Operational modelling for outbreak response

When are genomic data useful?

Thibaut Jombart

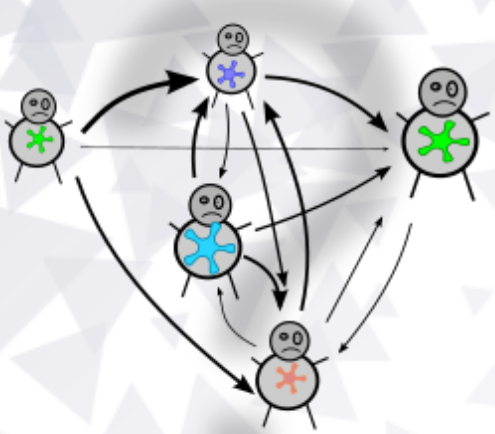
31st October 2018

London School of Hygiene and Tropical Medicine
Imperial College London

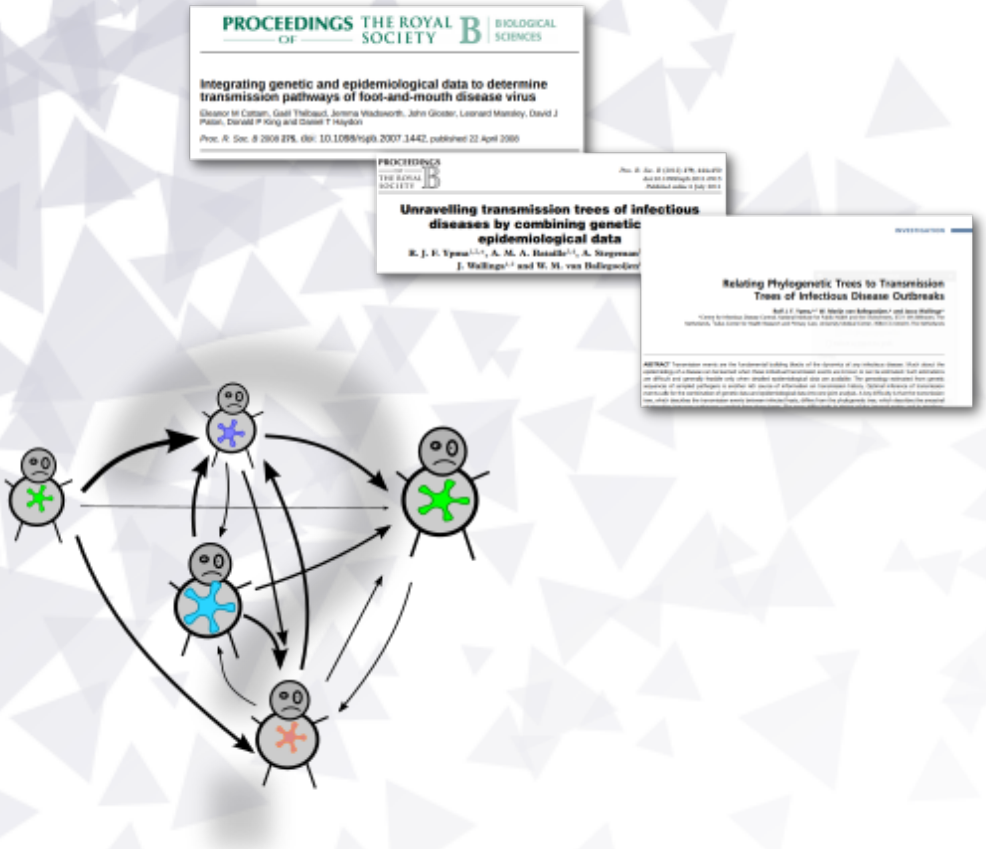

@teebzr

Reconstructing transmission trees

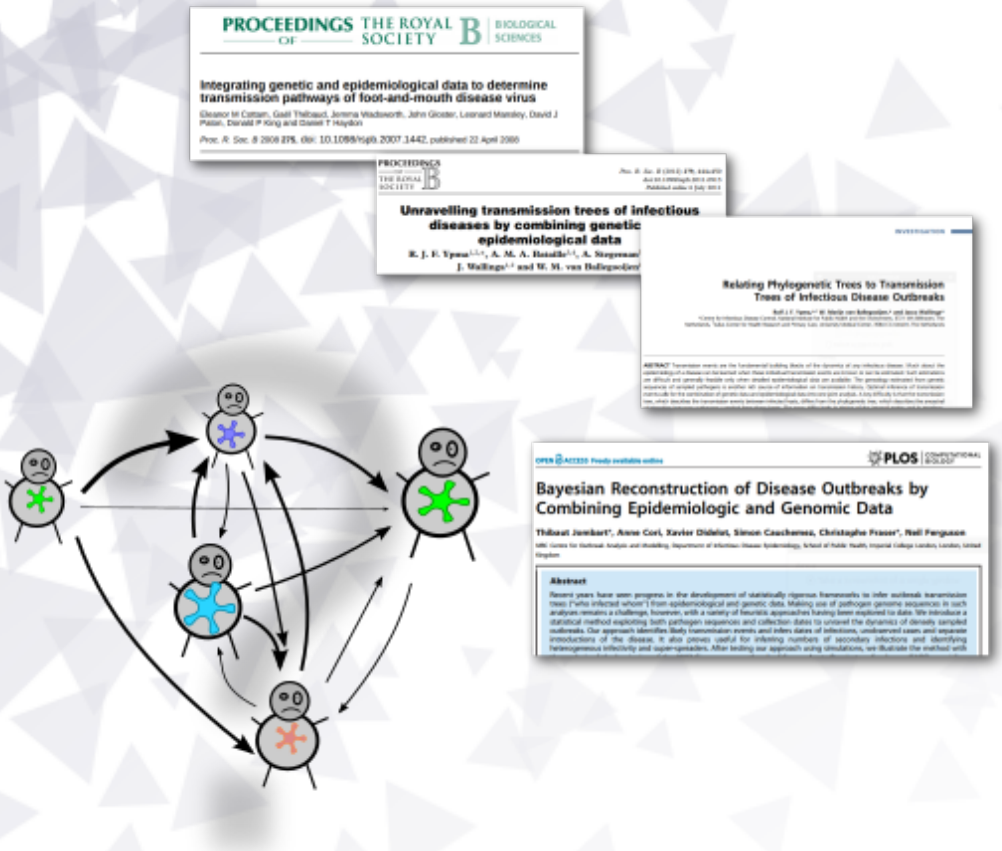
Who infects whom? Many answers for a single question



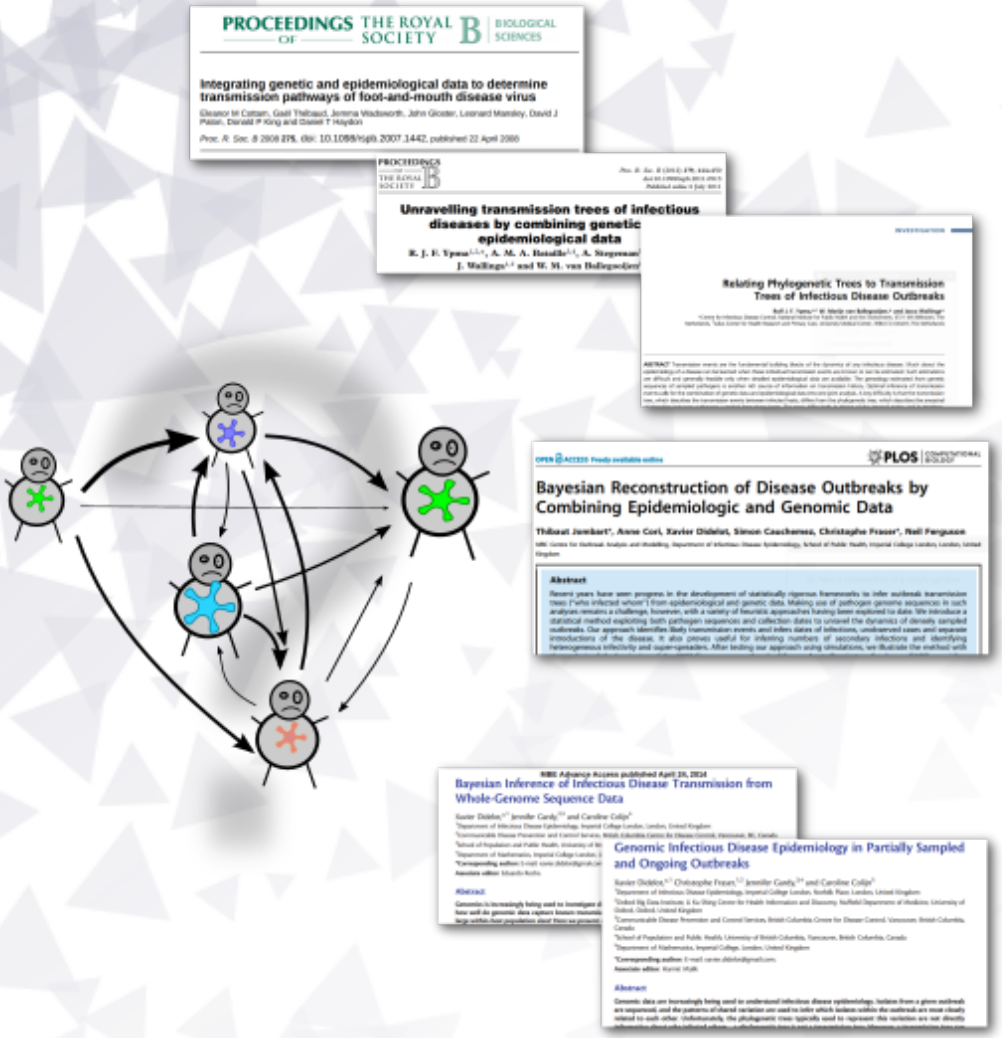
Who infects whom? Many answers for a single question



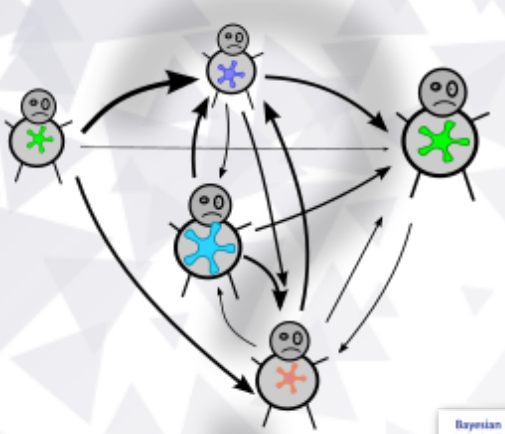
Who infects whom? Many answers for a single question



Who infects whom? Many answers for a single question



Who infects whom? Many answers for a single question



PROCEEDINGS OF THE ROYAL SOCIETY B BIOLOGICAL SCIENCES

Integrating genetic and epidemiological data to determine transmission pathways of foot-and-mouth disease virus

Unravelling transmission trees of infectious diseases by combining genetic epidemiological data

Relating Phylogenetic Trees to Transmission Trees of Infectious Disease Outbreaks

HHS Public Access

Author manuscript

Reconstructing transmission trees for communicable diseases using densely sampled genetic data

PLOS COMPUTATIONAL BIOLOGY

A Systematic Bayesian Integration of Epidemiological and Genetic Data

PLOS COMPUTATIONAL BIOLOGY

Bayesian Reconstruction of Disease Outbreaks by Combining Epidemiologic and Genomic Data

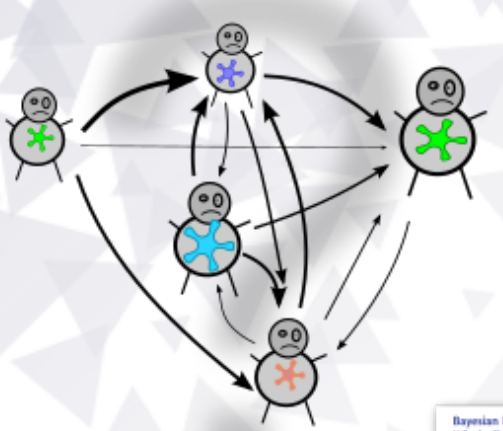
PROCEEDINGS OF THE ROYAL SOCIETY B BIOLOGICAL SCIENCES

A Bayesian approach for inferring the dynamics of partially observed endemic infectious diseases from space-time-genetic data

Bayesian Inference of Infectious Disease Transmission from Whole-Genome Sequence Data

Genomic Infectious Disease Epidemiology in Partially Sampled and Ongoing Outbreaks

Who infects whom? Many answers for a single question



PROCEEDINGS OF THE ROYAL SOCIETY B BIOLOGICAL SCIENCES

Integrating genetic and epidemiological data to determine transmission pathways of foot-and-mouth disease virus

Unravelling transmission trees of infectious diseases by combining genetic and epidemiological data

Relating Phylogenetic Trees to Transmission Trees of Infectious Disease Outbreaks

HHS Public Access

Author manuscript

Reconstructing transmission trees for communicable diseases using densely sampled genetic data

PLOS COMPUTATIONAL BIOLOGY

A Systematic Bayesian Integration of Epidemiological and Genetic Data

PLOS

Bayesian Reconstruction of Disease Outbreaks by Combining Epidemiologic and Genomic Data

PROCEEDINGS OF THE ROYAL SOCIETY B BIOLOGICAL SCIENCES

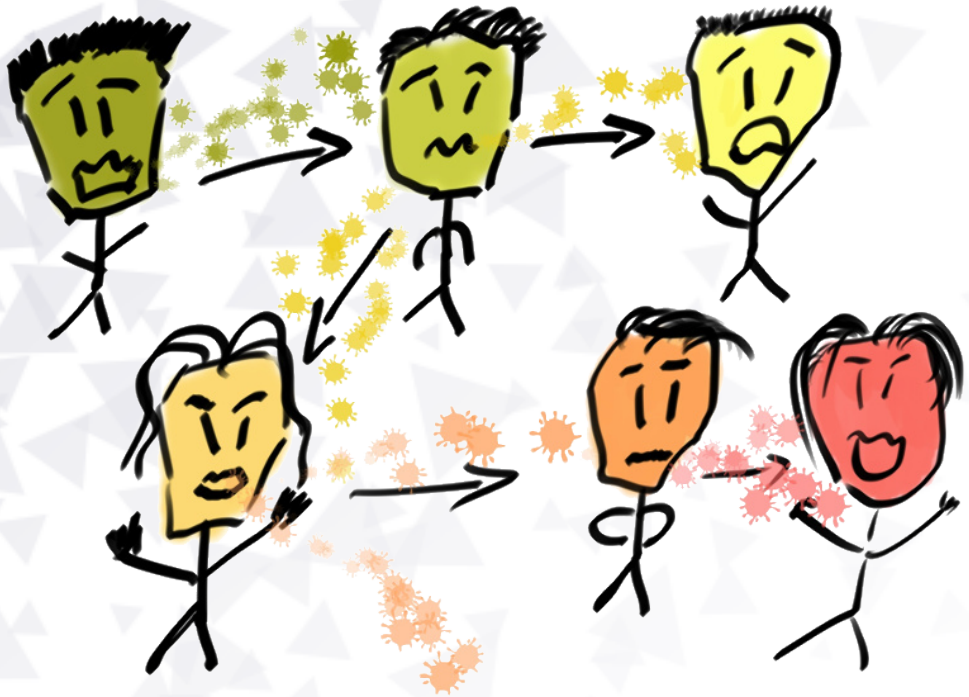
A Bayesian approach for inferring the dynamics of partially observed endemic infectious diseases from space-time-genetic data

Bayesian Inference of Infectious Disease Transmission from Whole-Genome Sequence Data

Genomic Infectious Disease Epidemiology in Partially Sampled and Ongoing Outbreaks

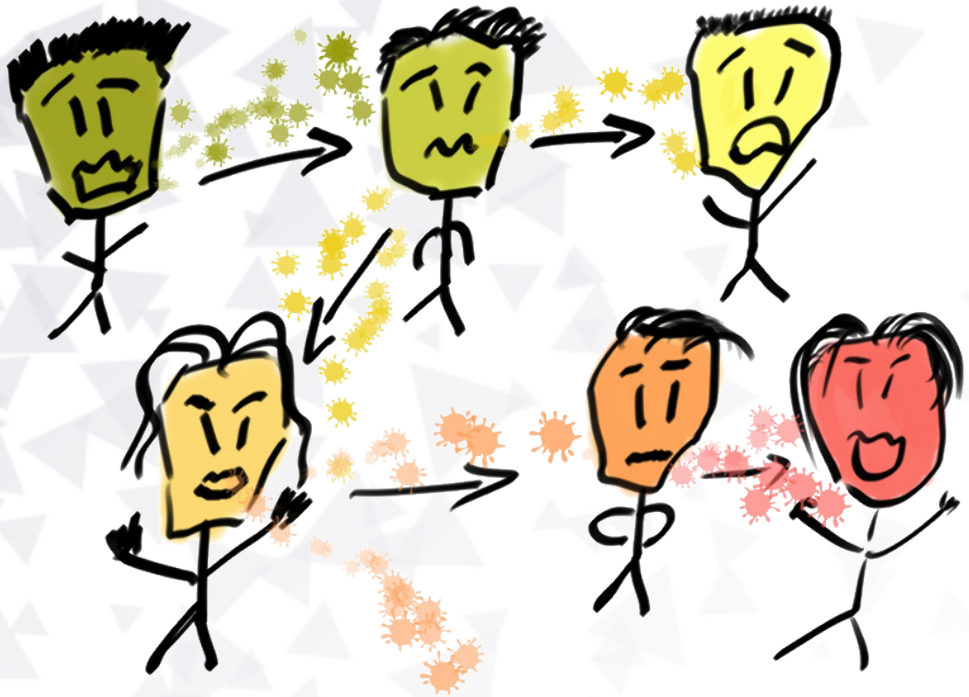
Methods heavily rely on whole genome sequence data

Using WGS to infer who infected whom



Mutations accumulate along transmission chains.

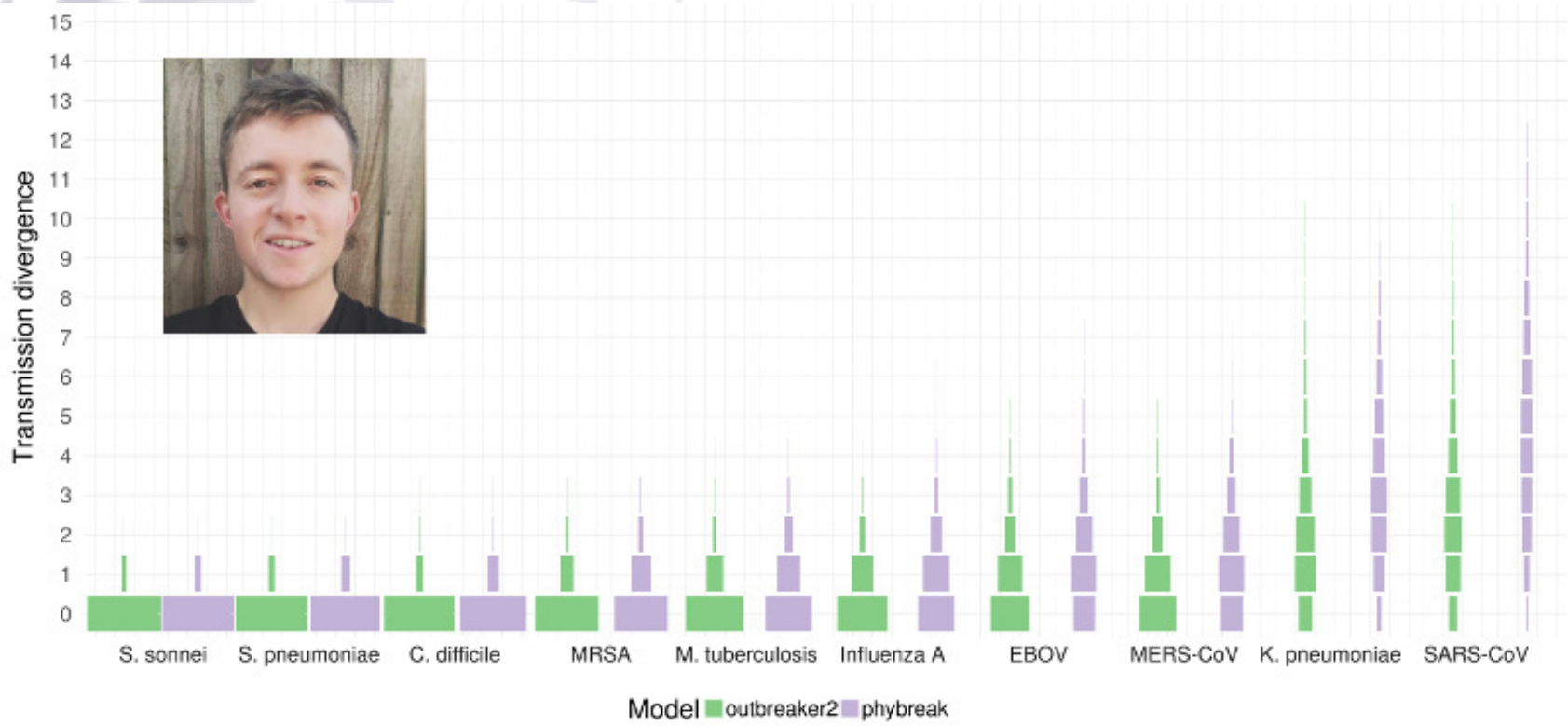
Using WGS to infer who infected whom



Mutations accumulate along transmission chains.

Can be used to reconstruct transmission trees.

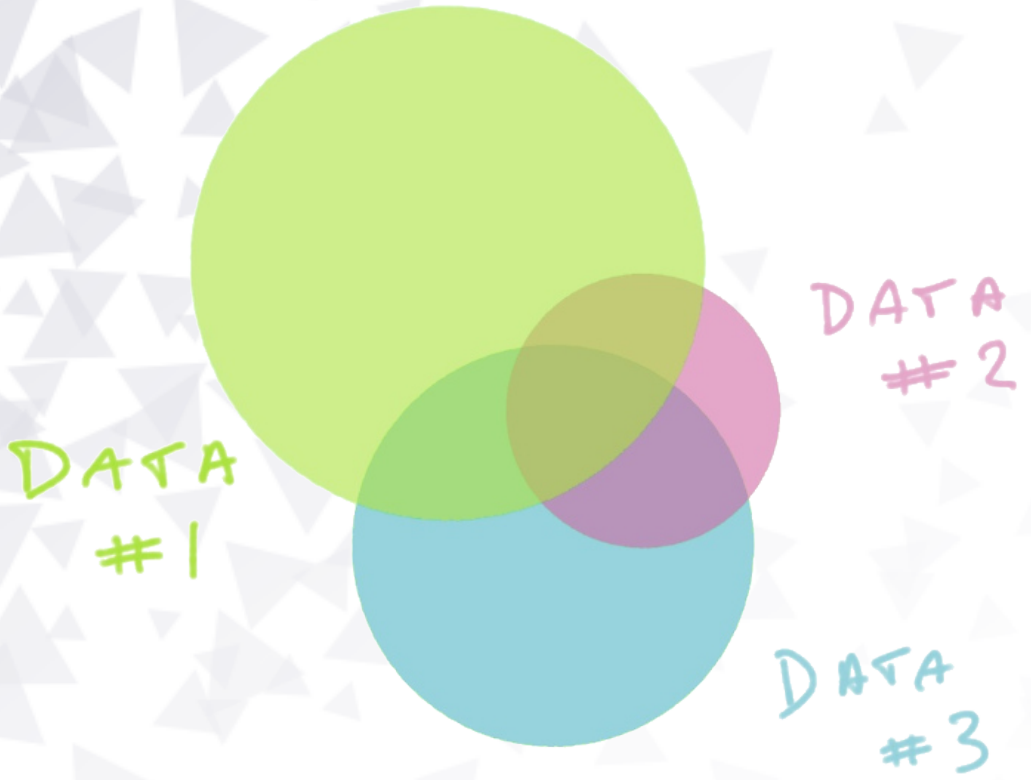
How informative are whole genome sequences?



[Campbell et al. (2018) PLoS Computational Biology]

Insufficient diversity for most diseases.

Evidence synthesis approach to outbreak reconstruction



Combine different data to shrink the set of plausible trees.

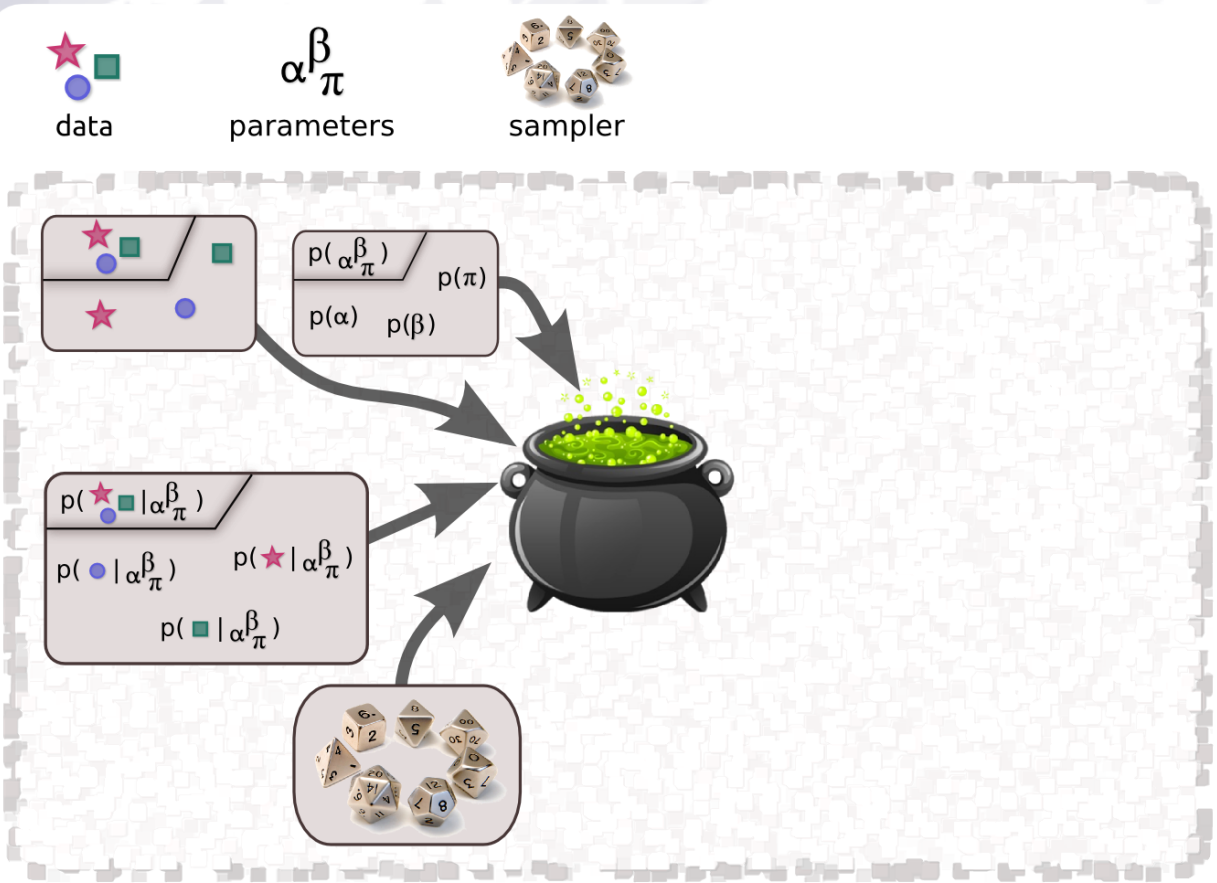
outbreaker2: evidence synthesis framework for outbreak reconstruction

Modularity: customise data, prior, likelihood, MCMC.

[Campbell et al. (2018) BMC Bioinformatics]

outbreaker2: evidence synthesis framework for outbreak reconstruction

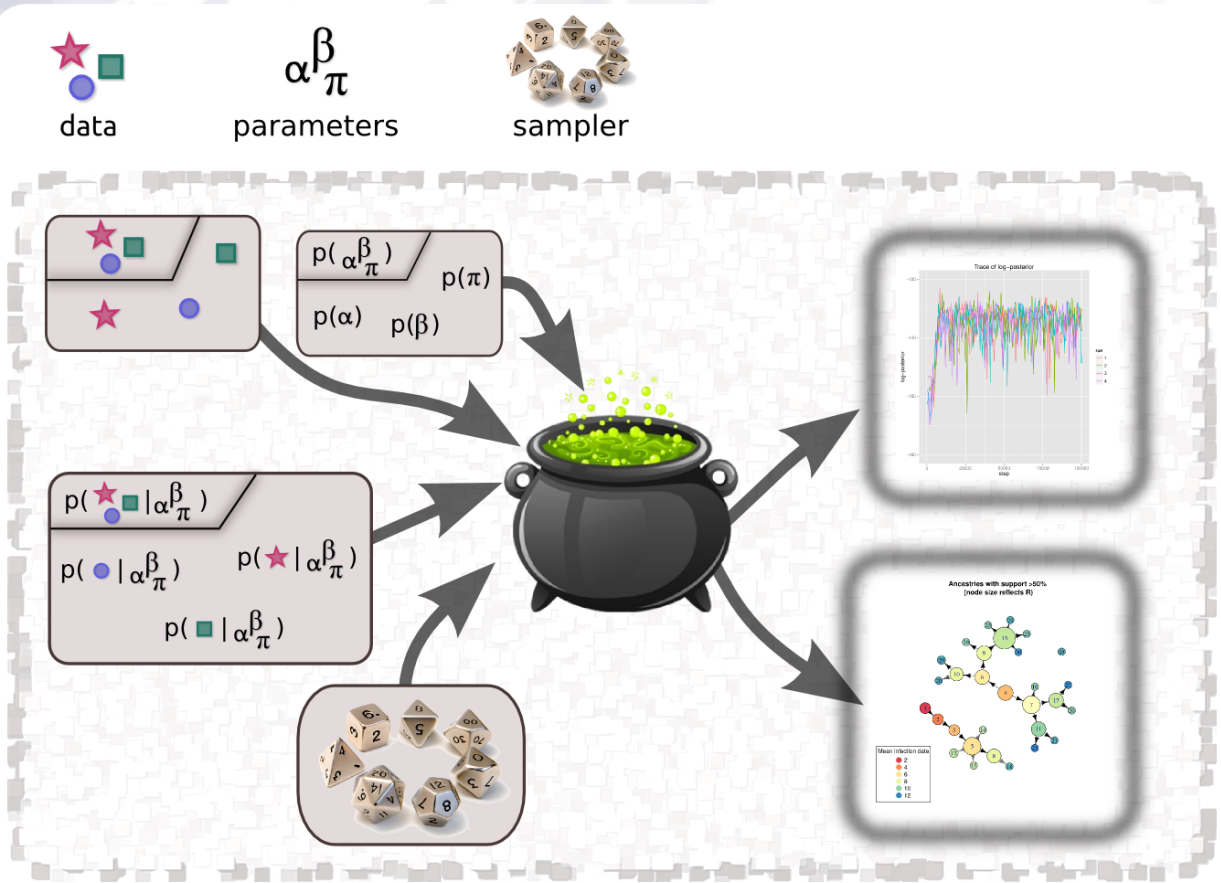
Modularity: customise data, prior, likelihood, MCMC.



[Campbell et al. (2018) BMC Bioinformatics]

outbreaker2: evidence synthesis framework for outbreak reconstruction

Modularity: customise data, prior, likelihood, MCMC.



[Campbell et al. (2018) BMC Bioinformatics]

Example: implementing *TransPhylo* in *outbreaker2*

outbreaker likelihood

- $p(s, t|\alpha, T^{inf}, \kappa, \mu, \pi) =$
  $p(T^{inf}|\alpha, \kappa)p(T^{inf}|t)p(s|\alpha, \kappa, \mu)p(\kappa|\pi)$
- i.e. *timing infection* x *incubation* x *genetic (simple)* x *missing cases*

Example: implementing *TransPhylo* in *outbreaker2*

outbreaker likelihood

- $p(s, t|\alpha, T^{inf}, \kappa, \mu, \pi) =$
  $p(T^{inf}|\alpha, \kappa)p(T^{inf}|t)p(s|\alpha, \kappa, \mu)p(\kappa|\pi)$
- i.e. *timing infection* x *incubation* x *genetic (simple)* x *missing cases*

TransPhylo likelihood

- $p(G|\beta, \gamma, N_e g, \alpha) = p(G|N_e g, \alpha) \times p(\alpha|\beta, \gamma)$
- i.e. *phylogeny (coalescent)* x *SIR*

Example: implementing *TransPhylo* in *outbreaker2*

outbreaker likelihood

- $p(s, t|\alpha, T^{inf}, \kappa, \mu, \pi) =$
  $p(T^{inf}|\alpha, \kappa)p(T^{inf}|t)p(s|\alpha, \kappa, \mu)p(\kappa|\pi)$
- i.e. *timing infection* x *incubation* x *genetic (simple)* x *missing cases*

TransPhylo likelihood

- $p(G|\beta, \gamma, N_e g, \alpha) = p(G|N_e g, \alpha) \times p(\alpha|\beta, \gamma)$
- i.e. *phylogeny (coalescent)* x *SIR*

Can we combine the two models?

TransPhylo module for *outbreaker2*

3 Custom likelihood

In order to calculate the likelihood under the TransPhylo model, we need to (i) extract the transmission tree from the outbreaker2 parameter, (ii) combine this transmission tree with the phylogenetic tree to form a colored tree, and (iii) calculate the likelihood of this colored tree. Step (i) is easy since transmission tree are encoded almost in the same way in TransPhylo and outbreaker2. For step (ii) we have to write the `combine` function which is tedious but not especially interesting (this function is included in this Rnw file but its code is not shown in the pdf). For step (iii) we only need to call the appropriate function of the TransPhylo package which is `probPTreeGivenTTree`. During step (ii) messages can arise indicating that the transmission tree and phylogenetic tree are in fact incompatible, in which case the likelihood is returned as -Inf.

```
lik_TransPhylo <- function(data, param) {
  ttree <- list(ttree = cbind(param$t_inf, data$dates, param$alpha),
                nam = data$ptree$nam)
  ttree$ttree[which(is.na(ttree$ttree[,3])),3] <- 0
  txt <- capture.output(ctree <- combine(ttree,data$ptree))
  if (length(txt)==0) {
    prob <- probPTreeGivenTTree(ctree, neg = 365 * 0.25)
    } else {
      prob <- -Inf
      }
  return(prob)
}
```

```
args(api$cpp_ll_all)

## function (data, param, i = NULL, custom_functions = NULL)
## NULL

new_move_tinf <- function(param, data, list_custom_ll = new_model) {
  for (i in 1:data$N) {
    current_ll <- api$cpp_ll_all(data,param, i = NULL, list_custom_ll)
    modif <- sample(c(-100:-1,1:100), 1)
    param$t_inf[i] <- param$t_inf[i] + modif
    new_ll <- api$cpp_ll_all(data, param, i = NULL, list_custom_ll)
    if (log(runif(1)) > (new_ll - current_ll)) {
      param$t_inf[i] <- param$t_inf[i] - modif
    }
  }
  return(param)
}

new_moves <- custom_moves(t_inf = new_move_tinf)
new_moves

##
##
##  ///// outbreaker movement functions ///
##
## class: outbreaker_moves list
## number of items: 8
##
## /// movement functions //
## $mu
```

[Campbell et al. (2018) BMC Bioinformatics]

TransPhylo module for *outbreaker2*

3 Custom likelihood

In order to calculate the likelihood under the TransPhylo model, we need to (i) extract the transmission tree from the outbreaker2 parameter, (ii) combine this transmission tree with the phylogenetic tree to form a colored tree, and (iii) calculate the likelihood of this colored tree. Step (i) is easy since transmission tree are encoded almost in the same way in TransPhylo and outbreaker2. For step (ii) we have to write the `combine` function which is tedious but not especially interesting (this function is included in this Rnw file but its code is not shown in the pdf). For step (iii) we only need to call the appropriate function of the TransPhylo package which is `probPTreeGivenTTree`. During step (ii) messages can arise indicating that the transmission tree and phylogenetic tree are in fact incompatible, in which case the likelihood is returned as -Inf.

```
lik_TransPhylo <- function(data, param) {
  ttree <- list(ttree = cbind(param$t_inf, data$dates, param$alpha),
                nam = data$ptree$nam)
  ttree$ttree[which(is.na(ttree$ttree[,3])),3] <- 0
  txt <- capture.output(ctree <- combine(ttree,data$ptree))
  if (length(txt)==0) {
    prob <- probPTreeGivenTTree(ctree, neg = 365 * 0.25)
    } else {
      prob <- -Inf
      }
  return(prob)
}
```

likelihood

```
args(api$cpp_ll_all)

## function (data, param, i = NULL, custom_functions = NULL)
## NULL

new_move_tinf <- function(param, data, list_custom_ll = new_model) {
  for (i in 1:data$N) {
    current_ll <- api$cpp_ll_all(data,param, i = NULL, list_custom_ll)
    modif <- sample(c(-100:-1,1:100), 1)
    param$t_inf[i] <- param$t_inf[i] + modif
    new_ll <- api$cpp_ll_all(data, param, i = NULL, list_custom_ll)
    if (log(runif(1)) > (new_ll - current_ll)) {
      param$t_inf[i] <- param$t_inf[i] - modif
    }
  }
  return(param)
}

new_moves <- custom_moves(t_inf = new_move_tinf)
new_moves

##
##
##  ///// outbreaker movement functions ///
##
## class: outbreaker_moves list
## number of items: 8
##
## /// movement functions //
## $mu
```

[Campbell et al. (2018) BMC Bioinformatics]

TransPhylo module for *outbreaker2*

3 Custom likelihood

In order to calculate the likelihood under the TransPhylo model, we need to (i) extract the transmission tree from the outbreaker2 parameter, (ii) combine this transmission tree with the phylogenetic tree to form a colored tree, and (iii) calculate the likelihood of this colored tree. Step (i) is easy since transmission tree are encoded almost in the same way in TransPhylo and outbreaker2. For step (ii) we have to write the `combine` function which is tedious but not especially interesting (this function is included in this Rnw file but its code is not shown in the pdf). For step (iii) we only need to call the appropriate function of the TransPhylo package which is `probPTreeGivenTTree`. During step (ii) messages can arise indicating that the transmission tree and phylogenetic tree are in fact incompatible, in which case the likelihood is returned as -Inf.

```
lik_TransPhylo <- function(data, param) {
  ttree <- list(ttree = cbind(param$t_inf, data$dates, param$alpha),
                nam = data$ptree$nam)
  ttree$ttree[which(is.na(ttree$ttree[,3])),3] <- 0
  txt <- capture.output(ctree <- combine(ttree,data$ptree))
  if (length(txt)==0) {
    prob <- probPTreeGivenTTree(ctree, neg = 365 * 0.25)
    } else {
      prob <- -Inf
      }
  return(prob)
}
```

likelihood

movement function

```
args(api$cpp_ll_all)

## function (data, param, i = NULL, custom_functions = NULL)
## NULL

new_move_tinf <- function(param, data, list_custom_ll = new_model) {
  for (i in 1:data$N) {
    current_ll <- api$cpp_ll_all(data,param, i = NULL, list_custom_ll)
    modif <- sample(c(-100:-1,1:100), 1)
    param$t_inf[i] <- param$t_inf[i] + modif
    new_ll <- api$cpp_ll_all(data, param, i = NULL, list_custom_ll)
    if (log(runif(1)) > (new_ll - current_ll)) {
      param$t_inf[i] <- param$t_inf[i] - modif
    }
  }
  return(param)
}

new_moves <- custom_moves(t_inf = new_move_tinf)
new_moves

##
##
##  ///// outbreaker movement functions ///
##
## class: outbreaker_moves list
## number of items: 8
##
## /// movement functions //
## $mu
```

[Campbell et al. (2018) BMC Bioinformatics]

TransPhylo module for *outbreaker2*

3 Custom likelihood

In order to calculate the likelihood under the TransPhylo model, we need to (i) extract the transmission tree from the outbreaker2 parameter, (ii) combine this transmission tree with the phylogenetic tree to form a colored tree, and (iii) calculate the likelihood of this colored tree. Step (i) is easy since transmission tree are encoded almost in the same way in TransPhylo and outbreaker2. For step (ii) we have to write the `combine` function which is tedious but not especially interesting (this function is included in this Rnw file but its code is not shown in the pdf). For step (iii) we only need to call the appropriate function of the TransPhylo package which is `probPTreeGivenTTree`. During step (ii) messages can arise indicating that the transmission tree and phylogenetic tree are in fact incompatible, in which case the likelihood is returned as -Inf.

```
lik_TransPhylo <- function(data, param) {
  ttree <- list(ttree = cbind(param$t_inf, data$dates, param$alpha),
                nam = data$ptree$nam)
  ttree$ttree[which(is.na(ttree$ttree[,3])),3] <- 0
  txt <- capture.output(ctree <- combine(ttree,data$ptree))
  if (length(txt)==0) {
    prob <- probPTreeGivenTTree(ctree, neg = 365 * 0.25)
    } else {
      prob <- -Inf
      }
  return(prob)
}
```

likelihood

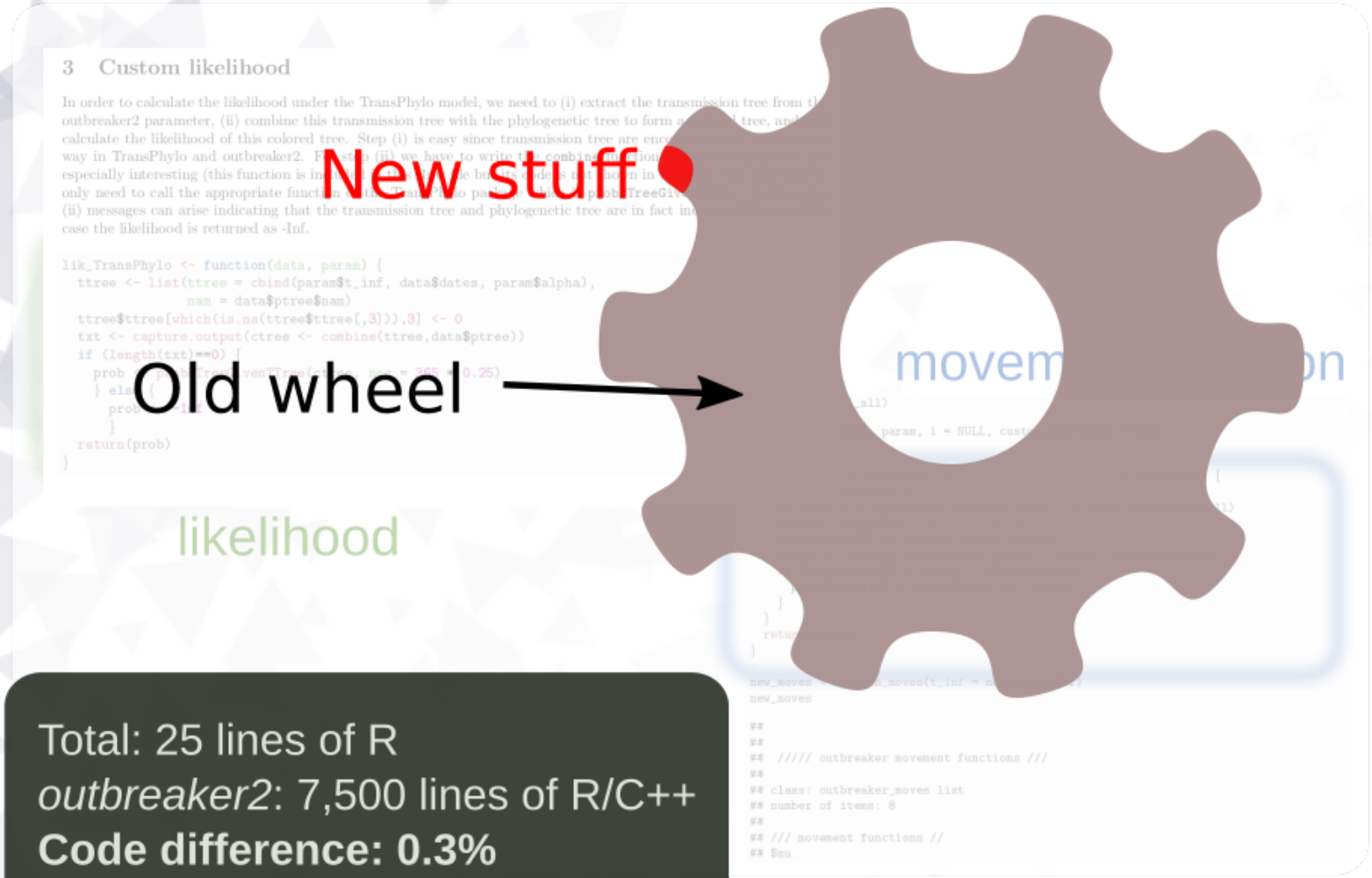
movement function

```
args(api$cpp_ll_all)

## function (data, param, i = NULL, custom_functions = NULL)
## NULL

new_move_tinf <- function(param, data, list_custom_ll = new_model) {
  for (i in 1:data$N) {
    current_ll <- api$cpp_ll_all(data,param, i = NULL, list_custom_ll)
    modif <- sample(c(-100:-1,1:100), 1)
    param$t_inf[i] <- param$t_inf[i] + modif
    new_ll <- api$cpp_ll_all(data, param, i = NULL, list_custom_ll)
    if (log(runif(1)) > (new_ll - current_ll)) {
      param$t_inf[i] <- param$t_inf[i] - modif
    }
  }
  return(param)
}

new_moves <- custom_moves(t_inf = new_move_tinf)
new_moves

##
##
##  ///// outbreaker movement functions ///
##
## class: outbreaker_moves list
## number of items: 8
##
## /// movement functions //
## $mu
```

Total: 25 lines of R
outbreaker2: 7,500 lines of R/C++
Code difference: 0.3%

[Campbell et al. (2018) BMC Bioinformatics]

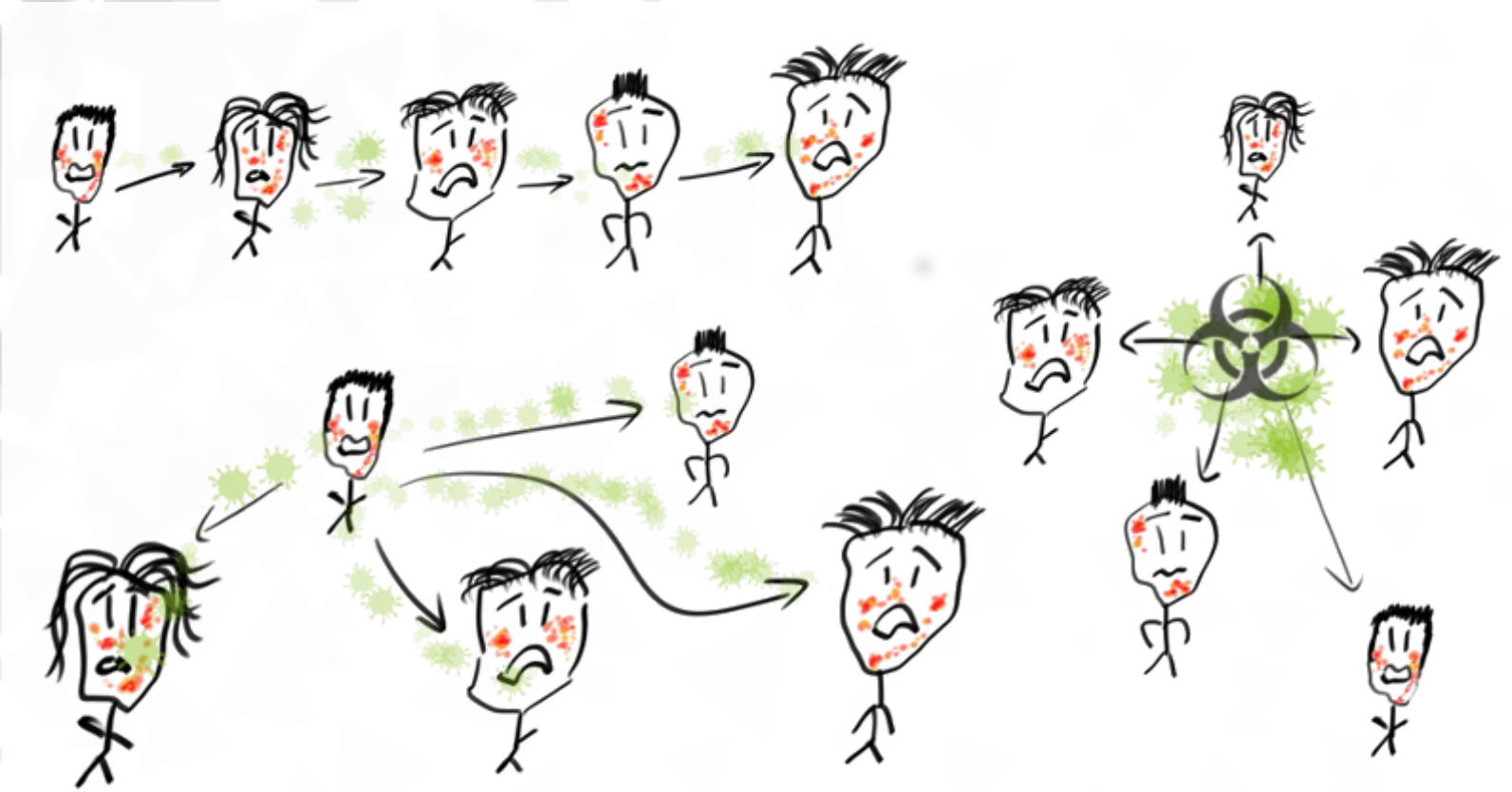
TransPhylo module for *outbreaker2*



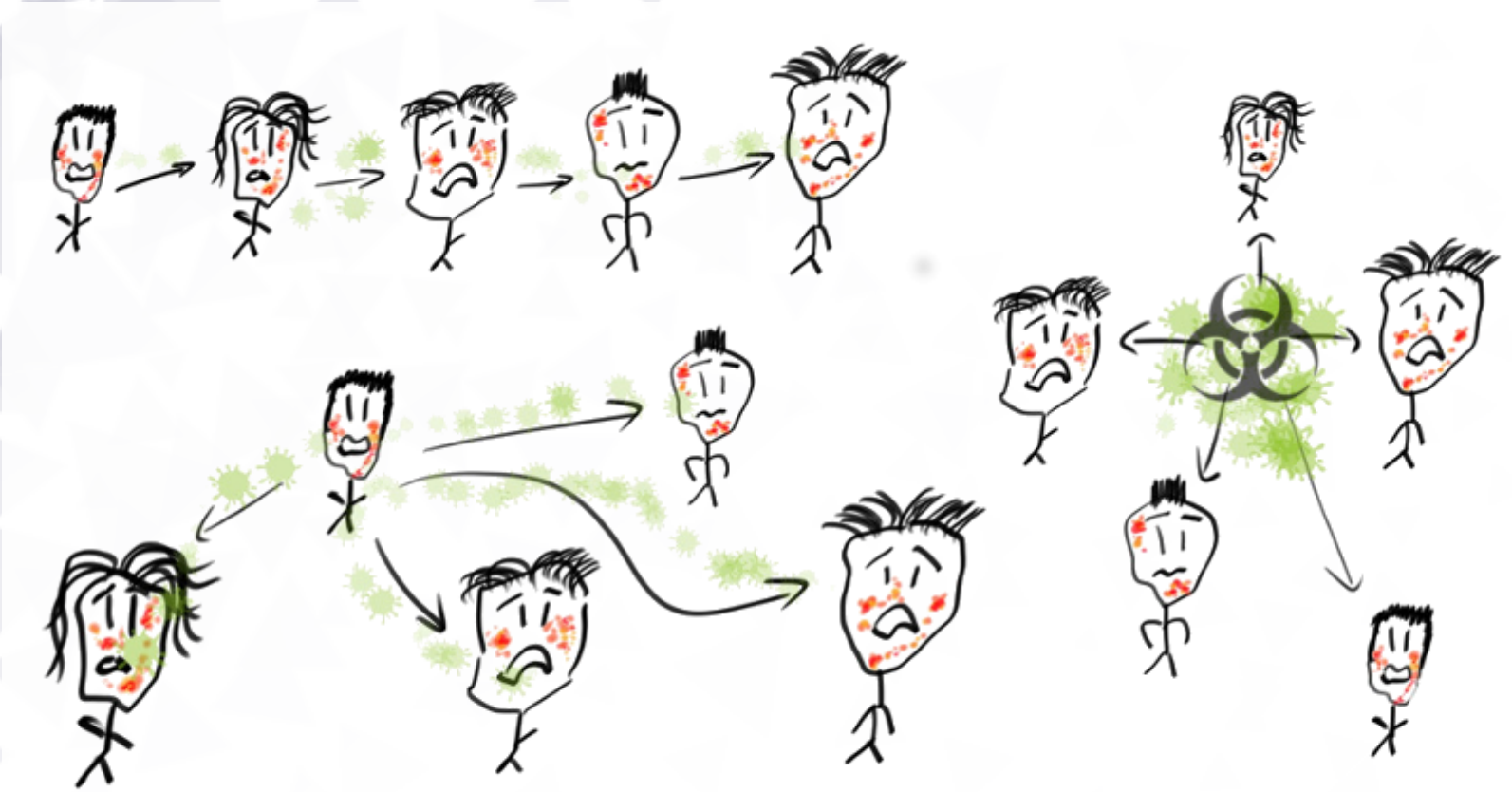
[Campbell et al. (2018) BMC Bioinformatics]

Identifying transmission clusters

Sustained transmission vs repeated introductions



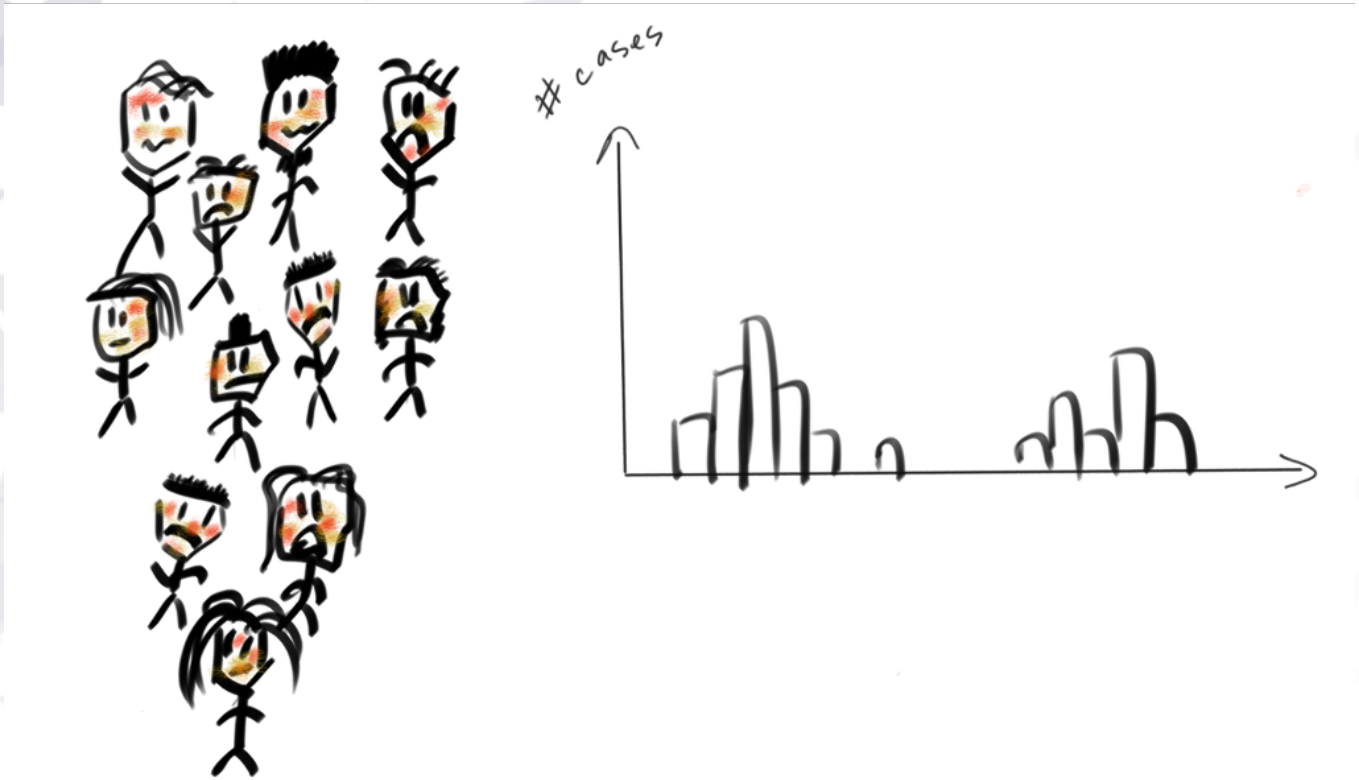
Sustained transmission vs repeated introductions



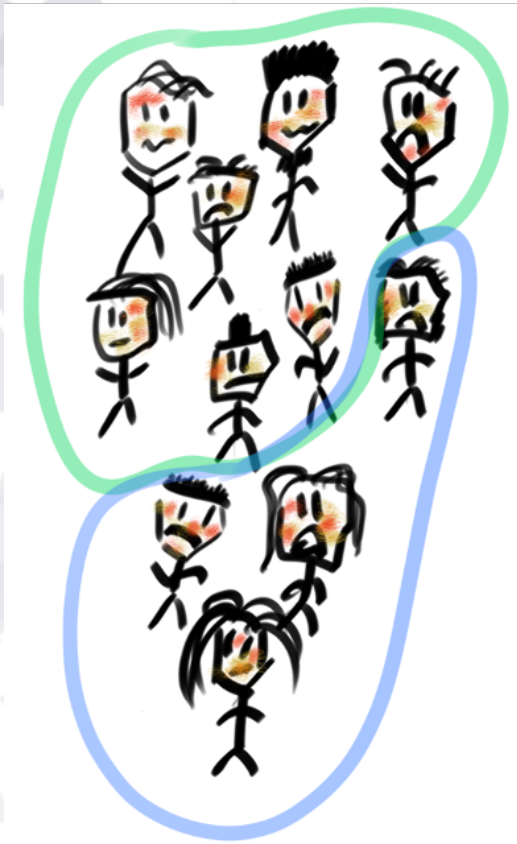
Different transmission patterns call for different interventions

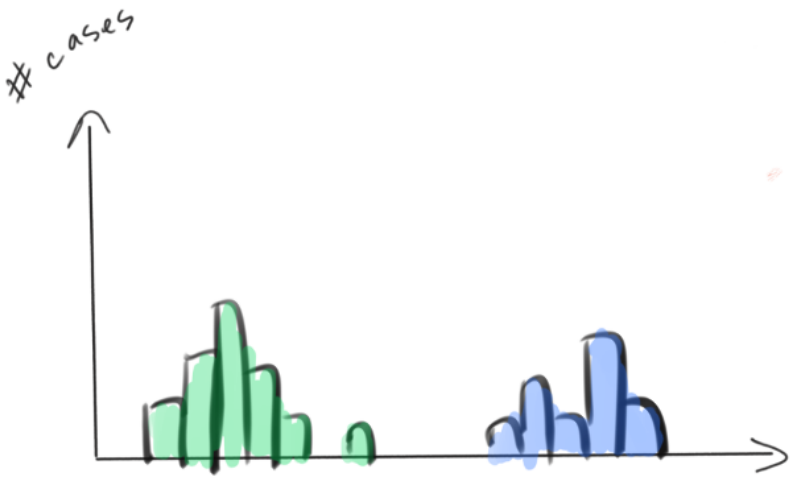
One outbreak.. how many introductions?

One outbreak.. how many introductions?

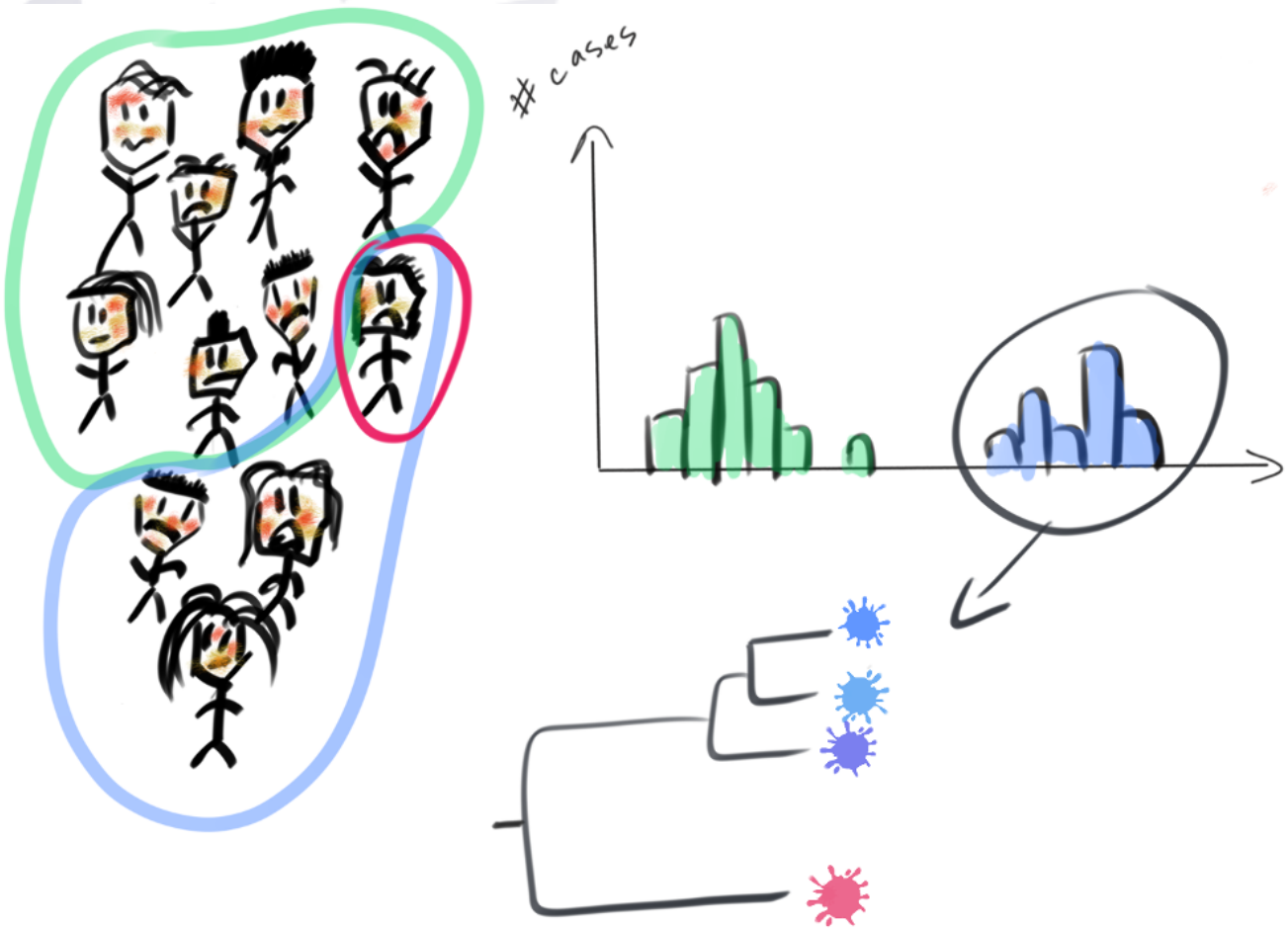


One outbreak.. how many introductions?





One outbreak.. how many introductions?



Combined data sources can detect outbreak clusters

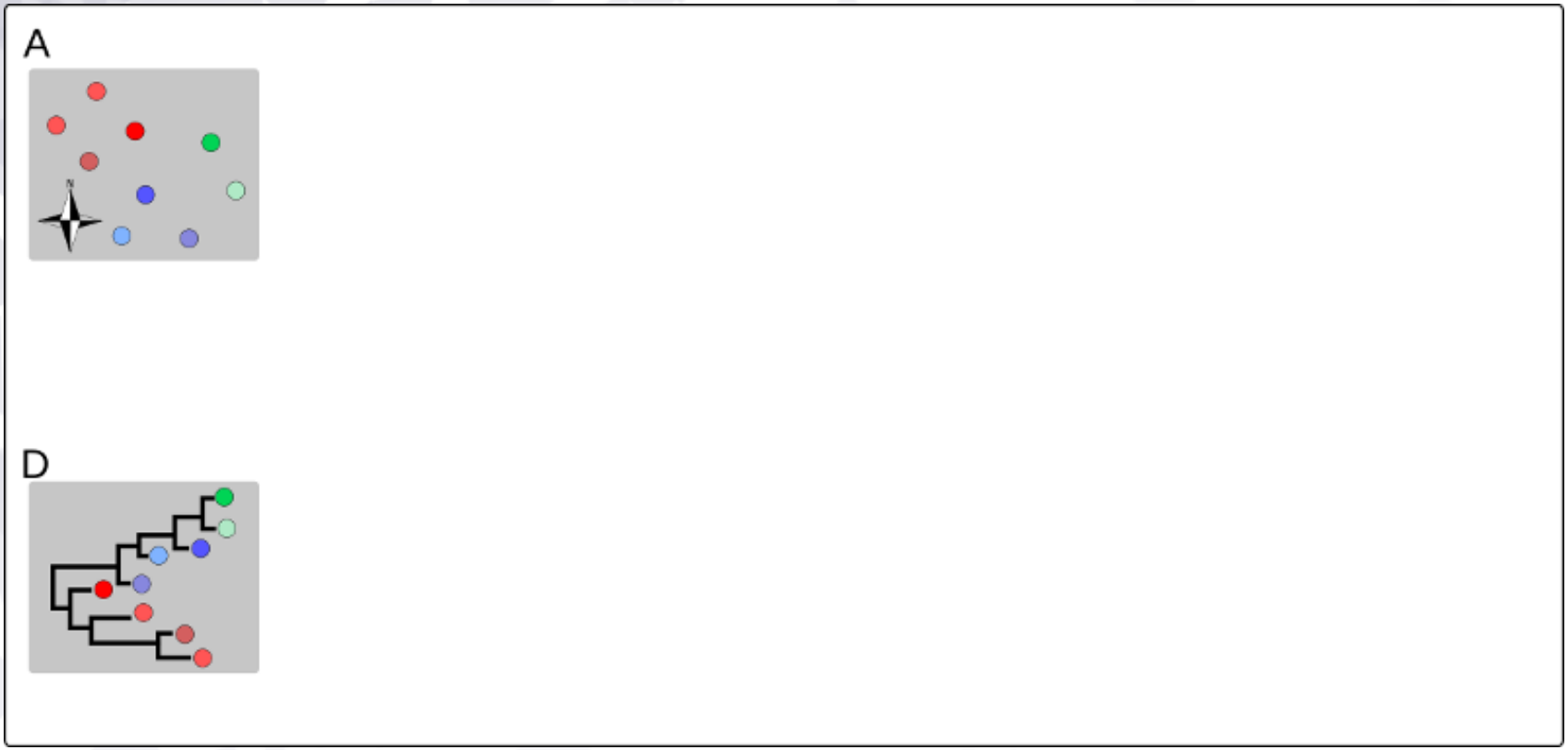
VIMES: VIsualisation and Monitoring of EpidemicS [1]

Aims: develop a new method which...

- detects outbreak clusters: cases stemming from same introduction (same transmission tree)
- integrates different data: temporal, spatial, genetic, etc.
- works fast, scales well: so that it can be used for real-time outbreak detection

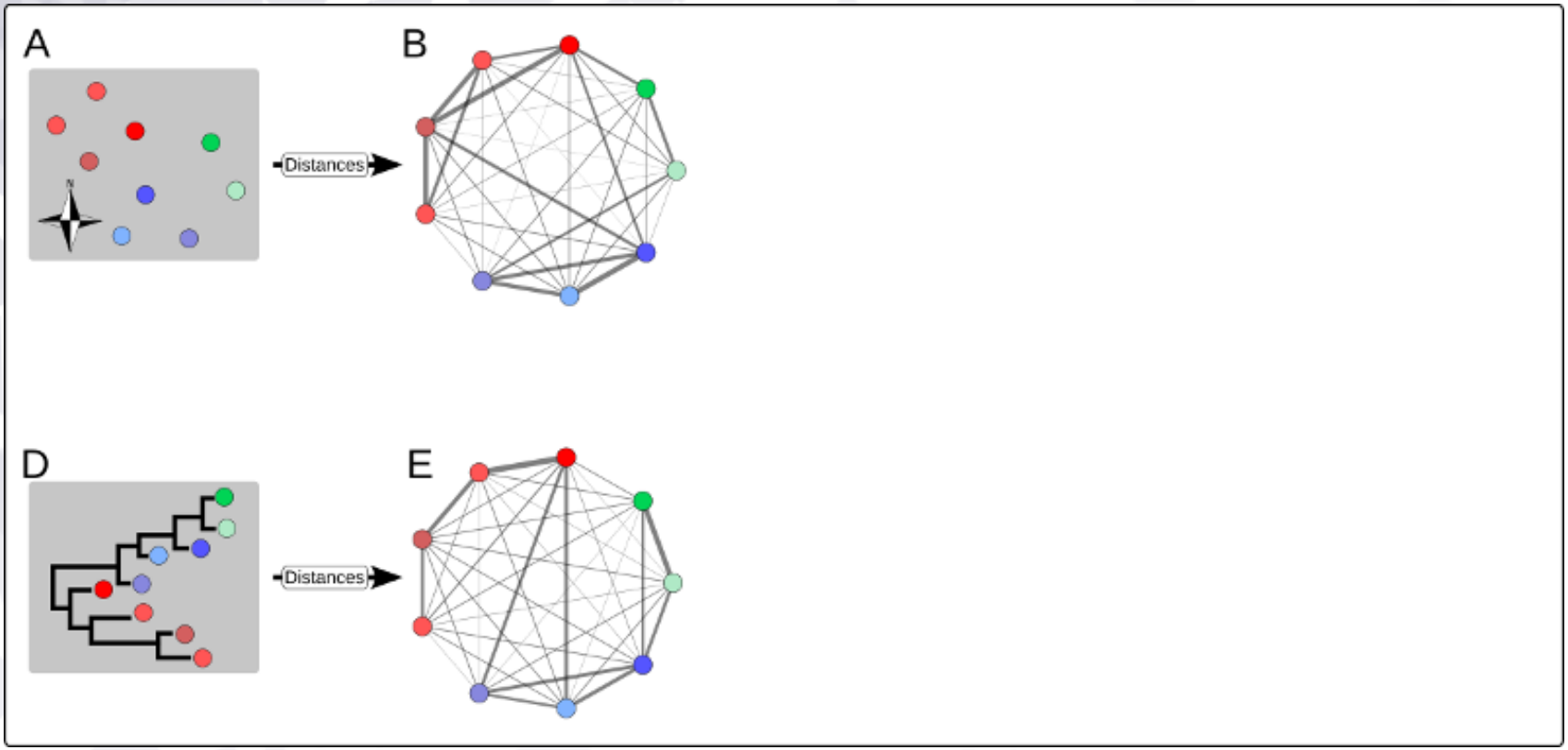
[1] well, really, I made that up because I was reading 'Snuff' by Terry Pratchett at the time; incidentally, Pratchett was a huge fan of using long footnotes in his novels, which are often quite entertaining to read; well, this does not apply here: if you are still reading this, you probably missed what I just said

A graph-based evidence synthesis approach



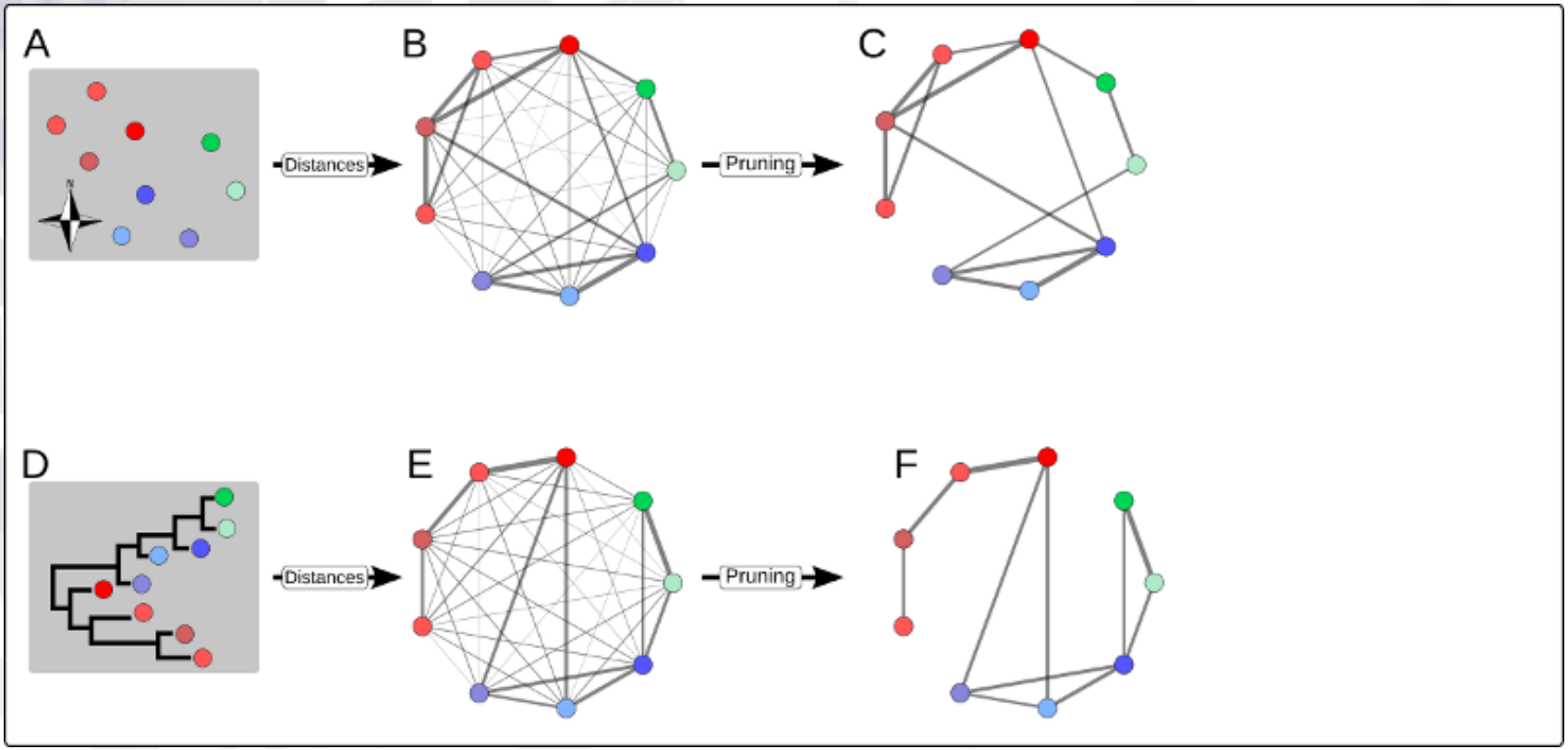
[Cori et al. (2018) PLoS Computational Biology]

A graph-based evidence synthesis approach



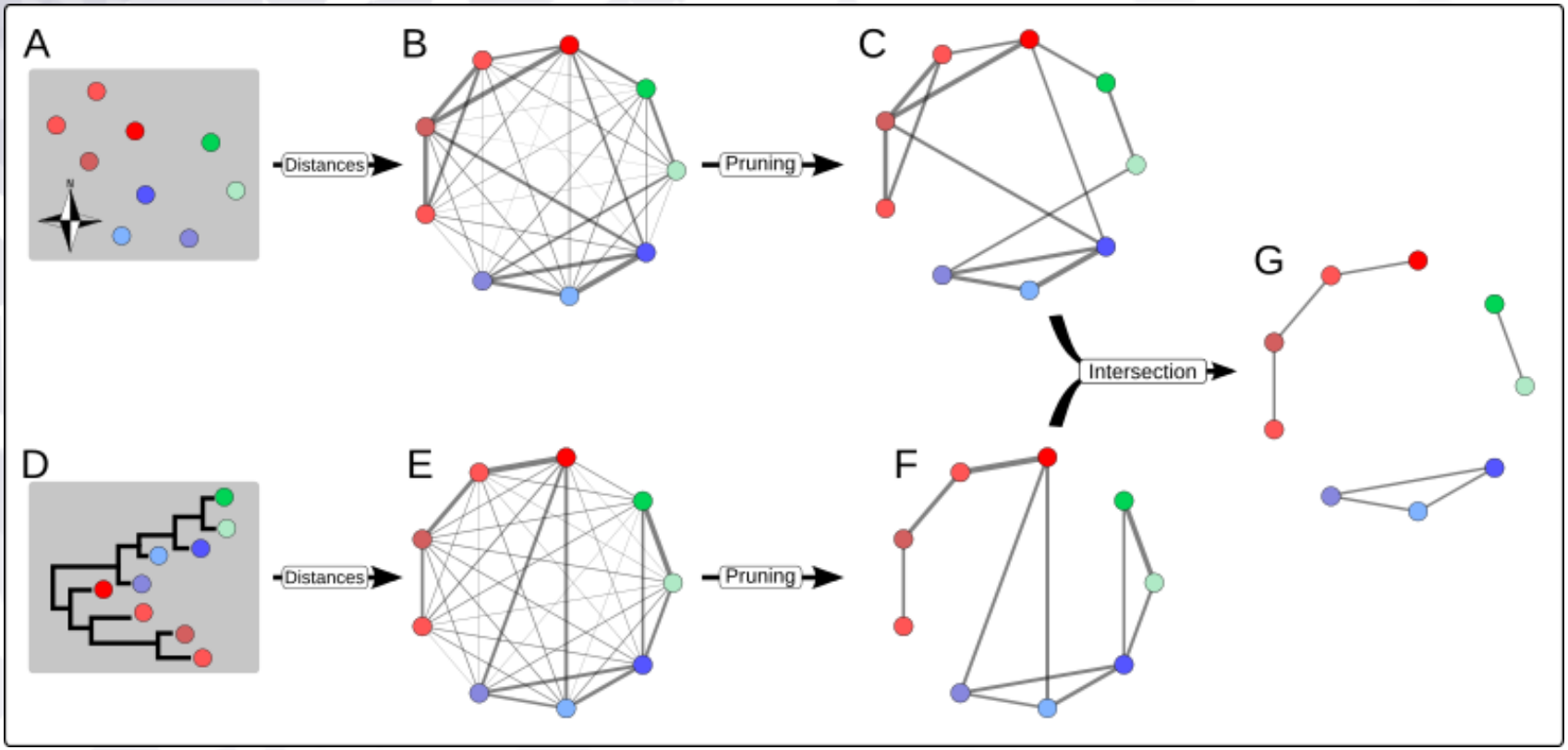
[Cori et al. (2018) PLoS Computational Biology]

A graph-based evidence synthesis approach



[Cori et al. (2018) PLoS Computational Biology]

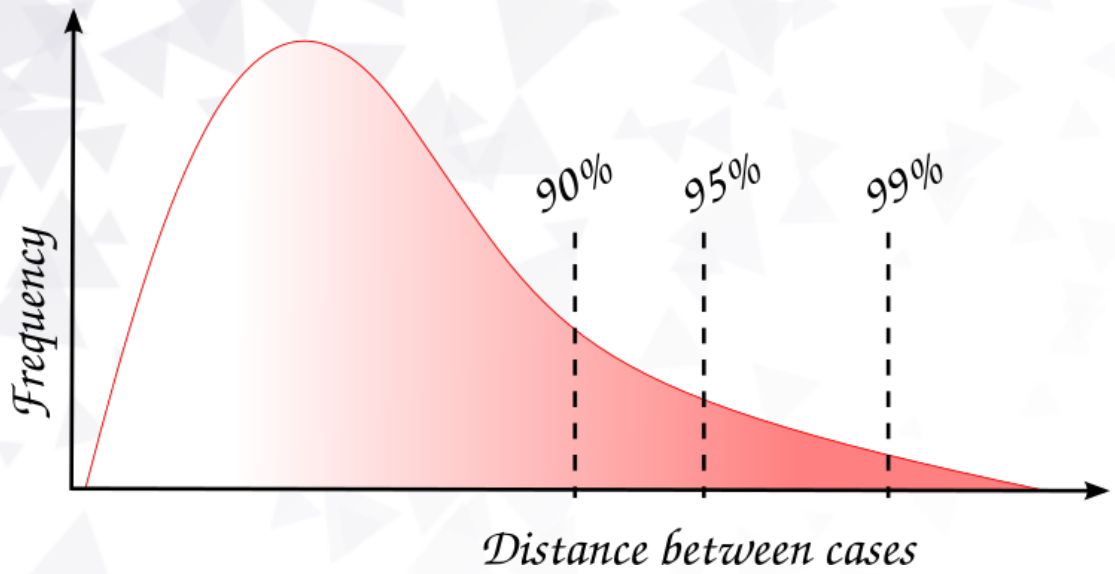
A graph-based evidence synthesis approach



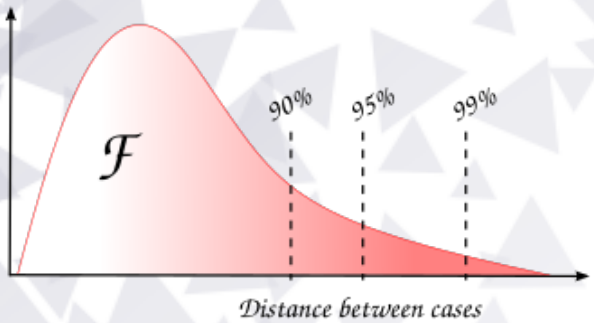
[Cori et al. (2018) PLoS Computational Biology]

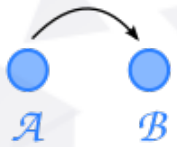
Pruning graphs: where to cut?

Assuming a known expected distribution between pairs of cases (e.g. serial interval, spatial kernel, molecular clock), different quantiles can be used:

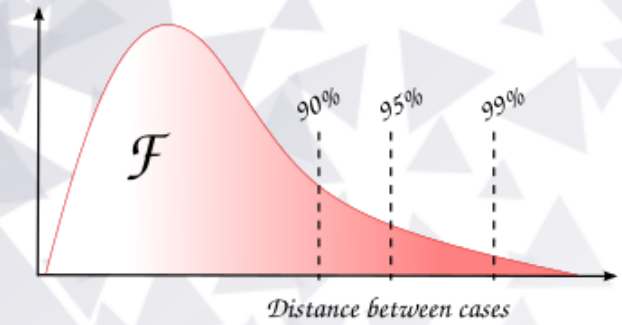


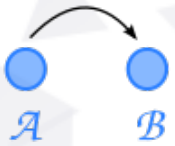
Pruning graphs: where to cut?

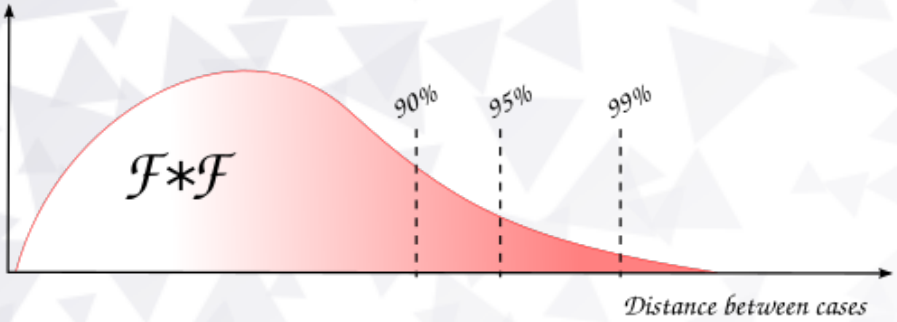




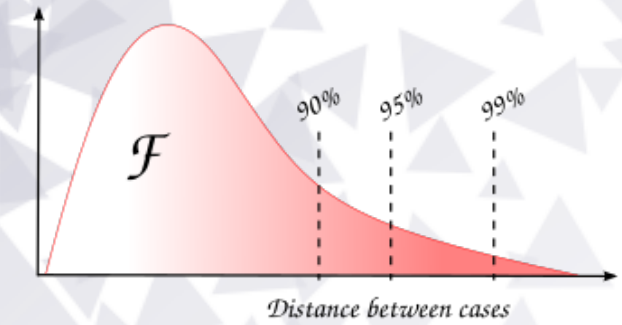
Pruning graphs: where to cut?

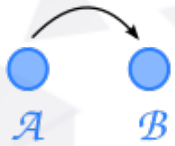


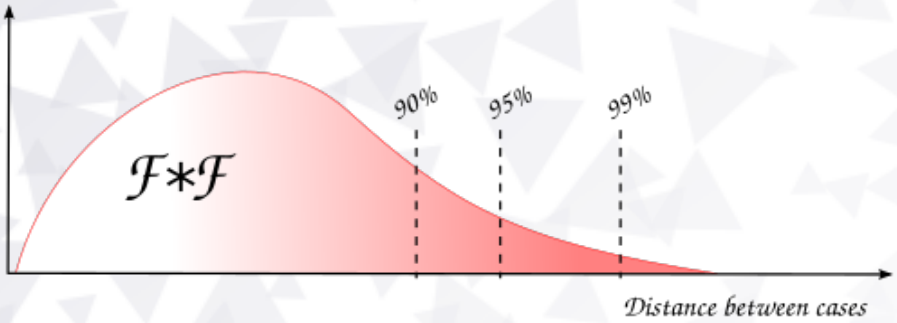


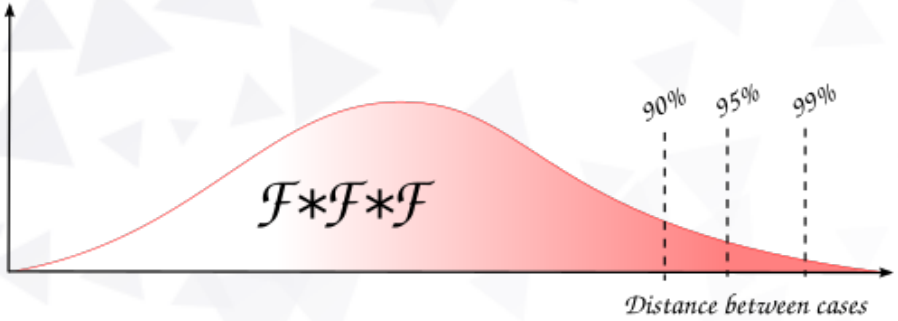


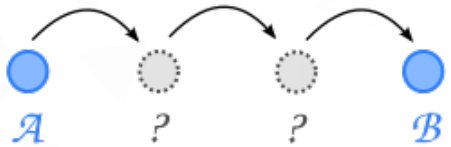
Pruning graphs: where to cut?



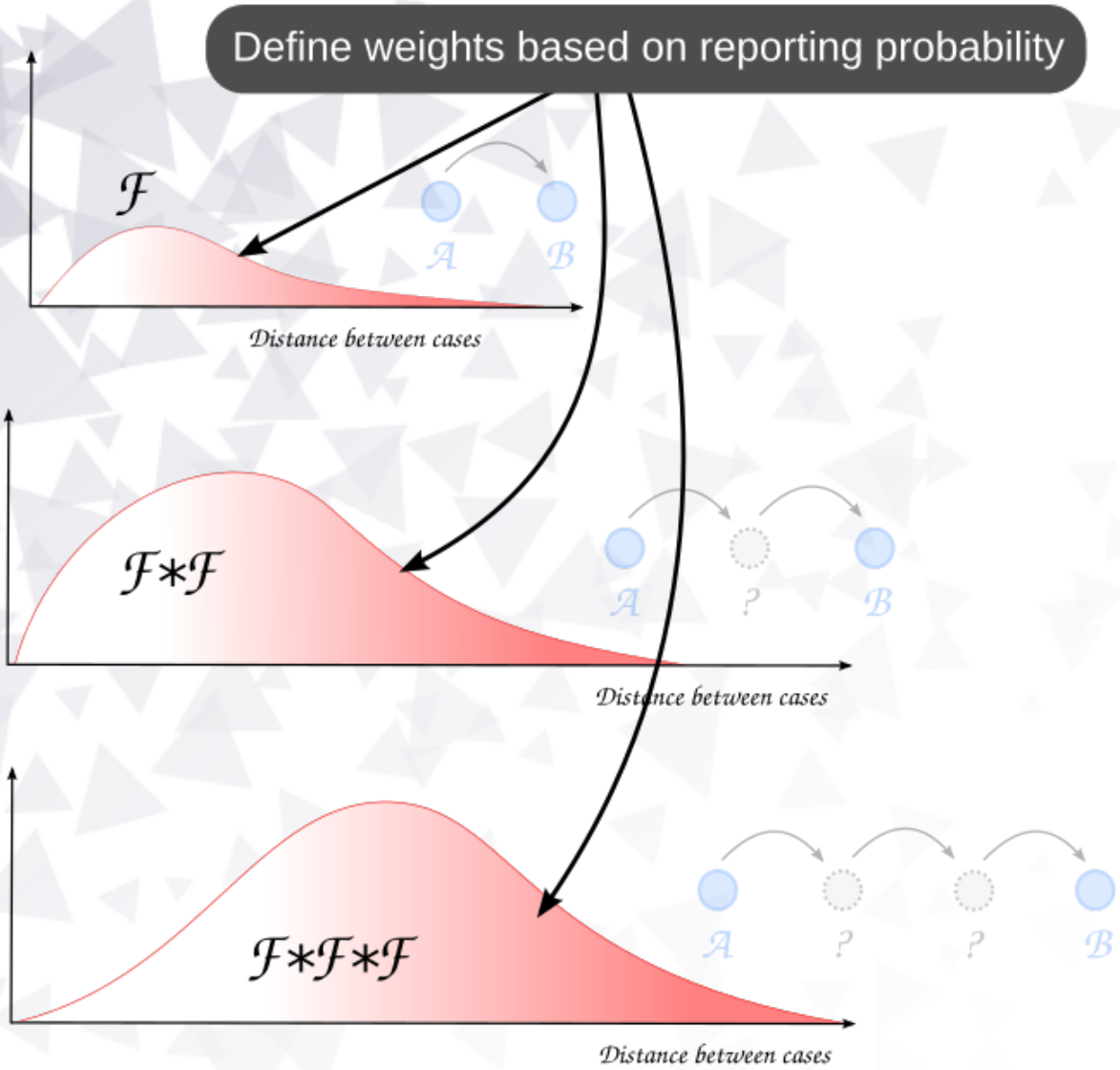




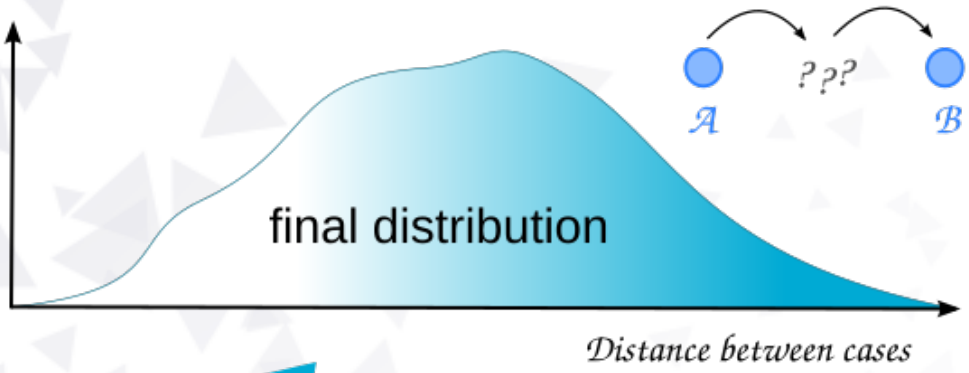


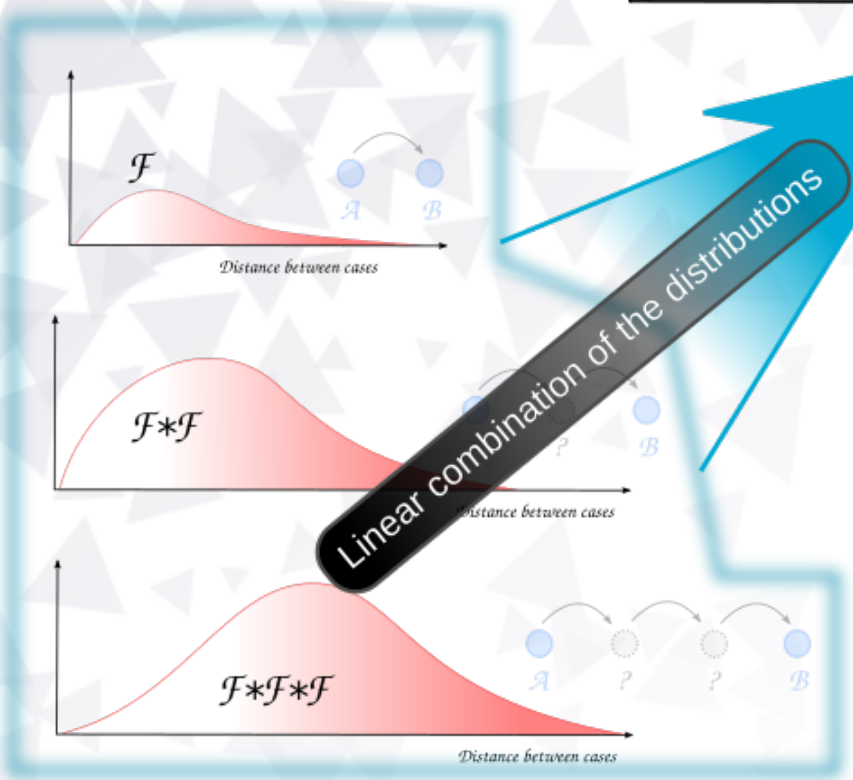


Pruning graphs: where to cut?



Pruning graphs: where to cut?





Linear combination of the distributions

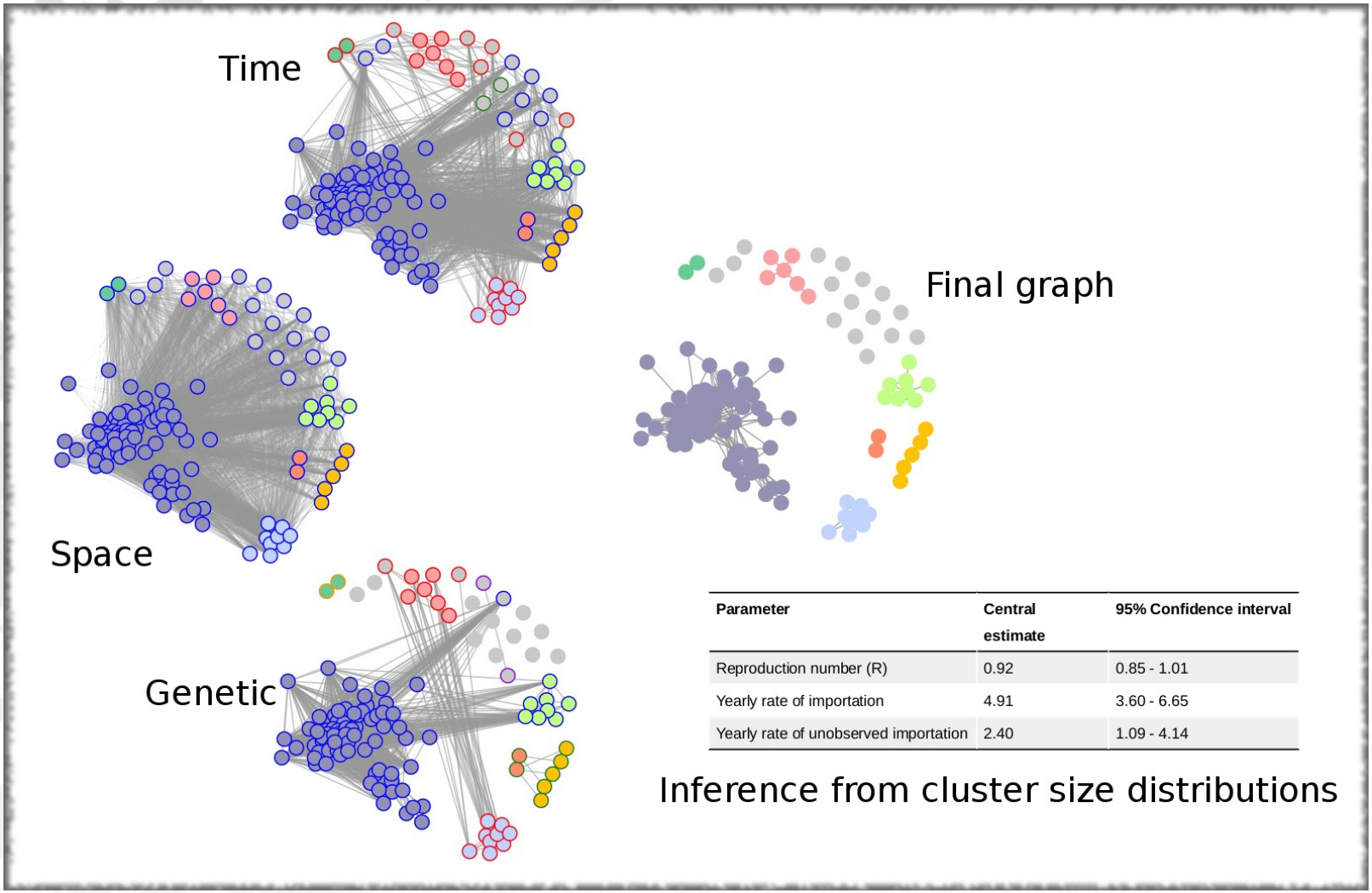
idea by
Pierre Nouvellet

analytical results by
Anne Cori

Application: dog rabies epidemics, Central African Republic

[Cori et al. (2018) PLoS Computational Biology]

Results



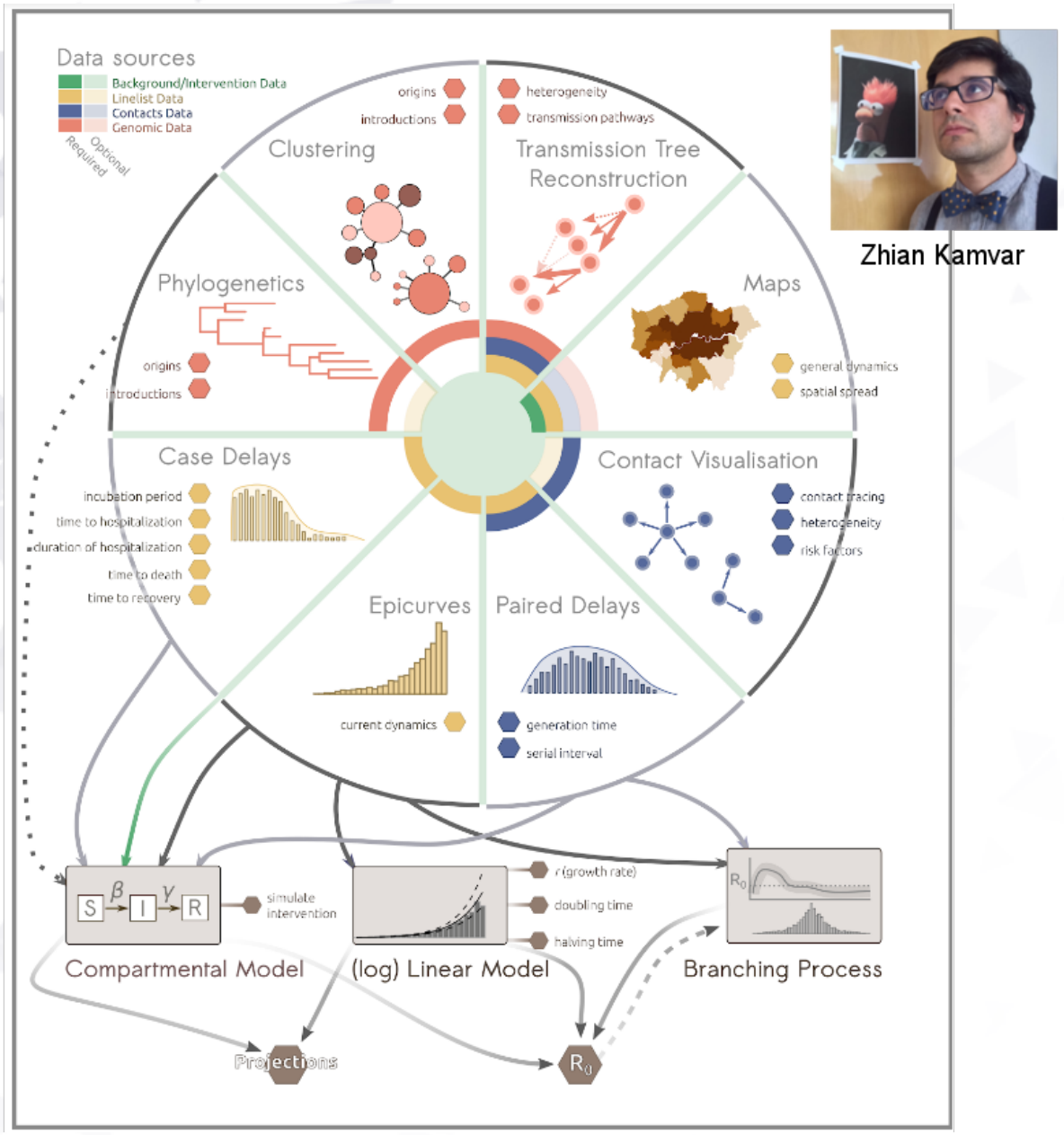
Parameter	Central estimate	95% Confidence interval
Reproduction number (R)	0.92	0.85 - 1.01
Yearly rate of importation	4.91	3.60 - 6.65
Yearly rate of unobserved importation	2.40	1.09 - 4.14

Inference from cluster size distributions

[Cori et al. (2018) PLoS Computational Biology]

When are genomics data useful?

The outbreak analytics toolbox



Zhian Kamvar

Pathogen genetics in outbreak response

- in general, not needed for forecasting or control

Pathogen genetics in outbreak response

- in general, not needed for forecasting or control
- useful to detect multiple introductions or superspreading

Pathogen genetics in outbreak response

- in general, not needed for forecasting or control
- useful to detect multiple introductions or superspreading
- can complement contact tracing data

Pathogen genetics in outbreak response

- in general, not needed for forecasting or control
- useful to detect multiple introductions or superspreading
- can complement contact tracing data
- WGS are costly: is it worth it?

Thanks to:

- **Session:** John Edmunds
- **Collaborators:** Finlay Campbell, Anne Cori, Pierre Nouvellet, Zhian Kamvar, Amrish Baidjoe, Roz Eggo, Tini Garske, Hervé Bourhy, Emmanuel Nakouné, Jimmy Whitworth, Dan Bausch, Bayard Roberts, Neil Ferguson
- **Groups:** R Epidemics Consortium
- **Funding:** GCRF project RECAP (ES/P010873/1), UK PH RST, HPRU-NIHR, MRC

www.repidemicsconsortium.org