type map of the SSAM-lite algorithm for the chosen tissue subsection. The cell-type map is useful to identify persisting noise in the output, which can be reduced by adjusting the

bandwidth and/or the expression threshold parameters. The cell-type map panel is updated whenever the parameters change.

2.1.3 Analysis and Visualization

The last section is dedicated to data analysis and the visualization of results. The section provides interfaces to the two more resource-intensive *TensorFlow.js* backend functions that perform the KDE and the correlation analysis.

2.1.3.1 Kernel Density Estimation (KDE)

Once the parameters are optimized, the user can perform the KDE, which typically takes below a minute to generate SSAM-lite's internal, pixel-based spatial model of local signal strength (**Figure 2C**). In a pre-processing step, the mRNA coordinates are rescaled linearly to fit the user-defined pixel width of the spatial model. The respective height is determined to match the vertical spread of the coordinate data and the bandwidth parameter is scaled accordingly to match the new internal unit of computation. SSAM-lite computes an independent local signal strength pixel matrix for each type of mRNA defined in the input data. For this, a large *TensorFlow.js* buffer is initiated by stacking all empty pixel matrices. The KDE implementation in SSAM-lite employs two heuristics to optimize computing performance. The first is to iterate over all mRNA locations and round them to their closest output pixel, allowing us to use a pre-calculated Gaussian mass function for all mRNA spots. The second is to ignore long tails of the Gaussian mass function by limiting its calculation to two bandwidths. This heuristics approximates the naive KDE implementation well, with negligible differences at reasonable bandwidth (**Supplementary Figure S2**). This new implementation results in a 1000-fold performance increase over the default SSAM implementation of the KDE step (**Supplementary Figure S3**).

Further differences to the original SSAM-guided mode implementation are described in the **Supplementary Material**.

After KDE computation is completed, the collected sum of all pixel matrices is displayed using a *plotly.js heatmap* layout analogous to the optimization panel. If the results do not match expectations, parameters can be adapted and the KDE function can be re-run. Otherwise, the user can move on to generate the cell-type map.

2.1.3.2 Correlation Analysis and Cell-Type Map Generation

As in the original SSAM algorithm, the last step of analysis computes the cell-type map through correlation analysis with known gene expression signatures (**Figure 2D**). The combined expression arrays of each x- and y-location in the stacked pixel matrixes are compared to the expression signature data and each pixel is assigned the cell type with the highest correlating signature. All pixels whose sum across matrices are below the user-defined expression threshold parameter are considered background and not assigned any cell type. The final result is displayed as a cell-type map using a modified version of *plotly.js*'s *heatmap* layout. The heatmap element is fed with a custom generated list of colors and altered to display the x- and y-coordinates and the assigned cell-type name during mouse hover events. The plot offers *plotly.js*'s elementary functions like zooming, panning, resetting as well as a save to disk option. Furthermore, a custom scale bar is added that adapts to the current zoom factor and displays the bar width in micrometers.

2.1.3.3 Cell-type Localization and Abundance

An important part of the downstream analysis of the cell-type map is the localization of cell types and the quantification of cell-type signals in the entire and parts of the tissue (**Figure 2D**; **Supplementary Figure S4A**). We therefore implement an interactive barplot that quantifies the relative cell-type abundance based on classified pixels in the current view of the cell-type map. This quantification is updated when zooming into or panning over different regions (**Supplementary Figures S4B,C**). The user can also provide custom color palettes and select only certain cell-types to be rendered by double-clicking the cell-type labels (**Supplementary Figure S5**).

The code itself is documented and organized according to the model-view-controller paradigm, which allows the user to easily adapt the code base to the needs of their own specific project. One example would be to use an alternative kernel shape, e.g., a circular Epanechnikov kernel could be achieved by adding a logical threshold expression to the *runKDE* function inside *model.js*. Any changes are integrated into the code execution right away and available after a simple browser page refresh.

2.1.4 SSAM-lite-Server

SSAM-lite is an efficient tool that is dependent on client-side hardware. While we demonstrate that a modest laptop is capable of processing real-world SRT datasets (**Figure 3**), we also recognize possible limitations due to client-side hardware constraints. To address this issue, we developed a server-side version called SSAM-lite-server (**Figure 1A**). SSAM-lite-server runs the computationally expensive KDE and cell assignment algorithms at the server-side. SSAM-lite-server preserves the overall implementation of SSAM-lite in Javascript, HTML, CSS, and allows a server running a Flask (v0.8) framework to take over computationally expensive functionalities of SSAM-lite. Flask was chosen due to its lightweight nature and extensibility. To further make the backend data structures memory-efficient we use Python's numerical library *NumPy* (v1.20.3). Python's *pandas* package (v1.3.2) is used to handle the signature data. For privacy preservation, the data streamed to the server for processing do not persist on the server file systems but is only stored in memory for the duration of the computation.

SSAM-lite-server runs the KDE algorithm by streaming variables such as coordinates, signature matrix, input and output image width, bandwidth, gene expression threshold to the server as an Ajax POST request, which then returns JSON objects to the user. The server-side computation includes the computation of KDE and the generation of the cell-type map.

To enhance the overall security, SSAM-lite-server offers the option to host all libraries locally, thus enabling SSAM-lite-server to run in closed networks without an internet connection.

2.1.5 Benchmarking

Benchmarking was carried out on a Lenovo X1 Carbon laptop with Intel Core i7-8565u CPU, 16GB of RAM, and Windows 10. We used Google Chrome (v93.0) to run SSAM-lite (v0.1.0). The benchmark was performed using the Chrome DevTools Performance monitor to evaluate the runtime of the *runFullKDE* function and the maximum memory heap while carrying out a complete analysis (not using the parameter preview) with the pixel width of the cell-type map set to 500, the kernel bandwidth to five and the expression threshold for assigning cell types to two.

To simulate different complexities of the mouse brain primary somatosensory cortex (SSp) data we performed downscaling and upscaling of the data. A 0.5× dataset was created by randomly downsampling to 50% of the molecules present in the coordinate file. A 2× data set was created by appending the mRNA coordinate locations to itself after carrying out a pixel shift of 1 µm along both axes to each of the molecules. A 3× dataset was created by pixel shifting the original coordinate matrix by -1 µm and appending it to the 2× coordinate matrix. Finally, a 5× dataset was created by appending the dataset to itself, the first time pixel-shifting +1 along x and y, the second time +2, and so on. Each of the above datasets was then tested in three replicates.

Furthermore, to demonstrate usable performance on modest hardware we report the runtimes of SSAM-lite on a Lenovo b570e with 4GB of RAM and a 2.20 GHz Intel dual-core processor running Windows 10, and a Samsung Galaxy S8+ Android 9 smartphone running Chrome v96 (**Supplementary Materials**).

3 RESULTS

To demonstrate equivalent cell-type map performance to our previously published SSAM algorithm, we applied SSAM-lite to two datasets using a laptop computer (**Section 2.1.5** in **Section 2**). The first dataset was mouse SSp profiled by osmFISH (Zeisel et al., 2015; Marques et al., 2016), profiling 1,802,589 mRNA spots for 33 genes and 31 cell-types signatures derived from scRNAseq (Zeisel et al., 2015; Marques et al., 2016). The coordinate matrix was uploaded and rendered in 4 s on average, and the uploading and rendering time for the signature matrix was negligible in comparison. The cell-type map width was set to 1,500, KDE bandwidth to 2.5, and the gene expression threshold to 13. The resultant image of the cell-type map was very similar to those previously published (**Figures 3A,B**). To demonstrate SSAM-lite's performance on a sparse dataset, we applied it to human pancreas profiled by ISS, profiling 461,078 mRNA spots for 138 genes and 16 cell-type signatures (Tosti et al., 2021). The cell-type map width was set to 750, KDE bandwidth to 22, and the gene expression threshold to 2.4. The resultant image of the cell-type map was highly comparable to those previously published (**Figures 3C,D**).

To investigate how SSAM-lite's performance scales with regards to memory requirements and CPU time, we performed a synthetic benchmark on the mouse brain SSp dataset with different dataset sizes (**Supplementary Figure S6**). Overall, the CPU time for calculating the KDE (**Supplementary Figure S6A**) scales linearly with the number of profiled mRNA molecules. Further, the total memory footprint for a complete analysis also depends linearly on the dataset size (**Supplementary Figure S6A**).

4 DISCUSSION

Analysis of spatial transcriptomics data was so far limited by excessive hardware requirements and an understanding of navigation in a terminal window using the Linux command line. With SSAM-lite we overcome these limitations by providing an easy-to-use graphical user interface that runs in any modern web browser on common laptop computers. Input files are text files that can be loaded by drag-and-drop into the browser window. This circumvents the need to provide certain command-line arguments or editing of configuration files. SSAM-lite makes the analysis of spatial transcriptomics data accessible to a broad range of researchers that may not have a high-performance computing cluster or experience with command-line tools. SSAM-lite was able to generate similar results to those previously published (Park et al., 2021; Tosti et al., 2021) in only a few minutes. SSAM-lite provides an easy-to-use interface to analyze high-dimensional SRT data to the wider biomedical research community. In addition, we see the additional utility in SSAM-lite for SRT data generators to perform rapid quality control of experiments and to provide customers with an easy-to-use exploratory tool. We also expect that specialized computational scientists may want to use SSAM-lite to rapidly identify optimal parameters for downstream analysis and to compare the resultant cell-type map of more parameterized and resource-hungry analysis tools.

In addition, SSAM-lite-server mitigates much of the computational burden to the server-side, enabling analysis of very large datasets, and also analysis of datasets on mobile devices. The stand-alone implementation of SSAM-lite-server is amenable to networks with limited access to the internet such as in many university hospitals.

AVAILABILITY AND IMPLEMENTATION

SSAM-lite is an open-source browser-based web application with source code freely available on Github *via* https://github.com/HiDiHlabs/ssam-lite. Stable releases can be accessed *via* https://ssam-lite.bihealth.org and https://ssam-lite.netlify.app, and developmental releases can be accessed *via* https://dev–ssam-lite.netlify.app. The source code for a locally deployable server version, SSAM-lite-server, is available on GitHub *via* https://github.com/HiDiHlabs/ssam-lite-server. Both versions require a modern browser with JavaScript and WebGL support. Detailed user guides and documentation can be found at https://ssam-lite.readthedocs.io.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: https://zenodo.org/record/5600532.

AUTHOR CONTRIBUTIONS

ST and NI conceived and designed the study. ST programmed the SSAM-lite and SSAM-lite-server software. SS made programming contributions to SSAM-lite-server. ST, SDM, and NI wrote the manuscript. SS and NM-B tested and documented the software, made programming contributions to SSAM-lite, wrote the user guide, and revised the manuscript. RE proofread and corrected the manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

This research received funding from the Federal Ministry of Education and Research of Germany in the framework of SAGE (Project Number 031L0265), the BMBF-funded de.NBI Cloud within the German Network for Bioinformatics Infrastructure (de.NBI) (031A532B, 031A533A, 031A533B, 031A534A, 031A535A, 031A537A, 031A537B, 031A537C, 031A537D, 031A538A), and from the European Commission EU Horizon 2020 research and innovation program (ESPACE, 874710; EASI-Genomics, 824110).

ACKNOWLEDGMENTS

We would like to thank Jeongbin Park and Wonyl Choi for conceiving the idea of SSAM.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2022.785877/full#supplementary-material

REFERENCES

Asp, M., Bergenstråhle, J., and Lundeberg, J. (2020). Spatially Resolved Transcriptomes-Next Generation Tools for Tissue Exploration. *Bioessays* 42, e1900221. doi:10.1002/bies.201900221

Burgess, D. J. (2019). Spatial Transcriptomics Coming of Age. *Nat. Rev. Genet.* 20, 317. doi:10.1038/s41576-019-0129-z

Chen, K. H., Boettiger, A. N., Moffitt, J. R., Wang, S., and Zhuang, X. (2015). RNA Imaging. Spatially Resolved, Highly Multiplexed RNA Profiling in Single Cells. *Science* 348, aaa6090. doi:10.1126/science.aaa6090

Codeluppi, S., Borm, L. E., Zeisel, A., La Manno, G., van Lunteren, J. A., Svensson, C. I., et al. (2018). Spatial Organization of the Somatosensory Cortex Revealed by osmFISH. *Nat. Methods* 15, 932–935. doi:10.1038/s41592-018-0175-z

Eng, C.-H. L., Lawson, M., Zhu, Q., Dries, R., Koulena, N., Takei, Y., et al. (2019). Transcriptome-scale Super-resolved Imaging in Tissues by RNA seqFISH+. *Nature* 568, 235–239. doi:10.1038/s41586-019-1049-y

He, Y., Tang, X., Huang, J., Zhou, H., Chen, K., Liu, A., et al. (2021). ClusterMap: Multi-Scale Clustering Analysis of Spatial Gene Expression. bioRxiv, 2021.02.18.431337. doi:10.1101/2021.02.18.431337

Lee, J. H. (2017). Quantitative Approaches for Investigating the Spatial Context of Gene Expression. *Wires Syst. Biol. Med.* 9. doi:10.1002/wsbm.1369

Littman, R., Hemminger, Z., Foreman, R., Arneson, D., Zhang, G., Gómez-Pinilla, F., et al. (2021). Joint Cell Segmentation and Cell Type Annotation for Spatial Transcriptomics. *Mol. Syst. Biol.* 17, e10108. doi:10.15252/msb.202010108

Marques, S., Zeisel, A., Codeluppi, S., van Bruggen, D., Mendanha Falcão, A., Xiao, L., et al. (2016). Oligodendrocyte Heterogeneity in the Mouse Juvenile and Adult central Nervous System. *Science* 352, 1326–1329. doi:10.1126/science.aaf6463

Marx, V. (2021). Method of the Year: Spatially Resolved Transcriptomics. *Nat. Methods* 18, 9–14. doi:10.1038/s41592-020-01033-y

Najman, L., and Schmitt, M. (1994). Watershed of a Continuous Function. *Signal. Process.* 38, 99–112. doi:10.1016/0165-1684(94)90059-0

Park, J., Choi, W., Tiesmeyer, S., Long, B., Borm, L. E., Garren, E., et al. (2021). Cell Segmentation-free Inference of Cell Types from *In Situ* Transcriptomics Data. *Nat. Commun.* 12, 3545. doi:10.1038/s41467-021-23807-4

Petukhov, V., Soldatov, R. A., Khodosevich, K., and Kharchenko, P. V. (2020). Bayesian Segmentation of Spatially Resolved Transcriptomics Data. bioRxiv, 2020.10.05.326777. doi:10.1101/2020.10.05.326777

Prabhakaran, S., Nawy, T., and Pe'er', D. (2021). Sparcle: Assigning Transcripts to Cells in Multiplexed Images. bioRxiv, 2021.02.13.431099. doi:10.1101/2021.02.13.431099

Qian, X., Harris, K. D., Hauling, T., Nicoloutsopoulos, D., Muñoz-Manchado, A. B., Skene, N., et al. (2020). Probabilistic Cell Typing Enables fine Mapping of Closely Related Cell Types *In Situ*. *Nat. Methods* 17, 101–106. doi:10.1038/s41592-019-0631-4

Thomas, R. M., and John, J. (2017). "A Review on Cell Detection and Segmentation in Microscopic Images," in 2017 International Conference on Circuit ,Power and Computing Technologies (ICCPCT) (IEEE) (IEEE). doi:10.1109/iccpct.2017.8074189

Tosti, L., Hang, Y., Debnath, O., Tiesmeyer, S., Trefzer, T., Steiger, K., et al. (2021). Single-Nucleus and *In Situ* RNA-Sequencing Reveal Cell Topographies in the Human Pancreas. *Gastroenterology* 160, 1330–1344.e11. doi:10.1053/j.gastro.2020.11.010

Zeisel, A., Muñoz-Manchado, A. B., Codeluppi, S., Lönnerberg, P., La Manno, G., Juréus, A., et al. (2015). Cell Types in the Mouse Cortex and hippocampus Revealed by Single-Cell RNA-Seq. *Science* 347, 1138–1142. doi:10.1126/science.aaa1934

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Advantages of publishing in Frontiers

OPEN ACCESS
Articles are free to read for greatest visibility and readership

FAST PUBLICATION
Around 90 days from submission to decision

HIGH QUALITY PEER-REVIEW
Rigorous, collaborative, and constructive peer-review

TRANSPARENT PEER-REVIEW
Editors and reviewers acknowledged by name on published articles

Frontiers
Avenue du Tribunal-Fédéral 34
1005 Lausanne | Switzerland

Visit us: www.frontiersin.org
Contact us: frontiersin.org/about/contact

REPRODUCIBILITY OF RESEARCH
Support open data and methods to enhance research reproducibility

DIGITAL PUBLISHING
Articles designed for optimal readership across devices

FOLLOW US
@frontiersin

IMPACT METRICS
Advanced article metrics track visibility across digital media

EXTENSIVE PROMOTION
Marketing and promotion of impactful research

LOOP RESEARCH NETWORK
Our network increases your article's readership