Reconstructing disease outbreaks from genetic data: a graph approach

Thibaut Jombart, Rosalind M. Eggo, Peter J. Dodd and François Balloux

Supplementary information

Contents

Simulating genetic data of disease outbreaks

Individual-based simulations were used to assess the ability of *SeqTrack* to recover the transmission tree of a pathogen during a densely sampled outbreak. We implemented a new simulation system in the *adegenet* package (http://adegenet.r-forge.r-project.org/) for the R software, which creates genealogies of haplotypes using a forward-time process. The function *haploGen* simulates haplotypes which replicate, mutate, and possibly disperse following stochastic processes, resulting in temporally and spatially referenced genealogies.

Each simulation begins with a randomly generated haplotype of determined length which forms the root of the genealogy. The number of descendents of this isolate is drawn from a (rounded) normal distribution with specified mean and standard deviation. Descendents are created by replicating the haplotype of their ancestor. During this process, the number of mutations from an ancestor to its descendent is drawn from a binomial distribution with the mutation rate and the length of the DNA sequence as parameters. The nature and positions of the mutations are then determined at random, with all positions and types of mutations having equal probabilities. The whole process of replication / mutation is repeated over a determined number of generations. The simulation output consists of haplotypes and dates of creation of the isolates, as well as a list of ancestries describing the genealogy.

One issue pertaining to the simulation of transmission trees of a disease outbreak is that the number of isolates grows exponentially, quickly leading to handling millions of genotypes and eventually saturating the memory. Practically, this means that only very short complete genealogies can be simulated. One convenient solution to overcome this issue lies in pruning the genealogy periodically during its expansion, as soon as the number of isolates simulated exceeds a given threshold. The resulting tree then becomes a sample of the complete genealogy, which is specifically the type of data that we aim to simulate.

Two types of simulations were performed in this study. The first type aimed at illustrating differences between *SeqTrack* and classical phylogenetic reconstruction using two very simple genealogies (main text, Figures 2-3). These simulations were performed using a mutation rate of $1x10^{-4}$, haplotypes of 10,000 nucleotides, a fixed generation time of one day, and a reproductive rate drawn from a rounded normal distribution N(1.5, 2). The second type of simulations was slightly more complex and computer-intensive, and aimed at quantifying the performances of *SeqTrack* for retrieving spatiotemporal dynamics of outbreaks using genetic data. Values of the parameters used in these simulations are provided in Table S1. Contrary to the previous simulations, these simulations were spatially explicit. The simulation system used is identical to the one described above, except that newly created isolates are assigned to a location on a spatial grid. The location of a new isolate is determined as a function of the location of its ancestor and a spatial model of dispersal. Currently, two spatial models are implemented. The first model uses a *random diffusion* process, where deviations from the ancestral location in x and y coordinates are drawn from a Poisson distribution of fixed parameter. The second spatial model consists in a *fully-parameterised dispersal* process, in which the probabilities of migration between every pair of locations are specified explicitly.

Table S1: values of the parameters used in simulated outbreaks

Parameters (units)	Values
Sequence length (number of nucleotides)	5,000
Mutation rate (number of mutations per year and per site)	0.008
Mean generation time (days)	5
Standard deviation of generation time (days)	1
Mean reproductive rate (number of descendents)	1.2
Standard deviation of reproductive rate (number of descendents)	0.5
Simulated time period (number of days after emergence)	100

A square grid with 25 locations was used in all spatially-explicit simulations. Spatial dispersal was modelled using the two afore-mentioned spatial models. For the random diffusion process (referred to as '*homogeneous dispersal*' in the main text), deviation from ancestral coordinates was drawn from a Poisson distribution with λ=0.5 (Figure S1). Structured dispersal (referred to as '*heterogeneous dispersal*' in the main text) was obtained by explicitly defining the matrix of connectivity between locations, *i.e.* the probabilities for the pathogen to be transmitted from one location to another (Figure S2). This matrix was designed so that i) one location attracted moderate migration from its neighbours, but systematically dispersed toward distant locations (the 'source') ii) one location attracted immigration, but did not allow resident isolates to seed other locations (the 'sink'), and iii) all other locations dispersed moderately to neighbouring locations.

Ten datasets were simulated for each spatial model, from which ten samples of 800 randomly chosen isolates were obtained. To recreate the possible uncertainty existing about collection dates observed during real outbreaks, we added random noise to dates of apparition of the isolates taken from a Poisson distribution (λ=1). The resulting 200 datasets were then analysed. Inferred ancestries were compared to the true ancestries, in terms of the proportion of successfully inferred ancestral haplotype and location. Only perfect matches were considered as successes in these computations: situations in which haplotypes or locations were close, but not strictly identical to the actual ancestors were counted as failures.

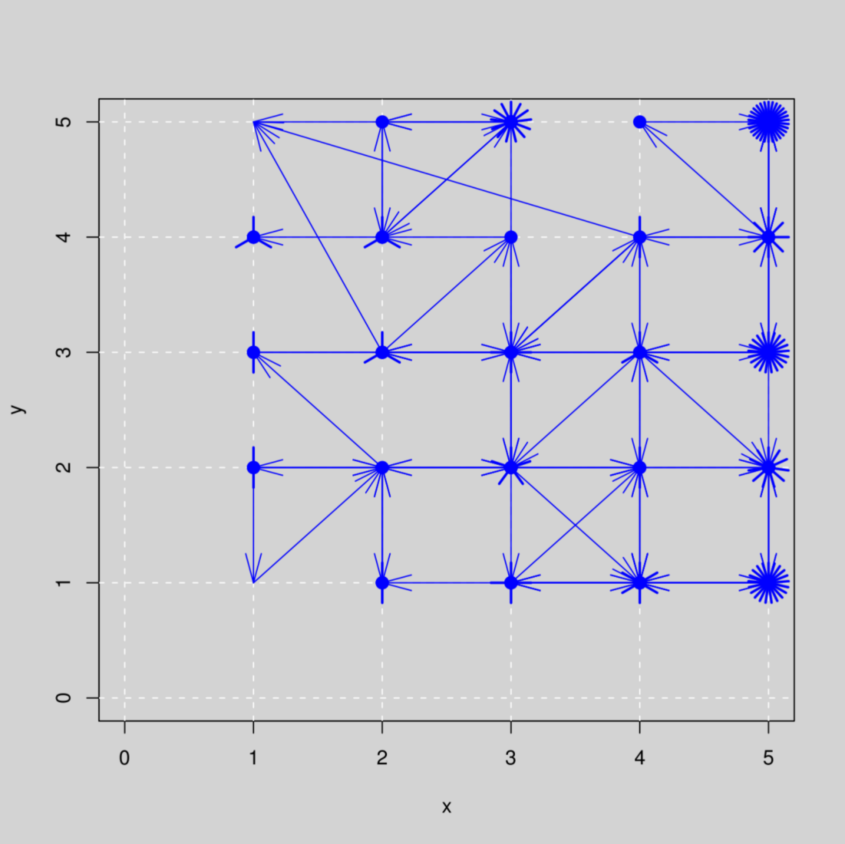

Figure S1: example of simulated outbreak data using *random diffusion* process. The figure maps ancestries of simulated data. Transmissions between locations are represented by arrows. Transmissions within a given location are represented by a sunflower (each segment represents one transmission event).

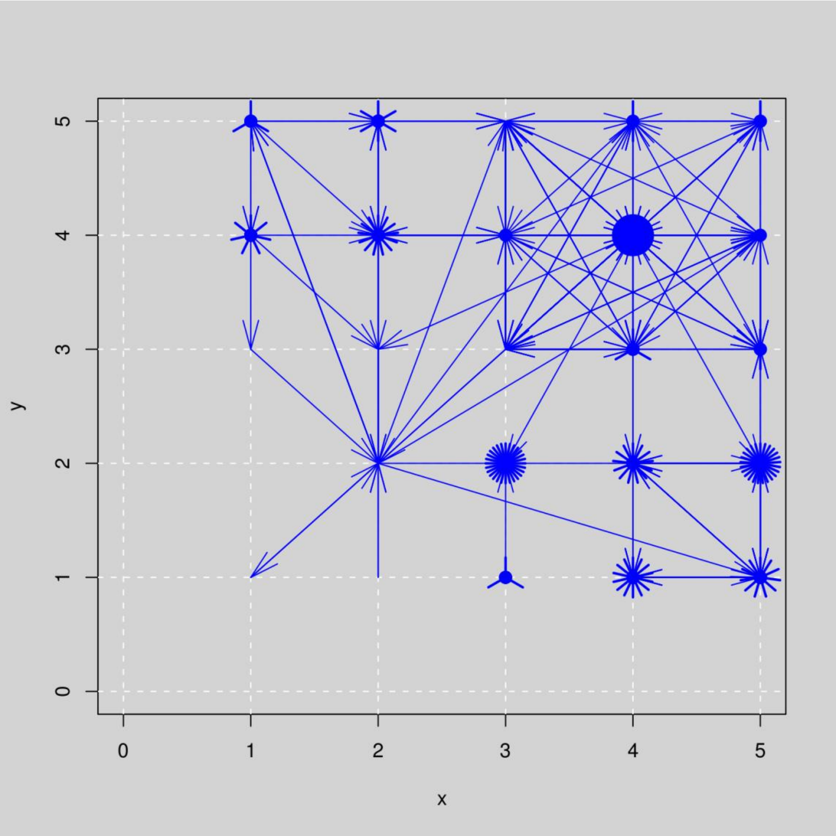

Figure S2: example of simulated outbreak data using *structured dispersal* process. The figure maps ancestries of simulated data. Transmissions between locations are represented by arrows. Transmissions within a given location are represented by a sunflower (each segment represents a single transmission event). The 'source' and the 'sink' are the populations with coordinates (2,2) and (4,4), respectively.

R Script used to simulate outbreak data

```
#### SIM 1: RANDOM DIFFUSION ####

## LIBRARY ##
library(MASS)
library(adegenet) # must be the devel version
library(ape)

## PARAMETERS FOR RANDOM DIFFUSION
SEQLENGTH <- 5000
TXMUT <- .008
GENTIME <- 5
GENTIME.SD <- 1
REPRO <- 1.2
REPRO.SD <- .5
TMAX <- 100
MAXNBSEQ <- 2000
DISP <- 0.3
GRID <- matrix(1:25,ncol=5)
GRID

NBSIM <- 10
NBSAMP <- 10
SAMPSIZE <- 800
INI.N <- 10


## SIMULATE DATA ##

for(simIdx in 1:NBSIM){
    mySim <- list(ances=NA) # initialization
    while(sum(!is.na(mySim$ances)) < 500){
        ## MAKE SIMULATION
        mySim <- haploGen(seq.le=SEQLENGTH, Tmax=TMAX, mu=TXMUT,
                          mean.gen=GENTIME, sd.gen=GENTIME.SD,
                          mean.repr=REPRO, sd.repro=REPRO.SD,
                          max.nb=MAXNBSEQ, lambda.xy=DISP, grid.size=5,
                          ini.n=INI.N)
     } # end while

    save(mySim, file=paste("sim.unif", simIdx, "RData", sep="."))

    ## GET SAMPLES
    for(sampIdx in 1:NBSAMP){
        mySamp <- sample.haploGen(mySim, SAMPSIZE, rDate=rpois,
arg.rDate=list(lambda=1))
        fileName <- paste("sim.unif", simIdx, sampIdx, "RData", sep=".")
        save(mySamp, file=fileName)
    }
}

#### SIM 2: STRUCTURED DISPERSAL ####

## PARAMETERS, RANDOM DIFFUSION
SEQLENGTH <- 5000
TXMUT <- .008
GENTIME <- 5
```

```
GENTIME.SD <- 1
REPRO <- 1.2
REPRO.SD <- .5
TMAX <- 100
MAXNBSEQ <- 2000
DISP <- 0.3
GRID <- matrix(1:25,ncol=5)
GRID
INI.N <- 10
INI.XY <- c(2,2)

DISPMAT <- matrix(0,ncol=25,nrow=25)
diag(DISPMAT) <- 1
## puits
DISPMAT[c(13,14,15,18,20,23,24,25),-19] <- 0
DISPMAT[c(13,14,15,18,20,23,24,25),19] <- 0.7 # apport puits
DISPMAT[c(13,14,15,18,20,23,24,25) , c(13,14,15,18,20,23,24,25)] <- 0.3
DISPMAT[19,] <- 0.01
DISPMAT[19,19] <- 0.96 # diffusion puits = 0
## source
DISPMAT[c(1,6,11,2,12),-7] <- 0 # apport pour la source
DISPMAT[c(1,6,11,2,12),7] <- 0.95
DISPMAT[c(1,6,11,2,12),c(1,6,11,2,12)] <- 0.05
DISPMAT[7, ] <- 0 # diffusion depuis la source
DISPMAT[7, c(4,5,10,15,20,21,22,23,24)] <- 1/9
## misc points with immediate neighbour connectivity
DISPMAT[3, c(2,8,4)] <- DISP/3 # 3
DISPMAT[3,3] <- 1-DISP
DISPMAT[4, c(3,5,9)] <- DISP/3 # 4
DISPMAT[4,4] <- 1-DISP
DISPMAT[5, c(4,9,10)] <- DISP/3 # 5
DISPMAT[5,5] <- 1-DISP
DISPMAT[9, c(4,8,10,14)] <- DISP/4 # 9
DISPMAT[9,9] <- 1-DISP
DISPMAT[10, c(5,9,15)] <- DISP/3 # 10
DISPMAT[10,10] <- 1-DISP
DISPMAT[16, c(11,12,17,21,22)] <- DISP/5 # 16
DISPMAT[16,16] <- 1-DISP
DISPMAT[17, c(18,12)] <- DISP/2 # 17
DISPMAT[17,17] <- 1-DISP
DISPMAT[21, c(16,17,22)] <- DISP/3 # 21
DISPMAT[21,21] <- 1-DISP
DISPMAT[22, c(17,18,23)] <- DISP/3 #
DISPMAT[22,22] <- 1-DISP
## DISPMAT[, c(,,)] <- .2/3 #
## DISPMAT[,] <- 0.8
## DISPMAT[, c(,,)] <- .2/3 #
## DISPMAT[,] <- 0.8
## DISPMAT[, c(,,)] <- .2/3 #
## DISPMAT[,] <- 0.8
## DISPMAT[, c(,,)] <- .2/3 #
## DISPMAT[,] <- 0.8


DISPMAT <- prop.table(DISPMAT,1)

NBSIM <- 10
```

```
NBSAMP <- 10
SAMPSIZE <- 800



## SIMULATE DATA ##

for(simIdx in 1:NBSIM){
    mySim <- list(ances=NA) # initialization
    while(sum(!is.na(mySim$ances)) < 500){
        ## MAKE SIMULATION
        mySim <- haploGen(seq.le=SEQLENGTH, Tmax=TMAX, mu=TXMUT,
                          mean.gen=GENTIME, sd.gen=GENTIME.SD,
                          mean.repr=REPRO, sd.repro=REPRO.SD,
                          max.nb=MAXNBSEQ, grid.size=5, matConnect=DISPMAT,
                          ini.n=INI.N, ini.xy=INI.XY)
    } # end while

    save(mySim, file=paste("sim.stru", simIdx, "RData", sep="."))

    ## GET SAMPLES
    for(sampIdx in 1:NBSAMP){
        mySamp <- sample.haploGen(mySim, SAMPSIZE, rDate=rpois,
arg.rDate=list(lambda=1))
        fileName <- paste("sim.stru", simIdx, sampIdx, "RData", sep=".")
        save(mySamp, file=fileName)
    }
}




## TO VISUALIZE THE LATEST DATASET  ##

plotHaploGen(mySamp)
```

Genetic distances of inferred ancestries for the pdH1N1 data

The number of mutations between a given isolate and its ancestor inferred by *SeqTrack* can be used to assess whether the correct ancestral haplotype has actually been sampled, and has been identified by the method. For instance, it is obvious that the correct ancestral haplotype has been sampled when the descendent differs from the inferred ancestor by a single mutation. Conversely, it is unlikely that the ancestral haplotype of a given isolate has been sampled whenever the closest haplotype to this isolate differs from it by several mutations. The analysis of swine-origin A/H1N1 pandemic influenza data by *SeqTrack* suggests that the correct ancestral haplotype had actually been sampled in a large number of cases, as descendents often differ from their ancestors by no or a single mutation (Figure S3).

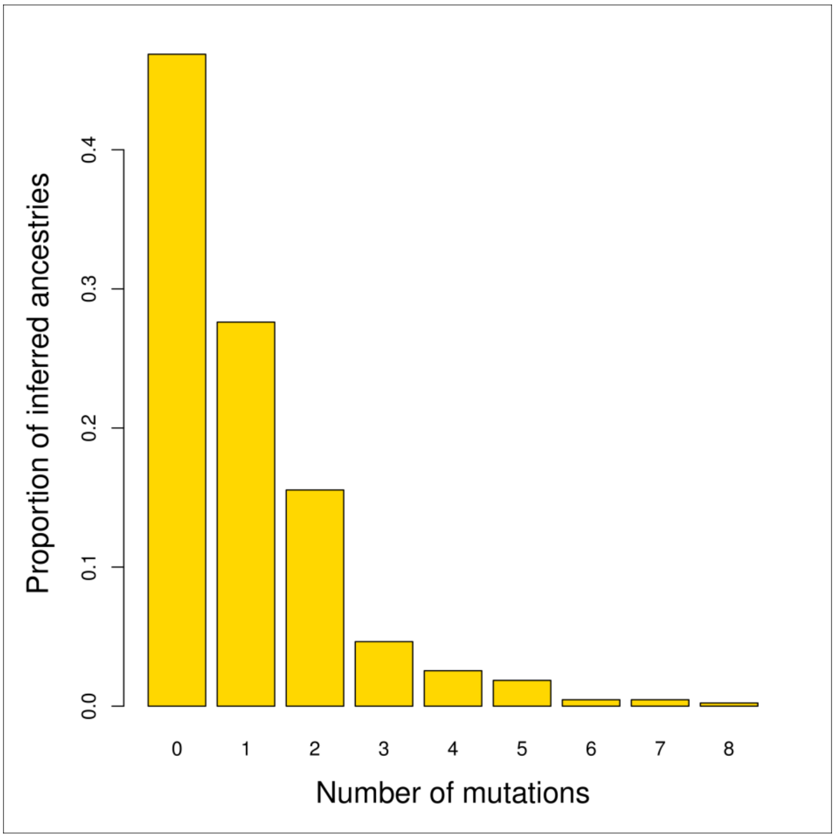

Figure S3: genetic distances of ancestries inferred by the *SeqTrack* analysis of swine-origin A/H1N1 pandemic influenza data. This figure shows the distribution of pairwise genetic distances (in number of mutations) between sampled isolates and their inferred ancestors.