Supplementary Material

1 INCREASING TEMPERATURE MEAN AND VARIABILITY

We denote the j^{th} temperature observation as T_j and the set t_d to be all temperatures collected on day d, where a day begins at 0:00 and ends at 23:59. We calculate the daily average temperature as

$$A_d = \frac{1}{96} \sum_{T_j \in t_d} T_j,$$

where 96 is the number of observations (taken every 15 minutes). Each temperature $T_j \in t_d$ is some difference $T_j - A_d$ from the daily average and can be represented as

$$T_j = A_d + (T_j - A_d).$$

To increase the average temperature by I degrees without changing difference from the average, we transform each temperature to be

$$T_j^{inc} = (A_d + I) + (T_j - A_d).$$

This is equivalent to shifting the temperature curve up by I. Every temperature observation will increase by I degrees. To increase the daily variability by a fraction V without changing the average, we transform each temperature to be

$$T_j^{var} = A_d + V(T_j - A_d).$$

This is equivalent to stretching the temperature curve vertically around A_d, so that the highs become higher and the lows become lower. However, any observation at the average ($T_j = A_d$) will remain at the average.

To combine these effects, we transform each temperature to be

$$\tilde{T}_j = (A_d + I) + V(T_j - A_d).$$

This is equivalent to shifting the temperature curve vertically, and then stretching its amplitude. We repeat this process for each day of observations, to obtain a temperature curve which incorporates increased temperatures with increased variability. This is shown for two days of temperatures in Figure S5, with a 2.5 degree average temperature increase and 10% added variability. We note that the increased average outweighs the increased variability for these observations, so the curve of $\tilde{T}_j$ is always above the curve of T_j.

2 SUPPLEMENTARY FIGURES

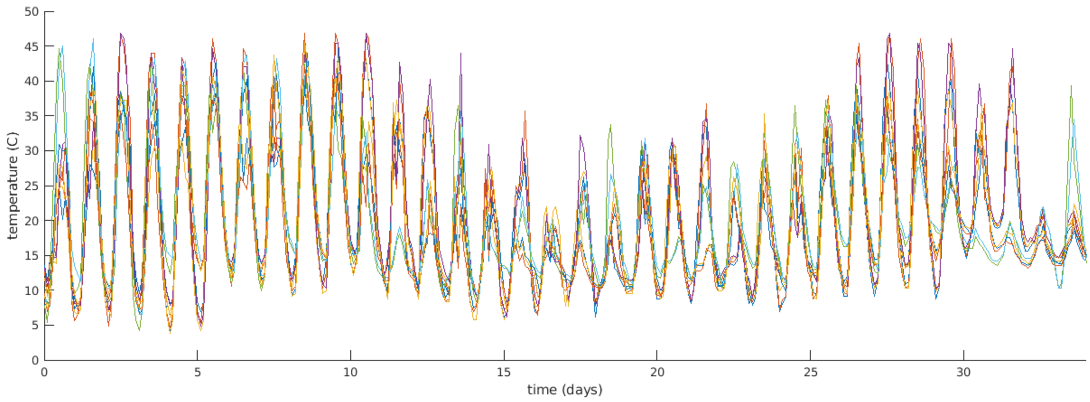

Figure S1. Temperature data from Curtsdotter et al. (2019), obtained every 15 minutes at 10 different field sites. The different fields are plotted with different line colors.

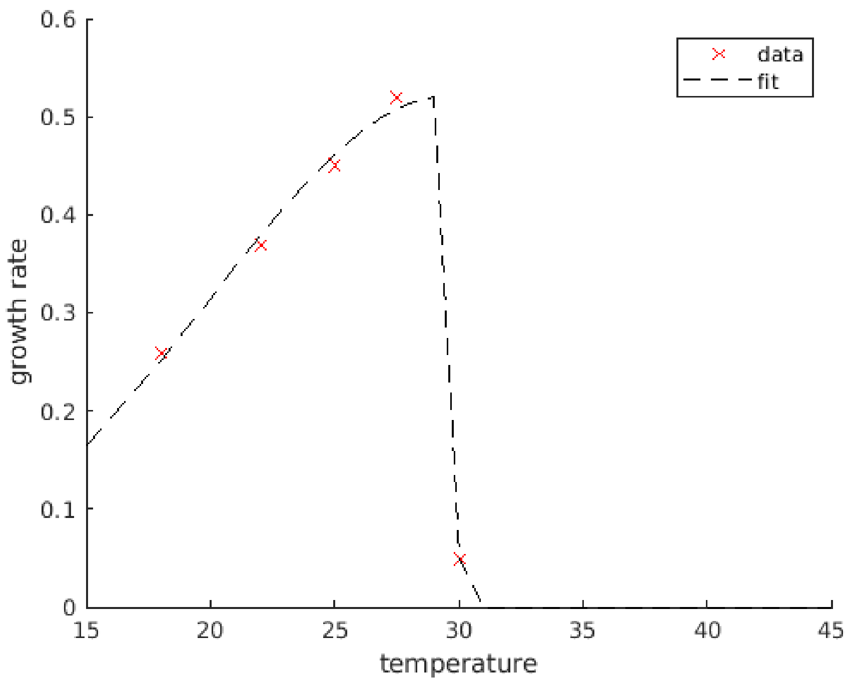

Figure S2. Aphid growth rate as a function of temperature, obtained using the model and data from Asin and Pons (2001). We minimized the least squares error between the data and proposed model, to recover unpublished model parameters.

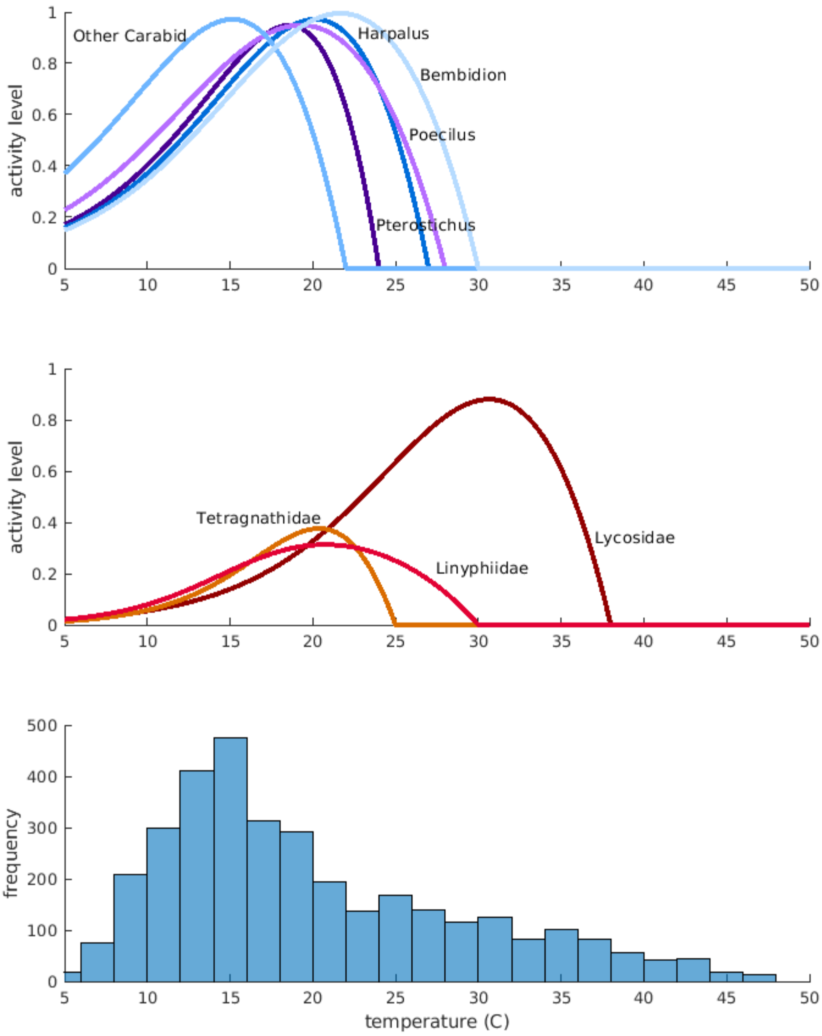

Figure S3. Activity curves ν_j for all predators, using values from Table S2. The top panel is for beetle predators and the lower panel is for spider predators. Darker colors denote larger predators of either category. Compared to Fig. 1 in the manuscript, these curves do not incorporate the effect of body size and prey preference on attack rates.

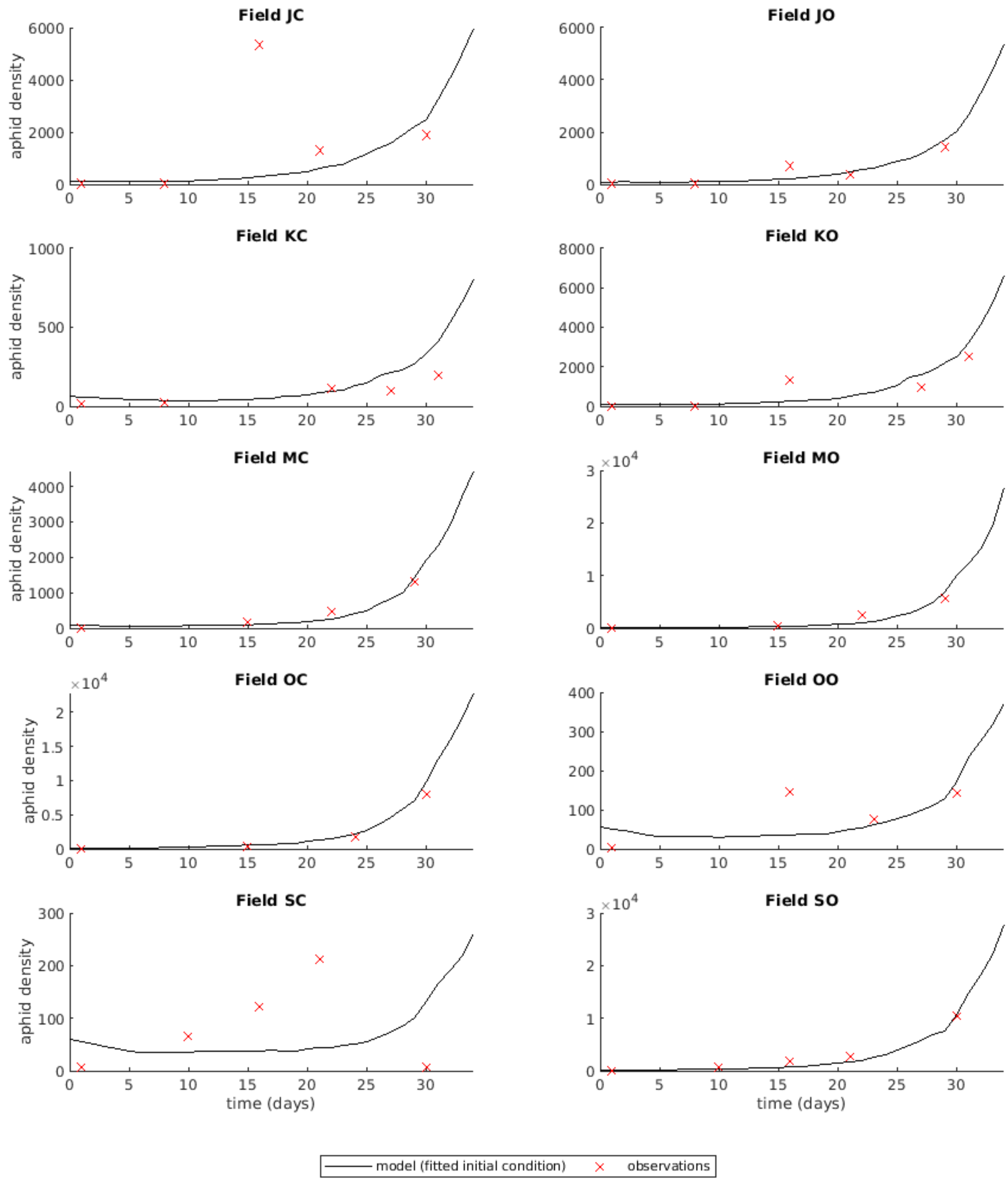

Figure S4. Model for aphid dynamics fit to field abundance data, by modifying initial aphid abundance. Each panel represents a different field. Solid lines denote model fit and red 'x' markers denote field data. Fits are obtained from the same model in each field, but initial aphid abundance varies across fields.

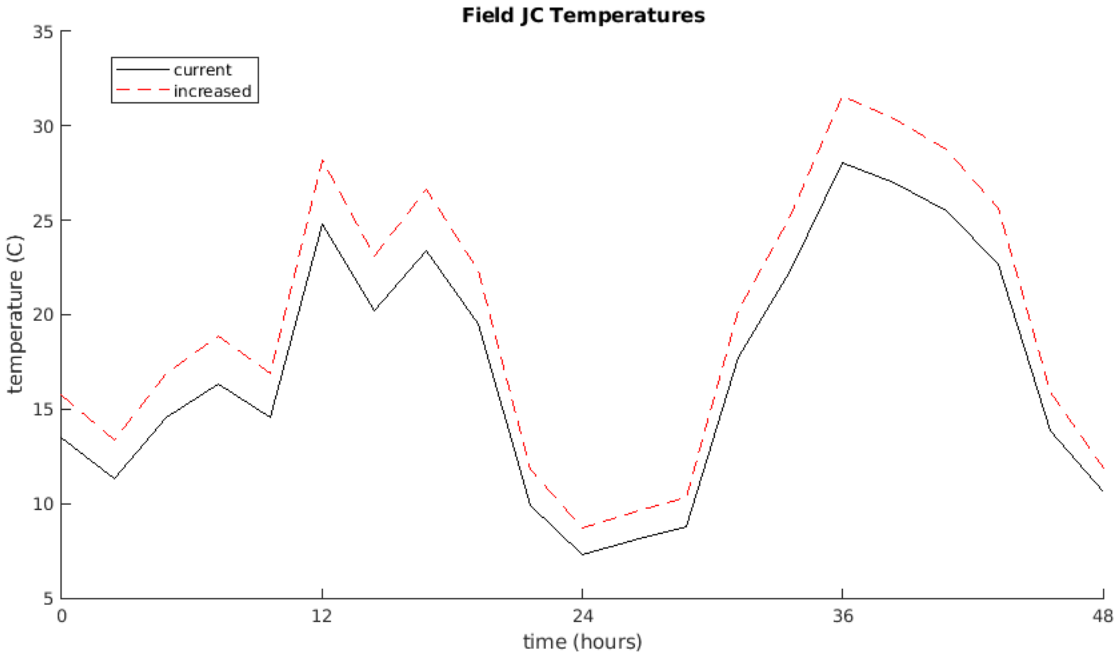

Figure S5. The effect of increasing average temperatures and variability from field observations, over two days in field JC. Current temperatures (T_j) are denoted by the solid black line and temperatures with increased average and variability ($\tilde{T}_j$) are denoted by the dashed red line.

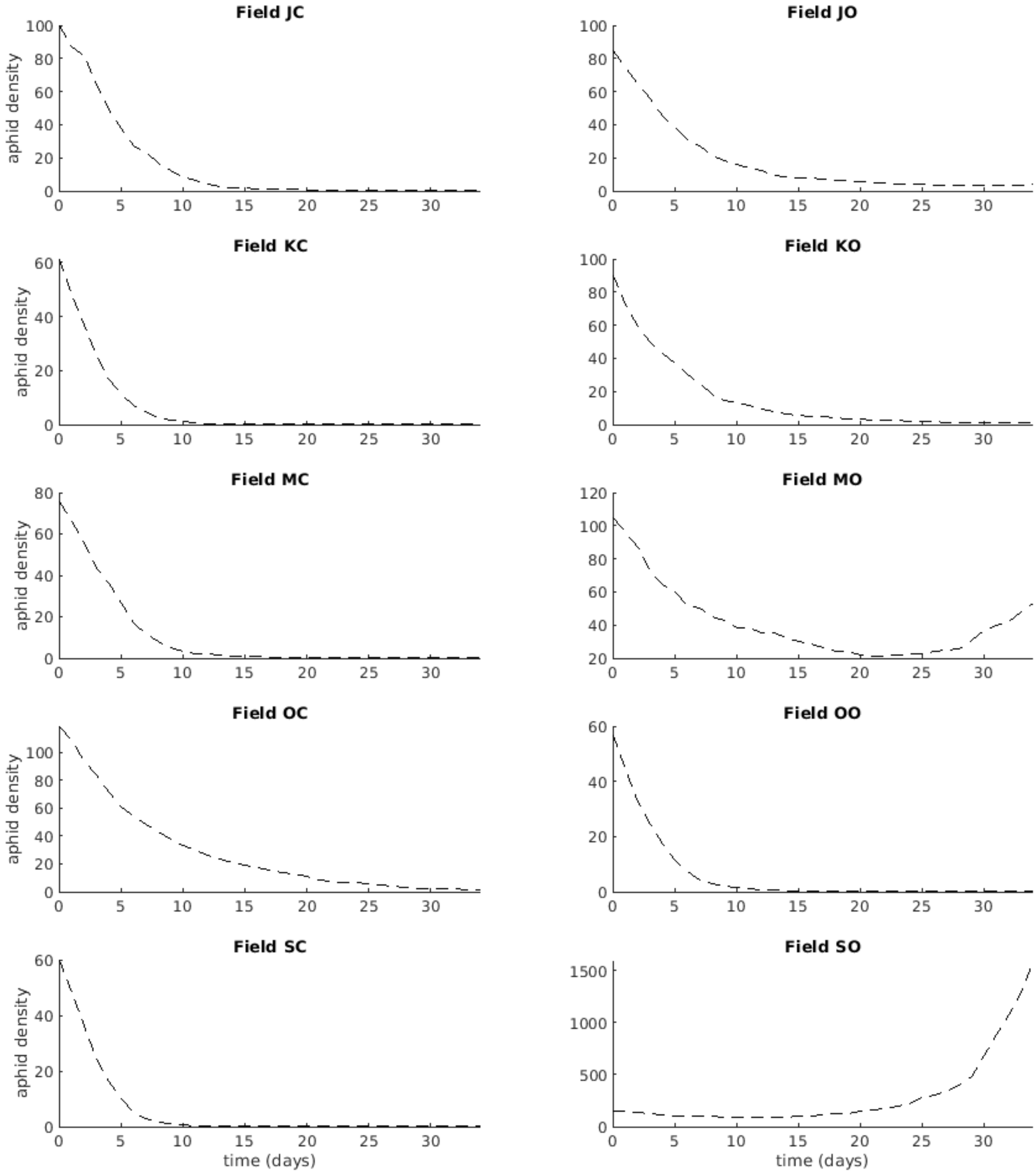

Figure S6. Model for aphid dynamics using optimal predator community in each field, under current field temperatures. Each panel represents a different field.

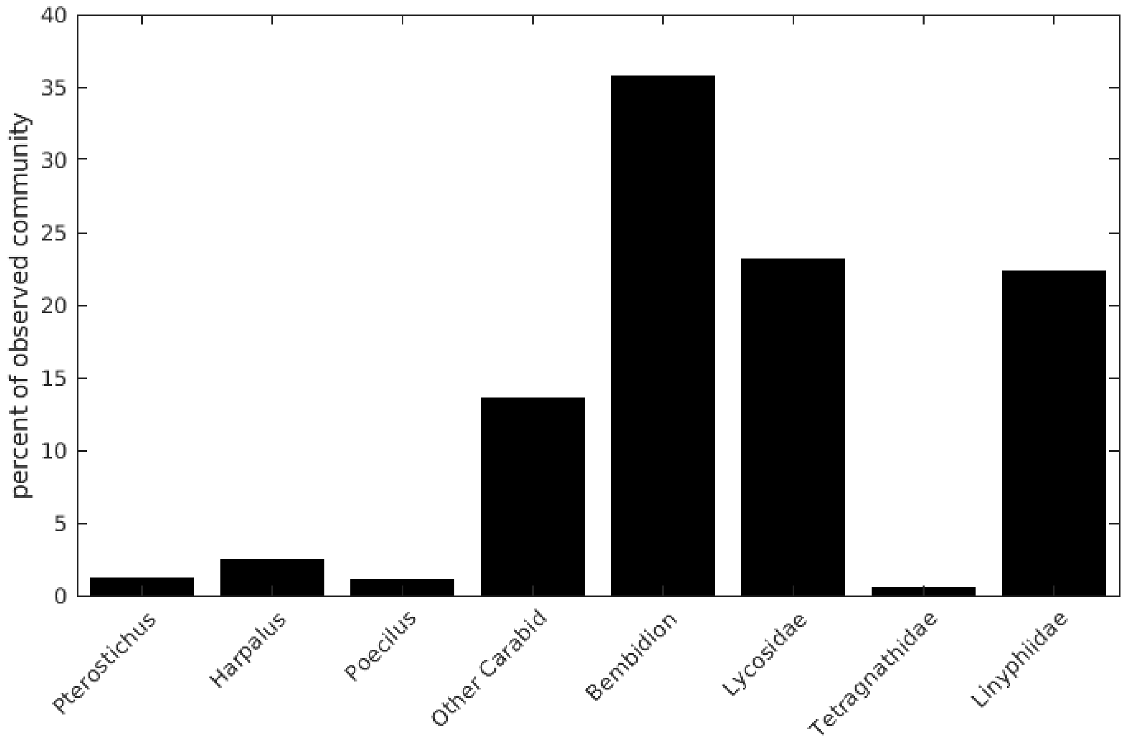

Figure S7. Percent biomass of predator groups, as observed across all field data.

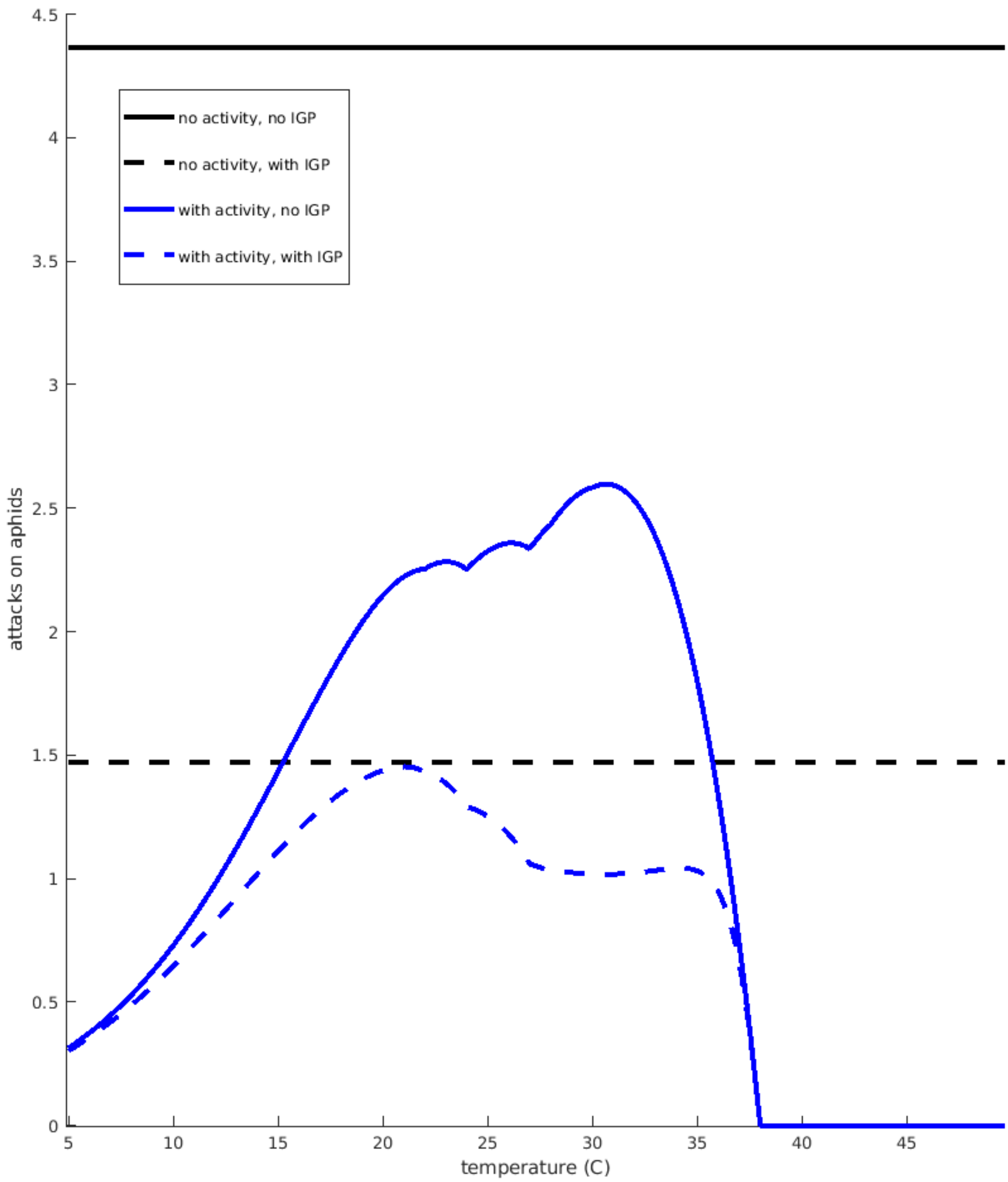

Figure S8. Community-level attack rates on aphids by the observed predator community, plotted against temperature. Solid lines denote potential attack rates and dashed lines denote effective attack rates accounting for interference due to intraguild predation. Black lines do not incorporate temperature-dependent activity levels and blue lines incorporate temperature-dependent activity levels. Compared to Fig. 4 in the manuscript, this figure uses the observed predator community instead of the optimized predator community.

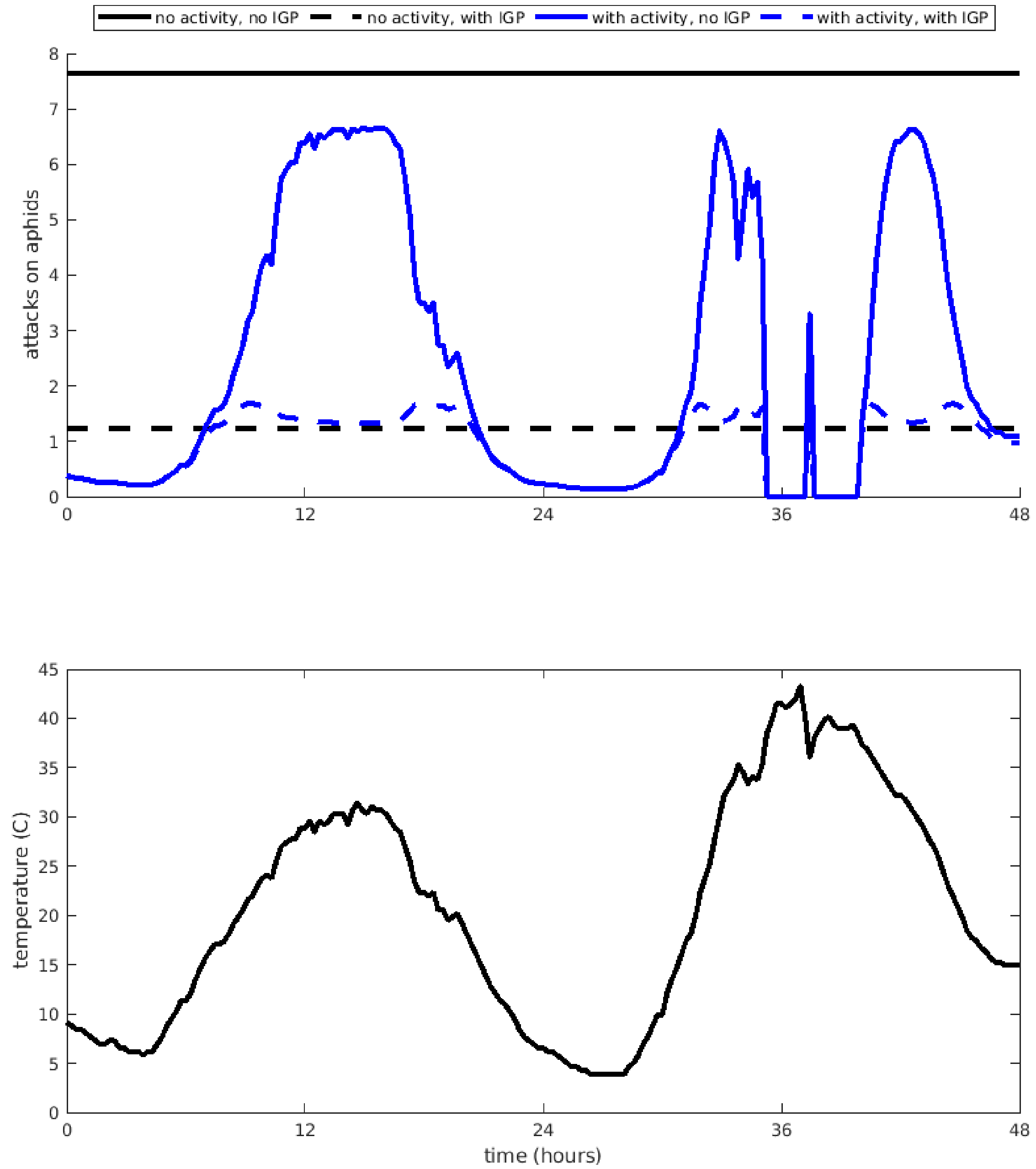

Figure S9. Per-capita aphid attack rates by the average optimal community, as a function of time. Solid lines denote potential attack rates and dashed lines denote effective attack rates accounting for interference due to intraguild predation. Black lines do not incorporate temperature-dependent activity levels and blue lines incorporate temperature-dependent activity levels. The lower panel indicates the corresponding temperatures for these attack rates, taken over two days in one of the fields.

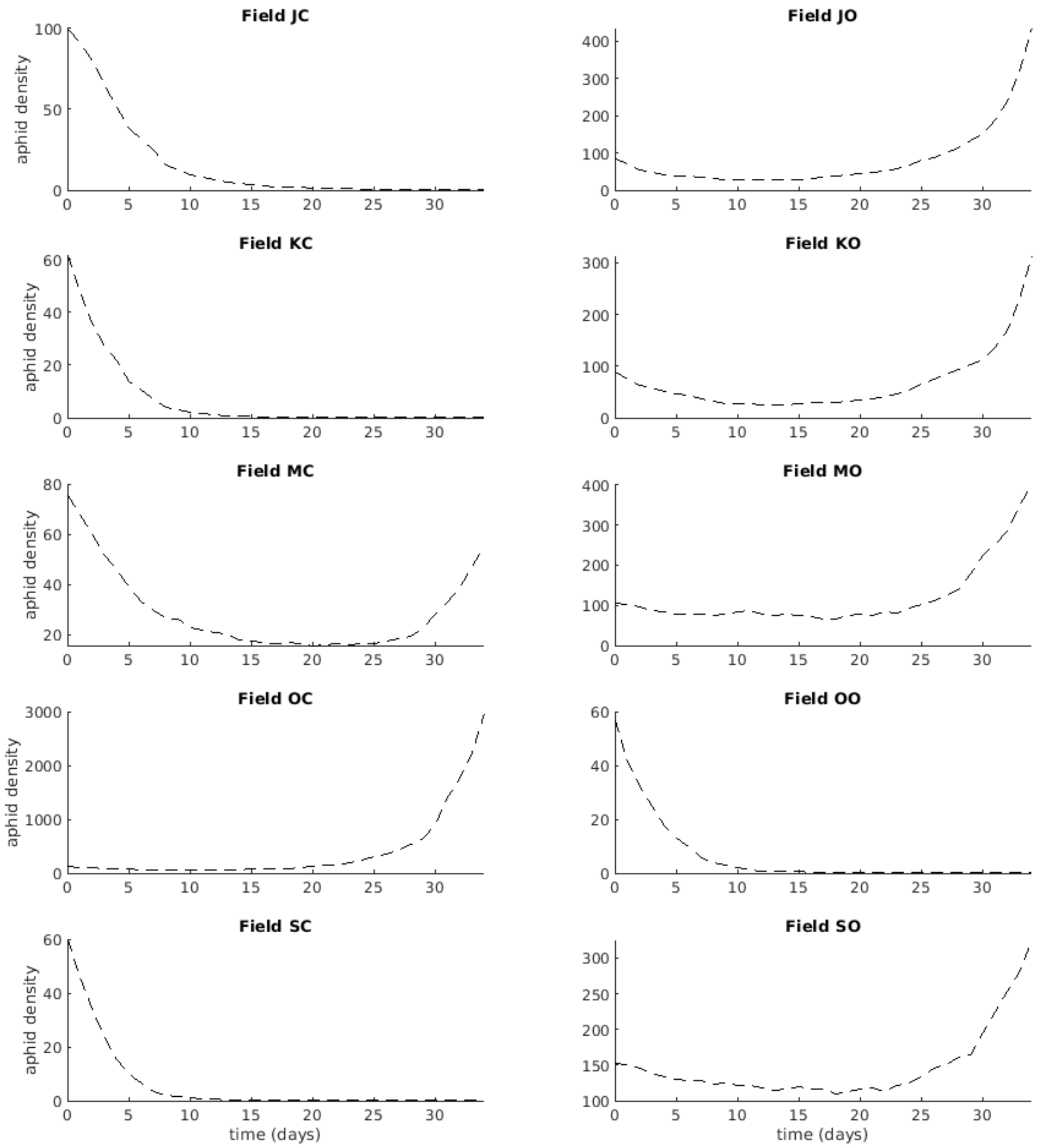

Figure S10. Model for aphid dynamics using optimal predator community in each field, under temperatures with 10% increased variability and an average increase of $2.5°C$. Each panel represents a different field.

3 SUPPLEMENTARY TABLES

parameter	value
r_{max} $[\frac{1}{\text{day}}]$	0.5238
T_u $[^{\circ}C]$	29.96
C_1 $[^{\circ}C]$	9.871
C_2 $[^{\circ}C]$	0.0166
a_0 $[\frac{1}{\text{prey}}\frac{1}{\text{day}}\frac{1}{\text{area}}\frac{1}{\text{mg}^2}]$	9.56
ϕ	1.34
h_0 $[\frac{\text{pred}}{\text{day}}]$	0.03
b_0 $[\frac{\text{pred}}{\text{day}}]$	10.83
$R_{\text{opt}}^{\text{beetle}}$	200
$R_{\text{opt}}^{\text{spider}}$	50
x_0 $[\frac{1}{\text{day}}\frac{1}{\text{mg}^{3/4}}]$	$4e^{19.75}$

Table S1. Model parameter values. Aphid growth parameters (r_{max}, t_u, C_1, C_2) were fit to data from Asin and Pons (2001). Allometric parameters were primarily estimated in Wootton et al. (2022), although R_{opt} values are rounded due to species differences. The value for x_0 was selected in Laubmeier et al. (2020), to balance metabolic penalties for body size across a 30 day simulation; due to the short duration of mesocosm experiments, this parameter was not estimated in Wootton et al. (2022). Parameter units are given in square brackets, and dimensionless parameters have no units

Predator	c_1	c_2	k_1 $[\frac{1}{^{\circ}C}]$	k_2 $[\frac{1}{^{\circ}C}]$	T_a $[^{\circ}C]$	T_b $[^{\circ}C]$
Pterostichus	4.2	3	-0.2	0.15	15	24
Harpalus	2.6	3	-0.2	0.15	13	27
Poecilus	1.8	3	-0.2	0.15	9	28
Other Carabid	2.6	3	-0.2	0.15	8	22
Bembidion	2.0	3	-0.2	0.15	12	30
Other Spider	2.1	5	-0.2	0.15	20	38
Lycosidae	2.1	5	-0.2	0.15	20	38
Tetragnathidae	1.5	5	-0.3	0.15	15	25
Linyphiidae	0.5	5	-0.3	0.15	10	30

Table S2. Predator activity parameter values, selected to create curves with ranges aligned with literature reported in Table 1 of the main text. Parameter units are given in square brackets, and dimensionless parameters have no units.

4 SUPPLEMENTARY CODE

MATLAB code (.m) can be downloaded at https://github.com/anlaubmeier/optpredtemps.git. It is also pasted below in case of any future linking issues.

```
function tempPredOpt
close all, clear all,

%firstrun;
load('constantvariables',...
    'a0','phi','h0','b0','x0','Ropts',...
    'R','I','W','N0s','names','fields',...
    'a','ah','x','Nu','tBd');
```

```
% for k=1:length(fields)
%     A=load(['Templog numDate comma ',fields{k},'.csv']); %load raw data
%     Ts=A(:,2);
%     t=A(:,1)-2;
%     [B,P,J]=solveProb(a,ah,b0,R,x,Nu,tempdat2fun(t,Ts),tBd,N0s{k},W);
%     figure, bar(P),
%     Tiv=tempchange(t,Ts,2.5,1.1);
%     [Biv,Piv,Jiv]=solveProb(a,ah,b0,R,x,Nu,tempdat2fun(t,Tiv),...
tBd,N0s{k},W);
%     Bs=[B, Biv];
%     Ps=[P, Piv];
%     Js=[J, Jiv];
%     save(['optimLump',fields{k}],'Bs','Ps','Js');
% end

%[Bs,Ps,Js]=solveMultField(a,ah,b0,R,x,Nu,tBd,N0s,W,fields)
%save('optimLumpAll','Bs','Ps','Js');

Tspan=5:2.5:37.5;
Bs=zeros(8,length(Tspan));
Ps=Bs;
Js=zeros(length(Tspan),1);

A0=0;
for k=1:length(fields)
    N0=N0s{k};
    A0=N0(1)+A0;
    P0=N0(2:end);
end
A0=A0/length(fields);
N0=[A0; P0];

for j=1:length(Tspan)
    Tc=Tspan(j);
    Tf=@(T) Tc;
    [B,P,J]=solveProb(a,ah,b0,R,x,Nu,Tf,tBd,N0,W);
    Bs(:,j)=B;
    Ps(:,j)=P;
    Js(j)=J;
    save('optimLumpTemps','Bs','Ps','Js');
end

end

%% Minimization functions
```

```
function [yOpt,pOpt,J]=solveProb(a,ah,b0,R,x,Nu,Tf,tBd,N0,W)

opts = optimoptions('fmincon','Algorithm','interior-point');
th0=N0(2:end); %true predator community
Acon=W(2:end);
Bcon=Acon*th0;

thLB=zeros(size(th0)); %lower bound is 'none of a predator'
thUB=ones(size(th0))*Bcon; %upper bound is 'total size of true community'

problem=createOptimProblem('fmincon','objective',...
    @(th) controlCost(a,ah,b0,R,x,Nu,Tf,[N0(1);th],tBd),...
    'x0',th0,'Aeq', Acon, 'beq',Bcon, 'lb',thLB, 'ub',thUB,'options',opts);
ms=MultiStart;
[yOpt,J]=run(ms,problem,20);

pOpt=round(100*(yOpt./sum(yOpt)));
%report 'percent of optimal predator community'

end

function [Bs,Ps,Js]=solveMultField(a,ah,b0,R,x,Nu,tBd,N0s,W,fields)
opts = optimoptions('fmincon','Algorithm','interior-point');
N0=N0s{1}; %doesn't matter which we use for this part
th0=N0(2:end); %true predator community
Acon=W(2:end);
Bcon=Acon*th0;

thLB=zeros(size(th0)); %lower bound is 'none of a predator'
thUB=ones(size(th0))*Bcon; %upper bound is 'total size of true community'

Tfs={};
Tivfs={};
A0s=zeros(length(fields),1);
for k=1:length(fields)
    T=load(['Templog numDate comma ',fields{k},'.csv']); %load raw data
    Ts=T(:,2);
    t=T(:,1)-2;
    Tfs{k}=tempdat2fun(t,Ts);
    Tiv=tempchange(t,Ts,2.5,1.1);
    Tivfs{k}=tempdat2fun(t,Ts);
    N0=N0s{k};
    A0s(k)=N0(1);
end
```

```
problem=createOptimProblem('fmincon','objective',...
    @(th) multCost(a,ah,b0,R,x,Nu,Tfs,th,A0s,tBd,fields),...
    'x0',th0,'Aeq', Acon, 'beq',Bcon, 'lb',thLB, 'ub',thUB,'options',opts);
ms=MultiStart;
[yOpt,J]=run(ms,problem,20);

pOpt=round(100*(yOpt./sum(yOpt)));
%report 'percent of optimal predator community'

problem=createOptimProblem('fmincon','objective',...
    @(th) multCost(a,ah,b0,R,x,Nu,Tivfs,th,A0s,tBd,fields),...
    'x0',th0,'Aeq', Acon, 'beq',Bcon, 'lb',thLB, 'ub',thUB,'options',opts);
ms=MultiStart;
[yOptiv,Jiv]=run(ms,problem,20);

pOptiv=round(100*(yOpt./sum(yOpt)));
%report 'percent of optimal predator community'

Bs=[yOpt, yOptiv];
Ps=[pOpt, pOptiv];
Js=[J, Jiv];
end

function [J,t,ys]=multCost(a,ah,b0,R,x,Nu,Tfs,N0,A0s,tBd,fields)
J=0;
for k=1:length(fields)
    Tf=Tfs{k};
    [t,ys]=ode45(@atnODE,tBd(1):1:tBd(2),[A0s(k); N0],[],a,ah,b0,R,x,Nu,Tf);
    J=J+sum(ys(:,1))/length(t); %compute average daily prey abundance
end
%could also have minimized:
%number of aphids at end of season: ys(end,1)
%maximum number of aphids within season: max(ys(:,1))
end

function [J,t,ys]=controlCost(a,ah,b0,R,x,Nu,Tf,N0,tBd)
[t,ys]=ode45(@atnODE,tBd(1):1:tBd(2),N0',[],a,ah,b0,R,x,Nu,Tf);
%solve the ODE
J=sum(ys(:,1))/length(t); %compute average daily prey abundance
%could also have minimized:
%number of aphids at end of season: ys(end,1)
%maximum number of aphids within season: max(ys(:,1))
end

%% Model functions
```

```
function dy=atnODE(t,y,a,ah,b0,R,x,Nu,Tf)
T=Tf(t);
nu=Nu(T);
r=zeros(size(y));
r(1)=R(T);
x=x*exp(-.69/(8.917e-5*(T+273.15)));
%compute metabolic death at this temperature
a=a.*nu;
ah=ah.*nu;
F=ones(size(y))+(ah'+b0*a)*y; %compute functional response
effy=y./F; %scale predators by response
dy=y.*(r-a*effy-x'); %compute ODE
end

function tempfun=tempdat2fun(xipl, yipl)
 tempstring=['@(t) ',num2str(yipl(1)),'.*(t<=',num2str(xipl(1)),')'];
for jj=2:length(yipl)
    newfun=[num2str(yipl(jj)),'.*','((',num2str(xipl(jj-1)),'<t).*...
    (t<=',num2str(xipl(jj)),'))'];
    tempstring=[tempstring,' + ',newfun];
end
tempstring=[tempstring,' + ',num2str(yipl(jj)),'.*(t>',num2str(xipl(jj)),...
')'];
tempfun=str2func(tempstring);
end

function Tiv=tempchange(t,T,m,v)
tcurr=t(1);
tmax=ceil(t(end));
ts=tmax-tcurr;
Dev=[];
Avg=[];
count=1;
for j=1:ts
    Tsum=0;
    day=floor(tcurr);
    Ts=[];
    Tnum=1;
    while tcurr<day+1
        Tsum=Tsum+T(count);
        Ts=[Ts, T(count)];
        count=count+1;
        Tnum=Tnum+1;
        tcurr=t(count);
```

```
        if count==length(t)
            Tsum+T(count);
            Ts=[Ts, T(count)];
            Tnum=Tnum+1;
            tcurr=day+1;
        end
    end
    Tsum=Tsum/(Tnum-1);
    Ts=Ts-Tsum;
    Ta=Tsum*ones(size(Ts));

    Dev=[Dev;Ts'];
    Avg=[Avg; Ta'];
end

Tiv=(Avg+m)+v*Dev;
end

%% First Run Functions (parameterization)

function firstrun
%parameter values from Wootton 2022
a0=9.56; phi=1.34; h0=0.03; b0=10.83; x0=4*exp(19.75);
Ropts=[1, 200, 200, 200, 200, 200, 50, 50, 50];
% aphid growth from Asin 2001
R=aphgrowthfun;
[I,W,N0,names,fields]=genFields;
[a,ah,x]=atnParams(I,W,a0,phi,h0,Ropts,x0);
Nu=habitatfun;
tBd=[0,34];

N0s=aphabunfit(fields,b0,R,N0,a,ah,x,Nu,tBd);

save('constantvariables',...
    'a0','phi','h0','b0','x0','Ropts',...
    'R','I','W','N0s','names','fields',...
    'a','ah','x','Nu','tBd');
end

function [a,ah,x]=atnParams(I,W,a0,phi,h0,Ropt,x0)
W2=W.^(1/2);
W4=W.^(1/4); %powers appearing in model
A=(W4'*W4).*I; %encounter term for aij.
%I makes this zero if no consumption occurs
H=repmat(W2',1,length(I)).*I; %for handling time
```

```
R=((W.^(-1))'*W).*I; %success term for aij
X=W.^(3/4); %metabolic death term

Rmat=repmat(Ropt,length(Ropt),1); %stretch out Ropt values
Rm=R./Rmat; %compute Wj/Wi/Ropt
rat=(Rm.*exp(1-Rm)).^phi; %compute success curve in aij
a=a0.*A.*rat; %compute aij
ah=h0.*a0.*H.*rat; %compute aij*hij
x=x0*X; x(1)=0; %compute death rate xj, but ignore for prey
end

function Nu=habitatfun

f2=@(T) max((4.2.*(1./(1+3.*exp(-.2.*(T-15)))).*(1-exp(.15.*(T-24)))),0);%pte
f3=@(T) max((2.6.*(1./(1+3.*exp(-.2.*(T-13)))).*(1-exp(.15.*(T-27)))),0);%har
f4=@(T) max((1.8.*(1./(1+3.*exp(-.2.*(T-9)))).*(1-exp(.15.*(T-28)))),0);%poec
f5=@(T) max((2.6.*(1./(1+3.*exp(-.2.*(T-8)))).*(1-exp(.15.*(T-22)))),0);%cara
f6=@(T) max((2.*(1./(1+3.*exp(-.2.*(T-12)))).*(1-exp(.15.*(T-30)))),0);%bembi
f7=@(T) max((2.1.*(1./(1+5.*exp(-.2.*(T-20)))).*(1-exp(.15.*(T-38)))),0);%lyc
f8=@(T) max((1.5.*(1./(1+5.*exp(-.3.*(T-15)))).*(1-exp(.15.*(T-25)))),0); %te
f9=@(T) max((.5.*(1./(1+5.*exp(-.3.*(T-10)))).*(1-exp((.15.*(T-30))))),0); %I

% %foraging similarity between prey and predators
Nu=@(T) [1 f2(T) f3(T) f4(T) f5(T) f6(T) f7(T) f8(T) f9(T);...
    f2(T) f2(T).*f2(T) f2(T).*f3(T) f2(T).*f4(T) f2(T).*f5(T) f2(T).*f6(T)...
    f2(T).*f7(T) f2(T).*f8(T) f2(T).*f9(T);...
    f3(T) f3(T).*f2(T) f3(T).*f3(T) f3(T).*f4(T) f3(T).*f5(T) f3(T).*f6(T)...
    f3(T).*f7(T) f3(T).*f8(T) f3(T).*f9(T);...
    f4(T) f4(T).*f2(T) f4(T).*f3(T) f4(T).*f4(T) f4(T).*f5(T) f4(T).*f6(T)...
    f4(T).*f7(T) f4(T).*f8(T) f4(T).*f9(T);...
    f5(T) f5(T).*f2(T) f5(T).*f3(T) f5(T).*f4(T) f5(T).*f5(T) f5(T).*f6(T)...
    f5(T).*f7(T) f5(T).*f8(T) f5(T).*f9(T);...
    f6(T) f6(T).*f2(T) f6(T).*f3(T) f6(T).*f4(T) f6(T).*f5(T) f6(T).*f6(T)...
    f6(T).*f7(T) f6(T).*f8(T) f6(T).*f9(T);...
    f7(T) f7(T).*f2(T) f7(T).*f3(T) f7(T).*f4(T) f7(T).*f5(T) f7(T).*f6(T)...
    f7(T).*f7(T) f7(T).*f8(T) f7(T).*f9(T);...
    f8(T) f8(T).*f2(T) f8(T).*f3(T) f8(T).*f4(T) f8(T).*f5(T) f8(T).*f6(T)...
    f8(T).*f7(T) f8(T).*f8(T) f8(T).*f9(T);...
    f9(T) f9(T).*f2(T) f9(T).*f3(T) f9(T).*f4(T) f9(T).*f5(T) f9(T).*f6(T)...
    f9(T).*f7(T) f9(T).*f8(T) f9(T).*f9(T)];
end

function [IM,W,y0,spec,fields]=genFields
%function outputs quantities needed for ATN model solutionos:
% interaction matrix (IM), vector of masses (W),
```

```
% initial abundancecs (y0), and names for all species (spec).

fields={'JC','JO','KC','KO','MC','MO','OC','OO','SC','SO'}; %field refs
specs={'Aphid','Bembidion','Harpalus','Poecilus','Pterostichus','Other_Carab
    'Linyphiidae','Lycosidae','Tetragnathidae','Other_Spider'}; %pred refs

Wtot=zeros(1,9); ytot=zeros(9,1); %initialize vectors

for k=1:length(fields) %loop over fields
    Wfull=load(['./AlvaDat/BM FoodWeb Fulltime ',fields{k},'.csv']); %load th
    W=[Wfull(1),Wfull(6:13)']; %save masses for prey, beetles, and spiders
    W(8)=(W(8)+Wfull(14))/2; %combine lycosidae with other spiders
    y0=zeros(length(W)+1,1); %initialize vector
    for j=2:length(specs) %loop over predators
        dats=load(['./AlvaDat/Density numDate ',specs{j},' Fulltime ',...
        fields{k},'.csv']); %load abundances
        pop=dats(:,2); %discard times
        averagePop=mean(pop); %compute average abundance
        y0(j)=averagePop; %save this predator's average abundance
    end
    y0(8)=(y0(8)+y0(10))/2; %combine lycosidae with other spiders
    y0=y0(1:end-1); %only keep information for the named predators
    aphs=load(['./AlvaDat/Density numDate Aphid Shifted Truncated ',...
    fields{k},'.csv']); %load abundances
    y0(1)=aphs(1,2); %save initial abundance
    Wtot=Wtot+W; %add up the masses we've found
    ytot=ytot+y0; %add up the initial abundances we've found
end
Wtot=Wtot./length(fields); %compute average mass over all fields

[WB,Bind]=sort(-1*Wtot(2:6)); %arrange beetles by size (easier to view)
[WS,Sind]=sort(-1*Wtot(7:end)); %arrange spiders by size (easier to view)
W=[Wtot(1),-1*WB,-1*WS]; %re-pack all species

ytot=ytot./length(fields); %compute average init. abund. over all fields
yB=ytot(2:6); yS=ytot(7:end); %separate beetles/spiders
y0=[ytot(1);yB(Bind);yS(Sind)]; %pack using ordering from above (by mass)

specs={'Aphid','Bembidion','Harpalus','Poecilus',...
'Pterostichus','Other Carabid',...
    'Linyphiidae','Lycosidae','Tetragnathidae'}; %readable species names

spec={specs{Bind(1)+1},specs{Bind(2)+1},specs{Bind(3)+1},specs{Bind(4)+1},..
specs{Bind(5)+1},...
    specs{Sind(1)+6},specs{Sind(2)+6},specs{Sind(3)+6}};
```

```
%pack using ordering from above (by mass)

IMfull=load('./AlvaDat/FoodWeb.csv'); %load interaction matrix
IMhalf=[IMfull(1,:);IMfull(6:13,:)]; %save the part for prey, beetles, spider
IM=[IMhalf(:,1),IMhalf(:,6:13)];
IM(9,2)=1; %real foodweb introduced inconsistencies
%the only non-consumption is Bembidion won't eat tetragnathidae,
%allow this link to make comparison possible between groups
%we don't need to reshape IM because it is all 1's except for aphid
end

function N0s=aphabunfit(fields,b0,R,N0,a,ah,x,Nu,tBd)
N0s={};
x0s=[95, 80, 55, 90, 75, 100, 120, 55, 50, 175];
%inital minimization ran over wide bounds
%but narrowing in now to get closer min

%aphid initial conditions to match observed abundances
for k=1:length(fields)
    T=load(['Templog numDate comma ',fields{k},'.csv']); %load raw data
    Ts=T(:,2);t=T(:,1)-2;
    Tfun=tempdat2fun(t,Ts);
    aphs=load(['./AlvaDat/Density numDate Aphid Shifted Truncated ',...
    fields{k},'.csv']); %load abundances
    if k==1
        aphs=aphs(1:end-1,:); %JC has more data than our temps
    elseif k==7
        aphs=aphs(1:end-1,:); %OC has more data than our temps
    end
    [A0,J]=fmincon(@(th) aphscale(th,aphs,Tfun,b0,R,N0,a,ah,x,Nu,tBd),x0s(k),
    [],[],[],[],x0s(k)-25,x0s(k)+25)
    N0s{k}=[A0; N0(2:end)];
end

end

function R=aphgrowthfun
Ts=[18, 22, 25, 27.5, 30]; %Asin, Pons, Env Ent 2001
rs=[0.26, 0.37, 0.45, 0.52, 0.05];
dat=[Ts; rs];
th0=[.55; 29; .5; .5];
thL=[.3; 25; .0001; .0001];
thU=[.7; 33; 10; 10];
th=fmincon(@(th) aphgrowthfit(dat, th(1), th(2), th(3), th(4)),...
    th0, [], [], [], [], thL, thU);
```

```
R=@(T) th(1)*(exp(-.5*((T-th(2))/th(3)).^2).*(T<=th(2))+...
    exp(-.5*((T-th(2))/th(4)).^2).*(T>th(2)));
end

function J=aphgrowthfit(dat, rm, tu, c1, c2)
Rtf=@(T) rm*(exp(-.5*((T-tu)/c1).^2).*(T<=tu)+...
    exp(-.5*((T-tu)/c2).^2).*(T>tu));
Rdat=Rtf(dat(1,:));
J=sum((Rdat-dat(2,:)).^2);
end

function diff=aphscale(A0,aphs,Tfun,b0,R,N0,a,ah,x,Nu,tBd)
[~,y]=ode45(@atnODE,tBd(1):1:tBd(2),[A0; N0(2:end)],[],a,ah,b0,R,x,Nu,Tfun);
diff=sum((y(aphs(:,1),1)-aphs(:,2)).^2);
end
```

REFERENCES

Asin, L. and Pons, X. (2001). Effect of high temperature on the growth and reproduction of corn aphids (homoptera: Aphididae) and implications for their population dynamics on the northeastern iberian peninsula. *Environmental Entomology* 30, 1127–1134

Curtsdotter, A., Banks, H. T., Banks, J. E., Jonsson, M., Jonsson, T., Laubmeier, A. N., et al. (2019). Ecosystem function in predator–prey food webs—confronting dynamic models with empirical data. *Journal of Animal Ecology* 88, 196–210

Laubmeier, A. N., Rebarber, R., and Tenhumberg, B. (2020). Towards understanding factors influencing the benefit of diversity in predator communities for prey suppression. *Ecosphere* 11, e03271

Wootton, K. L., Curtsdotter, A., Jonsson, T., Banks, H., Bommarco, R., Roslin, T., et al. (2022). Beyond body size—new traits for new heights in trait-based modelling of predator-prey dynamics. *PloS one* 17, e0251896