vimesMulti vignette

Sarah Hayes

2024-06-21

## vimesMulti

vimesMulti is a package for detecting disease outbreak clusters from temporal and spatial suveillance data using group-specific epidemiological parameters. There are four basic functions within the package. This vignette illustrates a basic workflow.

Load the package

devtools::install\_github("sarahhayes/vimesMulti",build = TRUE)  
library(vimesMulti)  
library(ggplot2)  
library(tidyverse)

The cluster detection method relies on cut-off values for pruning the graphs (for full details see….) These cut-off values are crucial in determining the size and composition of clusters.

*get\_quantiles\_multi* uses a simulation-based approach to generate these cut-off values using user-defined values for the number of cases in each group and user-specified values for epidemiological parameters.

The user enters:

* Number of observed cases in group 1 (g1\_obs) and group 2 (g2\_obs).
* Reporting probability for group 1 (g1\_rr) and group 2 (g2\_rr).
* Number of individuals to include in the simulation (n). It is recommended to use at least 1 million and to use higher numbers if reporting probabilities are very low.
* Percentile cut-off value (q).
* Data type (d\_type) which will be one of ‘spatial’ or ‘temporal’
* Distribution (distrib). This can be either ‘gamma’ or ‘lognormal’ for temporal data and ‘gamma’ or ‘rayleigh’ for spatial data.
* The summary statistics (mean and standard deviation) for the data. These may be the same for all transmission types or may vary by transmission types.
* Parameter vector (params\_temporal). In this vector, the summary statistics for each of the transmission types are assigned in the order of: transmission between members of group 1 (g1-g1), transmission between groups 1 and 2 (g1-g2 or g2-g1), transmission between members of group 2 (g2-g2). Summary statistics for spatial data should be entered in kilometres.
* Assortativity parameter (assort\_mix). This parameter assigns the degree of assortative transmission between groups. At 1, this assumes random transmission. As the assortativity parameter increases, the level of assortative transmission increases.

Example of generating cut-off values for temporal data using the lognormal distribution

g1\_obs <- 30  
g2\_obs <- 20  
g1\_rr <- 0.75  
g2\_rr <- 0.50  
n = 10000000  
q <- 0.95  
si\_mean <- 27.8  
si\_sd <- 36.8  
params\_temporal <- c(si\_mean, si\_sd, si\_mean, si\_sd, si\_mean\*2, si\_sd\*2)  
assort\_mix <- 1  
  
temporal\_cut\_off <- get\_quantiles\_multi(d\_type = "temporal", distrib = "lognormal",  
 obs = c(g1\_obs, g2\_obs), rr = c(g1\_rr, g2\_rr),  
 params = params\_temporal,  
 n = n, q = q, assort\_mix = assort\_mix)

Example of generating cut-off values for spatial data using the Rayleigh distribution. The Rayleigh distribution is a one-parameter distribution and so only the mean of the transmission distance is entered and NA entered for the standard deviation. Distances should be entered in kilometres.

dist\_mean <- 0.87  
params\_spatial <- c(dist\_mean, NA, dist\_mean, NA, dist\_mean, NA)  
  
  
spatial\_cut\_off <- get\_quantiles\_multi(d\_type = "spatial", distrib = "rayleigh",  
 obs = c(g1\_obs, g2\_obs), rr = c(g1\_rr, g2\_rr),  
 params = params\_spatial,  
 n = n, q = q, assort\_mix = assort\_mix)  
  
  
spatial\_cut\_off

## trans\_type threshold\_spatial proportion\_sim\_spatial  
## 1 g1g1 2.270445 0.3601802  
## 2 mixed 2.269434 0.4797613  
## 3 g2g2 2.273608 0.1600585

Alternatively, if the user already has the values they wish to use for the cut-off values, they can be manually specified using *set\_cut\_off*. When manually entering spatial cut-off values, these should be entered in kilometres.

manual\_spatial\_cut\_off <- set\_cutoff("spatial", 2,3,2)  
manual\_spatial\_cut\_off

## trans\_type threshold\_spatial proportion\_sim\_spatial  
## 1 g1g1 2 NA  
## 2 mixed 3 NA  
## 3 g2g2 2 NA

The cut-off values that have been generated for each transmission type can now be used for cluster detection.

Generate some example data

dat\_times <- c(1855, 1196, 711, 777, 780, 1332, 740, 1251, 1234, 804, 1789, 222, 345, 314, 465, 209, 253, 656,  
 1621, 455, 439, 439, 1414, 253, 896, 921, 410, 537, 954, 896, 980, 642, 119, 412, 413, 401, 401,  
 399, 398, 395, 398, 1321, 944, 709, 776, 769, 1067, 1261, 1534, 330)  
  
dat\_geo1 <- c(623626.8, 627471.6, 625338.6, 631570.6, 629642.1, 629805.5, 626784.1, 618404.2, 617202.1, 591642.1, 600856.2, 590948.1, 576259.7, 576802.9, 578319.8, 585333.7, 585286.0, 604260.9, 612423.8, 607130.3, 607068.2, 603192.5, 583602.8, 577951.7, 604808.5, 604798.9, 603662.1, 592264.8, 586373.1, 580764.4, 586428.5, 585508.1, 585520.0, 584640.1, 584640.1, 584640.1, 584640.1, 584640.1, 584640.1, 585215.0, 584640.1, 631706.5, 631875.6, 629265.6, 629382.1, 628769.7, 629680.1, 644871.1, 644841.1, 640511.1)  
  
dat\_geo2 <- c(8850794, 8864207, 8861898, 8862102, 8854782, 8864586, 8853986, 8858298, 8855642, 8860179, 8866007, 8860162, 8852528, 8853027, 8855598, 8855781, 8855480, 8868075, 8854970, 8848923, 8849040, 8843570, 8853303, 8847845, 8845847, 8845835, 8843812, 8840340, 8838607, 8833568, 8838656, 8842167, 8842038, 8841850, 8841850, 8841850, 8841850, 8841850, 8841850, 8842443, 8841850, 8849402, 8849364, 8851496, 8850060, 8849636, 8847645, 8842940, 8842928, 8850822)  
  
# Also need a vector specifying the group for each data entry  
set.seed(26)  
group\_vect <- sample(c(rep("g1", 30), rep("g2", 20)))

Use the *vimes\_multi* function to generate the clusters. The user will input: \* The temporal data (dat\_time) for each case. This should be entered as a numeric to allow computation of the distance matrix. Unit does not need to be specified but must be the same for all entries.

* The spatial data (dat\_geo\_1 and dat\_geo\_2) for each case. The package uses *rdist* from the fields package to calculate euclidean distance, thus locations should be entered in Universtal Transverse Mercator to ensure the calculated distance matrix is in metres. This is then converted to kilometres within the function.
* The temporal and spatial cut-off values that have either been generated using *get\_quantiles\_multi* or *set\_cutoff*
* A vector specifying which group each case belongs to

vimes\_multi\_results <- vimes\_multi(  
 dat\_time = dat\_times,  
 dat\_geo1 = dat\_geo1,  
 dat\_geo2 = dat\_geo2,  
 temporal\_cut\_offs = temporal\_cut\_off,  
 distance\_cut\_offs = spatial\_cut\_off,  
 group\_vect = group\_vect,  
 graph\_opt = vimes::vimes\_graph\_opt())

Summary table showing the number and proportion of each transmission type from the data clusters and from the original simulation

vimes\_multi\_results$combined\_results

## trans\_type sim\_proportion data\_count data\_proportion sim\_count  
## 1 g1g1 0.3600408 18 0.4615385 14  
## 2 mixed 0.4798512 16 0.4102564 19  
## 3 g2g2 0.1601081 5 0.1282051 6

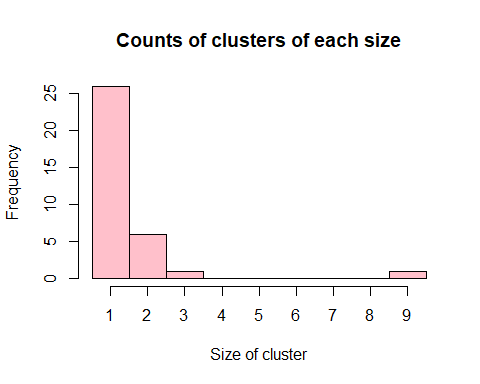
Summary table of size and composition of clusters of two or more.

vimes\_multi\_results$cluster\_size\_df

## # A tibble: 8 × 5  
## cluster\_no g1 g2 total trans\_type  
## <fct> <int> <int> <dbl> <chr>   
## 1 13 2 0 2 g1g1   
## 2 15 2 0 2 g1g1   
## 3 18 1 1 2 mixed   
## 4 19 2 0 2 g1g1   
## 5 22 1 1 2 mixed   
## 6 24 0 2 2 g2g2   
## 7 26 6 3 9 mixed   
## 8 30 1 2 3 mixed

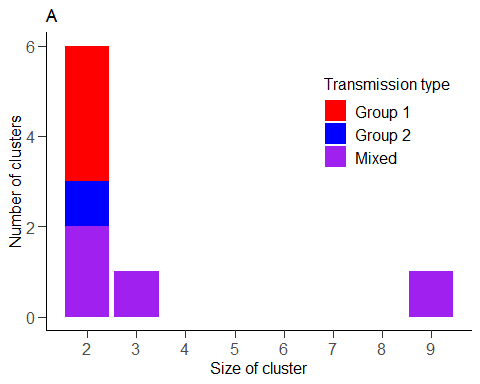
Histogram to show the cluster sizes

hist(vimes\_multi\_results$vimes\_results\_list$clusters$size, col = "pink", xlab = "Size of cluster",  
 xaxt = "n",  
 breaks = seq(0,max(vimes\_multi\_results$vimes\_results\_list$clusters$size),1),  
 main = "Counts of clusters of each size")  
axis(side = 1, at = seq(0.5,max(vimes\_multi\_results$vimes\_results\_list$clusters$size)-0.5),labels = seq(1,max(vimes\_multi\_results$vimes\_results\_list$clusters$size)))



Plot coloured by cluster composition

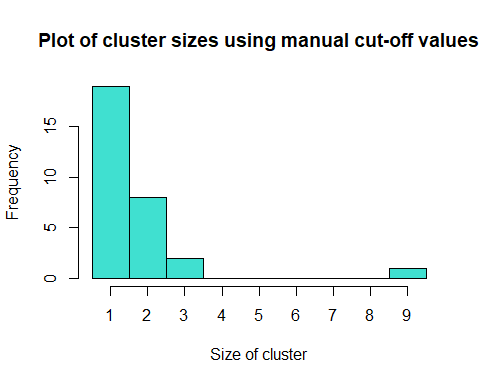
#extract the cluster details  
cluster\_counts <- vimes\_multi\_results$cluster\_size\_df %>%  
 dplyr::group\_by(total, trans\_type)%>%  
 dplyr::count()  
  
# plot of the clusters  
own\_cols <- c("red", "blue", "purple")  
own\_labels <- c("Group 1", "Group 2", "Mixed")  
  
  
gg\_res\_1 <- ggplot(cluster\_counts, aes(x = total, y = n, fill = trans\_type)) +  
 geom\_bar(stat = "identity") +  
 scale\_fill\_manual(values = own\_cols, name = "Transmission type",  
 labels = own\_labels) +  
 theme\_classic() +  
 theme(axis.title.x = element\_text(size=12, face="plain"),  
 axis.title.y = element\_text(size=12, face="plain"),  
 axis.text.x = element\_text(size = 12, face = "plain"),  
 axis.text.y = element\_text(size = 12, face = "plain"),  
 axis.ticks.length = unit(.2, "cm"),  
 legend.text = element\_text(size = 12, face = "plain"),  
 legend.title = element\_text(size = 12, face = "plain"),  
 legend.position = c(0.8,0.7)) +  
 scale\_x\_continuous(breaks = c(2,3,4,5,6,7,8,9), labels = c(2,3,4,5,6,7,8,9)) +  
 labs(title = "A", y = "Number of clusters", x = "Size of cluster")  
  
gg\_res\_1



The same process can also be undertaken using the user-defined cut-off values.

#specify the temporal cut-off values (already specified spatial cut-offs above)  
manual\_temporal\_cut\_off <- set\_cutoff("temporal",150,180,240)  
  
  
vimes\_multi\_results\_manual <- vimes\_multi(  
 dat\_time = dat\_times,  
 dat\_geo1 = dat\_geo1,  
 dat\_geo2 = dat\_geo2,  
 temporal\_cut\_offs = manual\_temporal\_cut\_off,  
 distance\_cut\_offs = manual\_spatial\_cut\_off,  
 group\_vect = group\_vect,  
 graph\_opt = vimes::vimes\_graph\_opt())

hist(vimes\_multi\_results\_manual$vimes\_results\_list$clusters$size, col = "turquoise", xlab = "Size of cluster",  
 breaks = seq(0,max(vimes\_multi\_results\_manual$vimes\_results\_list$clusters$size),1),  
 xaxt = "n",  
 main = "Plot of cluster sizes using manual cut-off values")  
axis(side = 1, at = seq(0.5,max(vimes\_multi\_results\_manual$vimes\_results\_list$clusters$size)-0.5),labels = seq(1,max(vimes\_multi\_results\_manual$vimes\_results\_list$clusters$size)))



The assortativity parameter sets the degree of assortative transmission. In the example above, we have set it to 1, which reflects random transmission between groups. Higher values will reflect more assortative transmission.

temporal\_cut\_off\_assort3 <- get\_quantiles\_multi(d\_type = "temporal", distrib = "lognormal",  
 obs = c(g1\_obs, g2\_obs), rr = c(g1\_rr, g2\_rr),  
 params = params\_temporal,  
 n = n, q = q, assort\_mix = 3)  
  
  
spatial\_cut\_off\_assort3 <- get\_quantiles\_multi(d\_type = "spatial", distrib = "rayleigh",  
 obs = c(g1\_obs, g2\_obs), rr = c(g1\_rr, g2\_rr),  
 params = params\_spatial,  
 n = n, q = q, assort\_mix = 3)

The differences in the parameter values and proportions between the random mixing scenario (assortativity parameter = 1) and that with assortativity parameter = 3 are shown below. You can see that the level of ‘mixed’ transmissions (between-group transmission) is higher when the assortativity parameter is set at a higher value

temporal\_cut\_off

## trans\_type threshold\_temporal proportion\_sim\_temporal  
## 1 g1g1 156.7844 0.3599014  
## 2 mixed 186.0619 0.4799410  
## 3 g2g2 246.7239 0.1601576

temporal\_cut\_off\_assort3

## trans\_type threshold\_temporal proportion\_sim\_temporal  
## 1 g1g1 126.5646 0.5135088  
## 2 mixed 202.6952 0.1731949  
## 3 g2g2 334.9332 0.3132963

spatial\_cut\_off

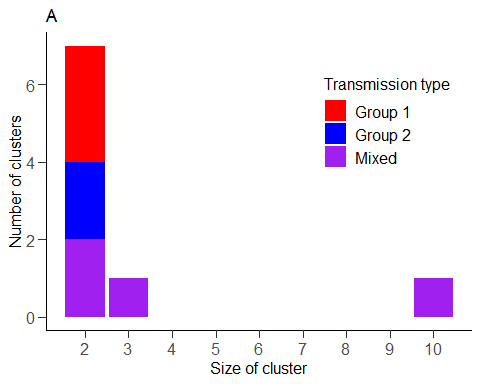
## trans\_type threshold\_spatial proportion\_sim\_spatial  
## 1 g1g1 2.270445 0.3601802  
## 2 mixed 2.269434 0.4797613  
## 3 g2g2 2.273608 0.1600585

spatial\_cut\_off\_assort3

## trans\_type threshold\_spatial proportion\_sim\_spatial  
## 1 g1g1 2.083237 0.5135143  
## 2 mixed 2.307082 0.1730194  
## 3 g2g2 2.533498 0.3134663

Changes in the cut-off values that occur as a result of changes to the assortativity parameter will affect the composition of clusters. If we rerun the script above with the new values we produced the clusters shown below. The size of the largest cluster has increased from 9 to 10 and there is a change to the composition of the clusters of 2.

## # A tibble: 9 × 5  
## cluster\_no g1 g2 total trans\_type  
## <fct> <int> <int> <dbl> <chr>   
## 1 13 2 0 2 g1g1   
## 2 15 2 0 2 g1g1   
## 3 18 1 1 2 mixed   
## 4 19 2 0 2 g1g1   
## 5 22 1 1 2 mixed   
## 6 24 0 2 2 g2g2   
## 7 26 6 4 10 mixed   
## 8 29 1 2 3 mixed   
## 9 31 0 2 2 g2g2



There is not a set range for the value of the assortativity parameter and users may wish to trial a range of values to see which reflects their situation best. The level of assortative transmission may be explored using the proportions of each type of transmission that are produced in the simulation and outputted by the *get\_quantiles\_multi* function. This can be compared to the proportion of each transmission type within the clusters.