Using `sourceR` for Bayesian non-parametric source attribution of zoonotic diseases

Poppy Miller

August 31, 2020

Zoonotic diseases are a major cause of morbidity, and productivity losses in both humans and animal populations. Identifying the source of food-borne zoonoses (e.g. an animal reservoir or food product) is crucial for the identification and prioritisation of food safety interventions. For many zoonotic diseases it is difficult to attribute human cases to sources of infection because there is little epidemiological information on the cases. However, microbial strain typing allows zoonotic pathogens to be categorised, and the relative frequencies of the strain types among the sources and in human cases allows inference on the likely source of each infection.

We introduce `sourceR`, an `R` package for quantitative source attribution, aimed at food-borne diseases. `sourceR` implements a fully joint non-parametric Bayesian model using strain-typed surveillance data from both human cases and source samples. The model attributes cases of infection in humans to putative sources of infection thus allowing the development of targeted interventions to reduce prevalence of the disease. Further source attribution models are planned to be added to the package. The model measures the force of infection from each source, allowing for varying survivability, pathogenicity and virulence of pathogen strains, and varying abilities of the sources to act as vehicles of infection. A Bayesian non-parametric (Dirichlet process) approach is used to cluster pathogen strain types by epidemiological behaviour, avoiding model overfitting and allowing detection of strain types associated with potentially high virulence, pathogenicity and survivability. This categorises pathogens by their risk to humans when detected in food sources.

HaldDP Model

To fit the model, strain typed samples are required from both humans and putative sources of infection. Often, human case data is associated with location such as urban/rural, or even GPS coordinates. On the other hand, food samples are likely to be less spatially constrained due to distances between production and sale locations, hence, the model currently does not allow source data to vary by location. Also, both human and source data may exist for multiple time-periods. We therefore denote the number of human cases of time i occurring in time-period t at location l by y_{itl}, the number of samples of source j in time-period t by s_{jt}, with the type counts x_{ijt}.

The number of human cases y_{itl} identified by isolation of subtype i in time-period t at location l is modelled as a Poisson random variable with mean given by a linear combination of source specific effects, type specific effects and source sample contamination prevalences.

$$y_{itl} \sim \mathsf{Poisson}\left(\lambda_{itl}\right) \tag{1}$$

$$\lambda_{itl} = q_i \sum_{j=1}^{m} a_{jtl} \cdot p_{ijt} \tag{2}$$

where p_{ijt} is the absolute prevalence of each pathogen type i in source j at time t. The unknown parameters in the model are the vectors $\boldsymbol{q}$ and $\boldsymbol{\alpha}$. Here, $\boldsymbol{q}$ represents the characteristics that determine a type's capacity to cause an infection (such as survivability during food processing, pathogenicity and virulence), and $\boldsymbol{\alpha}$ accounts for the ability of a particular source to act as a vehicle of infection or exposure to a given food source. We allow for different exposures of humans to sources in different locations, by allowing the source effects to vary between times and locations, α_{jtl}. Inference is performed in a Bayesian framework allowing the model to explicitly include and quantify the uncertainty surrounding each of the parameters.

For each source j, we model the number of positive source samples

$$\boldsymbol{x}_{jt} \sim \text{Multinomial}(s_{jt}^{+}, \boldsymbol{r}_{jt}) \tag{3}$$

where $\boldsymbol{x}_{jt} = (x_{ijt}, i = 1, ..., n)^T$ denotes the vector of type-counts in source j in time-period t, $s_{jt}^{+} = \sum_{i=1}^{n} x_{ijt}$ denotes the total number of positive samples obtained for source j, and $\boldsymbol{r}_{jt}$ denotes a vector of relative prevalences $Pr\left(\mathsf{type}_i|\mathsf{source}_j, \mathsf{time}_t\right)$. The source case model is then coupled to the human case model through the simple relationship

$$p_{ijt} = r_{ijt}k_{jt} \tag{4}$$

where $r_{ijt} = x_{ijt}/\sum_{i=1}^{n} x_{ijt}$ is the relative prevalence of type i given source j and time t and $k_{jt} = s_{jt}^{+}/s_{jt}$ is the prevalence of any isolate in source j in time-period t (note, s_{jt} is the total number of samples tested for source j, time t).

The type effects $\boldsymbol{q}$, which are assumed invariant across time or location, are drawn from a DP with base distribution Q_0 and a concentration parameter a_q

$$q_i \sim \text{DP}\left(a_q, Q_0\right). \tag{5}$$

The DP groups the elements of $\boldsymbol{q}$ into a finite set of clusters $1 : \kappa$ (unknown *a priori*) with values $\theta_1, ..., \theta_\kappa$ meaning bacterial types are clustered into groups with similar epidemiological behaviour.

The estimated number of cases attributed to a particular source j is

$$\hat{\xi}_{jtl} = a_{jtl} \sum_{i=1}^{n} q_i \cdot p_{ijt}. \tag{6}$$

Comparing the relative magnitudes of $\hat{\xi}_j$ provides a statistical method to prioritise intervention strategies to the most important sources of infection. The model is fitted in a Bayesian framework as posteriors for functions of parameters (such as ξ) are easily calculated, and to allow previous knowledge to be incorporated via informative priors.

Heterogeneity in the source matrix $\boldsymbol{x}$ is absolutely required to identify clusters from sources, which may not be guaranteed *a priori* due to the observational nature of the data collection. However, a sparse or highly unbalanced source matrix increases posterior correlations between some source and type effects. In our experience, the algorithm works best when the source matrix has a moderate amount of heterogeneity.

The interpretation of source $\boldsymbol{\alpha}$ and type effects $\boldsymbol{q}$ depends on the quality and type of data collected, the model specification, and the characteristics of the organism of interest. Source effects account for factors such as the amount of the food source consumed, the physical properties of the source and the environment provided for the bacteria through storage and preparation. Including an environmental source in the model can be thought of as grouping the (individually) unmeasured wildlife sources into one. It may also be a transmission pathway for pathogens present in livestock sources (for example, through the contamination of waterways) which complicates the interpretation meaning the source effects no longer directly summarise the ability of the source to act as a vehicle for food-borne infections [12]. The non-parametric clustering of the subtypes type effects groups the subtypes by epidemiological traits. This means subtypes assigned to the same group have similar pathogenicity, virulence and survivability.

Priors

The parameters $\boldsymbol{\alpha}_{tl}$ and $\boldsymbol{q}$ account for a multitude of source and type specific factors which are difficult to quantify *a priori*. Therefore, with no single real-world interpretation, the distributional form of the priors were chosen for their flexibility. A Dirichlet prior is placed on each $\boldsymbol{r}_{jt}$ which suitably constrains the individuals r_{ij}s such that $\sum_{i=1}^{n} r_{ijt} = 1$. A Dirichlet prior is also placed on each $\boldsymbol{\alpha}_{tl}$, with the constraint $\sum_{j=1}^{m} \alpha_{jtl} = 1$ aiding identifiability between the mean of the source and type effect parameters. In `sourceR`, the concentration parameter of the DP α_q is specified by the analyst as a modelling decision.

Specifying Dirichlet priors: The simplest Dirichlet priors for the source effects and relative prevalences are symmetric (meaning all of the elements making up the parameter vector $\boldsymbol{a}$ have the same value a, called the concentration parameter). Symmetric Dirichlet distributions are used as priors when there is no prior knowledge favouring one component over another. When a is equal to one, the symmetric Dirichlet distribution is uniform over all points in its support. Values of the concentration parameter above one prefer variates that are dense, evenly distributed distributions, whilst values of the concentration parameter below 1 prefer sparse distributions. Note, a prior of 1 for the relative prevalences is too strong (if a relatively non-informative prior is preferred) when there are many observed zero's in the source data. A more informative prior can be specified by using a non-symmetric Dirichlet distribution. The magnitude of the vector of $\boldsymbol{a}$ parameters corresponds to the strength of the prior. The relative values of the $\boldsymbol{a}$ vector corresponds to prior information on the comparative sizes of the parameters.

Dirichlet Process: The Dirichlet Process non-parametrically clusters the pathogen type effects providing an automatic data-driven way of reducing the dimensionality of $\boldsymbol{q}$ to aid model identifiability and identify groups of pathogens with similar virulence, pathogenicity and survivability. The Dirichlet Process is a random probability measure defined by a base distribution Q_0 and a concentration parameter a_q [25]. The base distribution constitutes a prior distribution in the values of each element of the type effects $\boldsymbol{q}$ whilst the concentration parameter encodes prior information on the number of groups K to which the pathogen types are assigned. For small values of a_q, samples from the DP are likely to have a small number of atomic measures with large weights. For large values, most samples are likely to be distinct, and hence, concentrated on Q_0. A value of 1 implies that, *a priori*, two randomly selected types have probability 0.5 of belonging to the same cluster [16].

The concentration parameter of the DP is specified by the analyst as a modelling decision. The concentration parameter specifies how strong the prior grouping is. In the limit $a \to 0$, all types will be assigned to one group, increasing a makes a larger number of groups increasingly likely. The Gamma base distribution Q_0 induces a prior for the cluster locations. This prior should not be too diffuse because if these locations are too spread out, the penalty in the marginal likelihood for allocating individuals to different clusters will be large, hence the tendency will be to overly favour allocation to a single cluster. However, the prior parameters may have a stronger effect than anticipated due to the small size of the relative prevalence and source effect parameters. This can been seen by considering the marginal posterior for θ_k

$$\theta_k \sim \mathsf{Gamma}\left(a_\theta + \sum_{i:S_i=k} y_i, b_\theta + \sum_{i:S_i=k}\sum_{j=1}^{m} \alpha_j \cdot p_{ij}\right)$$

The term $\sum_{i:S_i=k}\sum_{j=1}^{m} \alpha_j \cdot p_{ij}$ is very small (due to the Dirichlet priors on $\boldsymbol{\alpha}$ and $\boldsymbol{r}_j$), which can result in even a fairly small rate parameter (b_θ) dominating.

Fitting the model using sourceR

A simulated data set with data covering 2 times (1, 2) and 2 locations (A, B) is provided with the package, named `sim_SA`. In accordance with the HaldDP model, the source data varies over time, the source effects vary over times and locations, and the type effects are fixed over both times and locations. The priors are chosen to be minimally informative. The algorithm is run for a total of 500,000 iterations (with a burn in of 2000 iterations and thinning 500).

First the human case data, source sample data and prevalences are passed to methods `Y`, `X` and `Prev` to correctly format them. The first argument to each data formatting method gives the data in long format, the remaining arguments give the headers for the columns containing case or sample counts and identification variables (time, location, source, type).

```
y <- Y(                       # Cases
  data = sim_SA$cases,
  y = "Human",
  type = "Type",
  time = "Time",
  location = "Location"
)

x <- X(                       # Sources
  data = sim_SA$sources,
  x = "Count",
  type = "Type",
  time = "Time",
  source = "Source"
)

k <- Prev(                    # Prevalences
  data = sim_SA$prev,
  prev = "Value",
  time = "Time",
  source = "Source"
)
```

The model is the constructed using the above data, initial values and priors. Starting values are selected automatically unless provided via a list named `init` to the constructor. The priors for the $\boldsymbol{\alpha}$ and $\boldsymbol{R}$ can be specified as a dataframe with one value per time/ location/ type / source combination or as a single number (which is replicated for each α_{jtl} and r_{ijt} respecitvely).

```
## Create long-format Dirichlet(1) priors

## Create alpha prior data frame
prior_alpha_long <- expand.grid(
  Source   = unique(sim_SA$sources$Source),
  Time     = unique(sim_SA$sources$Time),
  Location = unique(sim_SA$cases$Location),
  Alpha    = 1
)
# Use the Alpha() constructor to specify alpha prior
prior_alpha <- Alpha(
  data     = prior_alpha_long,
  alpha    = ’Alpha’,
  source   = ’Source’,
  time     = ’Time’,
  location = ’Location’
)

## Create r prior data frame
prior_r_long <- expand.grid(
  Type   = unique(sim_SA$sources$Type),
  Source = unique(sim_SA$sources$Source),
  Time   = unique(sim_SA$sources$Time),
  Value  = 0.1
)
# Use X() constructor to specify r prior
prior_r <- X(
  data   = prior_r_long,
  x      = ’Value’,
  type   = ’Type’,
  time   = ’Time’,
  source = ’Source’
)

## Pack all priors into a list
priors <- list(
  a_theta = 0.01,
  b_theta = 0.00001,
  a_alpha = prior_alpha,
  a_r     = prior_r
)
```

Note, a shorthand specification of the priors is available when a single number is desired for all r's and α's.

```
## Equivalent result to the longform priors specified above
priors <- list(
  a_theta = 0.01,
  b_theta = 0.00001,
  a_alpha = 1,
  a_r     = 0.1
)
```

Having specified the priors, initial values for the Markov chain may be specified. This step is optional: in the absence of user-specified initial values, the chain will be automatically initialised. The user is at liberty to specify initial values for $\boldsymbol{\alpha}$, R, and/or $\boldsymbol{q}$ as desired. Values for $\boldsymbol{\alpha}$ and R are specified similarly to the prior hyperparameters as shown above, with directly equivalent use of the `Alpha()` and `X()` constructors. The `HaldDP()` model constructor (see below) will warn the user if sum-to-unity constraints in $\boldsymbol{\alpha}$ and R are not satisfied, and will automatically renormalise vectors where necessary. To specify starting values for $\boldsymbol{q}$, the `Q()` constructor may be used

```
types  <- unique(sim_SA$cases$Type)
q_long <- data.frame(q=rep(15, length(types)), Type=types)
init_q <- Q(q_long, q = 'q', type = 'Type')
```

with all specified initial values packed into a list, e.g.

```
inits <- list(q = init_q) # Pack starting values into a list
```

Armed with formatted data, prior hyperparameters, and initial values, construction of a `HaldDP()` object is a simple process:

```
my_model <- HaldDP(y = y, x = x, k = k, priors = priors, inits = inits, a_q = 0.1)
```

McMC control parameters are set via the `mcmc_params` method. Here, we request 1000 McMC iterations after 2000 iterations burn-in, and thinning by 500 (for a total of 502000 iterations). We accept the default number of elements of the R matrix to update per ‘sweep’ of the other parameters.

```
my_model$mcmc_params(n_iter = 1000, burn_in = 2000, thin = 500)
```

The model is run using the `update` function. Additional iterations may be appended using `append = TRUE`. Setting `append = FALSE` the or re-running the `mcmc_params` method will delete the previously saved posterior.

```
my_model$update()
my_model$update(n_iter = 100, append = T)
```

We provide the `extract` method for ease of access to the complex posterior. The `extract` function returns the posterior for the selected parameters as a list with a multidimensional array for each of `alpha`, `r`, `q`, `s`, `lambda_j` and `lambda_i`. This can be flattened to a list of long format data frames using the argument `flatten = T`. The posterior can be subset by parameter, time, location type or source.

```
## returns the posterior for the r, alpha, q, c,
## lambda_i, xi and xi_prop parameters,
## for all times, locations, sources and types
## the posterior is returned as a list or arrays
my_model$extract()

## returns the posterior for the r and alpha parameters,
## for time 1, location B, sources Source3, and Source4,
## types 5, 25, and 50, and iterations 200:300
## the posterior is returned as a list of dataframes
my_model$extract(params = c("r", "alpha"),
                 times = "1", location = "B",
                 sources = c("Source3", "Source4"),
                 types = c("5", "25", "50"),
                 iters = 200:300,
                 flatten = T)
```

Trace and autocorrelation plots for the parameters (Figure 1) indicate that the Markov chain is mixing well and has converged, and that thinning by 500 is adequate. The following R code demonstrates how to access and plot the marginal posteriors for some parameters.

```
## Plot the marginal posterior for source effect 2, time 1, location A
plot(my_model$extract(params = "alpha", times = "1", locations = "B",
     sources = "Source4")$alpha, type="l")

## Plot the marginal posterior for the type effect 21
plot(my_model$extract(params = "q", types = "21")$q, type="l")

## Plot the marginal posterior for the relative prevalence for
## Source5, type 17, at time 2
plot(my_model$extract(params = "r", times = "2", sources = "Source5",
     types = "17")$r, type="l")

## Plot the marginal posterior for xi Source1, time 1, location A
plot(my_model$extract(params = "xi", times = "1", locations = "A",
     sources = "Source1")$xi, type = "l")

## Plot the marginal posterior for lambda_i 10, time 2, location B
plot(my_model$extract(params = "lambda_i", times = "2", locations = "B",
     types = "10")$lambda_i, type="l")
```

The `summary()` function calculates medians and credible intervals calculated with three possible methods (percentile, SPIn [18], or chen-shao [19]). The output can be subset in the same way as `extract()`.

```
my_model$summary(alpha = 0.05, CI_type = "percentiles")

my_model$summary(alpha = 0.05, CI_type = "chen-shao",
                 params = c("r", "alpha"),
                 times = "1", location = "B",
                 sources = c("Source3", "Source4"),
                 types = c("5", "25", "50"),
                 iters = 200:300,
                 flatten = T)
```

The heatmap shows the grouping of the type effects (Figure 2) computed using a dissimilarity matrix from the clustering output of the McMC. The coloured bar under the dendrogram gives the correct grouping from the simulated data. This shows that the majority of types have been classified correctly. Care must be taken in performing marginal interpretations of the number of type parameters. It is much easier to split a group into two (with similar group means) than it is to merge two groups with clearly different means. Hence, a histogram of the number of groups per iteration is positively skewed compared to the true number of groups. When fitting the model with simulated data, visually assessing the dendrogram and heatmap to determine the number of groups usually provides a closer value to the true number of groups than looking at a histogram, particularly when the group means are well separated.

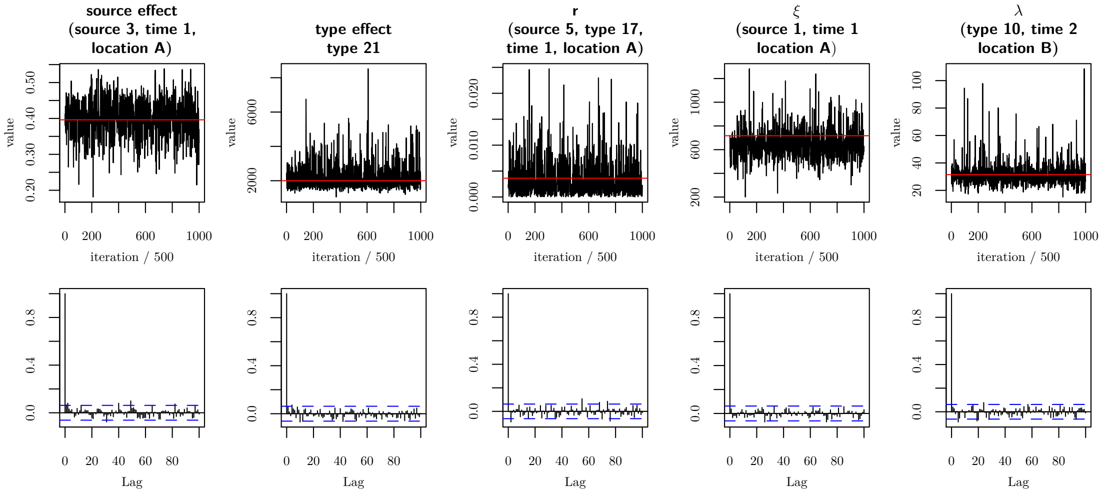

Figure 1: Trace and acf plots for a sample of the model parameters. True values of the parameters are shown in red.

```
my_model$plot_heatmap()
```

The violin plots of the number of cases attributed to each source at each time and location λ_{jtl} (Figure 3) and the number of cases attributed to each type λ_{itl} (Figure 4) show that the true values (shown by a red horizontal line on the graph) are being estimated well.

The `sourceR` package allows the relative prevalence matrix to be fixed at the maximum likelihood estimates which increases the posterior precision (and significantly reduces run time), but may bias the results if the source data is not of high quality. Reducing the number of elements in the relative prevalence matrix r that get updated at each iteration can significantly reduce computation time at the expense of convergence speed.

The data, initial values, prior values, acceptance rates, and McMC parameters, can be accessed using a set of `get()` methods.

```
my_model$get_data()
my_model$get_inits()
my_model$get_priors()
my_model$get_acceptance()
my_model$get_mcmc_params()
```

References

[1] Havelaar AH, Kirk MD, Torgerson PR, Gibb HJ, Hald T, Lake RJ, et al. World Health Organization Global Estimates and Regional Comparisons of the Burden of Foodborne Disease in 2010. PLoS Med. 2015;12(12):1–23. doi:10.1371/journal.pmed.1001923.

[2] World Health Organization. WHO estimates of the global burden of foodborne diseases: foodborne disease burden epidemiology reference group 2007-2015; 2015. available on the WHO web site (www.who.int) or can be purchased from WHO Press, World Health Organization, 20 Avenue Appia, 1211 Geneva 27, Switzerland. Available from: `http://apps.who.int/iris/bitstream/10665/199350/1/9789241565165_eng.pdf?ua=1`.

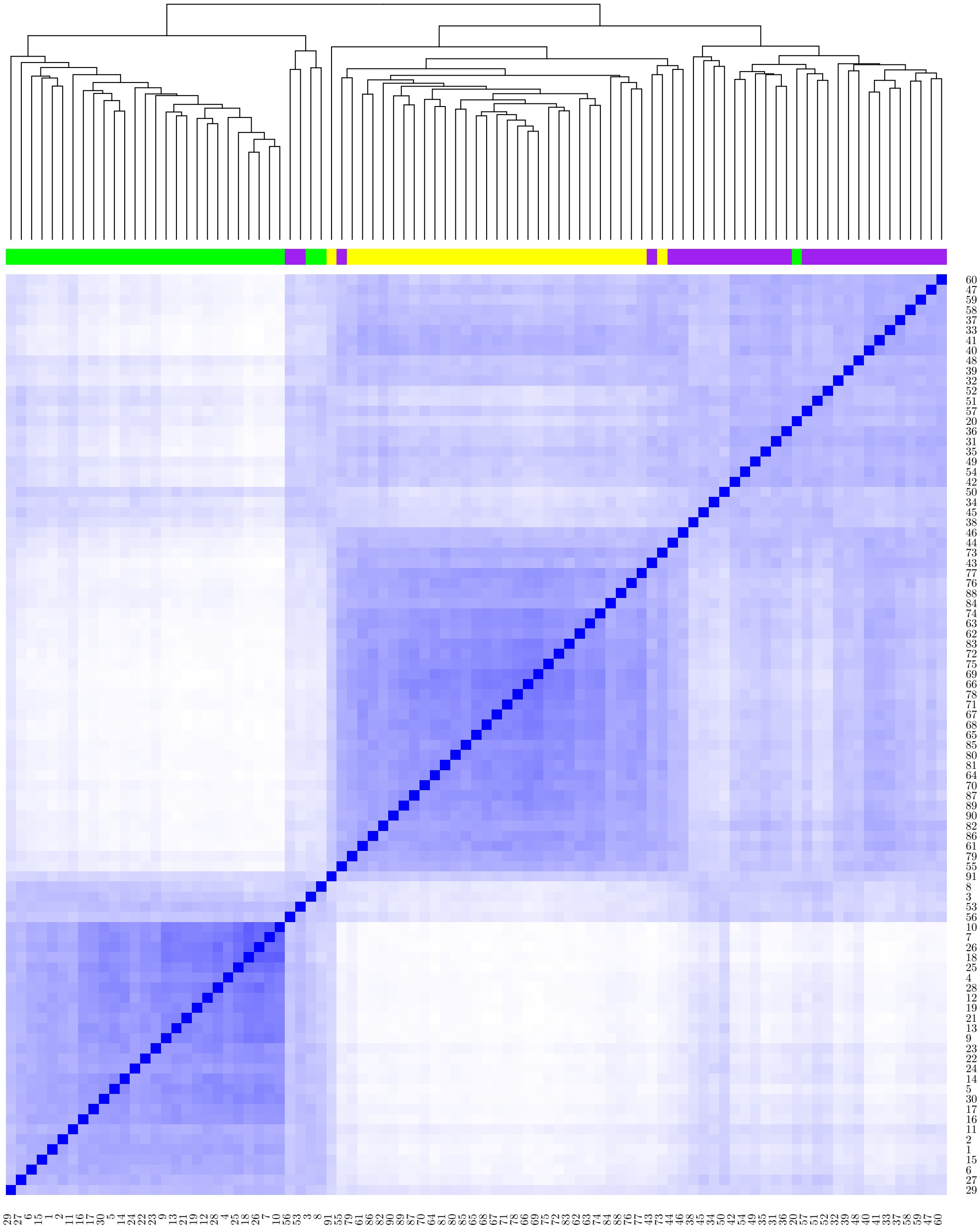

Figure 2: Heatmap showing the grouping of the type effects (q) using simulated data (true groupings given by the 3 colours in the bar under the dendrogram).

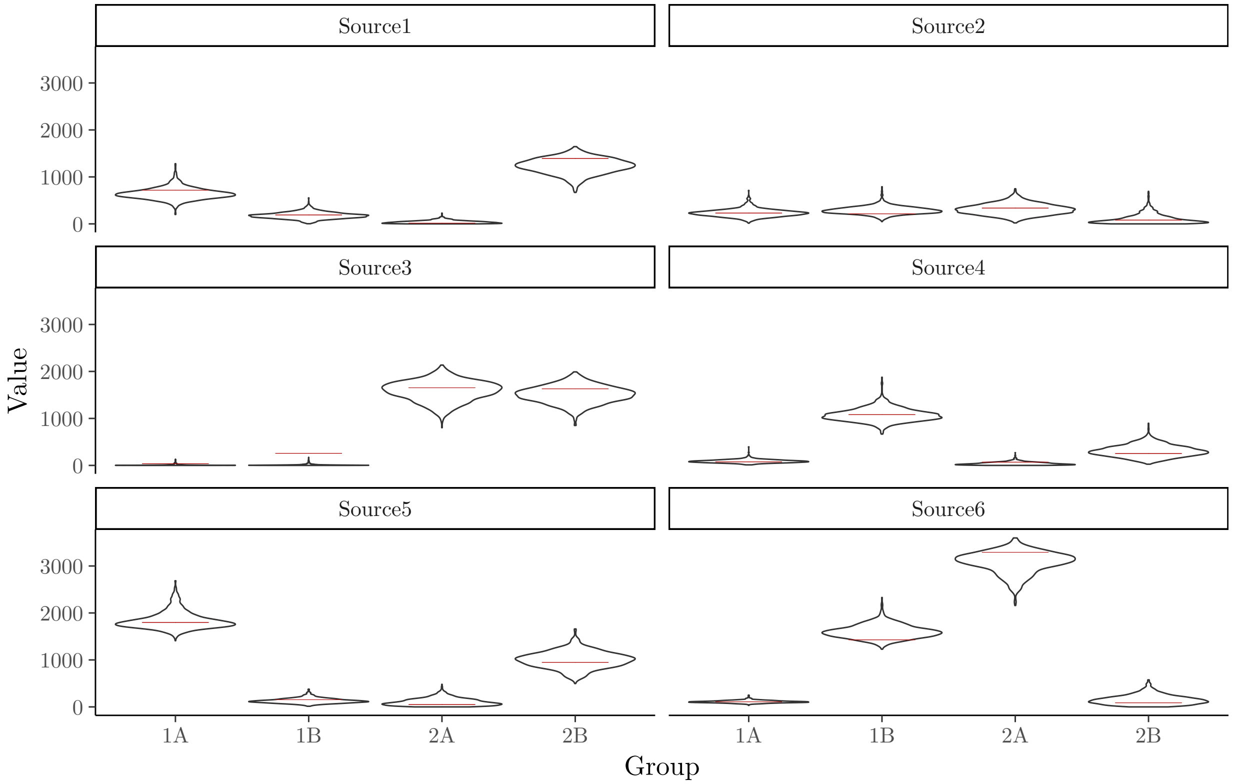

Figure 3: Violin plots showing marginal posteriors for each ξ (number of cases attributable to each source) for each time (1, 2) and location (A, B). True ξ values are shown as horizontal red lines.

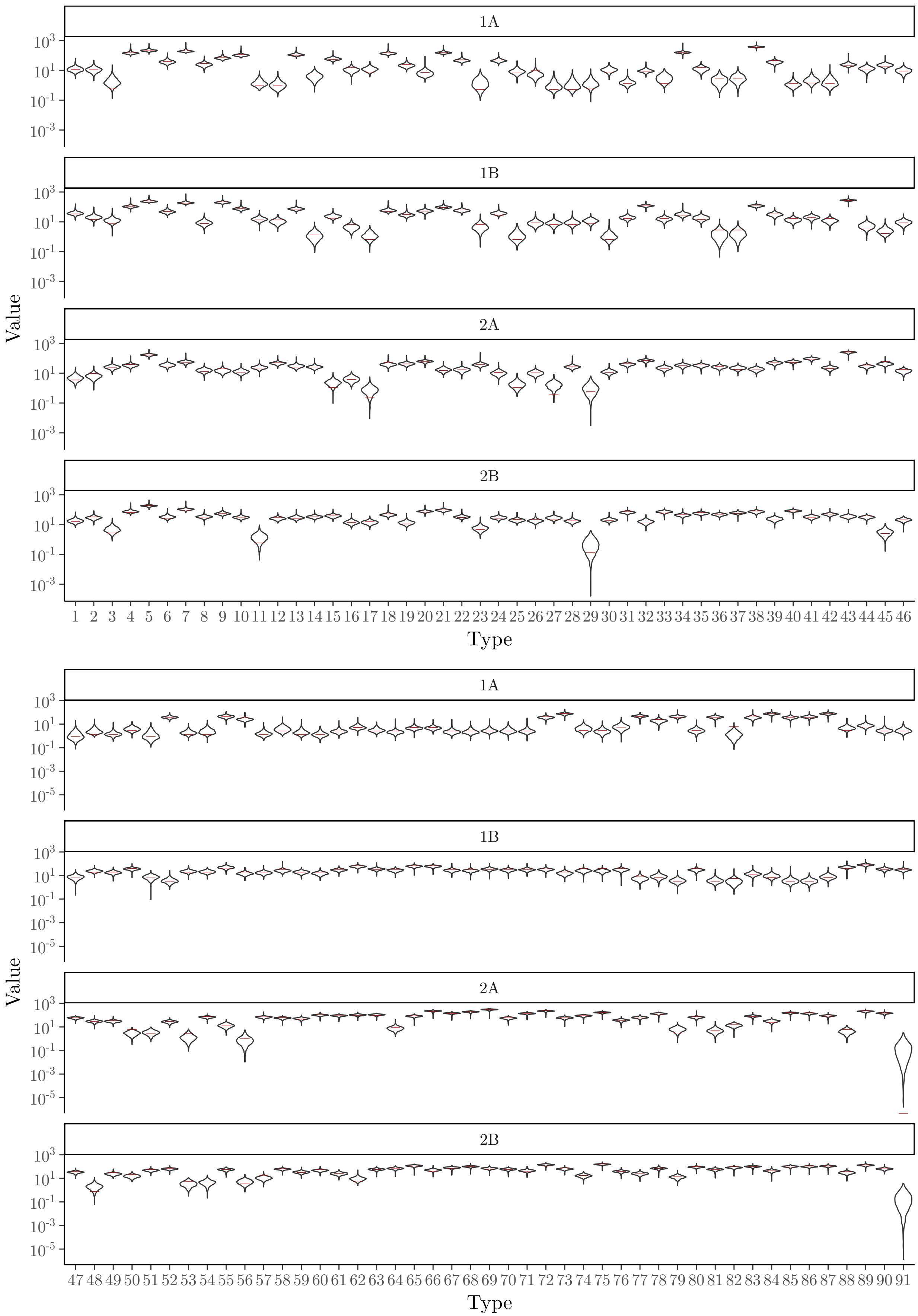

Figure 4: Violin plots showing the marginal posteriors for each λ_i (number of cases attributed to each type) for each time (1, 2) and location(A, B). True λ_i values are shown as horizontal red lines.

[3] Crump JA, Griffin PM, Angulo FJ. Bacterial Contamination of Animal Feed and Its Relationship to Human Foodborne Illness. Clinical Infectious Diseases. 2002;35(7):859–865. doi:10.1086/342885.

[4] Mullner P, Jones G, Noble A, Spencer S, Hathaway S, French N. Source Attribution of Food Borne Zoonoses in New Zealand: A Modified Hald Model. Risk Analysis. 2009;29(7).

[5] Urwin R, Maiden M. Multi-locus Sequence Typing: A Tool for Global Epidemiology. Trends in Microbiology. 2003;.

[6] Dingle K, Colles F, Wareing D, Ure R, Fox A, Bolton F, et al. Multilocus sequence typing system for Campylobacter jejuni. Journal of Clinical Microbiology. 2001;.

[7] Allos BM, Moore MR, Griffin PM, Tauxe RV. Surveillance for Sporadic Foodborne Disease in the 21st Century: The FoodNet Perspective. Clinical Infectious Diseases. 2004;38(Supplement 3):S115–S120. doi:10.1086/381577.

[8] Baker M, Wilson R, Ikram R, Chambers S, Shoemack S, Cook G. Regulation of Chicken Contamination Urgently Needed to Control New Zealand's Serious Campylobacteriosis Epidemic. The New Zealand Medical Journal. 2006;.

[9] Mullner P, Collins-Emerson J, Midwinter A, Carter P, Spencer S, van der Logt P, et al. Molecular Epidemiology of Campylobacter jejuni in a Geographically Isolated Country with a Uniquely Structured Poultry Industry. Applied and Environmental Microbiology. 2010;76(7):2145–2154.

[10] French N, Marshall J. Dynamic Modelling of Campylobacter Sources in the Manawatu. Hopkirk Institute, Massey University; 2009.

[11] French N, Marshall J. Completion of Sequence Typing of Human and Poultry Isolates and Source Attribution Modelling. Hopkirk Institute, Massey University; 2013.

[12] Hald T, Vose D, Wegener H, Koupeev T. A Bayesian Approach to Quantify the Contribution of Animal-Food Sources to Human Salmonellosis. Risk Analysis. 2004;24(1):255–269.

[13] Wilson D, Gabriel E, Leatherbarrow A, Cheesebrough J, Hart C, Diggle P. Tracing the Source of Campylobacteriosis. PLoS Genetics. 2008;.

[14] Wilson D. `iSource`; 2016. Available from: `http://www.danielwilson.me.uk/iSource.html`.

[15] Roberts G, Rosenthall J. Examples of Adaptive MCMC. University of Toronto Department of Statistics; 2006.

[16] Gelman A, Carlin J, Stern H, Dunson D, Vehtari A, Rubin D. Bayesian Data Analysis. Chapman & Hall/CRC Texts in Statistical Science; 2013.

[17] van Pelt W, van de Giessen A, van Leeuwen W, Wannet W, Henken A, Evers E. Oorsprong, Omvang en Kosten van Humane Salmonellose. Deel1. Oorsprong van Humane Salmonellose met Betrekking tot Varken, Rund, Kip, ei en Overige Bronnen. Infectieziekten Bull. 1999;.

[18] Liu Y, Gelman A, Zheng T. Simulation-efficient Shortest Probability Intervals. Statistics and Computing. 2015;.

[19] Chen M, Shao Q. Monte Carlo Estimation of Bayesian Credible and HPD Intervals. Journal of Computational and Graphical Statistics. 1991;.

[20] Gower JC. A general coefficient of similarity and some of its properties. Biometrics. 1971;.

[21] pubmlst. Campylobacter MLST; 2016. Available from: `http://pubmlst.org/campylobacter/`.

[22] Wilson DJ, Gabriel E, Leatherbarrow AJH, Cheesbrough J, Gee S, Bolton E, et al. Rapid Evolution and the Importance of Recombination to the Gastroenteric Pathogen Campylobacter jejuni. Molecular Biology and Evolution. 2009;26(2):385–397.

[23] Duane S, Kennedy AD, Pendleton BJ, Roweth D. Hybrid Monte Carlo. Physics Letters B. 1987;195(2):216 – 222. doi:http://dx.doi.org/10.1016/0370-2693(87)91197-X.

[24] Homan MD, Gelman A. The No-U-turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. J Mach Learn Res. 2014;.

[25] Ferguson T. Bayesian Analysis of some Nonparametric Problems. Ann Stat. 1973;1:209–230.