Practical course using the R software

Analysing outbreak data using R: some exploratory approaches

Thibaut Jombart (tjombart@imperial.ac.uk)

Imperial College London — RAPID-NGS Workshop

Abstract

This practical introduces some simple analyses of pathogen genome data collected during disease outbreaks, using the R software [4]. We illustrate how different approaches including phylogenetics, genetic clustering and *SeqTrack* [2] can be used to uncover the features of a disease outbreak, and possibly help designing containment strategies. This tutorial uses the packages *ape* [3] for phylogenetic analyses and *adegenet* [1] for genetic clustering and transmission tree reconstruction (*SeqTrack* algorithm). While the data and analysed outbreak are purely fictional, the methodology presented here will be useful for the first exploration of a range of actual disease outbreaks.

Contents

1 Introduction **3**

1.1 An emerging pathogen outbreak 3

1.2 Your objective 3

2 First look at the data **4**

3 Phylogenetic analysis **7**

4 Identifying clusters of cases **8**

5 Analysis using *SeqTrack* **10**

5.1 Transmission tree reconstruction using *SeqTrack* 10

5.2 Inference from the reconstructed tree 13

5.3 Update from detailed case investigations 18

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 possible 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 two R packages used for the analysis of the data, *ape* (for phylogenetics) and *adegenet* (for genetic clustering and *SeqTrack*).

```
> library(ape)
> library(adegenet)
```

The data consists of two files: one file `cases.csv` containing description 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 case description:

```
> 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 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"
```

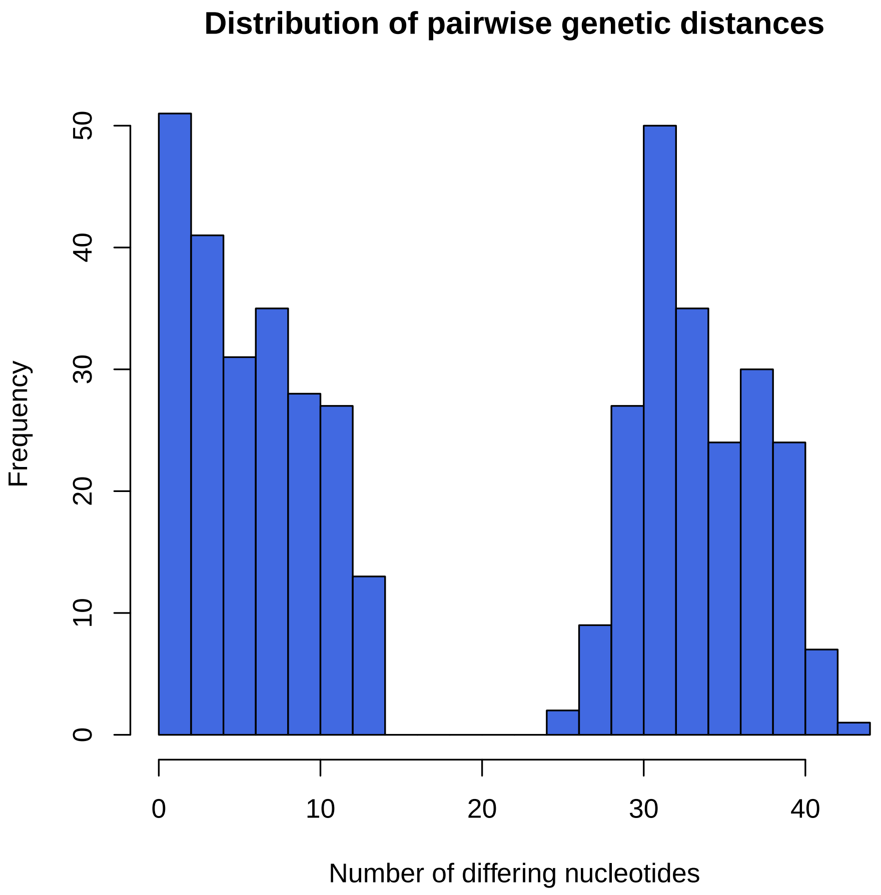
```
> days <- as.integer(difftime(dates, min(dates), unit="days"))
> days

 [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 `read.dna`, precising that the data are in fasta format (`format="fasta"`):

```
> dna <- read.dna("http://adegenet.r-forge.r-project.org/files/fakeOutbreak/alignment.fa",
+                 format="fasta")
> 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")
```



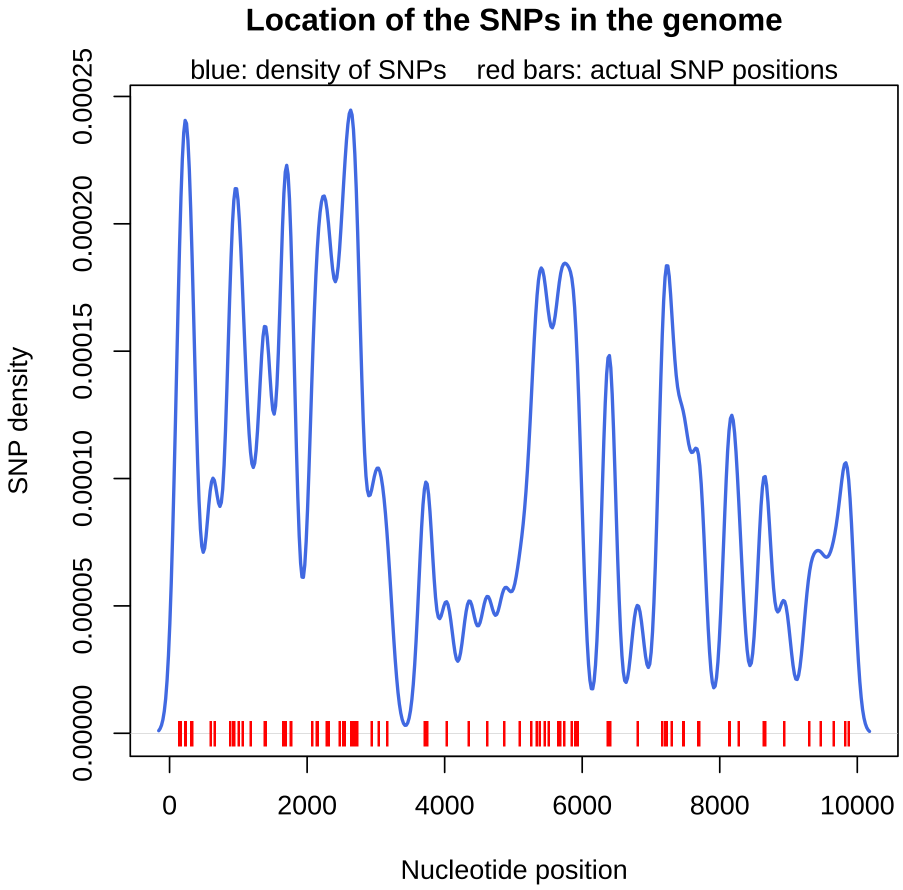
For such a small temporal scale and genome, the amount of diversity is considerable. The fact that the distribution is clearly bimodal suggests the existence of at least two clades (and possibly more).

It may be interesting to see if this remarkable polymorphism is distributed randomly across the genome. We can extract SNPs 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 often best inferred using phylogenetic trees. Here, we reconstruct a phylogenetic trees using the usual Neighbour-Joining algorithm on pairwise genetic distances. As the mere numbers 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`):

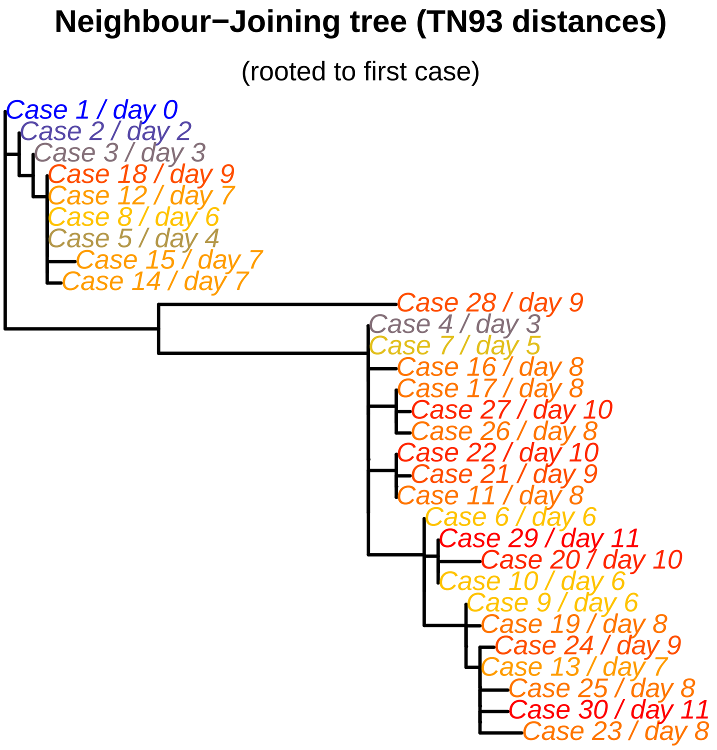
```
> tre <- nj(D.tn93)
> tre

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)")
```



The tree clearly shows at least two distinct clades, possibly three. This could be due to discontinuous sampling, but the dates/colors clearly show that this is not the case: case 4 was sampled on day 3, and is genetically very distinct from e.g. cases 1–3.

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)

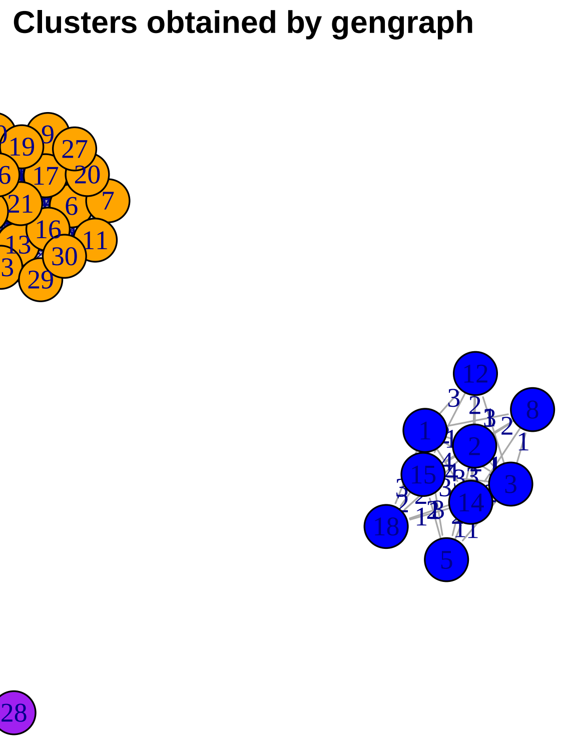
Try a few values; you should see that 3 groups are obtained for anything between 15 and 25 mutations, with the result looking like this:

```
> 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
"#0000FF" "#FFA500" "#A020F0"
```

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

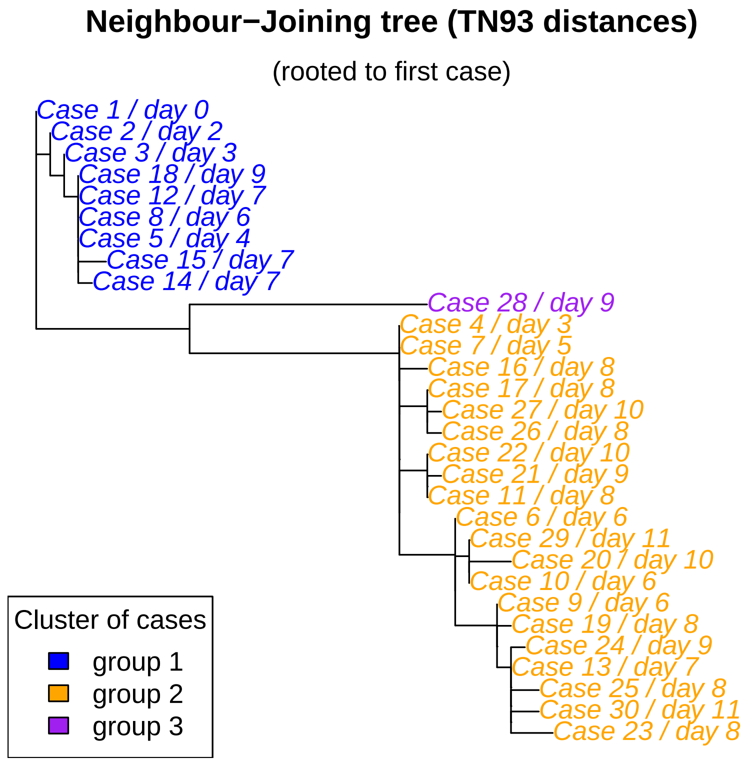


This confirms what the phylogeny suggested: there are two distinct clades, and one outlier (case 28), which is very likely an indication that this sample was indeed contaminated — as a reminder:

```
> 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$membership])
> title("Neighbour-Joining tree (TN93 distances)")
> mtext(side=3, text="(rooted to first case)")
> legend("bottomleft", fill=clust$col, legend=paste("group",1:3), title="Cluster of cases")
```



5 Analysis using *SeqTrack*

5.1 Transmission tree reconstruction 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)
> res <- seqTrack(distmat, x.names=cases$id, x.dates=dates)
> class(res)

[1] "seqTrack"   "data.frame"

> 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
```

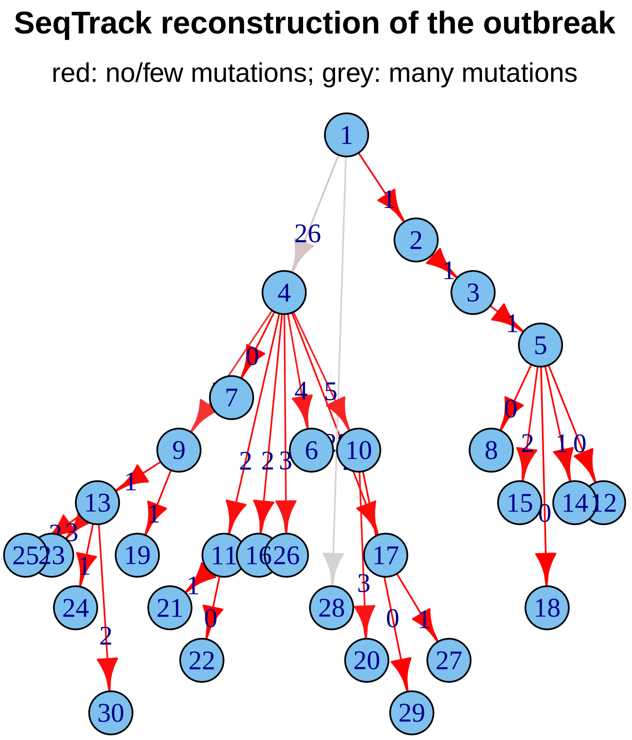
```
8   8     5      0 2013-02-24 2013-02-22
9   9     4      7 2013-02-24 2013-02-21
10 10     4      5 2013-02-24 2013-02-21
11 11     4      2 2013-02-26 2013-02-21
12 12     5      0 2013-02-25 2013-02-22
13 13     9      1 2013-02-25 2013-02-24
14 14     5      1 2013-02-25 2013-02-22
15 15     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 `res` is a `data.frame` with the special class `seqTrack`, containing the following information:

- ⋆ `res$id`: the indices of the cases.
- ⋆ `res$ances`: the indices of the putative ancestors of the cases.
- ⋆ `res$weight`: the number of mutations corresponding to the ancestries.
- ⋆ `res$date`: the collection dates of the cases.
- ⋆ `res$ances.date`: the collection dates of the putative ancestors.

`seqTrack` objects can be plotted simply using:

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


SeqTrack reconstruction of the outbreak
red: no/few mutations; grey: many mutations
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
26

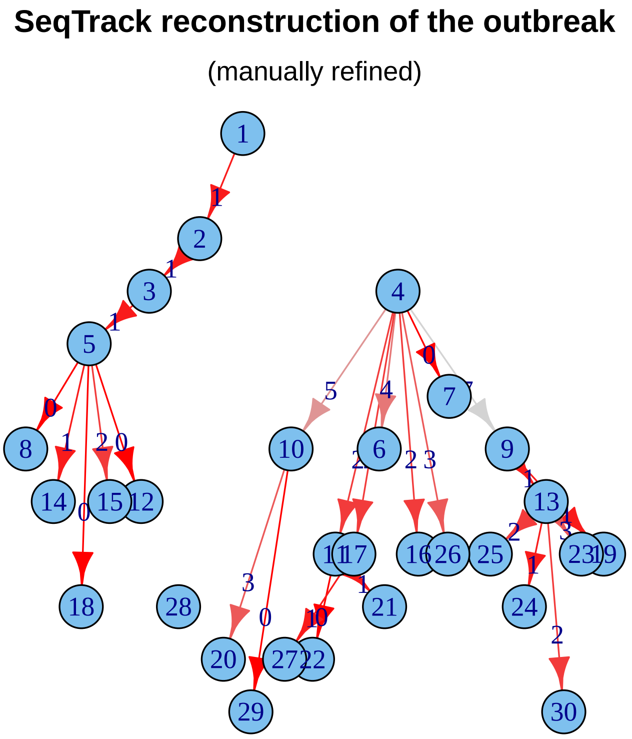
```
> g

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` here 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 basic 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). Sequence 28 cannot be trusted, so it is pointless to seek its ancestry. We can fix all this manually:

```
> res$ances[4] <- NA
> res$ances[28] <- NA

> plot(res, main="SeqTrack reconstruction of the outbreak")
> mtext(side=3, text="(manually refined)")
```



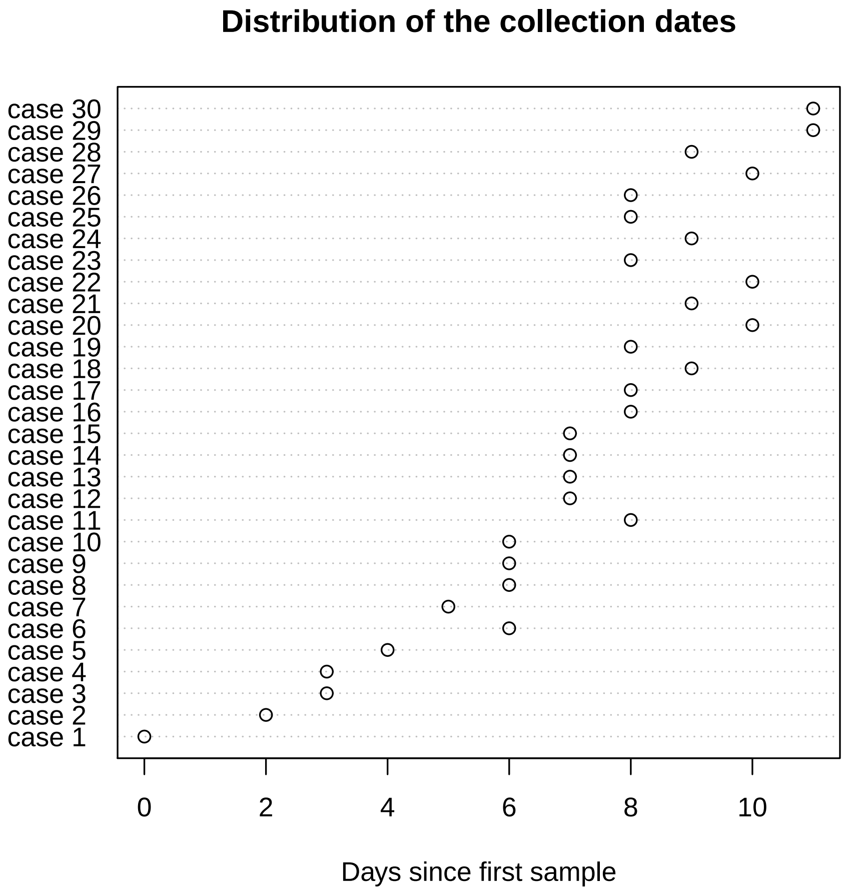
5.2 Inference from the reconstructed tree

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). We compute these values from the *SeqTrack* output:

```
> Rindiv <- sapply(1:30, function(i) sum(res$ances==i, na.rm=TRUE))
> names(Rindiv) <- paste("case",1:30,sep="")
> Rindiv[28] <- NA
> Rindiv

 case1  case2  case3  case4  case5  case6  case7  case8  case9 case10 case11
     1      1      1      8      5      0      0      0      2      2      2
case12 case13 case14 case15 case16 case17 case18 case19 case20 case21 case22
     0      4      0      0      0      1      0      0      0      0      0
case23 case24 case25 case26 case27 case28 case29 case30
     0      0      0      0      0     NA      0      0
```

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.

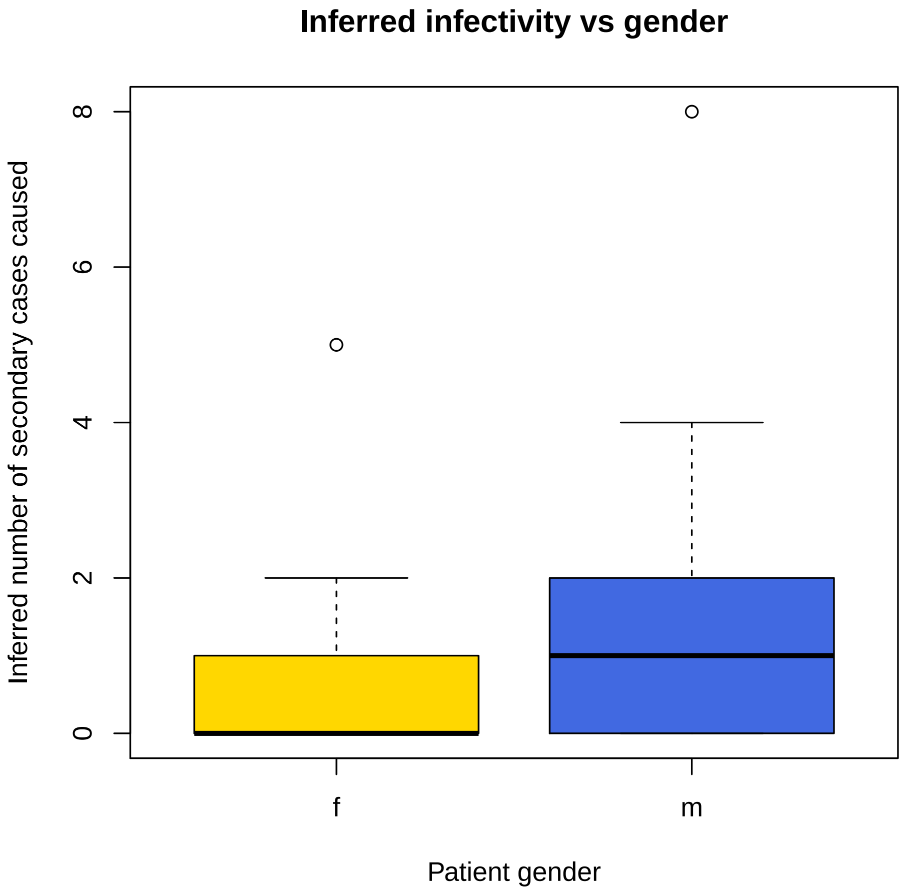
```
> toKeep <- days<9
```

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

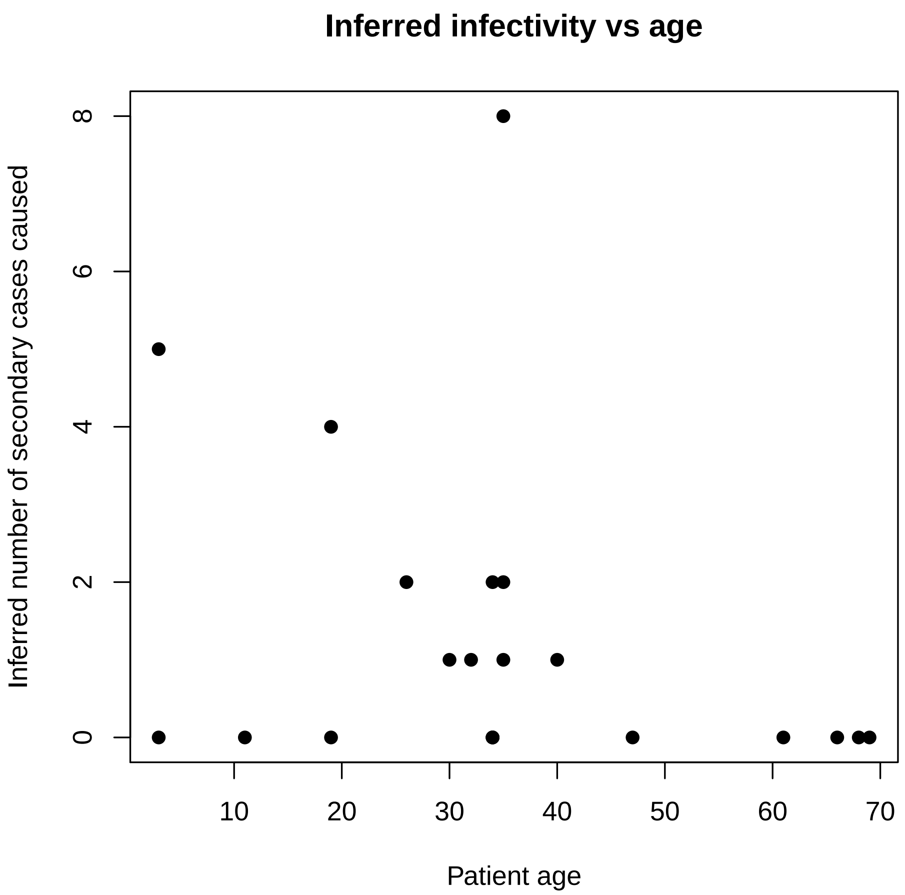
```
> head(cases)

  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:

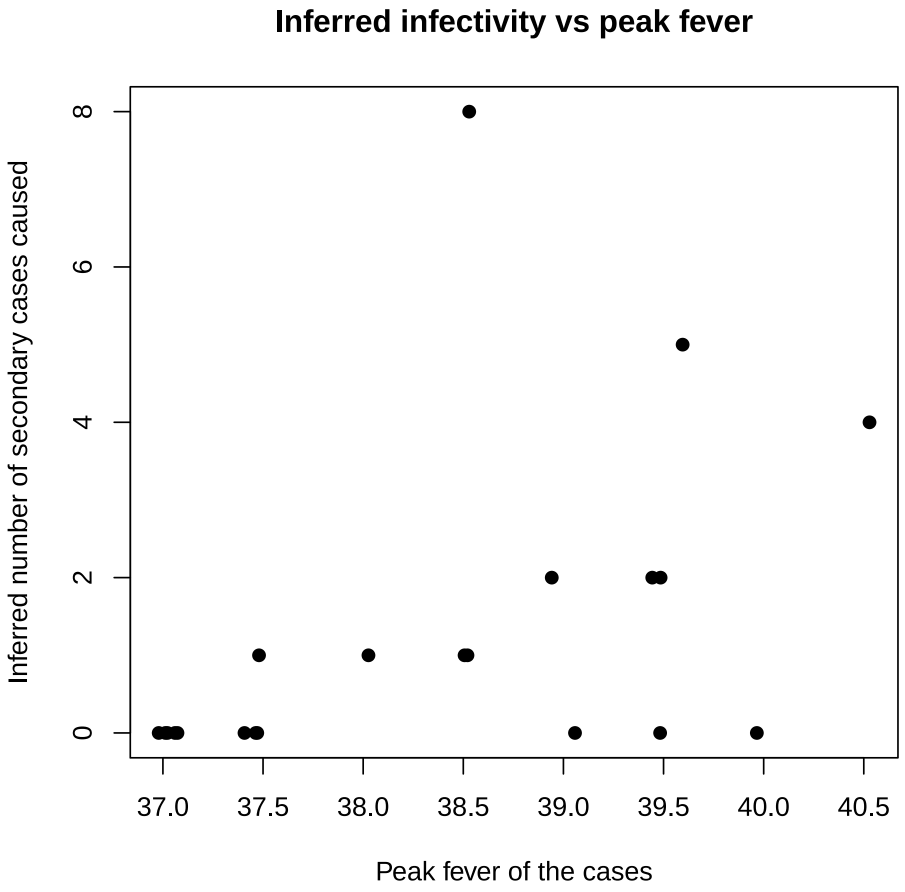
```
> boxplot(Rindiv[toKeep]~cases$sex[toKeep], xlab="Patient gender",
+      ylab="Inferred number of secondary cases caused", col=c("gold","royalblue"))
> title("Inferred infectivity vs gender")
```



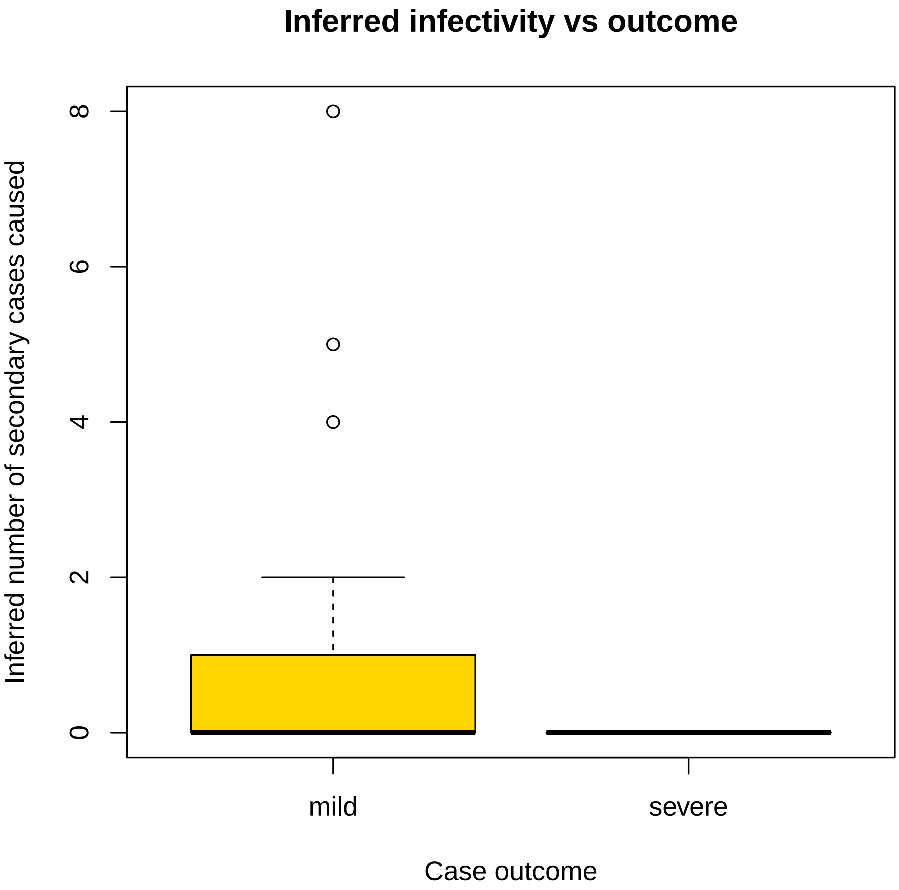
```
> t.test(Rindiv[toKeep]~cases$sex[toKeep])

        Welch Two Sample t-test

data:  Rindiv[toKeep] by cases$sex[toKeep]
t = -1.0619, df = 14.579, p-value = 0.3056
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2.9575159  0.9938795
sample estimates:
mean in group f mean in group m
      0.8181818       1.8000000


> 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 = 1960.259, p-value = 0.2314
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho
-0.2728957


> 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 = 661.06, p-value = 0.006892
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.5707403

> 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 = 2.7605, df = 24, p-value = 0.01088
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.2725258 1.8874742
sample estimates:
  mean in group mild mean in group severe
                1.08                 0.00
```

5.3 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 *SeqTrack*'s results 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.

```
> res$ances[is.na(res$ances)] <- 0
> newinfo$infectors[is.na(newinfo$infectors)] <- 0
> comp <- rbind(res$ances[1:25], newinfo$infectors)
> rownames(comp) <- c("SeqTrack","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      0      3      4      4      5      4
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
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
investigations       8       9      10      11      11      13      13      13
```

```
> mean(comp[1,]==comp[2,])
```

```
[1] 0.72
```

Not too bad: `SeqTrack` and the detailed field investigations agree in 72% of cases.

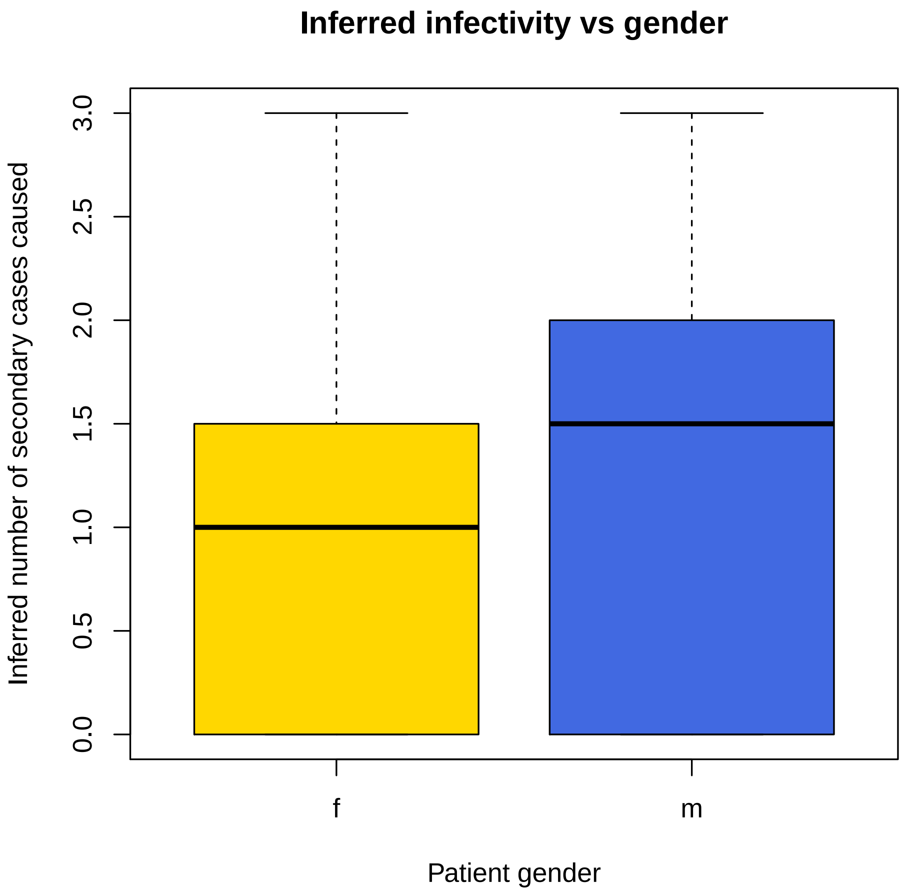
Let us examine again the possible effect of covariates on individual reproduction numbers R_i, this time computing R_i from the investigation data:

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

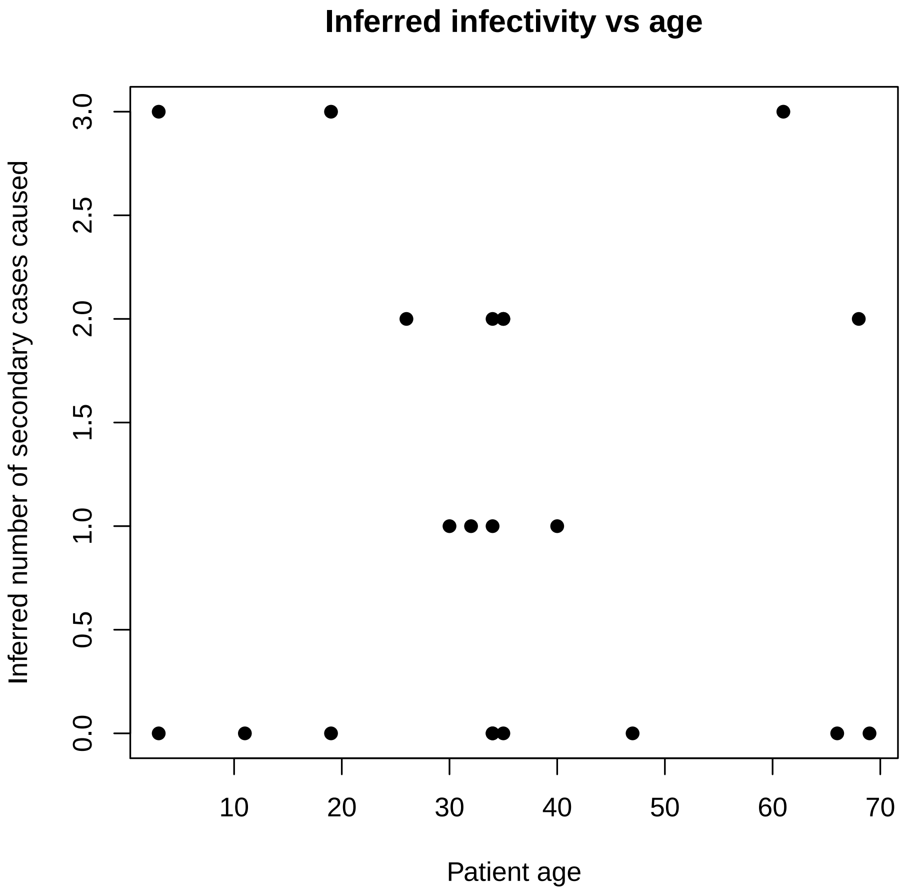
```
 case1  case2  case3  case4  case5  case6  case7  case8  case9 case10 case11 
     1      1      1      2      3      2      3      2      2      1      2 
case12 case13 case14 case15 case16 case17 case18 case19 case20 case21 case22 
     0      3      0      0      0      0      0      0      0      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:

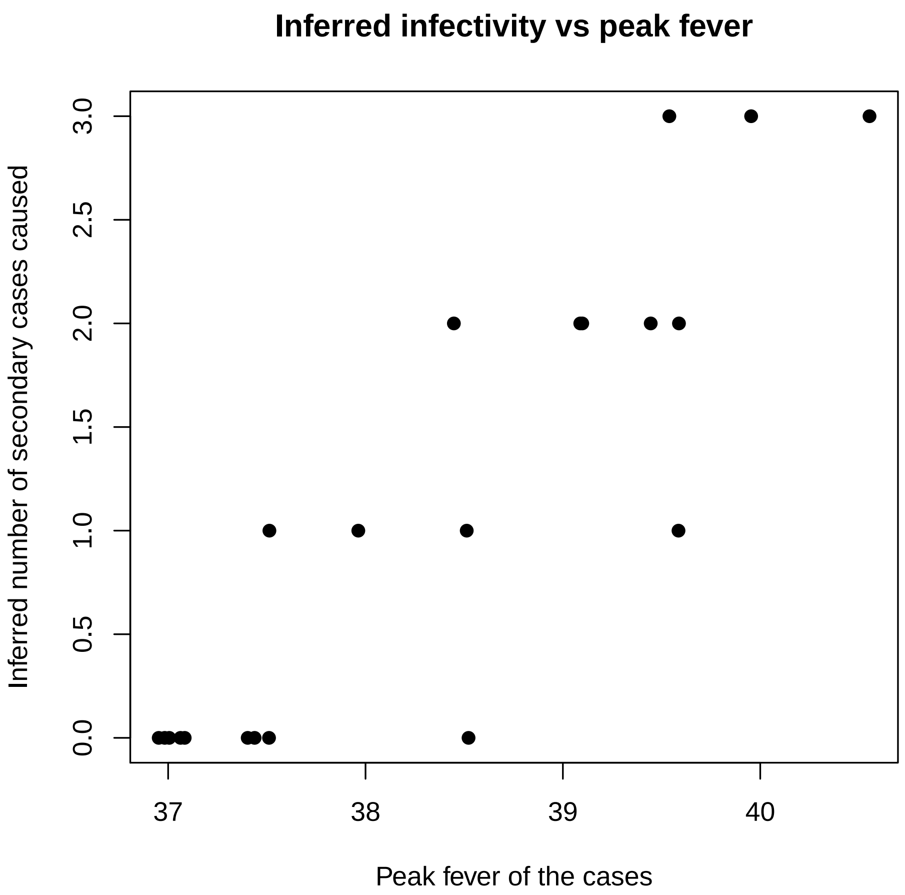
```
> boxplot(Rindiv2[toKeep]~cases$sex[toKeep], xlab="Patient gender",
+      ylab="Inferred number of secondary cases caused", col=c("gold","royalblue"))
> title("Inferred infectivity vs gender")
```



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

        Welch Two Sample t-test

data:  Rindiv2[toKeep] by cases$sex[toKeep]
t = -0.7728, df = 17.639, p-value = 0.4498
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.4551288  0.6733106
sample estimates:
mean in group f mean in group m
      0.9090909       1.3000000


> 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")
```

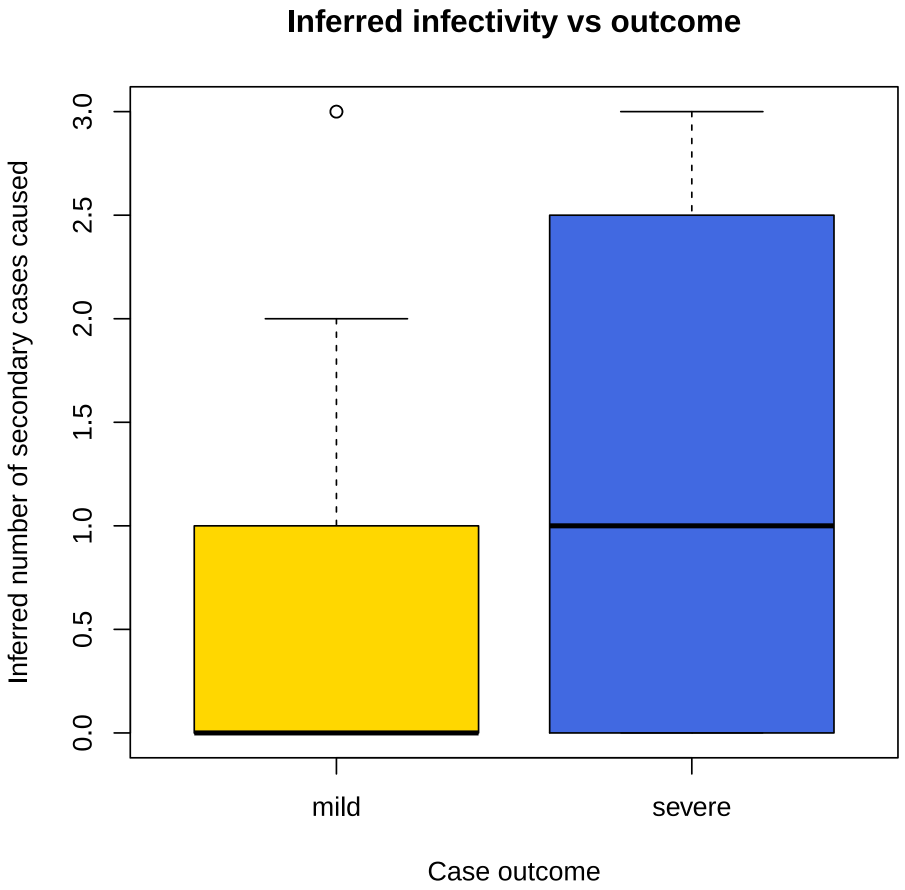


```
> cor.test(Rindiv2[toKeep],cases$age[toKeep], method="spearman")

        Spearman's rank correlation rho

data:  Rindiv2[toKeep] and cases$age[toKeep]
S = 1639.074, p-value = 0.7817
alternative hypothesis: true rho is not equal to 0
sample estimates:
        rho
-0.06433355

> 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 = 185.5836, p-value = 1.513e-07
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.8794912
```

```
> 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:
 -2.859747  1.744363
sample estimates:
  mean in group mild mean in group severe
           0.6923077            1.2500000
```

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?

References

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

[2] 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.

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

[4] 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.