Additional file 1: R codes for simulation experiments 1 and 2

```
###### load libraries
library(truncnorm)
library(changepoint)

################################################
### Simulation experiment 1 (Exp 1)
################################################

set.seed(101)

seg.length = c(90, 5, 20, 2, 100, 6, 3, 139)
exp.cpt = c(90,95,115, 117, 217, 223, 226)

mean.mig = matrix(NA,1000,2)
sd.mig = matrix(NA,1000,2)

mean.mvt = matrix(NA,1000,2)
sd.mvt = matrix(NA,1000,2)

mean.stag = matrix(NA,1000,4)
sd.stag = matrix(NA,1000,4)

mean.cat = matrix(NA, 1000,8)
sd.cat = matrix(NA, 1000,8)

seg.tot = matrix(NA,1000,365)
nb.ncpts = rep(NA,1)
loc.cpts = rep(NA,1)

#### constrain max daily distance for segment staging (stag), non-migratory
#### movement (mvt), migratory movement (mig): stag = 20   mvt = 100
#### mig = 800
ub.cat =   c(20, 100, 20, 800, 20,  100, 800, 20)

for (n in 1:1000)

  {
#sample mean and sd for each segment
# 1-mig = seg 4 and 7
mean.mig[n,] = sample(200:400, 2, replace=TRUE)
sd.mig[n,] = sample(10:200,2, replace=TRUE)

# 2-staging = seg 1, 3, 5, 8
mean.stag[n,] = sample(0.2: 2, 4, replace=TRUE)
sd.stag[n,] = sample(0.01: 4,4, replace=TRUE)
```

```
# 3- mvt = seg 2 and 6
mean.mvt[n,] = sample(20:50, 2, replace=TRUE)
sd.mvt[n,] = sample(5:10,2, replace=TRUE)

mean.cat[n,] = c(mean.stag[n,1], mean.mvt[n,1], mean.stag[n,2],
mean.mig[n,1], mean.stag[n,3],  mean.mvt[n,2], mean.mig[n,2], mean.stag[n,4])
sd.cat[n,] = c(sd.stag[n,1], sd.mvt[n,1], sd.stag[n,2], sd.mig[n,1],
sd.stag[n,3], sd.mvt[n,2], sd.mig[n,2], sd.stag[n,4])

############ sample each of the 8 distributions to create each segment

old.seg = rep(NA,1)

  for (x in 1:8)
  {
    seg = rtruncnorm(n=seg.length[x], a=0, b=ub.cat[x], mean=mean.cat[n,x],
            sd=sd.cat[n,x])
    old.seg = append(seg,old.seg)
  }

new.seg.rev = rev(old.seg)
new.seg.rev = new.seg.rev[-1]
seg.tot[n,] = new.seg.rev

###### Run the PELT on each of the 1000 times series (iterations)
dist.pelt=cpt.var(seg.tot[n,],method='PELT')

###### Store the position of all detected change points
nb.ncpts = append(ncpts(dist.pelt), nb.ncpts)
loc.cpts = append(cpts(dist.pelt),loc.cpts)

}

nb.ncpts = nb.ncpts[-length(nb.ncpts)]
loc.cpts = loc.cpts[-length(loc.cpts)]

### Results: occurrences of each expected change point within the 1000
### iterations

perc.90 = length(loc.cpts[which(loc.cpts == 90)])/1000
perc.95 = length(loc.cpts[which(loc.cpts == 95)])/1000
perc.115 = length(loc.cpts[which(loc.cpts == 115)])/1000
perc.117 = length(loc.cpts[which(loc.cpts == 117)])/1000
perc.217 = length(loc.cpts[which(loc.cpts == 217)])/1000
perc.223 = length(loc.cpts[which(loc.cpts == 223)])/1000
perc.226 = length(loc.cpts[which(loc.cpts == 226)])/1000
```

```
##########################################
### Simulation experiment 2 (Exp 2)
##########################################

set.seed(145)

seg.length = c(90, 5, 20, 2, 100, 6, 3, 139)

mean.mig = matrix(NA,1000,2)
sd.mig = matrix(NA,1000,2)

mean.mvt = matrix(NA,1000,2)
sd.mvt = matrix(NA,1000,2)

mean.stag = matrix(NA,1000,4)
sd.stag = matrix(NA,1000,4)

mean.cat = matrix(NA, 1000,8)
sd.cat = matrix(NA, 1000,8)
seg.tot = matrix(NA,1000,365)

#### constrain max daily distance for segment staging (stag), non-migratory
#### movement (mvt), migratory movement (mig): stag = 52   mvt = 262
#### mig = 850
ub.cat =   c(52, 262, 52, 850, 52,  262, 850, 52)

nb.ncpts = rep(NA,1)
loc.cpts = rep(NA,1)

for (n in 1:1000)

  {
#sample mean and sd for each segment
# 1-mig = seg 4 and 7
mean.mig[n,] = sample(99.091:466.22, 2)
sd.mig[n,] = sample(9.898:600,2)

# 2-staging = seg 1, 3, 5, 8
mean.stag[n,] = sample(0.12: 14.59, 4)
sd.stag[n,] = sample(0.12: 9.89,4)

# 3- mvt = seg 2 and 6
mean.mvt[n,] = sample(18.35:99.09, 2)
sd.mvt[n,] = sample(9.898:129.84,2)

mean.cat[n,] = c(mean.stag[n,1], mean.mvt[n,1], mean.stag[n,2],
mean.mig[n,1], mean.stag[n,3],  mean.mvt[n,2], mean.mig[n,2], mean.stag[n,4])
```

```
sd.cat[n,] = c(sd.stag[n,1], sd.mvt[n,1], sd.stag[n,2], sd.mig[n,1],
sd.stag[n,3], sd.mvt[n,2], sd.mig[n,2], sd.stag[n,4])

############ sample each of the 8 distributions to create each segment

old.seg = rep(NA,1)

  for (x in 1:8)
  {
    seg = rtruncnorm(n=seg.length[x], a=0, b=ub.cat[x], mean=mean.cat[n,x],
            sd=sd.cat[n,x])
    old.seg = append(seg,old.seg)
  }

new.seg.rev = rev(old.seg)
new.seg.rev = new.seg.rev[-1]
seg.tot[n,] = new.seg.rev

###### Run the PELT on each of the 1000 times series (iterations)
dist.pelt=cpt.var(seg.tot[n,],method='PELT')

###### Store the position of all detected change points
nb.ncpts = append(ncpts(dist.pelt), nb.ncpts)
loc.cpts = append(cpts(dist.pelt),loc.cpts)

}

nb.ncpts = nb.ncpts[-length(nb.ncpts)]
loc.cpts = loc.cpts[-length(loc.cpts)]
```