

Data analytics in early outbreak response

Academic exercise or operational tool?

Thibaut Jombart

31st October 2018

London School of Hygiene and Tropical Medicine
Imperial College London



Emerging disease, early outbreak response context



- situational awareness urgently needed
- limited data available
- questions focus on delays, risk factors, transmissibility
- reproducibility and reliability » refinement and complexity

Emerging disease, early outbreak response context

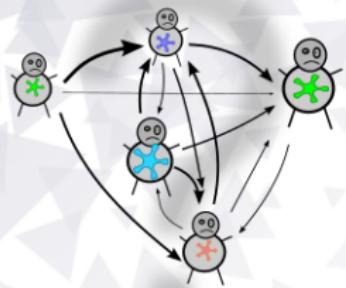


- situational awareness urgently needed
- limited data available
- questions focus on delays, risk factors, transmissibility
- reproducibility and reliability » refinement and complexity

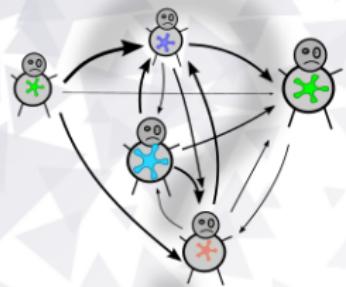
How can data analytics / modelling help?

Reconstructing transmission trees

Using genomics to infer who infects whom?



Using genomics to infer who infects whom?

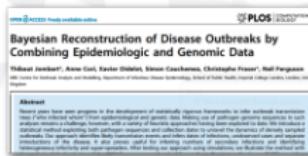
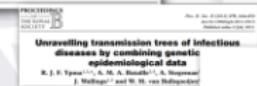
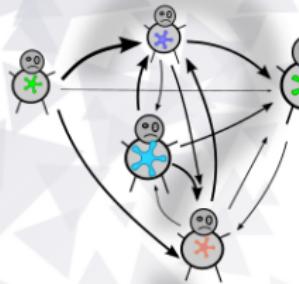


Unravelling transmission trees of infectious diseases by combining genetic and epidemiological data

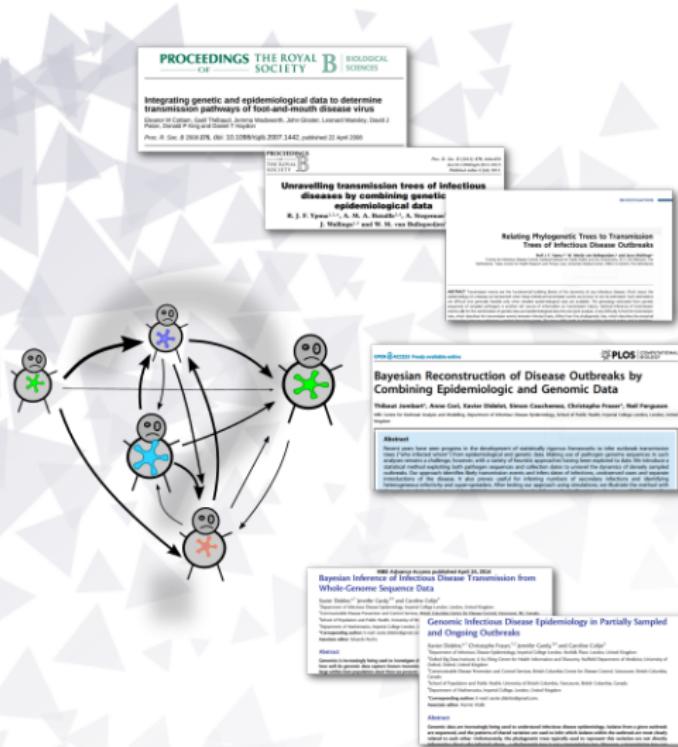
R. J. E. Verwoerd, A. M. A. Bouwink¹, J. A. Nijhuis², J. H. Hulst¹ & W. H. van den Heuvel¹



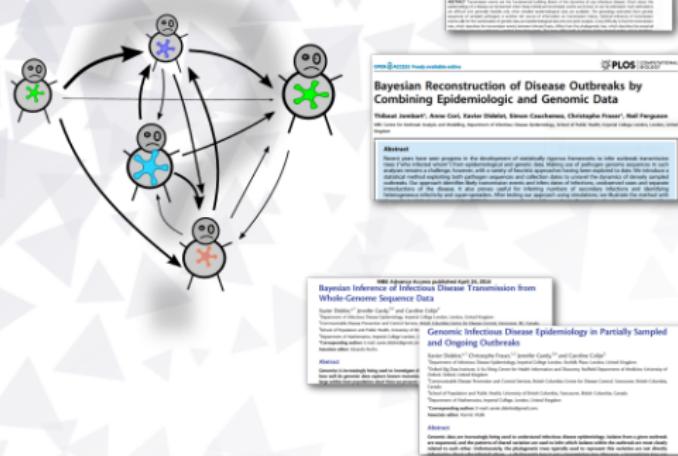
Using genomics to infer who infects whom?



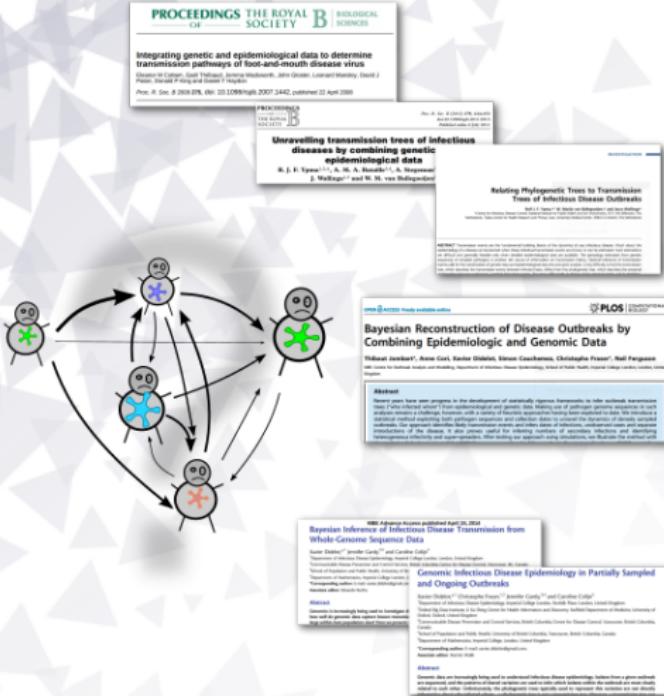
Using genomics to infer who infects whom?



Using genomics to infer who infects whom?



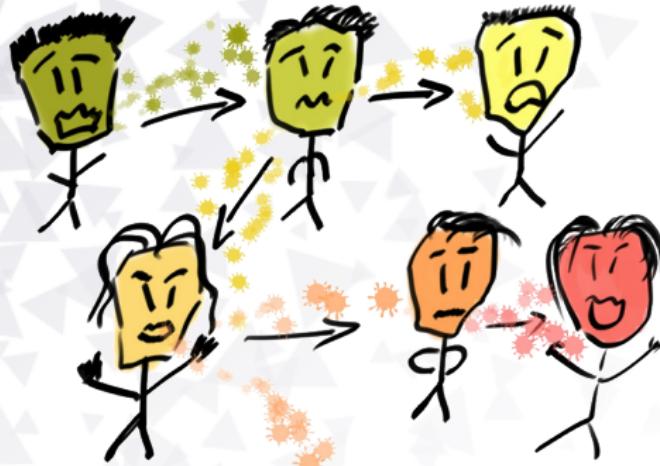
Using genomics to infer who infects whom?



Methods heavily
rely on whole genome
sequence data

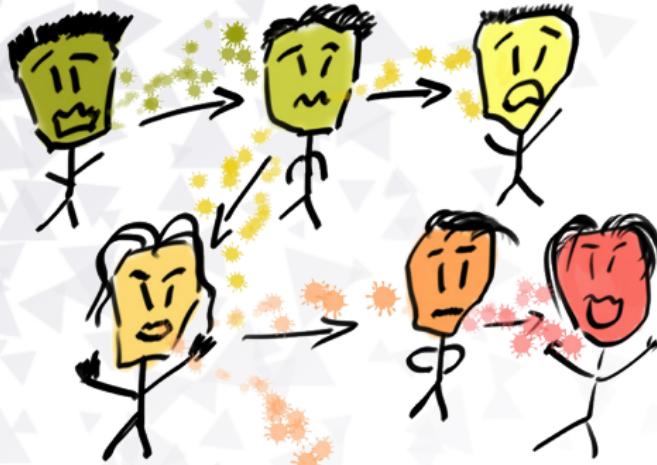
3

Using WGS to infer who infected whom



Mutations accumulate along transmission chains.

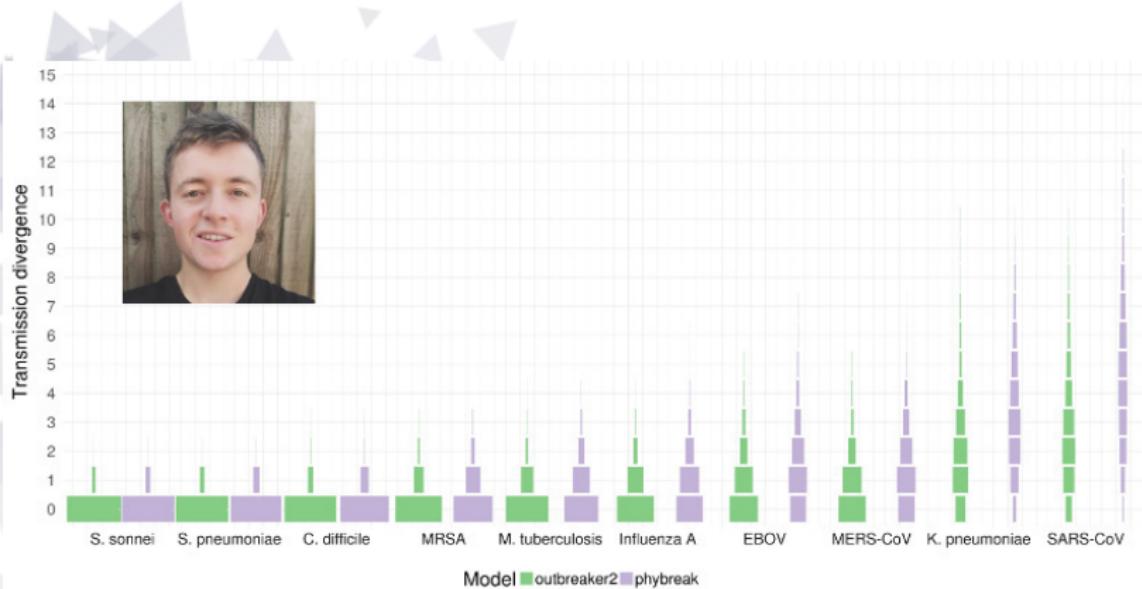
Using WGS to infer who infected whom



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

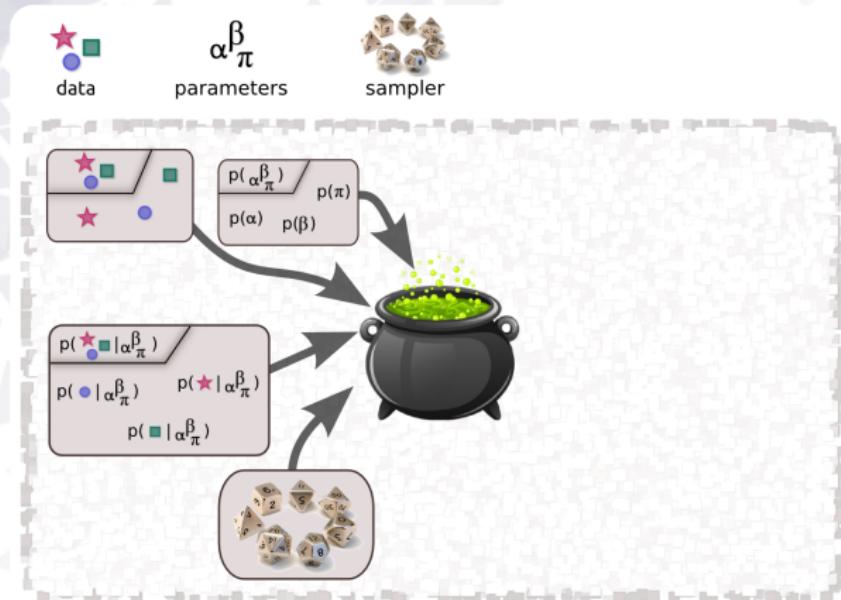
Modularity: customise data, prior, likelihood, MCMC.



[Campbell et al. (2018) BMC Bioinformatics]

outbreaker2: evidence synthesis framework for outbreak reconstruction

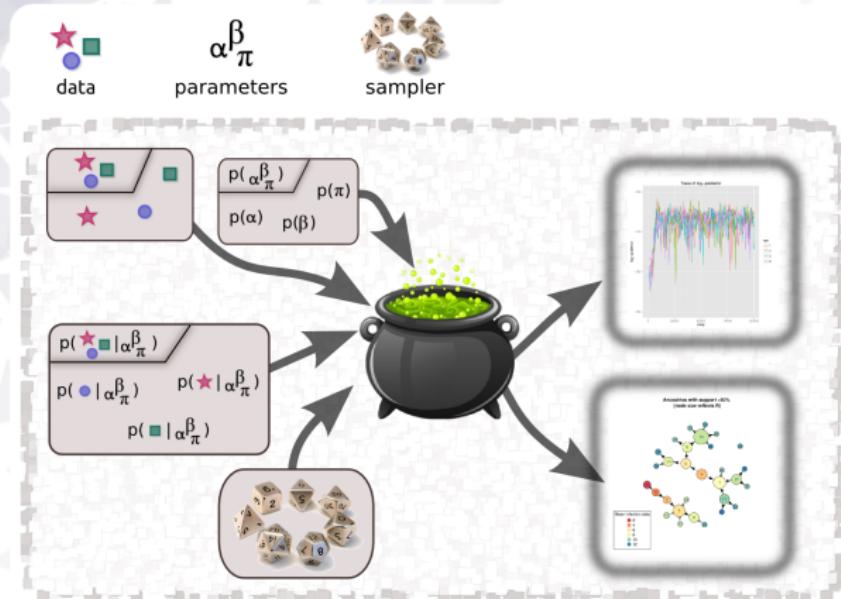
Modularity: customise data, prior, likelihood, MCMC.



[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* \times *incubation* \times *genetic (simple)* \times *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* \times *incubation* \times *genetic (simple)* \times *missing cases*

TransPhylo likelihood

- $p(G | \beta, \gamma, N_{eg}, \alpha) = p(G | N_{eg}, \alpha) \times p(\alpha | \beta, \gamma)$
- i.e. *phylogeny (coalescent)* \times *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* \times *incubation* \times *genetic (simple)* \times *missing cases*

TransPhylo likelihood

- $p(G | \beta, \gamma, N_{eg}, \alpha) = p(G | N_{eg}, \alpha) \times p(\alpha | \beta, \gamma)$
- i.e. *phylogeny (coalescent)* \times *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 `probPTreeGivenTree`. 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 = chind(param$t.inf, data$dates, param$alpha),
               nam = data$pstree$nam)
  ttrees$tree(which(is.na(ttrees$ttree[, 3])), 3) <- 0
  txt <- capture.output(ttree <- combine(ttrees, data$pstree))
  if (length(txt)==0) {
    prob <- probPTreeGivenTree(ttree, neg = 366 * 0.26)
  } else {
    prob <- -Inf
  }
  return(prob)
}

lik_transphylo <- function(date, param, i = NULL, custom_functions = NULL)
{
  ## function (date, param, i = NULL, custom_functions = NULL)

  new_move_tinf <- function(param, data, list_custom_ll = new_model) {
    for (i in 1:date$N) {
      current_ll <- api$cpp_ll_all(data, param, i = NULL, list_custom_ll)
      modif <- sample(c(-100:-1, 1:100), 1)
      param$ll.inf[i] <- param$ll.inf[i] + modif
      new_ll <- api$cpp_ll_all(data, param, i = NULL, list_custom_ll)
      if (log10(modif[i]) > log10(ll.current_ll)) {
        param$ll.inf[i] <- param$ll.inf[i] + modif
      }
    }
    return(param)
  }

  new_moves <- custom_moves(t.inf = new_move_tinf)
  new_moves

  ##
  ## //////////////////////////////////////////////////////////////////
  ## class: outbreaker_moves list
  ## number of items: 8
  ## //////////////////////////////////////////////////////////////////
  ## movement functions //
  ## See
}
```

[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 `probPTreeGivenTree`. 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 = chind(param$t.inf, data$dates, param$alpha),
               nam = data$pstree$nam)
  ttrees$tree(which(is.na(ttrees$ttree[,3]))[,3] <- 0)
  txt <- capture.output(ttree <- combine(ttrees,data$pstree))
  if (length(txt)==0) {
    prob <- probPTreeGivenTree(ttree, neg = 366 * 0.26)
  } else {
    prob <- -Inf
  }
  return(prob)
}

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

new_move_tinf <- function(param, data, list_custom_ll = new_model) {
  for (i in 1:date$N) {
    current_ll <- api$pp_ll_all(data,param, i = NULL, list_custom_ll)
    modif <- sample(c(-100:-1,1:100), 1)
    param$ll.inf[i] <- param$ll.inf[i] + modif
    new_ll <- api$pp_ll_all(data,param, i = NULL, list_custom_ll)
    if (log10(modif[i]) > log10(ll - current_ll)) {
      param$ll.inf[i] <- param$ll.inf[i] + modif
    }
  }
  return(param)
}

new_moves <- custom_moves(t.inf = new_move_tinf)
new_moves

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

likelihood

[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 `probPTreeGivenTree`. 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 = chind(param$t.inf, data$dates, param$alpha),
               nam = data$ptree$nam)
  ttree$tree$which(is.na(ttree$tree[,3]))[,3] <- 0
  txt <- capture.output(ctree <- combine(ttree, data$ptree))
  if (length(txt)==0) {
    prob <- probPTreeGivenTree(ctree, neg = 366 * 0.26)
  } else {
    prob <- -Inf
  }
  return(prob)
}
```

likelihood

movement function

```
args(api$cpp_ll_all)
## function (date, param, i = NULL, custom_functions = NULL)
## NULL

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

new_moves <- custom_moves(t_inf = new_move_tinf)
new_moves

##
## //////////////////////////////////////////////////////////////////
## class: outbreaker_moves list
## number of items: 8
## //////////////////////////////////////////////////////////////////
## movement functions //
## See
```

[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 `probPTreeGivenTree`. 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 = chind(param$t.inf, data$dates, param$alpha),
               nam = data$ptree$nam)
  ttrees$tree(which(is.na(ttrees$ttree[, 3])), 3) <- 0
  txt <- capture.output(ctree <- combine(ttrees, data$ptree))
  if (length(txt) == 0) {
    prob <- probPTreeGivenTree(ctree, neg = 366 * 0.26)
  } else {
    prob <- -Inf
  }
  return(prob)
}
```

likelihood

Total: 25 lines of R

outbreaker2: 7,500 lines of R/C++

Code difference: 0.3%

movement function

```
args(api$cpp_ll_all)
## function (date, param, i = NULL, custom_functions = NULL)
## NULL

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

new_moves <- custom_moves(t_inf = new_move_tinf)
new_moves

## 
## ///////////////////////////////////////////////////////////////////
## class: outbreaker_moves list
## number of items: 8
## // movement functions //
## @na
```

[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 as a list in *outbreaker2* and phylogenetic tree are encoded as a list in *TransPhylo*. Step (ii) we have to write a simple function to combine these two trees. This is especially interesting if this function is integrated into *outbreaker2*. Step (iii) is the most difficult part in *outbreaker2* and it only need to call the appropriate function in *TransPhylo*. In *outbreaker2*, there are many messages that can be received via (B) messages can arise indicating that the transmission tree and phylogenetic tree are in fact in. In this case the likelihood is returned as -Inf.

```
lik_TransPhylo <- function(data, param) {
  ttree <- list(ttree = chid(param$inf, data$data, param$alpha),
               ann = data$ptree$ann)
  ttrees@treenode[[which(is.na(ttree@ttree[,3]))]] <- 0
  ttrees <- capture.output(ttree <- combine(ttree, data$ptree))
  if (length(ttree)==0) {
    prob <- 0.0001
  } else {
    prob <- 1
  }
  return(prob)
}
```

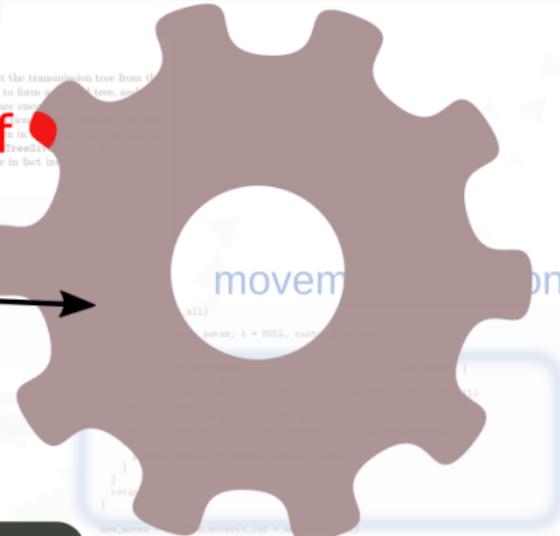
likelihood

Total: 25 lines of R

outbreaker2: 7,500 lines of R/C++

Code difference: 0.3%

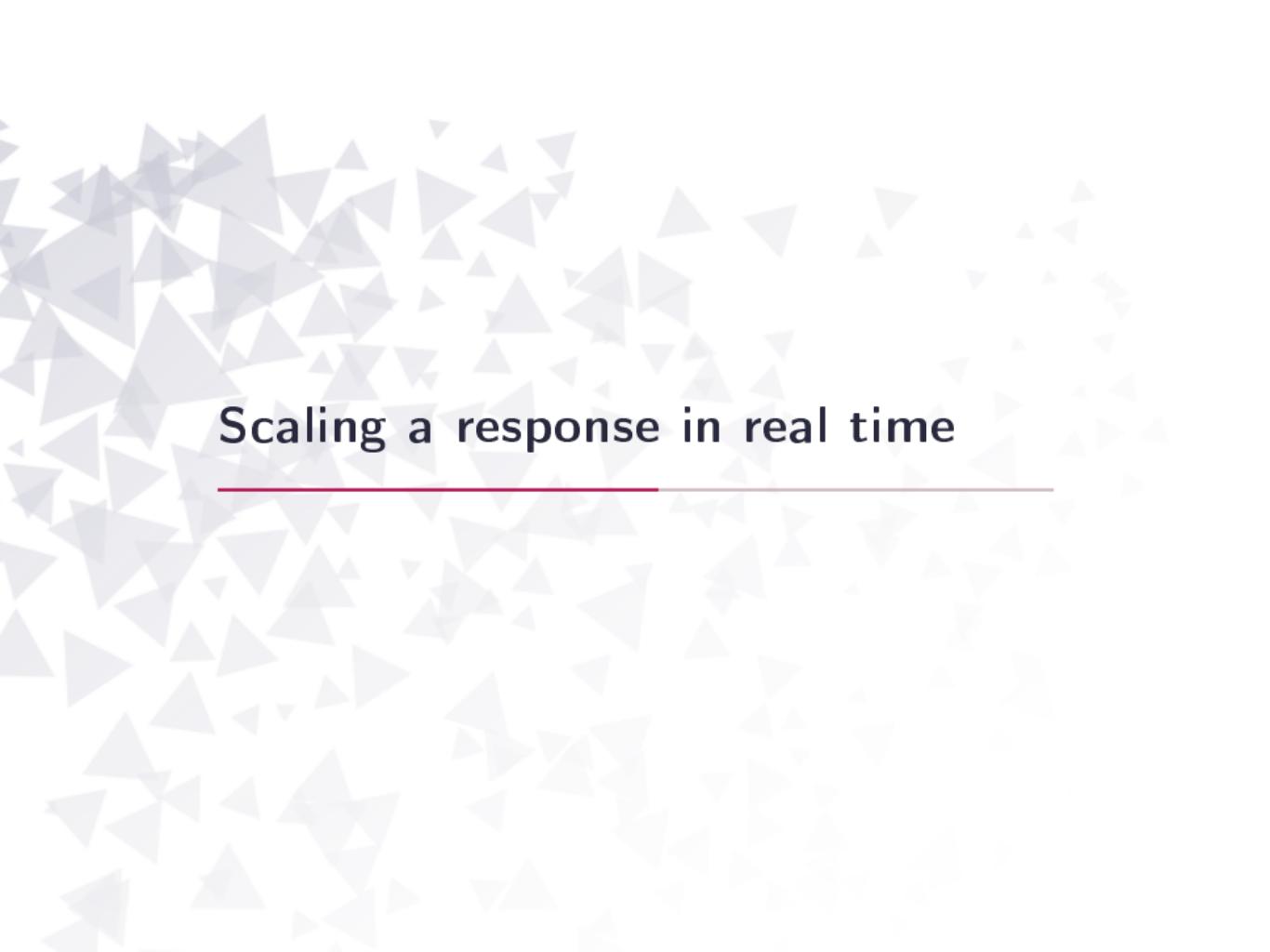
[Campbell et al. (2018) BMC Bioinformatics]



Who infects whom: when do we care?



- complex methods, WGS data costly: **is it worth it?**
- in general, not useful for **forecasting**
- useful to detect **multiple introductions** or **superspreading**
- complement **exposure / contact tracing** data

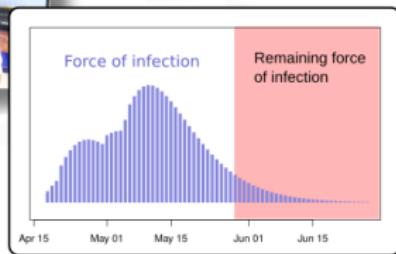


Scaling a response in real time

Ebola outbreak, Likati (DRC) 2017

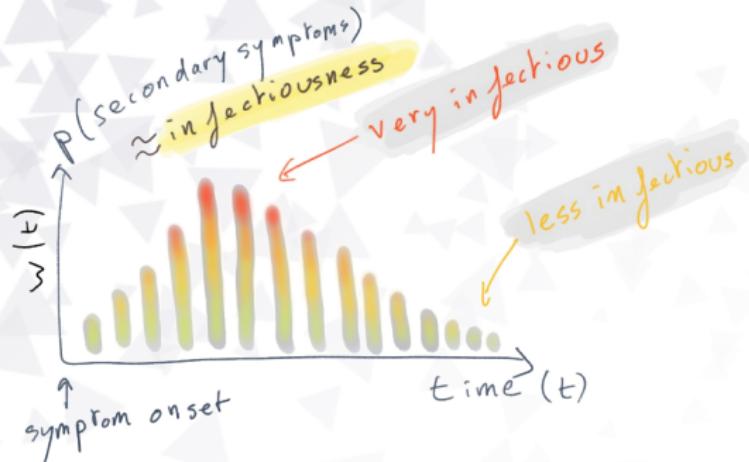


- EVD outbreak May 2017
- contact data visualisation tools used in contact tracing
- simple model informed response (scaling)
- end: 2nd July 2017



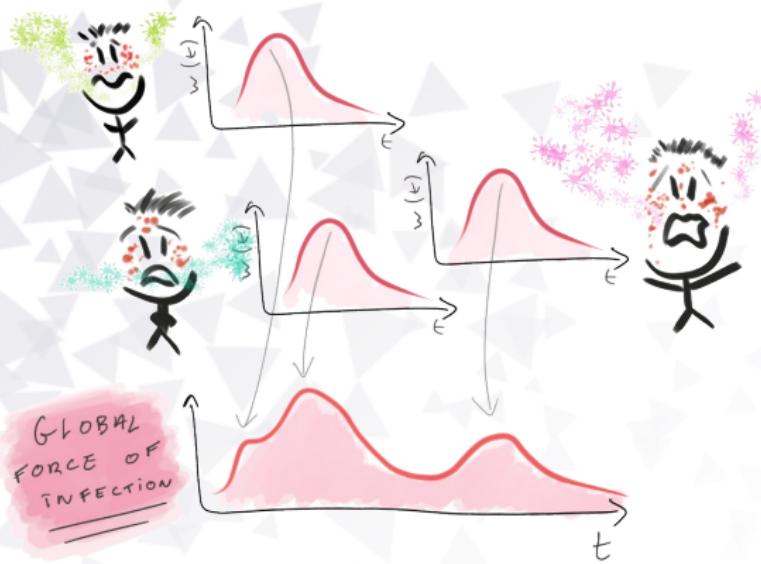
Individual infectiousness over time

Serial interval: delay between symptom onset in infector and infectees



Indicates when we expect new cases, if there are any.

A “simple” branching process model



$$y_t \sim \mathcal{P}(\lambda_t) \quad ; \quad \lambda_t = R_0 \times \sum_i w(t - t_i)$$

y_t : incidence at time t ; $\mathcal{P}()$: Poisson distribution; λ_t : **global force of infection**; $w()$: serial interval distribution; t_i : date of symptom onset

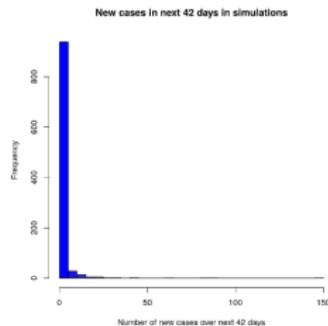
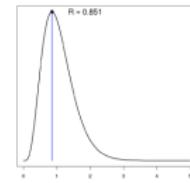
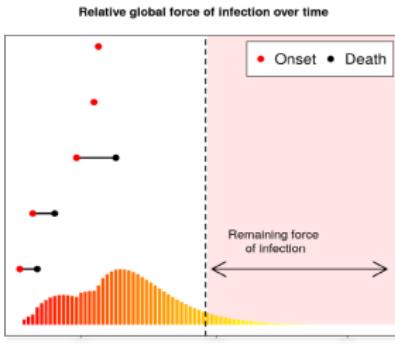
A model for short-term forecasting



1. estimate R from incidence y_1, \dots, y_t until time t
2. simulate incidence $y_{t+1} \sim \mathcal{P}(\lambda_{t+1})$
3. increase t by one day, repeat

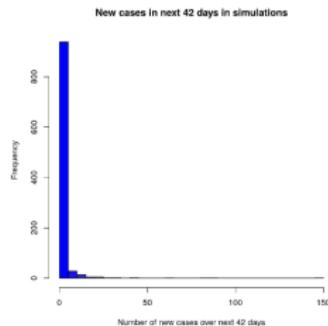
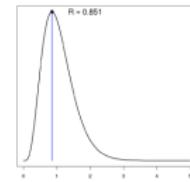
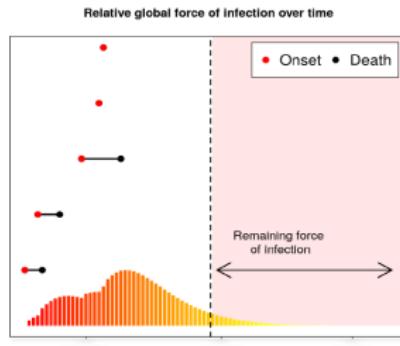
Scaling the response in real-time

Estimating remaining force of infection,
transmissibility (R), predicting new cases



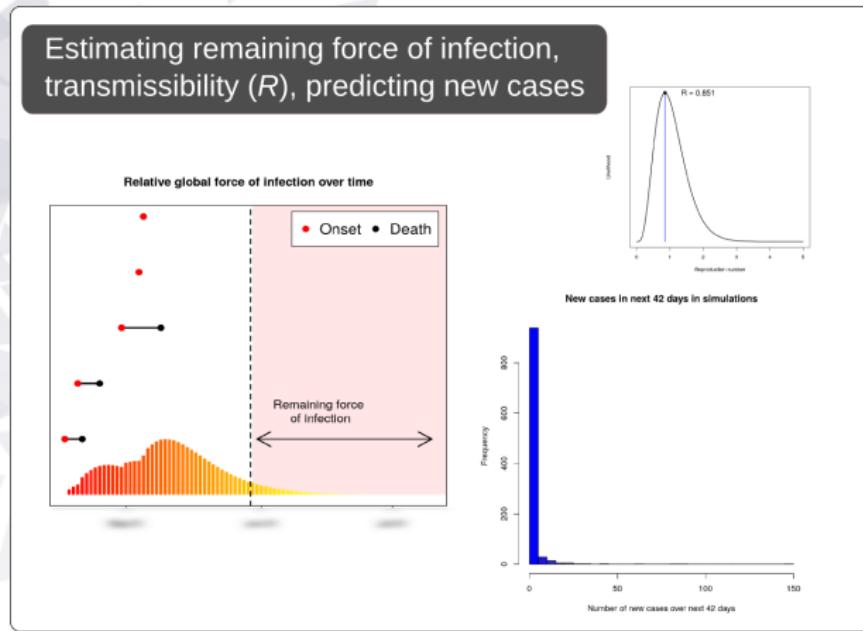
Scaling the response in real-time

Estimating remaining force of infection,
transmissibility (R), predicting new cases



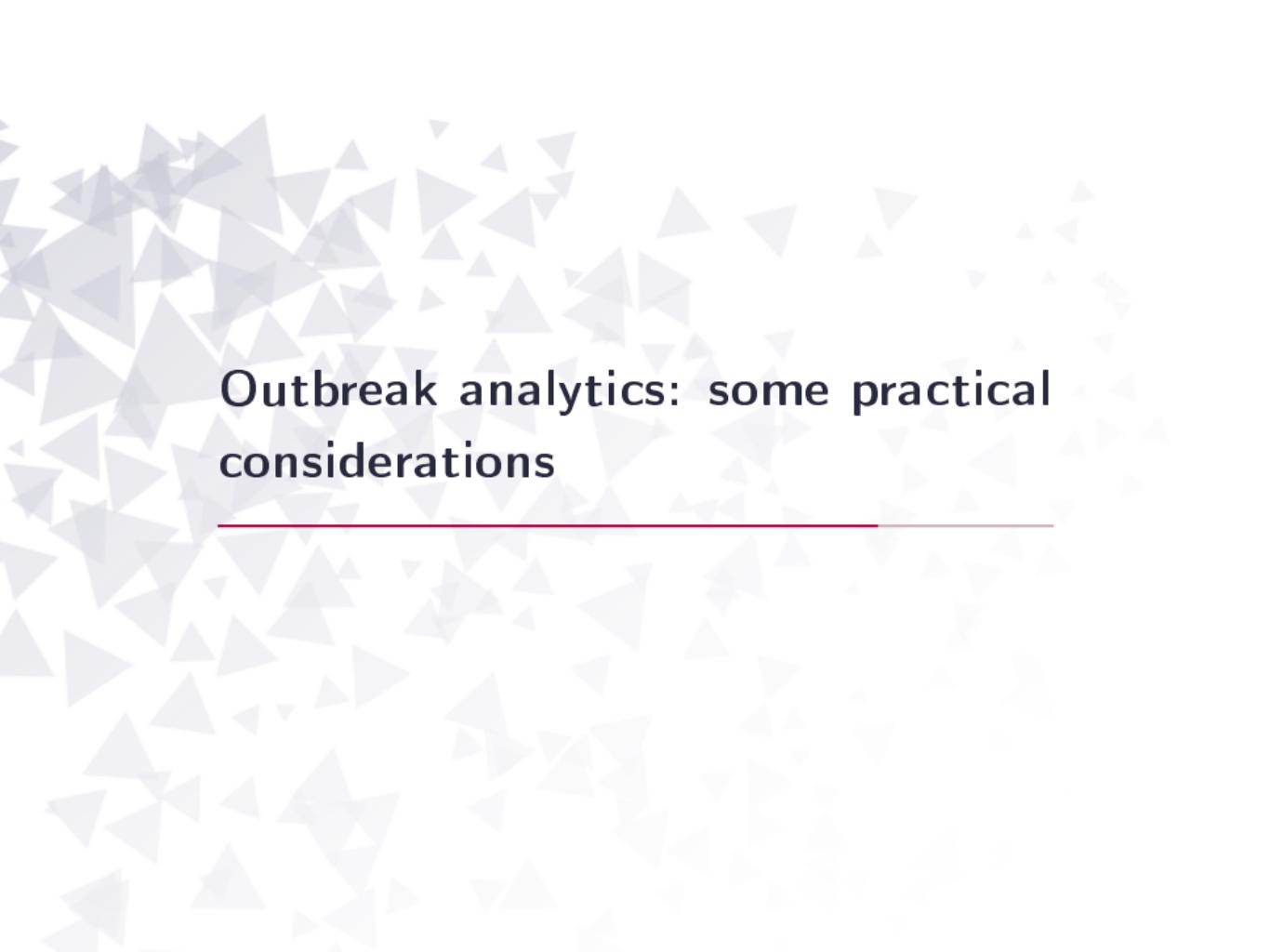
Despite uncertainty in R_0 , new cases were unlikely.

Scaling the response in real-time



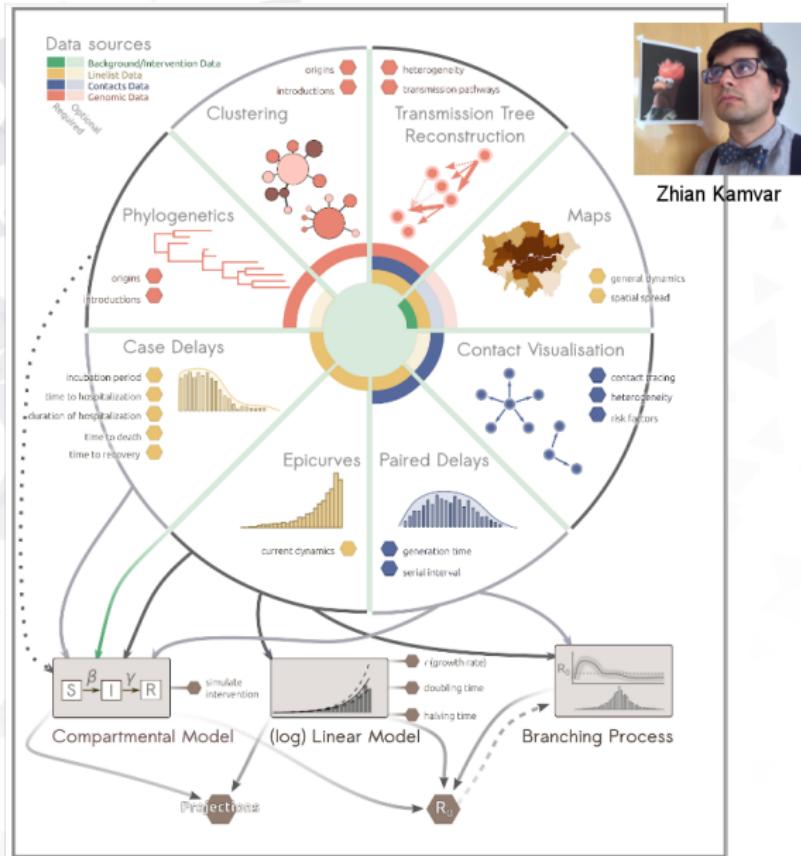
Despite uncertainty in R_0 , new cases were unlikely.

Discouraged scaling up in resource-limited context.



Outbreak analytics: some practical considerations

Cost-effective analyses: data needs vs actionable intel



Centralised analyses, distributed delays



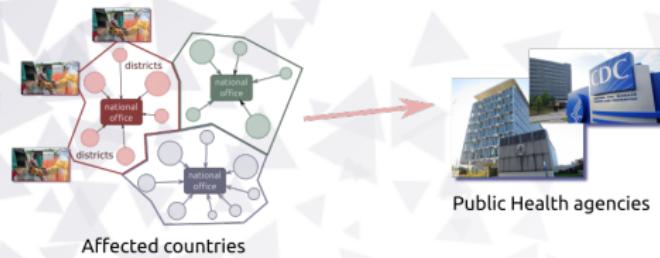
Centralised analyses, distributed delays



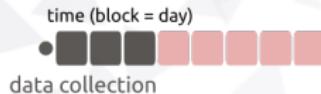
Centralised analyses, distributed delays



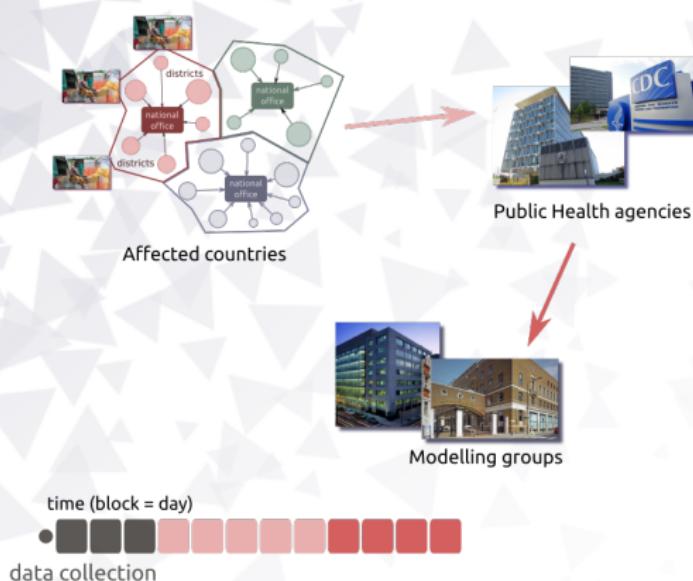
Centralised analyses, distributed delays



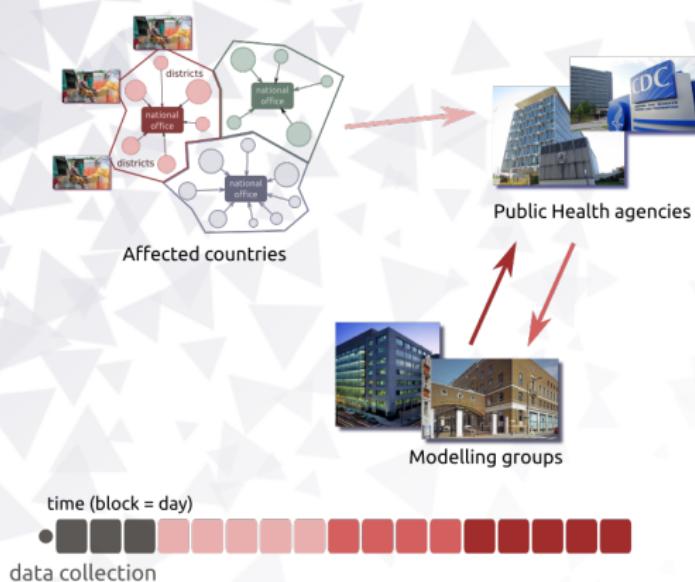
Public Health agencies



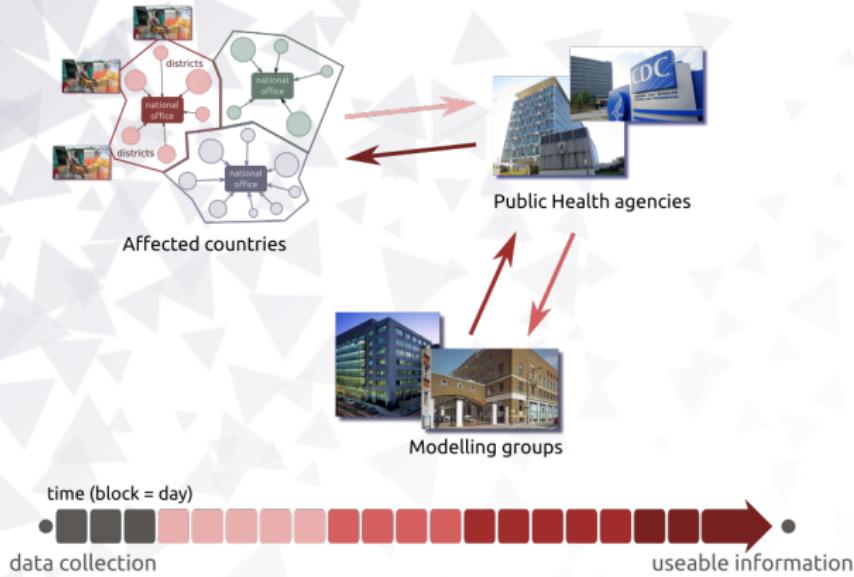
Centralised analyses, distributed delays



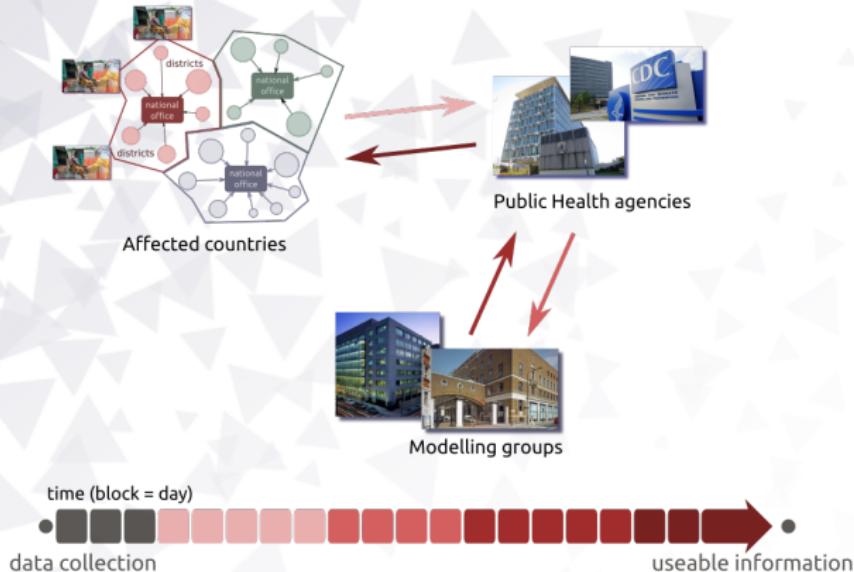
Centralised analyses, distributed delays



Centralised analyses, distributed delays



Centralised analyses, distributed delays



Timeliness is key: need to bring analytics to the field

Bringing analytics resources where they are needed

RECON

www.repidemicsconsortium.org

- an NGO for free, open **health crisis analytics**
- 100-150 subscribers, ~30 active members

RECON

www.repidemicsconsortium.org

- an NGO for free, open **health crisis analytics**
- 100-150 subscribers, ~30 active members
- development of **free**, open-source **analysis tools** (using )
- **10 packages released**, ~15 in development

RECON

www.repidemicsconsortium.org

- an NGO for free, open **health crisis analytics**
- 100-150 subscribers, ~30 active members
- development of **free**, open-source **analysis tools** (using )
- **10 packages released**, ~15 in development
- **short courses** with partner institutions (CDC, MSF, WHO, EAN, ...)

RECON

www.repidemicsconsortium.org

- an NGO for free, open **health crisis analytics**
- 100-150 subscribers, ~30 active members
- development of **free**, open-source **analysis tools** (using )
- **10 packages released**, ~15 in development
- **short courses** with partner institutions (CDC, MSF, WHO, EAN, ...)
- support **field deployment**

Thanks to:

- **Session:** John Edmunds
- **Collaborators:** Finlay Campbell, Anne Cori, Pierre Nouvellet, Zhian Kamvar, Amrish Baidjoe, Neil Ferguson, Dan Bausch, Jimmy Whitworth, Bayard Roberts
- **Groups:** WHO Ebola Likati Response Team
- **Funding:** GCRF project RECAP (ES/P010873/1), UK PH RST, HPRU-NIHR, MRC

Job real-time modelling at LSHTM



RECON

www.repidemicsconsortium.org