Genetic analysis of disease outbreaks using 

Thibaut Jombart *

Imperial College London
MRC Centre for Outbreak Analysis and Modelling

November 4th 2013

Abstract

This tutorial introduces different tools, from exploratory approaches to model-based methods, for the analysis of pathogen genome data collected during disease outbreaks, using the `R` software [5]. We illustrate how different approaches including phylogenetics, genetic clustering, *SeqTrack* [3] and *outbreaker* [2] can be used to uncover the features of a disease outbreak, and possibly help designing containment strategies. This tutorial uses the packages *ape* [4] for phylogenetic analyses, *adegenet* [1] for genetic clustering and quick transmission tree reconstruction (*SeqTrack* algorithm), and *outbreaker* [2] for advanced reconstruction of epidemics. While the data and analysed outbreak are purely fictional, the methodology presented here will be useful for a range of actual disease outbreaks.

*`tjombart@imperial.ac.uk`

Contents

1 Introduction

1.1 An emerging pathogen outbreak

A new virus has just emerged in the small city of Arkham, Massachusetts (USA), causing an outbreak of a very peculiar and unique disease. The most common symptoms include dementia and to varying extent fever, resulting in frequently attempted cannibalism and subsequent isolation of the patients (Figure 1).

Figure 1: Example of a "mild" case.

Unfortunately, in a smaller number of more concerning cases the patients were seen to grow fangs, claws, and various numbers of tentacles and pseudopods, and were subsequently shot by the police forces (Figure 2). Authorities refer to the two types of cases as "mild" and "severe", respectively.

Figure 2: Example of a "severe" case.

1.2 Your objective

An expert in the analysis of disease outbreaks, you have been mandated for the analysis of the first collected data. So far, the mode of transmission of the disease is not obvious, but the pathogen has been identified as a virus, and its genome sequenced. Your task is to exploit this information to cast some light on who infected whom.

2 First look at the data

We first load the R packages used for the analysis of the data: *ape* (for phylogenetics), *adegenet* (for genetic clustering and *SeqTrack*), and *outbreaker* (for detailed outbreak reconstruction).

```
library(ape)
library(adegenet)
library(igraph)
library(outbreaker)
```

The data consist of two files: the file `cases.csv` containing descriptions of the first 30 cases sampled so far, and a DNA alignment in fasta format (`alignment.fa`) containing one viral genome sequence for each case. We read these data directly from the server where they are available, starting with cases descriptions:

```
cases <- read.csv("http://adegenet.r-forge.r-project.org/files/fakeOutbreak/cases.csv")
cases

##    id collec.dates sex age peak.fever outcome                  notes
## 1   1   2013-02-18   m  30       37.5    mild
## 2   2   2013-02-20   f  40       38.5    mild
## 3   3   2013-02-21   f  32       38.0    mild
## 4   4   2013-02-21   m  35       38.5    mild
## 5   5   2013-02-22   f   3       39.5    mild
## 6   6   2013-02-24   f  34       39.0    mild
## 7   7   2013-02-23   m  61       40.0  severe
## 8   8   2013-02-24   f  68       39.5  severe
## 9   9   2013-02-24   m  35       39.5    mild
## 10 10   2013-02-24   f  34       39.5    mild
## 11 11   2013-02-26   m  26       39.0    mild
## 12 12   2013-02-25   f  69       37.5  severe
## 13 13   2013-02-25   m  19       40.5    mild
## 14 14   2013-02-25   f  66       37.5    mild
## 15 15   2013-02-25   f   3       37.0    mild
## 16 16   2013-02-26   m  19       37.0    mild
## 17 17   2013-02-26   m  35       38.5    mild
## 18 18   2013-02-27   m  37       37.0    mild
## 19 19   2013-02-26   m  11       37.5    mild
## 20 20   2013-02-28   m  35       37.5    mild
## 21 21   2013-02-27   m  49       37.0    mild
## 22 22   2013-02-28   m  35       37.0    mild
## 23 23   2013-02-26   m  34       37.0    mild
## 24 24   2013-02-27   m  59       37.5  severe
## 25 25   2013-02-26   f  47       37.0    mild
## 26 26   2013-02-26   f  34       37.0    mild
## 27 27   2013-02-28   f  26       37.5    mild
## 28 28   2013-02-27   f  16       37.0    mild possible-contamination
## 29 29   2013-03-01   f  15       41.0    mild
## 30 30   2013-03-01   m  40       37.0    mild
```

The data contain the following fields: `id` is the identifier of the cases, `collec.dates` are collection dates (in format *yyyy-mm-dd*), the gender (`sex`) and age (`age`) of the patients, the highest temperature of the case (`peak.fever`), and the outcome of the case (`outcome`). The additional field `notes` has been used for notes on the samples, and indicates that sample 28 might have experienced DNA contamination (possible mixture of different samples).

As operations on the collection dates will be useful, we convert the dates (originally as character strings) into `Date` objects; we also create a new object `days`, which gives collection times in number of days after the first sample (which has been sampled, by definition, on day 0):

```
dates <- as.Date(cases$collec.dates)
head(dates)

## [1] "2013-02-18" "2013-02-20" "2013-02-21" "2013-02-21" "2013-02-22"
## [6] "2013-02-24"

range(dates)

## [1] "2013-02-18" "2013-03-01"

days <- as.integer(difftime(dates, min(dates), unit="days"))
days

##  [1]  0  2  3  3  4  6  5  6  6  6  8  7  7  7  7  8  8  9  8 10  9 10  8  9  8
## [26]  8 10  9 11 11
```

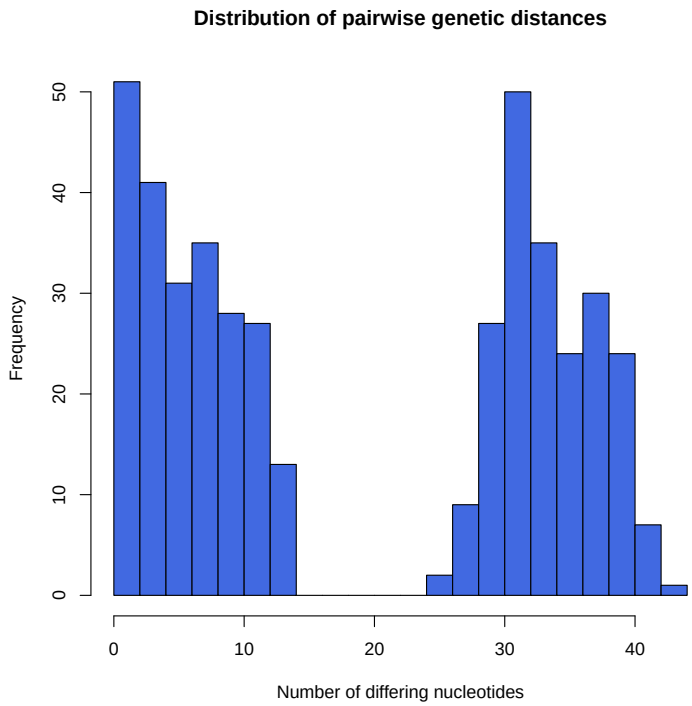
DNA sequences for the 30 cases are read from the server using `fasta2DNAbin`:

```
dna <- fasta2DNAbin("http://adegenet.r-forge.r-project.org/files/fakeOutbreak/alignment.fa")

##
##  Converting FASTA alignment into a DNAbin object...
##
##
##  Finding the size of a single genome...
##
##
##  genome size is: 10,000 nucleotides
##
## ( 168  lines per genome )
##
##  Importing sequences...
## ..........................................................
##  Forming final object...
##
## ...done.

dna

## 30 DNA sequences in binary format stored in a matrix.
##
## All sequences of same length: 10000
##
## Labels: 1 2 3 4 5 6 ...
##
## Base composition:
##     a     c     g     t
## 0.251 0.242 0.251 0.256
```

To have an idea of the existing diversity in these sequences, we compute the simple pair-wise Hamming distances and plot their distribution:

```
D <- dist.dna(dna, model="N")
```

```
hist(D, col="royalblue", nclass=30,
     main="Distribution of pairwise genetic distances",
     xlab="Number of differing nucleotides")
```



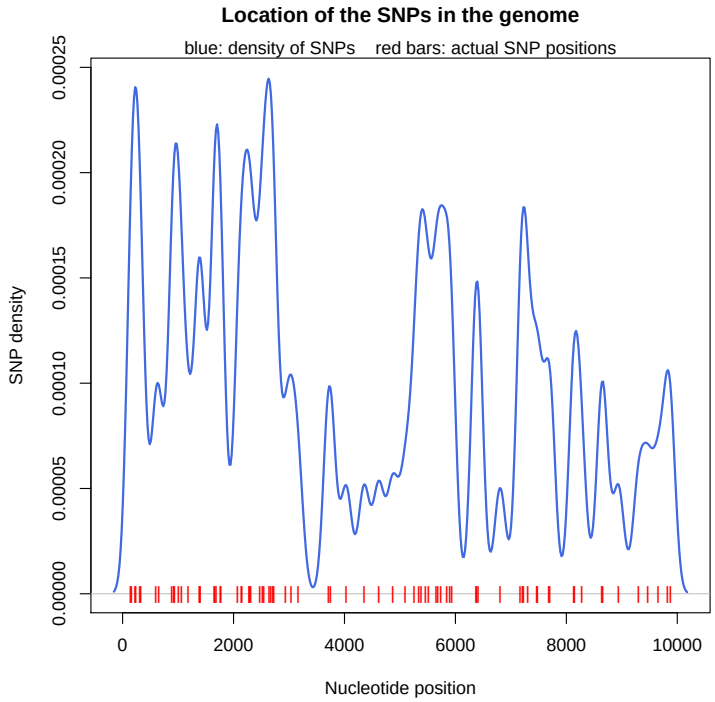
For such a small temporal scale and genome, the amount of diversity is considerable. The fact that the distribution is clearly bimodal suggests the existence of at least two clades (and possibly more).

It may be interesting to see if this remarkable polymorphism is distributed randomly across the genome. We can extract SNP positions very simply from the DNA sequences using `seg.sites`:

```
snps <- seg.sites(dna)
head(snps)

## [1] 142 161 226 236 313 331

length(snps)

## [1] 79
```

There are 79 polymorphic sites in the sample. We can visualize their position, and try to detect hotspots of polymorphism by computing the density of SNPs as we move along the genome:

```
plot(density(snps, bw=100), col="royalblue",
     xlab="Nucleotide position", ylab="SNP density",
     main="Location of the SNPs in the genome", lwd=2)
points(snps, rep(0, length(snps)), pch="|", col="red")
mtext(side=3, text="blue: density of SNPs    red bars: actual SNP positions")
```



Here, the polymorphism seems to be distributed fairly randomly.

3 Phylogenetic analysis

The genetic relationships between a set of taxa are typically inferred using phylogenetic trees. Here, we reconstruct a phylogeny using the usual Neighbour-Joining algorithm on pairwise genetic distances. As the mere number of differing nucleotides may be too crude a measure of genetic differentiation, we use Tamura and Nei's distance, which handles different rates for transitions and transversions (see `?dist.dna` for other available distances):

```
D.tn93 <- dist.dna(dna, model="TN93")
```

The package *ape* makes the construction of phylogenies from distances matrices easy; in the following, we create a Neighbour-Joining tree (`nj`) based on our new distance matrix (`D.tn93`), we root this tree to the first sample (`root`), and ladderize it to make it prettier (`ladderize`):

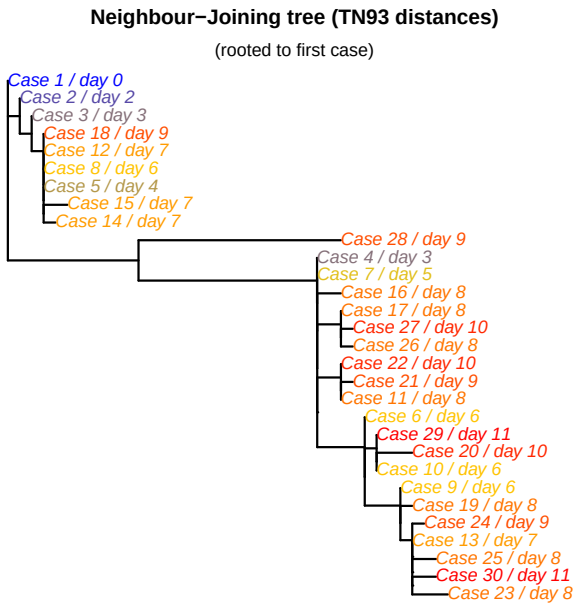
```
tre <- nj(D.tn93)
tre

##
## Phylogenetic tree with 30 tips and 28 internal nodes.
##
## Tip labels:
##  1, 2, 3, 4, 5, 6, ...
##
## Unrooted; includes branch lengths.

tre <- root(tre,1)
tre <- ladderize(tre)
```

We also rename the tips of the tree (`tre$tip.label`) to include the collection dates after the case indices:

```
tre$tip.label <- paste("Case ",1:30, " / day ", days, sep="")
```

Finally, we plot the resulting tree, using colors to represent collection dates (blue: ancient; red: recent):

```
plot(tre,edge.width=2, tip.col=num2col(days, col.pal=seasun))
title("Neighbour-Joining tree (TN93 distances)")
mtext(side=3, text="(rooted to first case)")
```



How many clades are there? Is this tree clock-like? Is rooting to the first sample appropriate? What other graphical representation would you recommend?

4 Identifying clusters of cases

Identifying clusters of cases from a phylogeny is not always straightforward. Adegenet implements a simple clustering approach based on the number of mutations separating sequences, classifying them in the same cluster if their distance is less than a given threshold. This function is called `gengraph`, and can be used with an interactive mode (by default), using:

```
clust <- gengraph(D)
```

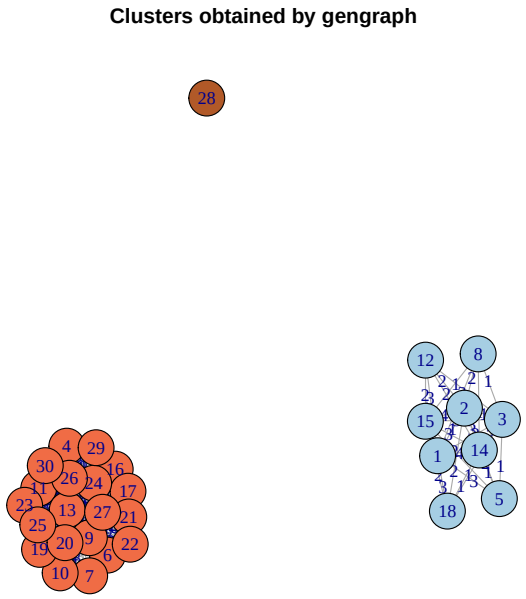
(legend: sequences are the nodes of the graphs; edges link sequences from the same cluster; numbers on the edges indicate numbers of mutations)

Try a few values; you should see that 3 groups are obtained for anything between 15 and 25 mutations, with the result looking like this:

```
clust
```

```
## $graph
## IGRAPH UNW- 30 217 --
## + attr: name (v/c), color (v/c), label (v/c), weight (e/n), label (e/n)
##
## $clust
## $clust$membership
##  [1] 1 1 1 2 1 2 2 1 2 2 2 1 2 1 1 2 2 1 2 2 2 2 2 2 2 2 2 3 2 2
##
## $clust$csize
## [1]  9 20  1
##
## $clust$no
## [1] 3
##
##
## $cutoff
## [1] 20
##
## $col
##         1         2         3
## "#A6CEE3" "#F06C45" "#B15928"
```

```
plot(clust$g, main="Clusters obtained by gengraph")
```

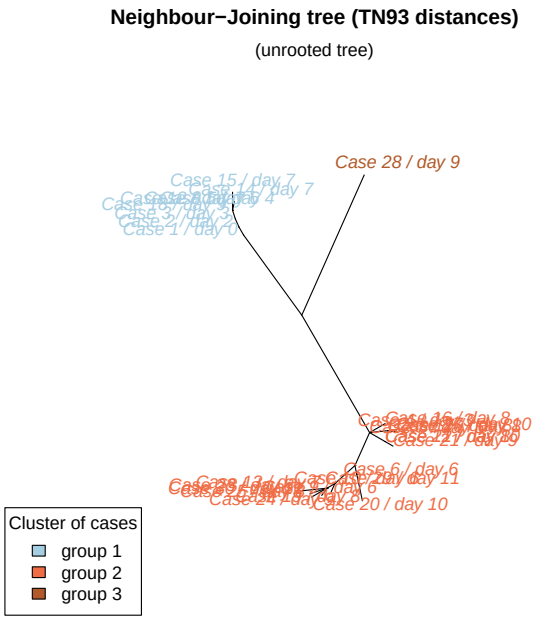


This confirms what the phylogeny suggested: there are two distinct clades, and one outlier (case 28), which is very likely an indication that this sample was indeed contaminated — as a reminder:

```
cases[28,]

##    id collec.dates sex age peak.fever outcome                  notes
## 28 28   2013-02-27   f  16         37    mild possible-contamination
```

We can verify the congruence of the groups and the phylogeny easily:

```
plot(tre, tip.color=clust$col[clust$clust$membership], type="unrooted")
title("Neighbour-Joining tree (TN93 distances)")
mtext(side=3, text="(unrooted tree)")
legend("bottomleft", fill=clust$col, legend=paste("group",1:3), title="Cluster of cases")
```



5 Analysis using *SeqTrack*

The phylogenetic tree gives us an idea of the possible chains of transmissions, but overlooks the collection dates. The *SeqTrack* algorithm has been designed to fill this gap. It aims to reconstruct ancestries between the sampled sequences based on their genetic distances and collection dates, so that the obtained tree has maximum parsimony. It is implemented in *adegenet* by the function `seqTrack` (see `?seqTrack`). Here, we use *SeqTrack* on the matrix of pairwise distances (`distmat`), indicating the labels of the cases (`x.names=cases$id`) and the collection dates (`x.dates=dates`):

```
distmat <- as.matrix(D)
sqtk.res <- seqTrack(distmat, x.names=cases$id, x.dates=dates)
class(sqtk.res)

## [1] "seqTrack"   "data.frame"

sqtk.res
```

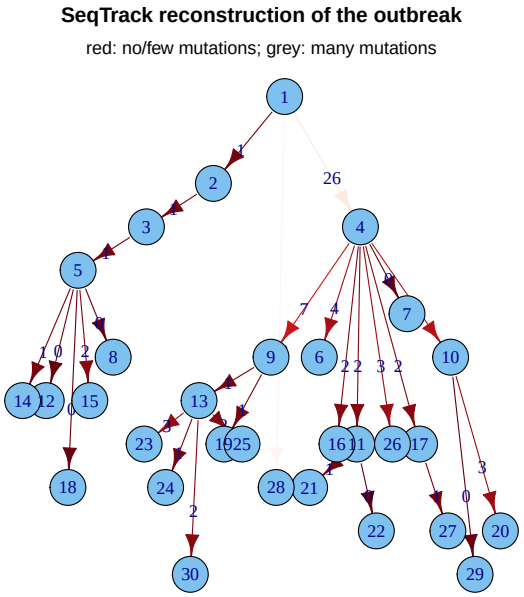
```
##    id ances weight       date ances.date
## 1   1    NA     NA 2013-02-18       <NA>
## 2   2     1      1 2013-02-20 2013-02-18
## 3   3     2      1 2013-02-21 2013-02-20
## 4   4     1     26 2013-02-21 2013-02-18
## 5   5     3      1 2013-02-22 2013-02-21
## 6   6     4      4 2013-02-24 2013-02-21
## 7   7     4      0 2013-02-23 2013-02-21
## 8   8     5      0 2013-02-24 2013-02-22
## 9   9     4      7 2013-02-24 2013-02-21
## 10 10     4      5 2013-02-24 2013-02-21
## 11 11     4      2 2013-02-26 2013-02-21
## 12 12     5      0 2013-02-25 2013-02-22
## 13 13     9      1 2013-02-25 2013-02-24
## 14 14     5      1 2013-02-25 2013-02-22
## 15 15     5      2 2013-02-25 2013-02-22
## 16 16     4      2 2013-02-26 2013-02-21
## 17 17     4      2 2013-02-26 2013-02-21
## 18 18     5      0 2013-02-27 2013-02-22
## 19 19     9      1 2013-02-26 2013-02-24
## 20 20    10      3 2013-02-28 2013-02-24
## 21 21    11      1 2013-02-27 2013-02-26
## 22 22    11      0 2013-02-28 2013-02-26
## 23 23    13      3 2013-02-26 2013-02-25
## 24 24    13      1 2013-02-27 2013-02-25
## 25 25    13      2 2013-02-26 2013-02-25
## 26 26     4      3 2013-02-26 2013-02-21
## 27 27    17      1 2013-02-28 2013-02-26
## 28 28     1     28 2013-02-27 2013-02-18
## 29 29    10      0 2013-03-01 2013-02-24
## 30 30    13      2 2013-03-01 2013-02-25
```

The result `sqtk.res` is a `data.frame` with the special class `seqTrack`, containing the following information:

- `sqtk.res$id`: the indices of the cases.
- `sqtk.res$ances`: the indices of the putative ancestors of the cases.
- `sqtk.res$weight`: the number of mutations for each putative ancestry.
- `sqtk.res$date`: the collection dates of the cases.
- `sqtk.res$ances.date`: the collection dates of the putative ancestors.

`seqTrack` objects can be plotted simply using:

```
g <- plot(sqtk.res, main="SeqTrack reconstruction of the outbreak")
mtext(side=3, text="red: no/few mutations; grey: many mutations")
```




```
g

## IGRAPH DNW- 30 29 --
## + attr: name (v/c), dates (v/n), weight (e/n), label (e/n), color (e/c)
```

Each sequence/case is a node of the graph, and arrows model putative ancestries/transmissions. The number of mutations between ancestors and descendents are indicated by the color of the arrows (red = no/few mutations; light grey= many mutations) and the numbers in blue. Time is represented on the y axis (up: ancient; down: recent). Note that the function `plot` invisibly returns a `graph` object which can be used for further visualization. In particular, `tkplot` offers a basic interface for interactive graphics which you can try using:

```
tkplot(g)
```

One of the (several) limitations of *SeqTrack* is made quite obvious here: all sequences are forced to coalesce to the initial one, while there are some clearly distinct clusters indicative of two separate introductions (cases 1 and 4).

6 Detailed outbreak reconstruction using *outbreaker*

6.1 *outbreaker* analysis

The *SeqTrack* algorithm provides a single, parsimonious reconstruction of the disease transmission tree, based on the pathogen sequences and the temporal ordering of their collection dates. While useful, it still represents a very incomplete reconstruction of the outbreak: it assumes a single introduction of the disease, does not take the time interval between cases into account, does not estimate dates of infections, overlooks alternative yet likely transmissions, and does not provide statistical support for individual ancestries. *outbreaker* [2] has been developed to address these issues.

This method exploits information on pathogen genomes, collection dates and (possibly vague) information about the generation time (time interval between primary and secondary cases) to reconstruct the outbreak,

including likely dates of infections and ancestries. In the present case, we do not possess external sources of information about the generation time distribution, but a fairly uninformative distribution can be constructed. Here, we want this distribution to meet two properties:

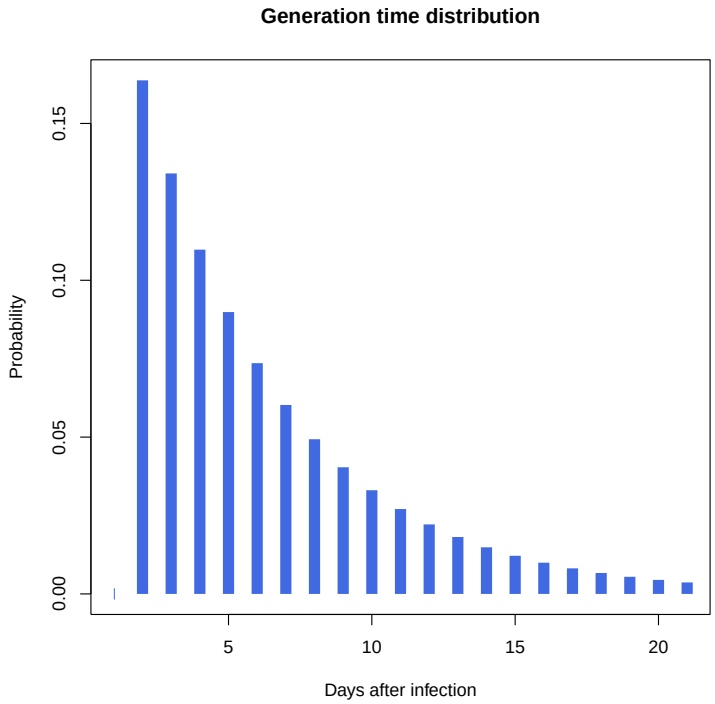
- not exclude *a priori* any transmission in the outbreak; in other words, the probability of the longest possible time interval must be non-zero.
- be monotonically decreasing, i.e. longer time intervals between transmissions are given smaller probabilities; this will force shorter transmission chains, which is consistent with the very short time intervals observed between the first cases of the outbreak.

Given the short time span of the outbreak:

```
range(days)

## [1]  0 11

head(days)

## [1] 0 2 3 3 4 6
```

we use an exponential distribution to model the generation time distribution:

```
w <- c(0,dexp(1:20, rate=1/5))
plot(w, main="Generation time distribution",
     xlab="Days after infection", ylab="Probability",
     type="h", lwd=10, lend=1, col="royalblue")
```



We now run `outbreaker`, providing as input the DNA sequences, their collection dates (in days since the first sample), and generation time distribution; as we are confident that every cases have been observed, we fix the number of generations between consecutive cases to be one (`move.kappa=FALSE` and `init.kappa` fixed to 1); finally, we use a star-like tree to initialize the MCMC:

```
obkr.res <- outbreaker(dna=dna, dates=days, w.dens=w,
                       move.kappa=FALSE, init.kappa=rep(1,30), init.tree="star")
```

```
class(obkr.res)

## [1] "list"

names(obkr.res)

##  [1] "chains"         "collec.dates"   "w"              "f"
##  [5] "D"              "idx.dna"        "tune.end"       "find.import"
##  [9] "burnin"         "find.import.at" "n.runs"         "call"
```

The object `obkr.res` is a list with a number of named items, described in `?outbreaker`. The most important one is `obkr.res$chains`, containing the MCMC outputs:

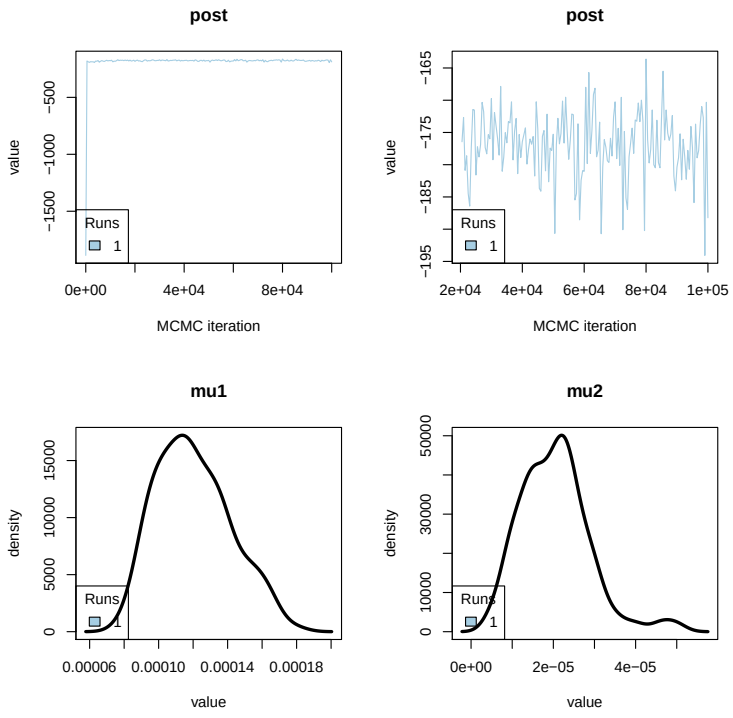
```
class(obkr.res$chains)

## [1] "data.frame"

dim(obkr.res$chains)

## [1] 201  99

names(obkr.res$chains)

##  [1] "step"     "post"     "like"     "prior"    "mu1"      "mu2"
##  [7] "gamma"    "pi"       "Tinf_1"   "Tinf_2"   "Tinf_3"   "Tinf_4"
## [13] "Tinf_5"   "Tinf_6"   "Tinf_7"   "Tinf_8"   "Tinf_9"   "Tinf_10"
## [19] "Tinf_11"  "Tinf_12"  "Tinf_13"  "Tinf_14"  "Tinf_15"  "Tinf_16"
## [25] "Tinf_17"  "Tinf_18"  "Tinf_19"  "Tinf_20"  "Tinf_21"  "Tinf_22"
## [31] "Tinf_23"  "Tinf_24"  "Tinf_25"  "Tinf_26"  "Tinf_27"  "Tinf_28"
## [37] "Tinf_29"  "Tinf_30"  "alpha_1"  "alpha_2"  "alpha_3"  "alpha_4"
## [43] "alpha_5"  "alpha_6"  "alpha_7"  "alpha_8"  "alpha_9"  "alpha_10"
## [49] "alpha_11" "alpha_12" "alpha_13" "alpha_14" "alpha_15" "alpha_16"
## [55] "alpha_17" "alpha_18" "alpha_19" "alpha_20" "alpha_21" "alpha_22"
## [61] "alpha_23" "alpha_24" "alpha_25" "alpha_26" "alpha_27" "alpha_28"
## [67] "alpha_29" "alpha_30" "kappa_1"  "kappa_2"  "kappa_3"  "kappa_4"
## [73] "kappa_5"  "kappa_6"  "kappa_7"  "kappa_8"  "kappa_9"  "kappa_10"
## [79] "kappa_11" "kappa_12" "kappa_13" "kappa_14" "kappa_15" "kappa_16"
## [85] "kappa_17" "kappa_18" "kappa_19" "kappa_20" "kappa_21" "kappa_22"
## [91] "kappa_23" "kappa_24" "kappa_25" "kappa_26" "kappa_27" "kappa_28"
## [97] "kappa_29" "kappa_30" "run"

obkr.res$chains[1:10,1:10]

##    step    post    like  prior       mu1       mu2  gamma     pi Tinf_1 Tinf_2
## 1     1 -1887.9 -1888.6 0.6317 5.000e-05 5.000e-05 1.0000 0.9771     -1      1
## 2   500  -180.8  -181.6 0.8604 5.015e-05 2.069e-05 0.4125 0.9947     -3      1
## 3  1000  -187.8  -188.2 0.4055 5.007e-05 8.664e-06 0.1730 0.9910    -11     -4
## 4  1500  -192.9  -193.3 0.4486 4.985e-05 1.609e-05 0.3227 0.9576     -7     -4
## 5  2000  -186.2  -186.7 0.5200 4.932e-05 3.005e-05 0.6092 0.9544     -8     -2
## 6  2500  -188.2  -189.0 0.8333 4.971e-05 2.004e-05 0.4030 0.9922     -9     -8
## 7  3000  -187.3  -188.2 0.9215 4.990e-05 2.305e-05 0.4619 0.9993     -3     -1
## 8  3500  -193.9  -194.5 0.6641 4.969e-05 1.790e-05 0.3602 0.9769     -5     -4
## 9  4000  -181.6  -182.3 0.6391 5.224e-05 2.208e-05 0.4227 0.9700     -2      0
## 10 4500  -182.5  -183.0 0.4630 5.557e-05 1.028e-05 0.1850 0.9918     -6     -2
```

The columns of this `data.frame` store the following outputs:

- `step`: the MCMC iteration of the sample
- `post/like/prior`: log values for posterior, likelihood, and prior densities
- `mu1`: rate of transitions, per site and generation
- `mu2`: rate of transversions, per site and generation
- `gamma`: the ratio between transversions and transitions (μ_2/μ_1)
- `pi`: the proportion of the transmission tree sampled
- `Tinf_[number]`: dates of infection
- `alpha_[number]`: the index of the ancestral cases (infectors)
- `kappa_[number]`: the number of generations between cases and their most recent sampled ancestor (here, fixed to 1)
- `run`: for parallel runs, the index of the run.

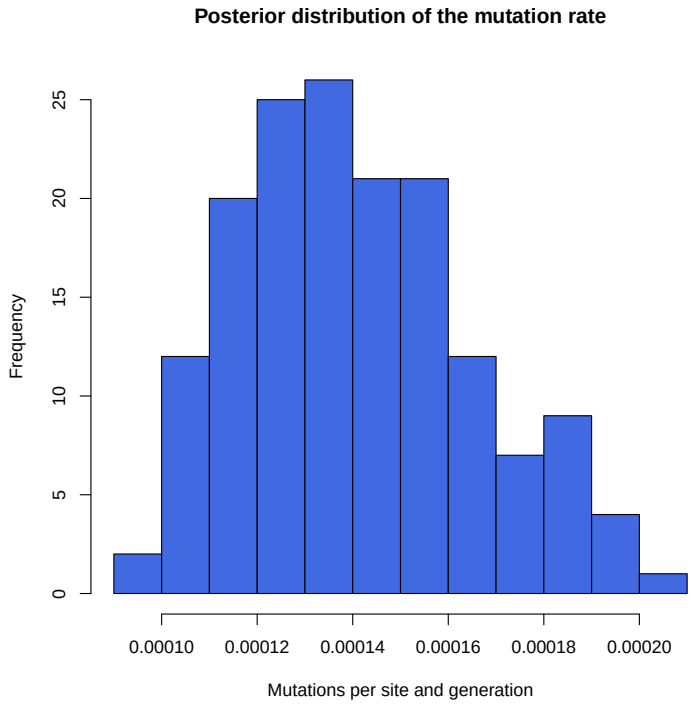
Convergence of the chain and distributions of parameters can be visualized using `plotChains` (see `?plotChains` for details):

```
par(mfrow=c(2,2))
plotChains(obkr.res)
plotChains(obkr.res, burnin=2e4)
plotChains(obkr.res, burnin=2e4, type="dens", what="mu1")
plotChains(obkr.res, burnin=2e4, type="dens", what="mu2")
```



One can easily derive statistics from posterior samples to get e.g. credibility intervals. For instance, the overall mutation rate (sum of the two rates) is obtained, after discarding a burnin of 20,000 iterations, by:

```
mu <- (obkr.res$chains$mu1+obkr.res$chains$mu2)[obkr.res$chains$step>2e4]
hist(mu, col="royalblue", main="Posterior distribution of the mutation rate",
     xlab="Mutations per site and generation")
```



Its mean and 95% credibility interval are:

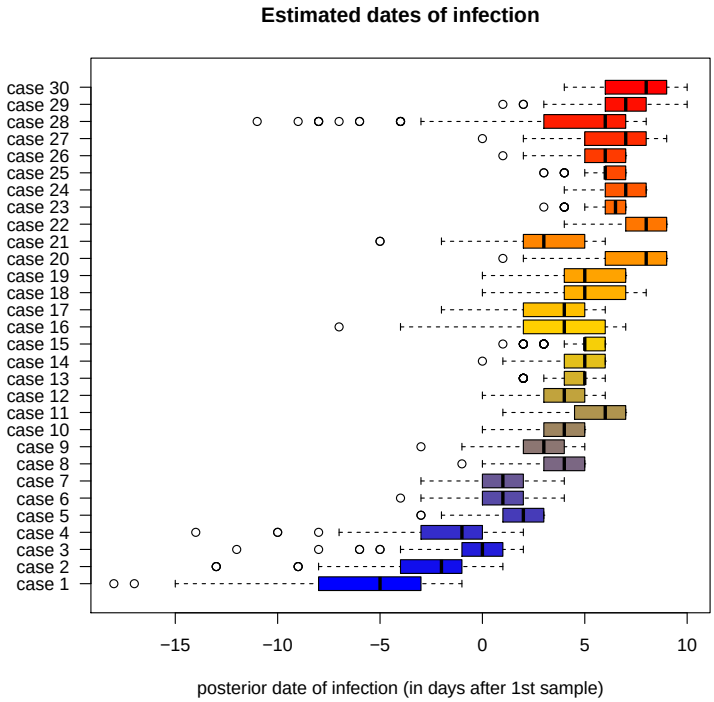
```
mean(mu)

## [1] 0.0001407

quantile(mu, c(0.025, 0.975))

##      2.5%     97.5%
## 0.0001019 0.0001907
```

which is, in number of mutations per genome and per transmission event:

```
mean(mu*ncol(dna))

## [1] 1.407

quantile(mu*ncol(dna), c(0.025, 0.975))

##  2.5% 97.5%
## 1.019 1.907
```

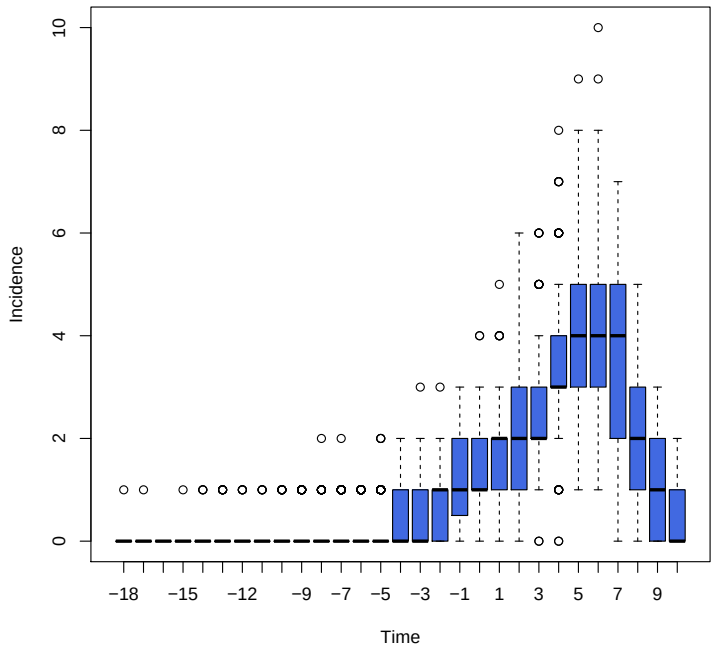
Likely dates of infection are probably better visualized as boxplots:

```
Tinf <- obkr.res$chains[obkr.res$chains$step>2e4,grep("Tinf",names(obkr.res$chains))]
boxplot(Tinf, horizontal=TRUE, col=seasun(30),las=1,
        xlab="posterior date of infection (in days after 1st sample)",
        names=paste("case",1:30), main="Estimated dates of infection")
```



The overall dynamics corresponding to these dates of infection can be summarized as incidence curves:

```
incid <- get.incid(obkr.res, burnin=2e4, fill.col="royalblue",
                   main="Inferred incidence")
```

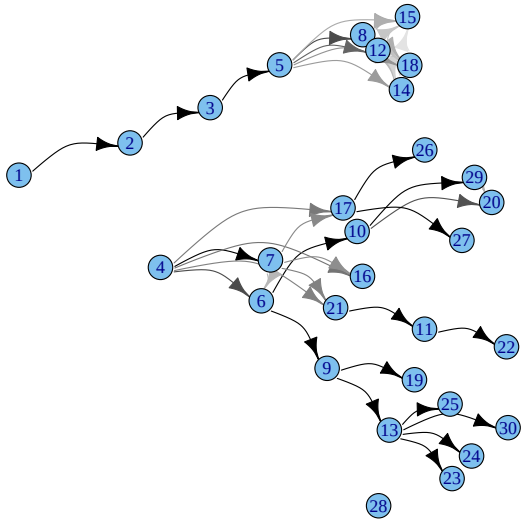


```
class(incid)

## [1] "matrix"

dim(incid)

## [1]  29 160
```

`incid` is a matrix containing one time series of incidence for each posterior tree. The displayed boxplot summarizes the entire posterior distribution. When did the outbreak take off? Can we trust the decrease at the end of the time series?

The posterior ancestries (columns "`alpha`" in `res$chains`) define a graph of possible ancestries, which can be extracted and plotted using `transGraph`; here, we remove annotations from the edges (`annot=""`) and retain only ancestries with a frequency $\geq 5\%$ (`thres=.05`):

```
g2 <- transGraph(obkr.res, vertex.size=10, annot="", thres=.05)
```



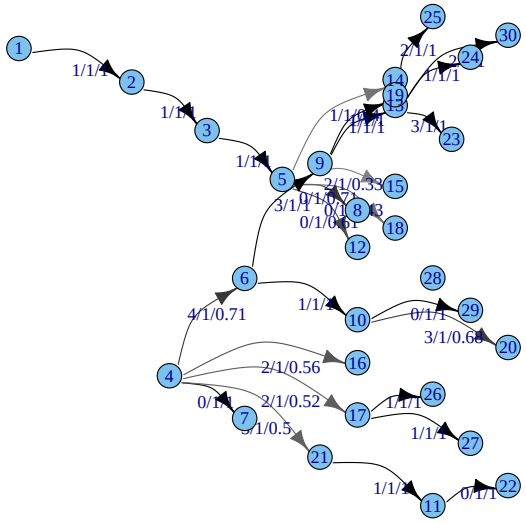

```
g2

## IGRAPH DN-- 30 44 --
## + attr: name (v/c), dates (v/n), label (v/n), color (e/c), support
##   (e/n), curved (e/x), nb.mut (e/n), label (e/c)
```

Darker arrows correspond to well-supported ancestries, while lighter ones are more ambiguous. The object `g2` is an `igraph`, which can be visualized interactively like before using `tkplot`:

```
tkplot(g2)
```

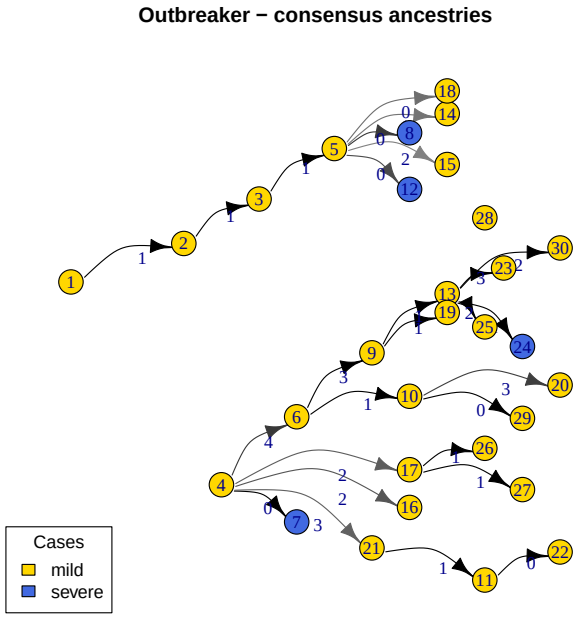
For some purposes, one may wish to retain only the best supported ancestries of each case. This can be achieved by `get.tTree`:

```
obkr.tre <- get.tTree(obkr.res)
class(obkr.tre)

## [1] "tTree"

plot(obkr.tre, vertex.size=10, edge.curved=TRUE)
```



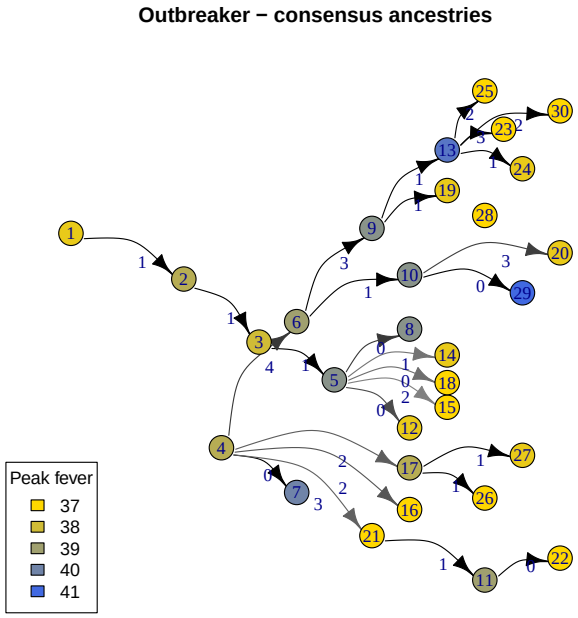
By default, the annotations of the edges indicate *number of mutations/number of generations/posterior support*. See `?get.tTree` for a detail of the content of `obkr.tre`. Note that these objects can also be converted to `igraph` objects using `as.igraph`, and implicitly using the `plot` method. The latter also allows for representing additional information, such as features of the cases using colors. For instance, we can represent the consensus ancestries with the outcome of the different cases:

```
plot(obkr.tre, vertex.size=10, annot="dist",
     vertex.color=fac2col(cases$outcome, col.pal=azur),
     edge.curved=TRUE)
title("Outbreaker - consensus ancestries")
legend("bottomleft", fill=azur(2), legend=unique(cases$outcome),
       title="Cases")
```



Similarly, we can represent the peak temperature of the patients:

```
plot(obkr.tre, vertex.size=10, annot="dist",
     vertex.color=num2col(cases$peak.fever, col.pal=azur),
     edge.curved=TRUE)
title("Outbreaker - consensus ancestries")
leg.val <- 37:41
leg.col <- num2col(leg.val, col.pal=azur, x.min=min(cases$peak.fever),
                   x.max=max(cases$peak.fever))
legend("bottomleft", fill=leg.col, legend=leg.val, title="Peak fever")
```



While trends may be suggested from such plot, a more quantitative assessment of potential determinants of infectiousness is needed before drawing conclusions.

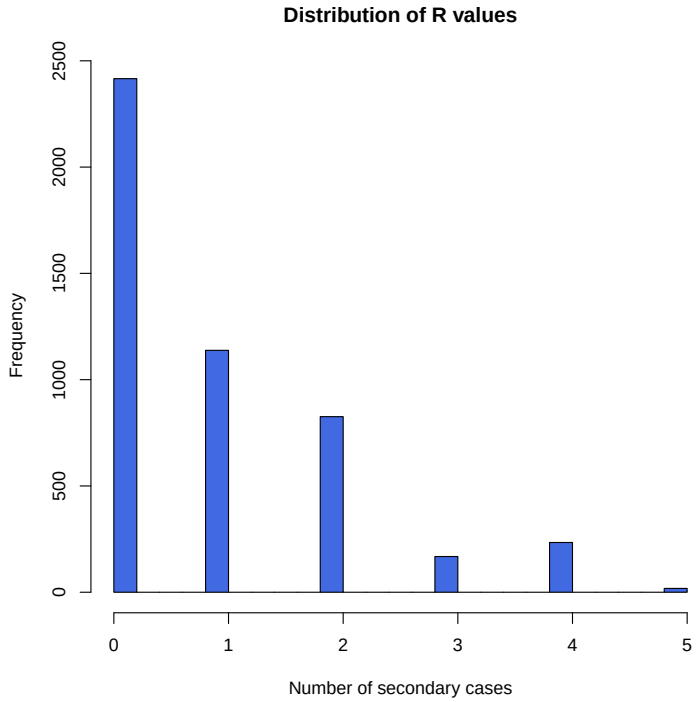
6.2 Inference from the reconstructed ancestries

One of the first concerns once we inferred a transmission tree is the identification of key individuals for the spread of the epidemic. This can be assessed by computing the number of secondary cases per infected individual, that is, the individual effective reproduction numbers (R_i). The posterior distribution of these values can be obtained using `get.R`:

```
Rmat <- get.R(obkr.res,burnin=2e4)
class(Rmat)

## [1] "matrix"

ncol(Rmat)

## [1] 30
```

`Rmat` is a matrix of posterior values of R_i for each of the 30 cases (in column). The distribution of R_i is obtained by:

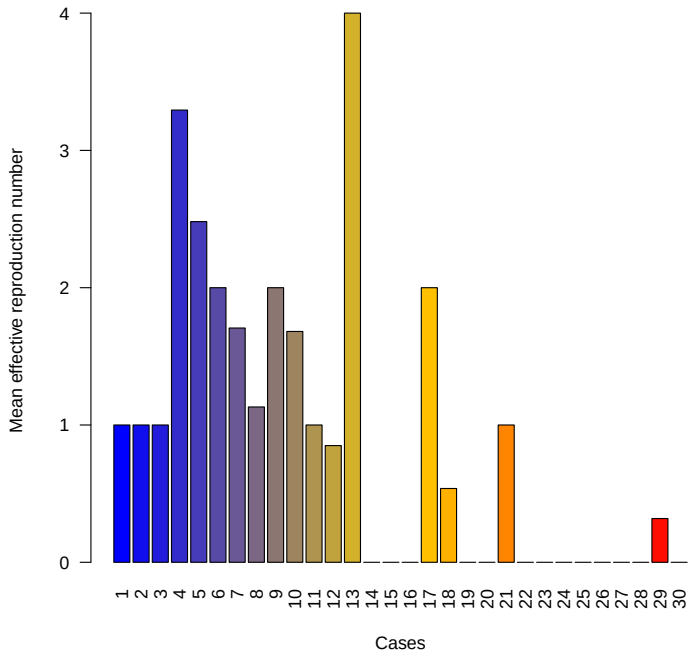
```
hist(Rmat, col="royalblue", nclass=20, main="Distribution of R values",
     xlab="Number of secondary cases")
```



Shall we be looking for individuals driving the epidemic?

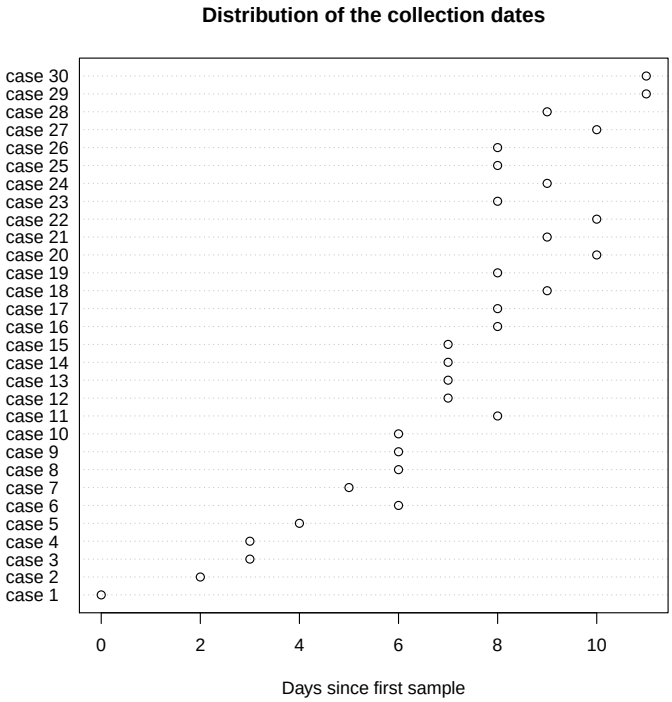
The average R_i over all chains is computed using:

```
Rindiv <- apply(Rmat, 2, mean)
Rindiv

##      1      2      3      4      5      6      7      8      9     10     11
## 1.0000 1.0000 1.0000 3.2938 2.4813 2.0000 1.7063 1.1313 2.0000 1.6812 1.0000
##     12     13     14     15     16     17     18     19     20     21     22
## 0.8500 4.0000 0.0000 0.0000 0.0000 2.0000 0.5375 0.0000 0.0000 1.0000 0.0000
##     23     24     25     26     27     28     29     30
## 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.3187 0.0000

barplot(Rindiv, col=seasun(30),las=2, xlab="Cases",
        ylab="Mean effective reproduction number")
```



Now that we have this proxy for the "infectiousness" of individuals, we can try to correlate it to other factors such as age, sex, or other measured covariates. Note that we only have a snapshot of an ongoing epidemic, so we probably have not measured the infectiousness of the last infected individuals. Let us first have another look at the distribution of the collection dates:

```
dotchart(days,labels=paste("case", 1:30),
         xlab="Days since first sample",
         main="Distribution of the collection dates")
```



There is no obvious way of defining a threshold date, but keeping all cases until day 8 (included) seems to exclude most recent cases while conserving a fair portion of the sample.

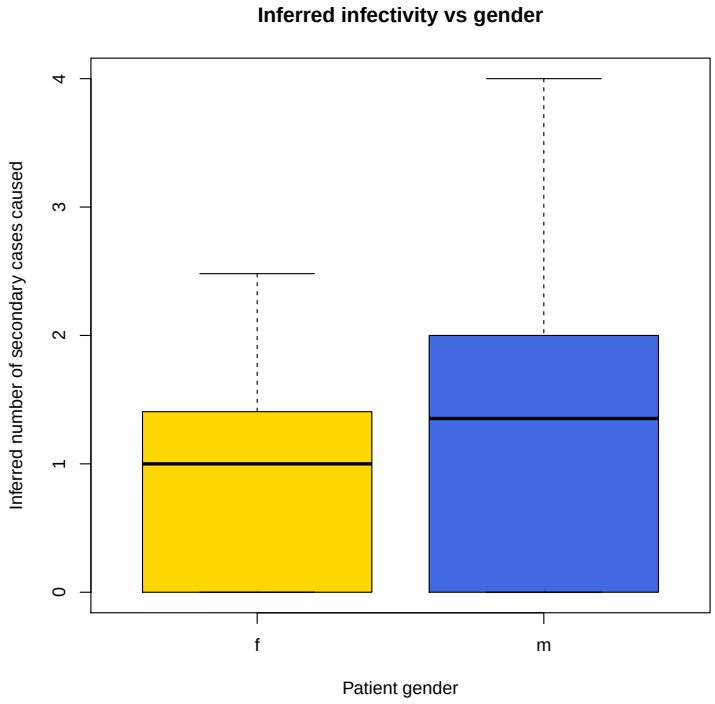
```
toKeep <- days<9
```

We can now examine and test possible relationships between R_i (object `Rindiv`) and covariates in `cases`. For a reminder:

```
head(cases)

##   id collec.dates sex age peak.fever outcome notes
## 1  1   2013-02-18   m  30       37.5    mild
## 2  2   2013-02-20   f  40       38.5    mild
## 3  3   2013-02-21   f  32       38.0    mild
## 4  4   2013-02-21   m  35       38.5    mild
## 5  5   2013-02-22   f   3       39.5    mild
## 6  6   2013-02-24   f  34       39.0    mild
```

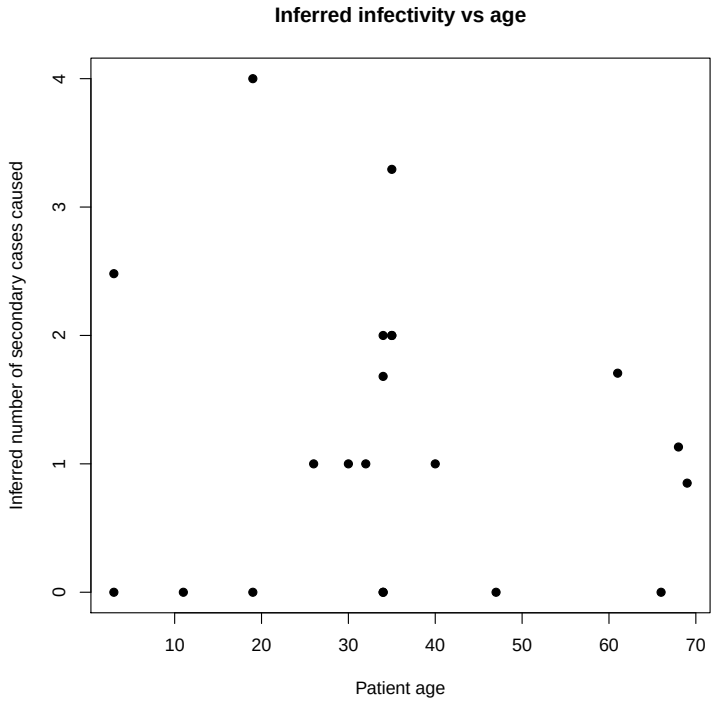
Interprete the following graphs and tests:

```
boxplot(Rindiv[toKeep]~cases$sex[toKeep], xlab="Patient gender",
     ylab="Inferred number of secondary cases caused", col=c("gold","royalblue"))
title("Inferred infectivity vs gender")
```



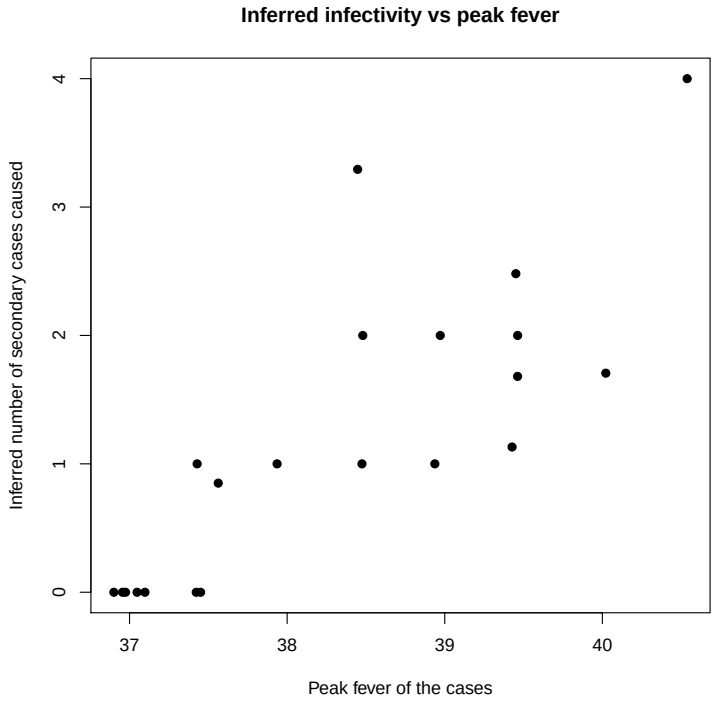

```
t.test(Rindiv[toKeep]~cases$sex[toKeep])

##
##  Welch Two Sample t-test
##
## data:  Rindiv[toKeep] by cases$sex[toKeep]
## t = -1.131, df = 14.93, p-value = 0.276
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.6675  0.5118
## sample estimates:
## mean in group f mean in group m
##          0.9222          1.5000
```

```
plot(Rindiv[toKeep]~cases$age[toKeep], xlab="Patient age",
     ylab="Inferred number of secondary cases caused",
     pch=20, cex=1.5)
title("Inferred infectivity vs age")
```



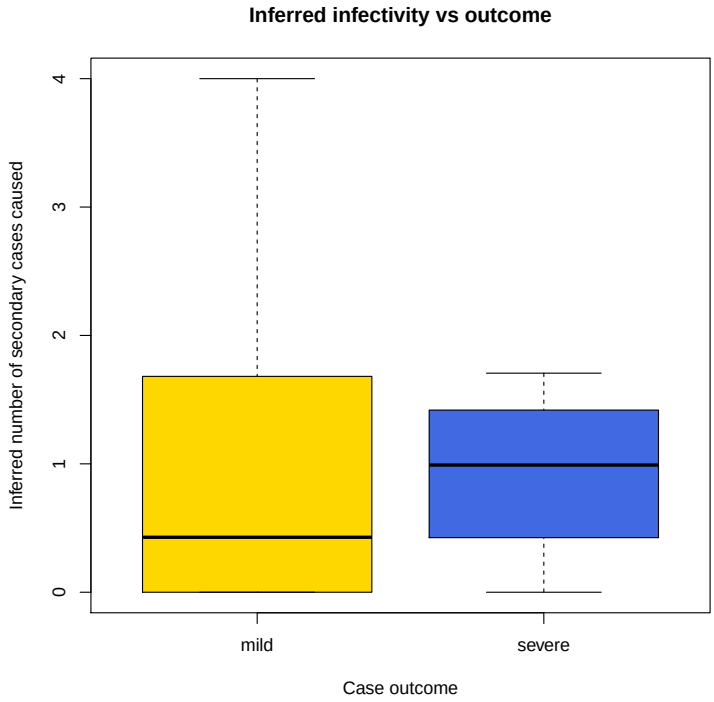

```
cor.test(Rindiv[toKeep],cases$age[toKeep], method="spearman")

##
##  Spearman's rank correlation rho
##
## data:  Rindiv[toKeep] and cases$age[toKeep]
## S = 1511, p-value = 0.9359
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##     rho
## 0.01871
```

```
plot(Rindiv[toKeep]~jitter(cases$peak.fever[toKeep]), xlab="Peak fever of the cases",
     ylab="Inferred number of secondary cases caused",
     pch=20, cex=1.5)
title("Inferred infectivity vs peak fever")
```




```
cor.test(Rindiv[toKeep],cases$peak.fever[toKeep], method="spearman")

##
##  Spearman's rank correlation rho
##
## data:  Rindiv[toKeep] and cases$peak.fever[toKeep]
## S = 220.7, p-value = 7.152e-07
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##    rho
## 0.8567
```

```
boxplot(Rindiv~cases$outcome, xlab="Case outcome",
     ylab="Inferred number of secondary cases caused", col=c("gold","royalblue"))
title("Inferred infectivity vs outcome")
```




```
t.test(Rindiv~cases$outcome)

##
##  Welch Two Sample t-test
##
## data:  Rindiv by cases$outcome
## t = -0.0602, df = 5.724, p-value = 0.9541
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.064  1.013
## sample estimates:
##   mean in group mild mean in group severe
##               0.8966               0.9219
```

What can you say about the transmissibility of this disease? Should prophylaxis target specific groups of individuals? Looking back at the data, especially the most recent cases:

```
tail(cases, 10)

##    id collec.dates sex age peak.fever outcome                  notes
## 21 21   2013-02-27   m  49       37.0    mild
## 22 22   2013-02-28   m  35       37.0    mild
## 23 23   2013-02-26   m  34       37.0    mild
## 24 24   2013-02-27   m  59       37.5  severe
## 25 25   2013-02-26   f  47       37.0    mild
## 26 26   2013-02-26   f  34       37.0    mild
## 27 27   2013-02-28   f  26       37.5    mild
## 28 28   2013-02-27   f  16       37.0    mild possible-contamination
## 29 29   2013-03-01   f  15       41.0    mild
## 30 30   2013-03-01   m  40       37.0    mild
```

Which individual(s) would you recommend isolating in priority?

7 Update from detailed case investigations

As you were finishing your analyses, you have been updated on the situation by the authorities. Apparently, detailed investigations have helped casting light on the transmissions that took place for the first 25 cases. Information on likely infectors is contained in the following file:

```
newinfo <- read.csv("http://adegenet.r-forge.r-project.org/files/fakeOutbreak/update.csv")
newinfo

##    infection.dates infectors
## 1       2013-02-15        NA
## 2       2013-02-17         1
## 3       2013-02-19         2
## 4       2013-02-19        NA
## 5       2013-02-21         3
## 6       2013-02-21         4
## 7       2013-02-21         4
## 8       2013-02-22         5
## 9       2013-02-22         6
## 10      2013-02-23         6
## 11      2013-02-23         7
## 12      2013-02-23         8
## 13      2013-02-23         9
## 14      2013-02-24         5
## 15      2013-02-24         5
## 16      2013-02-24         7
## 17      2013-02-24         7
## 18      2013-02-25         8
## 19      2013-02-25         9
## 20      2013-02-25        10
## 21      2013-02-25        11
## 22      2013-02-25        11
## 23      2013-02-25        13
## 24      2013-02-25        13
## 25      2013-02-25        13
```

It is fairly straightforward to compare the results of *SeqTrack* (`sqtk.res`) and *outbreaker* (`obkr.tre`) to this new data; we just need to avoid comparing NAs (as `NA==NA` is `NA`, not `TRUE`), so we replace unknown ancestries (`NA`) with 0.

```
sqtk.res$ances[is.na(sqtk.res$ances)] <- 0
obkr.ances <- obkr.tre$ances[1:25]
obkr.ances[is.na(obkr.ances)] <- 0
newinfo$infectors[is.na(newinfo$infectors)] <- 0
comp <- rbind(sqtk.res$ances[1:25], obkr.ances, newinfo$infectors)
rownames(comp) <- c("seqTrack", "outbreaker","investigations")
colnames(comp) <- paste("case", 1:25)
comp

##                case 1 case 2 case 3 case 4 case 5 case 6 case 7 case 8 case 9
## seqTrack            0      1      2      1      3      4      4      5      4
## outbreaker          0      1      2      0      3      4      4      5      6
```

```
## investigations      0      1      2      0      3      4      4      5      6
##                case 10 case 11 case 12 case 13 case 14 case 15 case 16 case 17
## seqTrack            4      4      5      9      5      5      4      4
## outbreaker          6     21      5      9      5      5      4      4
## investigations      6      7      8      9      5      5      7      7
##                case 18 case 19 case 20 case 21 case 22 case 23 case 24 case 25
## seqTrack            5      9     10     11     11     13     13     13
## outbreaker          5      9     10      4     11     13     13     13
## investigations      8      9     10     11     11     13     13     13
```

```
mean(comp[1,]==comp[3,])
```

```
## [1] 0.68
```

```
mean(comp[2,]==comp[3,])
```

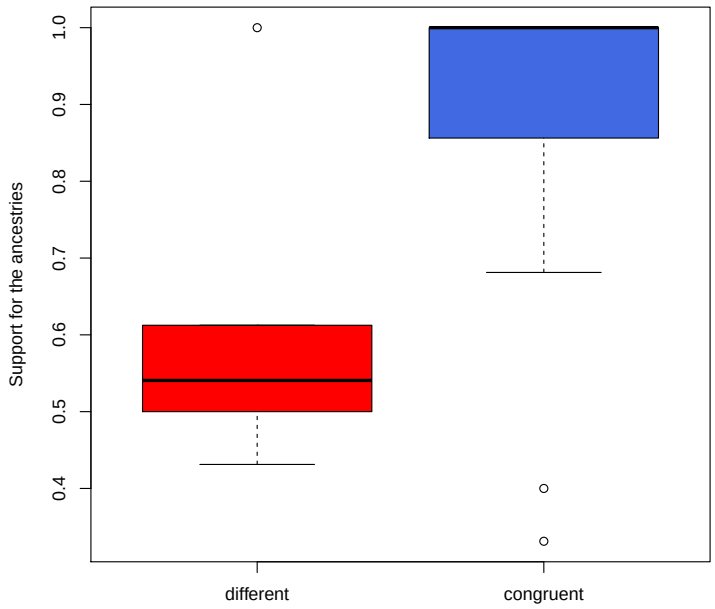
```
## [1] 0.76
```

Which method is most congruent with the new field observations, and what are the respective proportions of congruent results (inferred transmissions)? For `outbreaker`, one can assess easily if discrepancies are associated with weakly supported ancestries:

```
split(obkr.tre$p.ances[1:25], obkr.ances[1:25]==newinfo$infectors)
```

```
## $`FALSE`
## alpha_11 alpha_12 alpha_16 alpha_17 alpha_18 alpha_21
##   1.0000   0.6125   0.5563   0.5250   0.4313   0.5000
##
## $`TRUE`
##  alpha_1  alpha_2  alpha_3  alpha_4  alpha_5  alpha_6  alpha_7  alpha_8
##   1.0000   1.0000   1.0000   1.0000   1.0000   0.7125   1.0000   0.7063
##  alpha_9 alpha_10 alpha_13 alpha_14 alpha_15 alpha_19 alpha_20 alpha_22
##   1.0000   1.0000   1.0000   0.4000   0.3312   1.0000   0.6813   1.0000
## alpha_23 alpha_24 alpha_25
##   1.0000   1.0000   1.0000
```

```
boxplot(split(obkr.tre$p.ances[1:25], obkr.ances[1:25]==newinfo$infectors),
        col=c("red1","royalblue"), names=c("different","congruent"),
        ylab="Support for the ancestries")
```

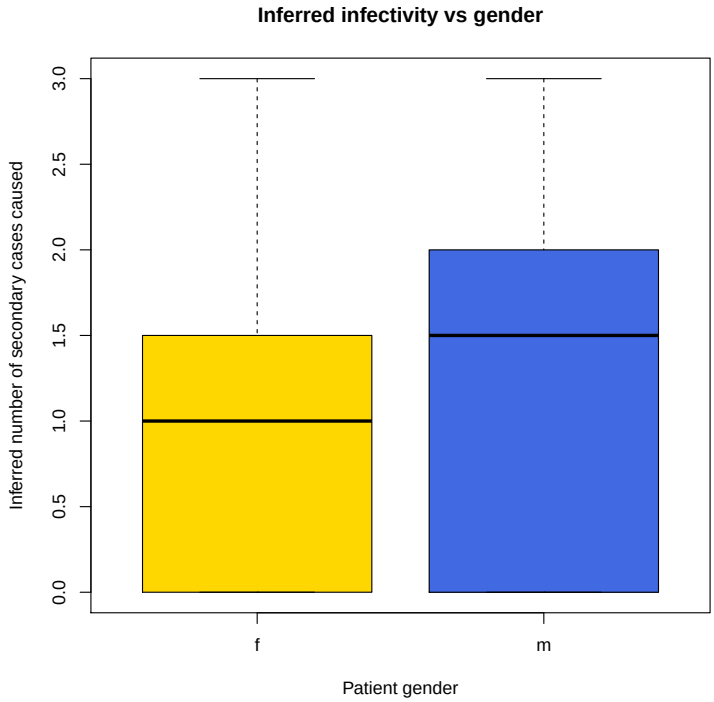


Let us examine again the possible effect of covariates on individual reproduction numbers R_i, this time computing R_i from the investigation data:

```
Rindiv2 <- sapply(1:30, function(i) sum(newinfo$infectors==i, na.rm=TRUE))
names(Rindiv2) <- paste("case",1:30,sep="")
Rindiv2

##  case1  case2  case3  case4  case5  case6  case7  case8  case9 case10 case11
##      1      1      1      2      3      2      3      2      2      1      2
## case12 case13 case14 case15 case16 case17 case18 case19 case20 case21 case22
##      0      3      0      0      0      0      0      0      0      0      0
## case23 case24 case25 case26 case27 case28 case29 case30
##      0      0      0      0      0      0      0      0
```

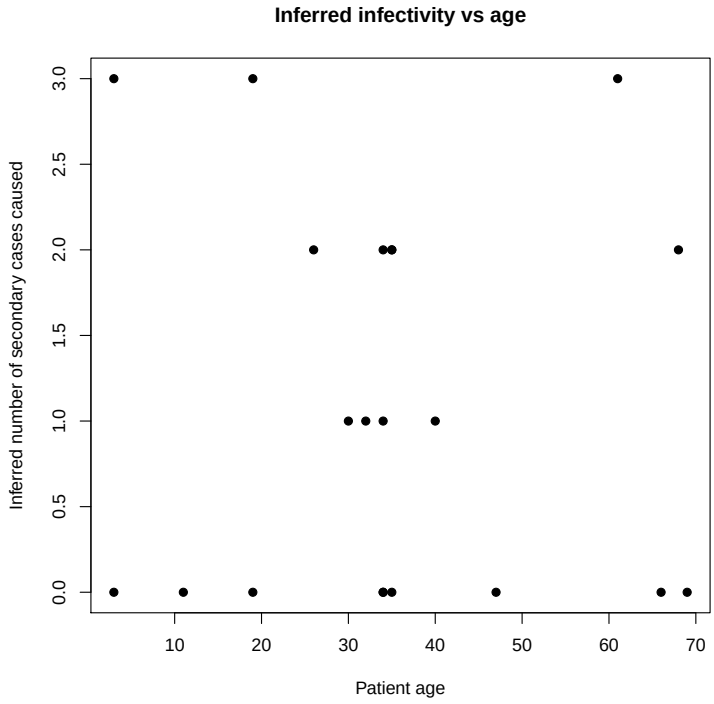
Again, we discard the most recent cases (collection on day 9 and later; this information is still in `toKeep`). What can you conclude from the following graphs and tests:

```
boxplot(Rindiv2[toKeep]~cases$sex[toKeep], xlab="Patient gender",
     ylab="Inferred number of secondary cases caused", col=c("gold","royalblue"))
title("Inferred infectivity vs gender")
```



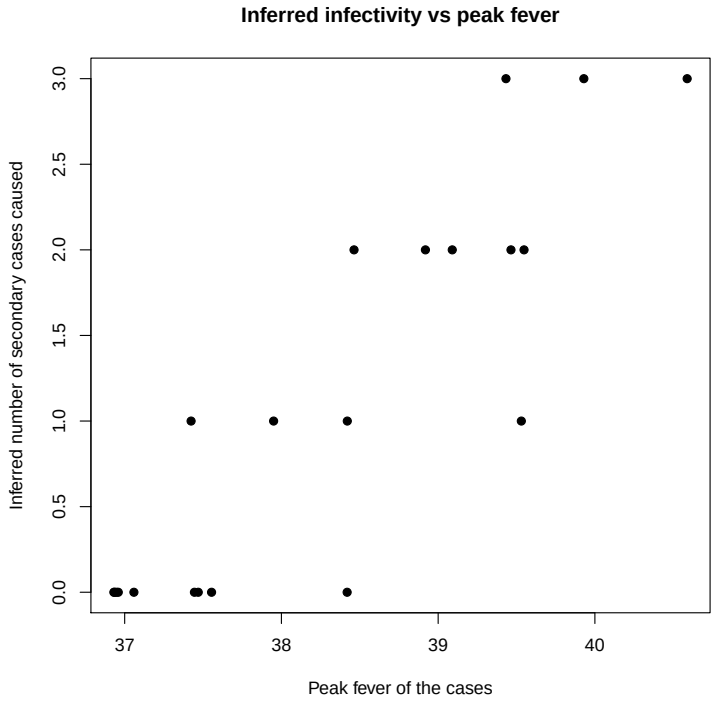

```
t.test(Rindiv2[toKeep]~cases$sex[toKeep])

##
##  Welch Two Sample t-test
##
## data:  Rindiv2[toKeep] by cases$sex[toKeep]
## t = -0.7728, df = 17.64, p-value = 0.4498
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.4551  0.6733
## sample estimates:
## mean in group f mean in group m
##          0.9091          1.3000
```

```
plot(Rindiv2[toKeep]~cases$age[toKeep], xlab="Patient age",
     ylab="Inferred number of secondary cases caused",
     pch=20, cex=1.5)
title("Inferred infectivity vs age")
```



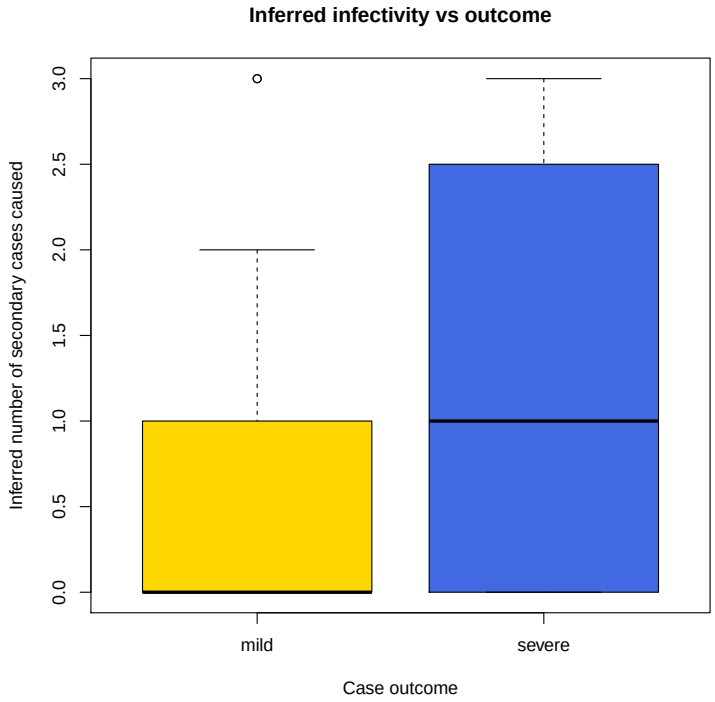

```
cor.test(Rindiv2[toKeep],cases$age[toKeep], method="spearman")

##
##  Spearman's rank correlation rho
##
## data:  Rindiv2[toKeep] and cases$age[toKeep]
## S = 1639, p-value = 0.7817
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## -0.06433
```

```
plot(Rindiv2[toKeep]~jitter(cases$peak.fever[toKeep]), xlab="Peak fever of the cases",
     ylab="Inferred number of secondary cases caused",
     pch=20, cex=1.5)
title("Inferred infectivity vs peak fever")
```




```
cor.test(Rindiv2[toKeep],cases$peak.fever[toKeep], method="spearman")

##
##  Spearman's rank correlation rho
##
## data:  Rindiv2[toKeep] and cases$peak.fever[toKeep]
## S = 185.6, p-value = 1.513e-07
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##    rho
## 0.8795
```

```
boxplot(Rindiv2~cases$outcome, xlab="Case outcome",
     ylab="Inferred number of secondary cases caused", col=c("gold","royalblue"))
title("Inferred infectivity vs outcome")
```




```
t.test(Rindiv2~cases$outcome)

##
##  Welch Two Sample t-test
##
## data:  Rindiv2 by cases$outcome
## t = -0.7189, df = 3.432, p-value = 0.5181
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.860  1.744
## sample estimates:
##   mean in group mild mean in group severe
##               0.6923               1.2500
```

Are the conclusions based on these new data consistent with *outbreaker*'s results?

References

[1] T. Jombart. adegenet: a R package for the multivariate analysis of genetic markers. *Bioinformatics*, 24:1403–1405, 2008.

[2] T. Jombart, A. Cori, X. Didelot, S. Cauchemez, C. Fraser, and N. Ferguson. Bayesian reconstruction of disease outbreaks by combining epidemiologic and genomic data. *Nature Communications*, submitted.

[3] T. Jombart, R. M. Eggo, P. J. Dodd, and F. Balloux. Reconstructing disease outbreaks from genetic data: a graph approach. *Heredity*, 106:383–390, 2010.

[4] E. Paradis, J. Claude, and K. Strimmer. APE: analyses of phylogenetics and evolution in R language. *Bioinformatics*, 20:289–290, 2004.

[5] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.