Heatmaps

```
rm(list = ls())

sapply(c("knitr", "buds", "coda", "dplyr", "ggplot2", "MCMCglmm",
         "rstan", "viridis"), require, character.only = TRUE)
```

```
## Loading required package: knitr
## Loading required package: buds
## Loading required package: Rcpp
## Loading required package: coda
## Loading required package: dplyr
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
## Loading required package: ggplot2
## Loading required package: MCMCglmm
## Loading required package: Matrix
## Loading required package: ape
## Loading required package: rstan
## Loading required package: StanHeaders
## rstan (Version 2.15.1, packaged: 2017-04-19 05:03:57 UTC, GitRev: 2e1f913d3ca3)
## For execution on a local, multicore CPU with excess RAM we recommend calling
## rstan_options(auto_write = TRUE)
## options(mc.cores = parallel::detectCores())
##
## Attaching package: 'rstan'
## The following object is masked from 'package:coda':
##
##     traceplot
## Loading required package: viridis
## Loading required package: viridisLite
##    knitr     buds     coda    dplyr  ggplot2 MCMCglmm    rstan  viridis
##     TRUE     TRUE     TRUE     TRUE     TRUE     TRUE     TRUE     TRUE
```

```
# Save generated figures
opts_chunk$set(fig.path = paste0("./heatmaps/"), dev='png')
# Functions
```

```
source("../R/distcomps.R")
source("../R/get_data_to_plot.R")
source("../R/plot_utils.R")
```

```
# Options
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
theme_set(theme_classic())
theme_update(text=element_text(size=15),
             legend.title.align = 0.5,
             legend.title = element_text(size=12))

# Parameters
min_row_sum <- 100
min_row_prevalence <- 5
B <- 100
min_sigma <- 0.05
hparams <- list(
  "gamma_tau"= 2.5,
  "gamma_epsilon" = 2.5,
  "gamma_bias" = 2.5,
  "gamma_rho" = 2.5,
  "min_sigma" = min_sigma
)
nfeatures <- 500
K <- 10
```

Generating heatmaps using alternative methods for comparison with BUDS ordering.

Frog data

```
# Load data files
sampleData_default_file <- "../data/frog_sample_data.csv"
countTable_default_file <- "../data/frog_processed_counts.csv"

sampleData <- read.csv(sampleData_default_file, row.names = 1)
covariate_name <- "hpf"
sample_covariate <- sampleData[, covariate_name]

X <- read.csv(countTable_default_file, row.names = 1)

D0 <- cor_dist(X, log_trans = FALSE)
D <- D0
```

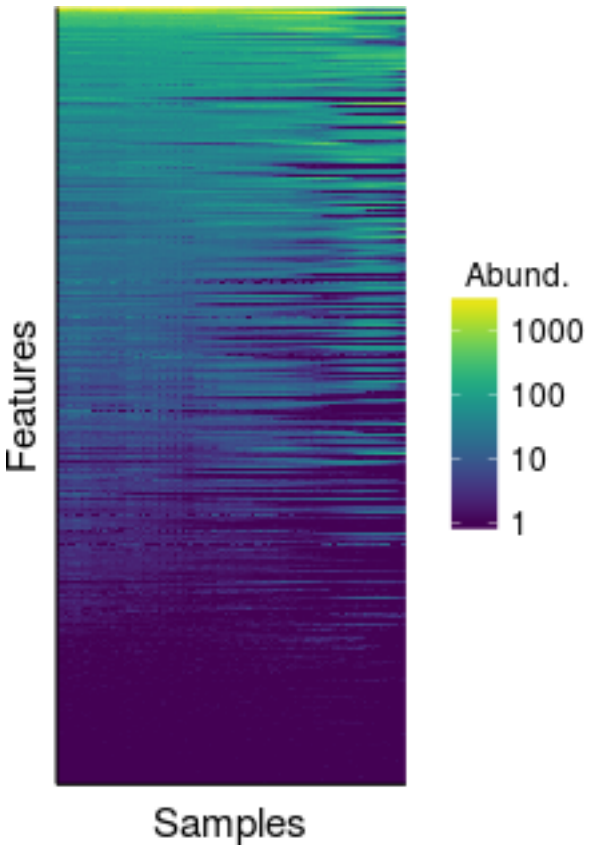
```
set.seed(1)
buds_seed <- sample.int(.Machine$integer.max, 1)
budsFit <- buds::fit_buds(D, K = K, method = "vb", hyperparams = hparams,
                          init_from = "random", seed = buds_seed,
                          tol_rel_obj = 0.005)
```

```
## ------------------------------------------------------------
## EXPERIMENTAL ALGORITHM:
##   This procedure has not been thoroughly tested and may be unstable
```

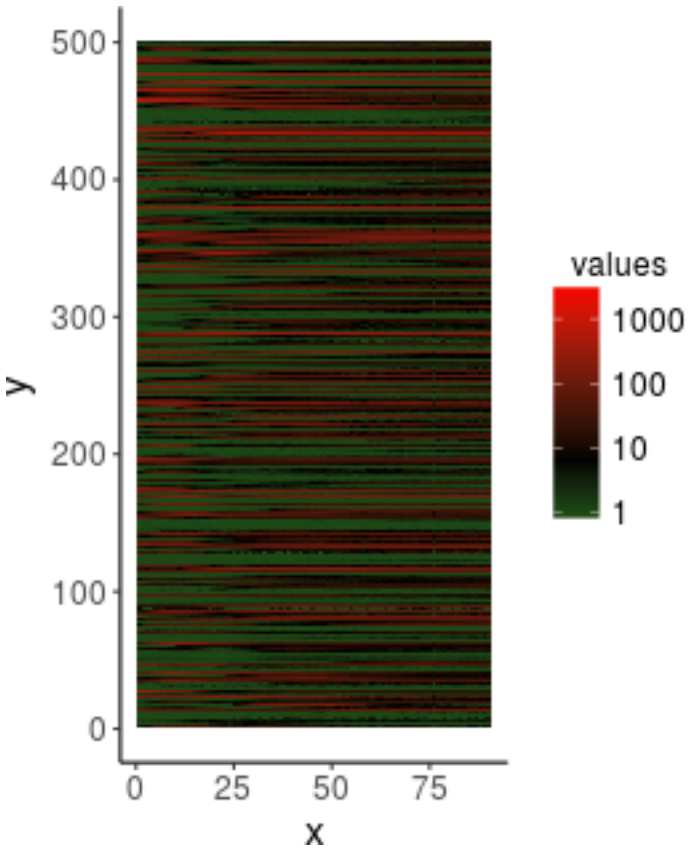
```
##   or buggy. The interface is subject to change.
## ------------------------------------------------------------
##
##
##
## Gradient evaluation took 0.002899 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 28.99 seconds.
## Adjust your expectations accordingly!
##
##
## Begin eta adaptation.
## Iteration:   1 / 250 [  0%]  (Adaptation)
## Iteration:  50 / 250 [ 20%]  (Adaptation)
## Iteration: 100 / 250 [ 40%]  (Adaptation)
## Iteration: 150 / 250 [ 60%]  (Adaptation)
## Iteration: 200 / 250 [ 80%]  (Adaptation)
## Success! Found best value [eta = 1] earlier than expected.
##
## Begin stochastic gradient ascent.
##   iter       ELBO   delta_ELBO_mean   delta_ELBO_med   notes
##    100     -7e+04             1.000            1.000
##    200     -3e+03            10.365           19.730
##    300      2e+03             7.917            3.021
##    400      6e+03             6.124            3.021
##    500      8e+03             4.944            1.000
##    600      8e+03             4.121            1.000
##    700      8e+03             3.532            0.747
##    800      8e+03             3.091            0.747
##    900      8e+03             2.748            0.221
##   1000      8e+03             2.473            0.221
##   1100      8e+03             2.373            0.007   MAY BE DIVERGING... INSPECT ELBO
##   1200      8e+03             0.400            0.002   MEDIAN ELBO CONVERGED
##
## Drawing a sample of size 1000 from the approximate posterior...
## COMPLETED.
```

```
budsParams <- (rstan::extract(budsFit$fit_buds))
tau_df <- get_tau_df(budsParams, prob = 0.95)
```

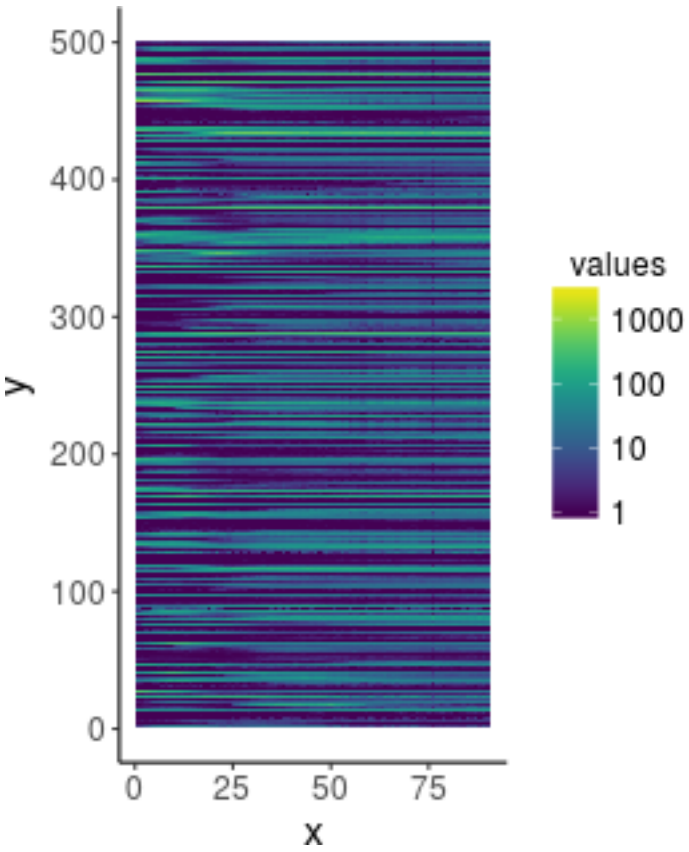
```
set.seed(1)
idx <- sample(1:nrow(X), nfeatures)
(plt <- plot_ordered_matrix(X, tau_df$tau,
                            log_trans = TRUE,
                            keep_fatures = idx) +
    coord_fixed(0.40))
```




```
Xsmall <- as.matrix(X[idx, ])
NeatMap::heatmap1(Xsmall + 1) + coord_fixed(0.35) +
  scale_fill_gradient2(low = "green", high = "red", mid = "black",
                       midpoint = mean(log10(Xsmall + 1), na.rm = TRUE),
                       trans = "log10")
```




```
Xsmall <- X[idx, ]
NeatMap::heatmap1(Xsmall+1) + coord_fixed(0.35) +
  scale_fill_viridis(trans = "log10")
```



TARA Oceans

```
# Load data files
sampleData_default_file <- "../data/tara_sample_data.csv"
countTable_default_file <- "../data/tara_processed_counts.csv"

sampleData <- read.csv(sampleData_default_file, row.names = 1)
covariate_name <- "Log10_Depth"
sample_covariate <- sampleData[, covariate_name]

X <- read.csv(countTable_default_file, row.names = 1)

D0 <- generic_dist(X, method = "jaccard",
                   min_row_sum = min_row_sum,
                   min_row_prevalence = min_row_prevalence)
D <- D0
```

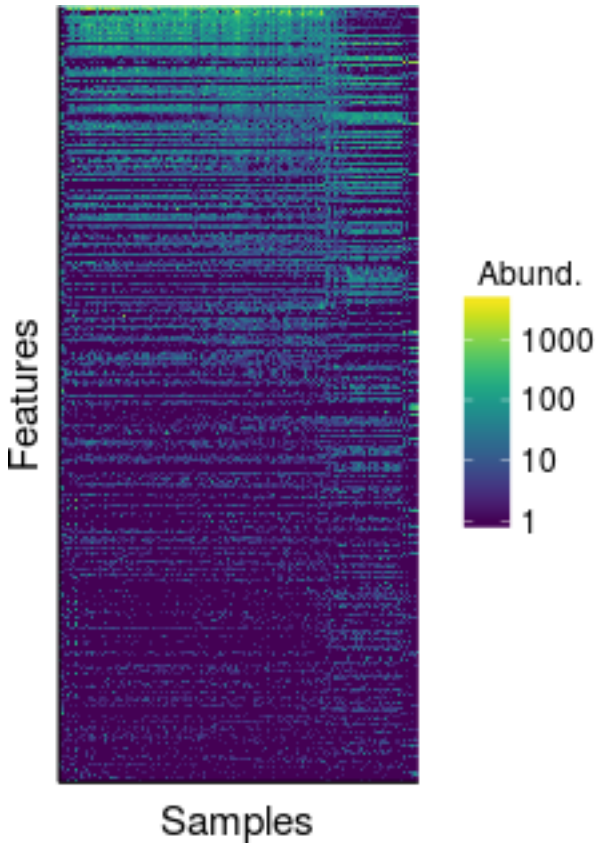
```
set.seed(1)
buds_seed <- sample.int(.Machine$integer.max, 1)
budsFit <- buds::fit_buds(D, K = K, method = "vb", hyperparams = hparams,
                          init_from = "random", seed = buds_seed,
                          tol_rel_obj = 0.005)
```

```
## ------------------------------------------------------------
## EXPERIMENTAL ALGORITHM:
```

```
##   This procedure has not been thoroughly tested and may be unstable
##   or buggy. The interface is subject to change.
## ------------------------------------------------------------
##
##
##
## Gradient evaluation took 0.005841 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 58.41 seconds.
## Adjust your expectations accordingly!
##
##
## Begin eta adaptation.
## Iteration:   1 / 250 [  0%]  (Adaptation)
## Iteration:  50 / 250 [ 20%]  (Adaptation)
## Iteration: 100 / 250 [ 40%]  (Adaptation)
## Iteration: 150 / 250 [ 60%]  (Adaptation)
## Iteration: 200 / 250 [ 80%]  (Adaptation)
## Success! Found best value [eta = 1] earlier than expected.
##
## Begin stochastic gradient ascent.
##   iter       ELBO   delta_ELBO_mean   delta_ELBO_med   notes
##    100     -8e+04             1.000            1.000
##    200     -2e+04             1.652            2.304
##    300      3e+03             4.322            2.304
##    400      6e+03             3.377            2.304
##    500      9e+03             2.765            1.000
##    600      1e+04             2.328            1.000
##    700      1e+04             2.011            0.542
##    800      1e+04             1.761            0.542
##    900      1e+04             1.569            0.315
##   1000      1e+04             1.414            0.315
##   1100      1e+04             1.315            0.147   MAY BE DIVERGING... INSPECT ELBO
##   1200      1e+04             1.085            0.107   MAY BE DIVERGING... INSPECT ELBO
##   1300      1e+04             0.119            0.027
##   1400      1e+04             0.065            0.022
##   1500      1e+04             0.034            0.013
##   1600      1e+04             0.020            0.012
##   1700      1e+04             0.010            0.009
##   1800      1e+04             0.010            0.009
##   1900      1e+04             0.008            0.006
##   2000      1e+04             0.006            0.005
##   2100      1e+04             0.005            0.004   MEAN ELBO CONVERGED   MEDIAN ELBO CONVERGED
##
## Drawing a sample of size 1000 from the approximate posterior...
## COMPLETED.
```

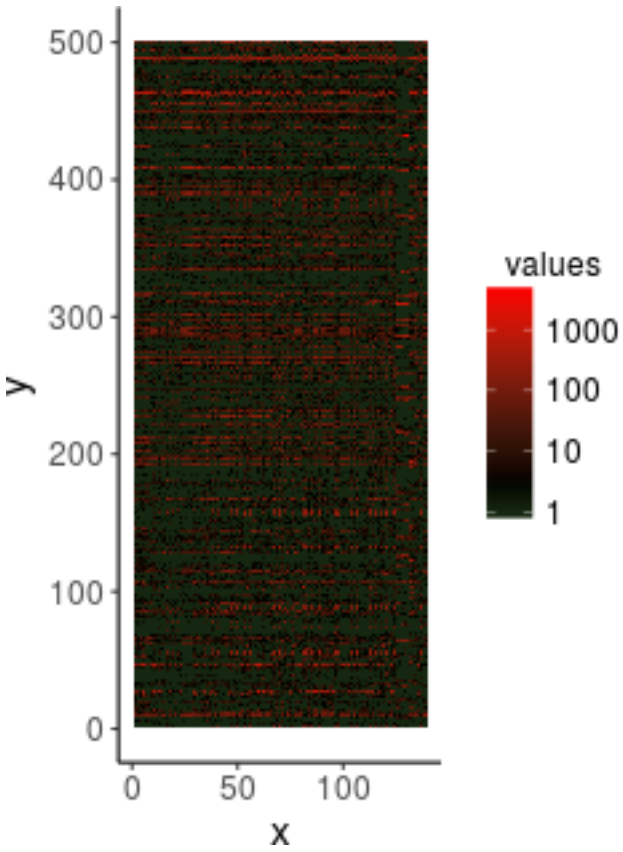
```
budsParams <- (rstan::extract(budsFit$fit_buds))
tau_df <- get_tau_df(budsParams, prob = 0.95)
```

```
set.seed(1)
idx <- sample(1:nrow(X), nfeatures)
(plt <- plot_ordered_matrix(X, tau_df$tau,
                            log_trans = TRUE,
                            keep_fatures = idx) +
    coord_fixed(0.6))
```

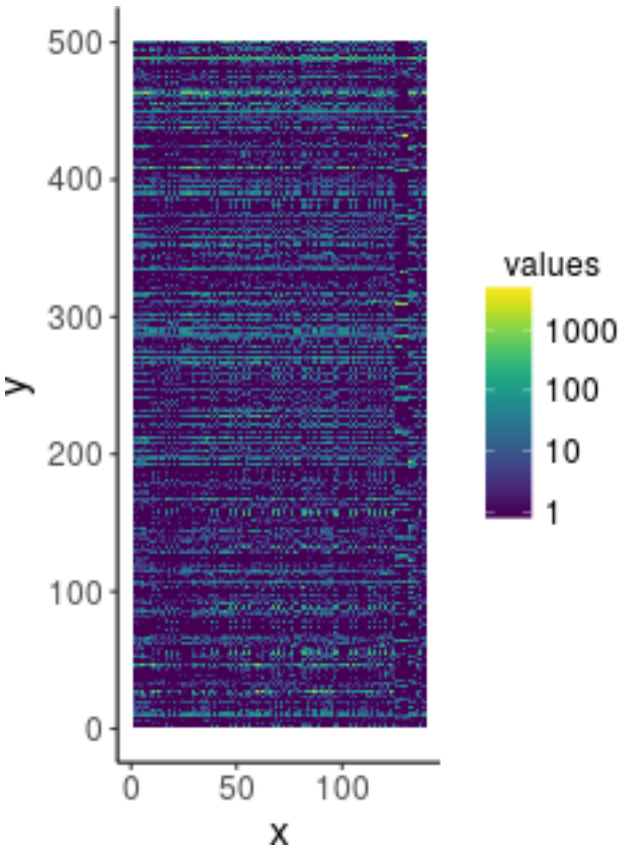



```
Xsmall <- as.matrix(X[idx, ])
NeatMap::heatmap1(Xsmall + 1) + coord_fixed(0.65) +
  scale_fill_gradient2(low = "green", high = "red", mid = "black",
                       midpoint = mean(log10(Xsmall + 1), na.rm = TRUE),
                       trans = "log10")
```

```
## Scale for 'fill' is already present. Adding another scale for 'fill',
## which will replace the existing scale.
```




```
Xsmall <- X[idx, ]
NeatMap::heatmap1(Xsmall+1) + coord_fixed(0.65) +
  scale_fill_viridis(trans = "log10")
```



DIABIMMUNE

```
# Load data files
countTable_default_file <- "../data/diabimmuneT1D_count_table_subset.csv"
sampleData_default_file <- "../data/diabimmuneT1D_sample_data_subset.csv"

sampleData <- read.csv(sampleData_default_file, row.names = 1)
covariate_name <- "Age_at_Collection"
sample_covariate <- sampleData[, covariate_name]

X <- read.csv(countTable_default_file, row.names = 1)
D0 <- generic_dist(X, method = "jaccard",
                   min_row_sum = min_row_sum,
                   min_row_prevalence = min_row_prevalence)
D <- transform_dist(D0, threshold = FALSE)
```

```
set.seed(1)
buds_seed <- sample.int(.Machine$integer.max, 1)
budsFit <- buds::fit_buds(D, K = K, method = "vb", hyperparams = hparams,
                          init_from = "random", seed = buds_seed,
                          tol_rel_obj = 0.005)
```

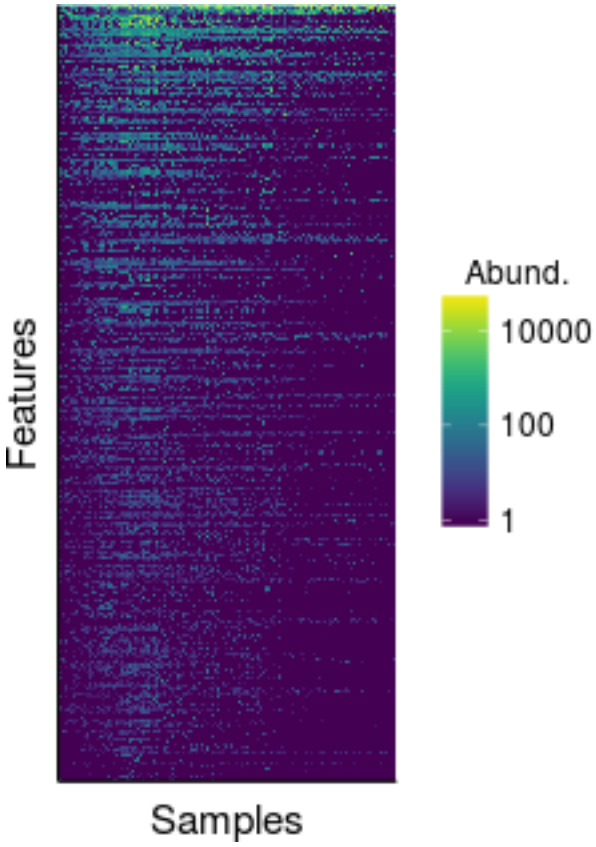
```
## ------------------------------------------------------------
## EXPERIMENTAL ALGORITHM:
##   This procedure has not been thoroughly tested and may be unstable
```

```
##   or buggy. The interface is subject to change.
## ------------------------------------------------------------
##
##
##
## Gradient evaluation took 0.030584 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 305.84 seconds.
## Adjust your expectations accordingly!
##
##
## Begin eta adaptation.
## Iteration:   1 / 250 [  0%]  (Adaptation)
## Iteration:  50 / 250 [ 20%]  (Adaptation)
## Iteration: 100 / 250 [ 40%]  (Adaptation)
## Iteration: 150 / 250 [ 60%]  (Adaptation)
## Success! Found best value [eta = 10] earlier than expected.
##
## Begin stochastic gradient ascent.
##   iter       ELBO   delta_ELBO_mean   delta_ELBO_med   notes
## ------------------------------------------------------------
## EXPERIMENTAL ALGORITHM:
##   This procedure has not been thoroughly tested and may be unstable
##   or buggy. The interface is subject to change.
## ------------------------------------------------------------
##
##
##
## Gradient evaluation took 0.039373 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 393.73 seconds.
## Adjust your expectations accordingly!
##
##
## Begin eta adaptation.
## Iteration:   1 / 250 [  0%]  (Adaptation)
## Iteration:  50 / 250 [ 20%]  (Adaptation)
## Iteration: 100 / 250 [ 40%]  (Adaptation)
## Iteration: 150 / 250 [ 60%]  (Adaptation)
## Iteration: 200 / 250 [ 80%]  (Adaptation)
## Success! Found best value [eta = 1] earlier than expected.
##
## Begin stochastic gradient ascent.
##   iter       ELBO   delta_ELBO_mean   delta_ELBO_med   notes
##    100     -2e+05             1.000            1.000
##    200     -9e+03             9.876           18.751
##    300      4e+03             7.585            3.004
##    400      1e+04             5.856            3.004
##    500      3e+04             4.797            1.000
##    600      4e+04             4.032            1.000
##    700      4e+04             3.469            0.668
##    800      4e+04             3.041            0.668
##    900      5e+04             2.707            0.564
##   1000      5e+04             2.442            0.564
##   1100      5e+04             2.342            0.205   MAY BE DIVERGING... INSPECT ELBO
##   1200      5e+04             0.469            0.088
```

```
##   1300      5e+04             0.171            0.051
##   1400      5e+04             0.108            0.051
##   1500      5e+04             0.052            0.042
##   1600      5e+04             0.033            0.035
##   1700      5e+04             0.025            0.026
##   1800      5e+04             0.020            0.021
##   1900      5e+04             0.017            0.010
##   2000      5e+04             0.012            0.008
##   2100      5e+04             0.014            0.010
##   2200      5e+04             0.012            0.008
##   2300      5e+04             0.010            0.006
##   2400      6e+04             0.008            0.006
##   2500      6e+04             0.008            0.006
##   2600      6e+04             0.007            0.006
##   2700      6e+04             0.006            0.005   MEDIAN ELBO CONVERGED
##
## Drawing a sample of size 1000 from the approximate posterior...
## COMPLETED.
```

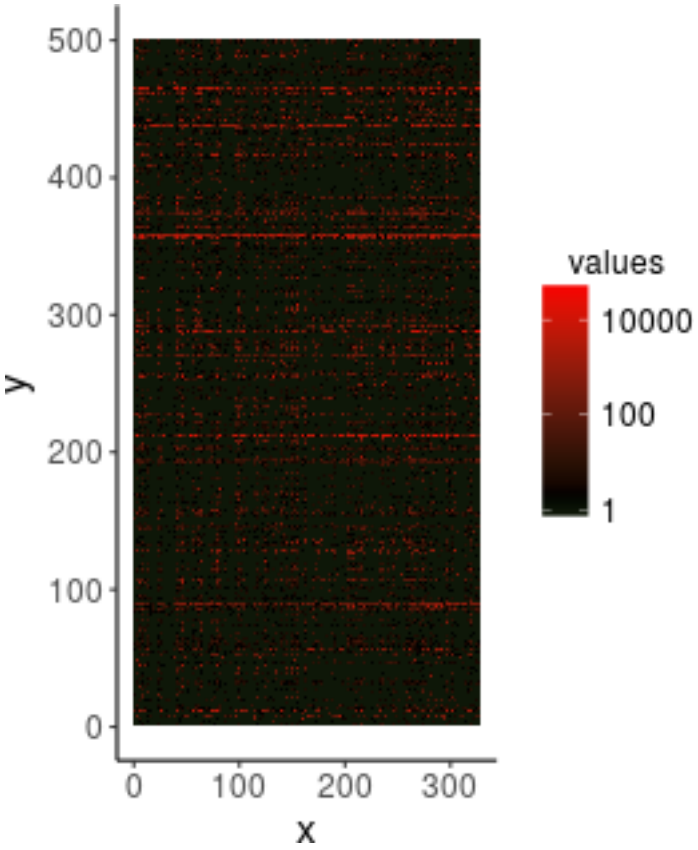
```
budsParams <- (rstan::extract(budsFit$fit_buds))
tau_df <- get_tau_df(budsParams, prob = 0.95)
```

```
set.seed(1)
idx <- sample(1:nrow(X), nfeatures)
(plt <- plot_ordered_matrix(X, tau_df$tau,
                            log_trans = TRUE,
                            keep_fatures = idx) +
    coord_fixed(1.5))
```

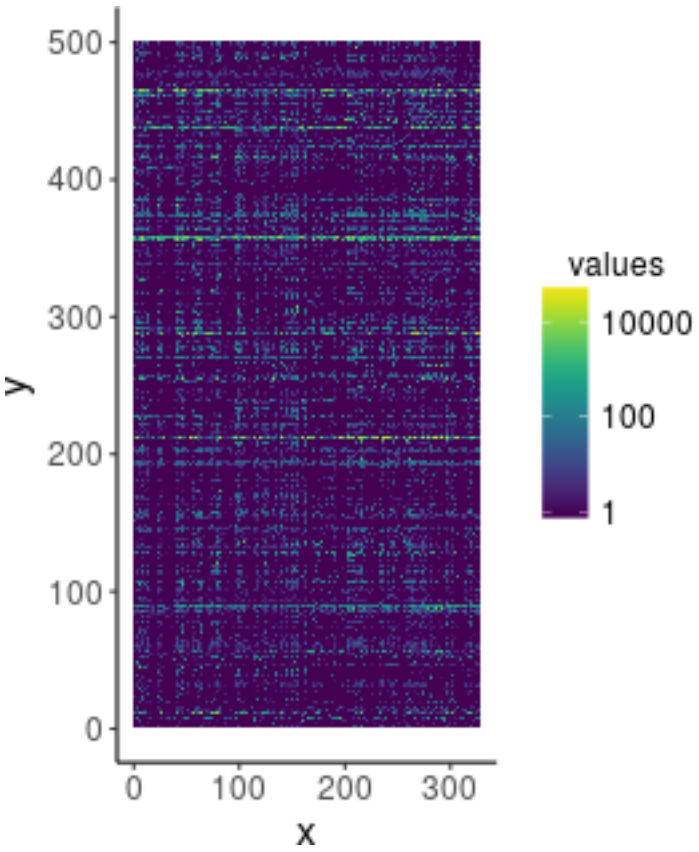


```
Xsmall <- as.matrix(X[idx, ])
NeatMap::heatmap1(Xsmall + 1) + coord_fixed(1.3) +
  scale_fill_gradient2(low = "green", high = "red", mid = "black",
                       midpoint = mean(log10(Xsmall + 1), na.rm = TRUE),
                       trans = "log10")
```

```
## Scale for 'fill' is already present. Adding another scale for 'fill',
## which will replace the existing scale.
```




```
Xsmall <- X[idx, ]
NeatMap::heatmap1(Xsmall+1) + coord_fixed(1.3) +
  scale_fill_viridis(trans = "log10")
```



Roll Call

```
# Load data files
countTable_default_file <- "../data/114_US_Senate_binVotes.csv"
sampleData_default_file <- "../data/114_US_Senate_legisData.csv"

sampleData <- read.csv(sampleData_default_file, row.names = 1)
covariate_name <- "party"
sample_covariate <- sampleData[, covariate_name]

X <- read.csv(countTable_default_file, row.names = 1)
D0 <- generic_dist(X, method = "exp manhattan", log_trans = FALSE)
D <- D0
```

```
set.seed(1)
buds_seed <- sample.int(.Machine$integer.max, 1)
budsFit <- buds::fit_buds(D, K = K, method = "vb", hyperparams = hparams,
                          init_from = "random", seed = buds_seed,
                          tol_rel_obj = 0.005)
```

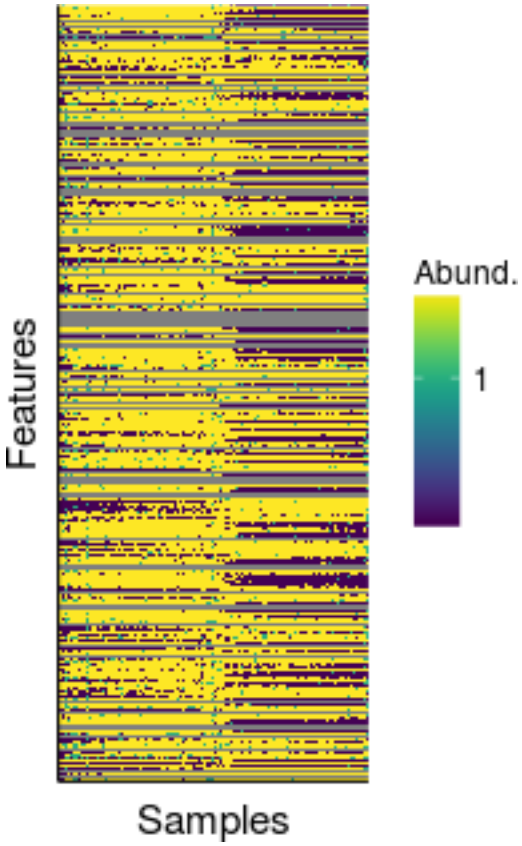
```
## ------------------------------------------------------------
## EXPERIMENTAL ALGORITHM:
##   This procedure has not been thoroughly tested and may be unstable
##   or buggy. The interface is subject to change.
## ------------------------------------------------------------
```

```
##
##
##
## Gradient evaluation took 0.002954 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 29.54 seconds.
## Adjust your expectations accordingly!
##
##
## Begin eta adaptation.
## Iteration:   1 / 250 [  0%]  (Adaptation)
## Iteration:  50 / 250 [ 20%]  (Adaptation)
## Iteration: 100 / 250 [ 40%]  (Adaptation)
## Success! Found best value [eta = 100] earlier than expected.
##
## Begin stochastic gradient ascent.
##   iter       ELBO   delta_ELBO_mean   delta_ELBO_med   notes
## ------------------------------------------------------------
## EXPERIMENTAL ALGORITHM:
##   This procedure has not been thoroughly tested and may be unstable
##   or buggy. The interface is subject to change.
## ------------------------------------------------------------
##
##
##
## Gradient evaluation took 0.00358 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 35.8 seconds.
## Adjust your expectations accordingly!
##
##
## Begin eta adaptation.
## Iteration:   1 / 250 [  0%]  (Adaptation)
## Iteration:  50 / 250 [ 20%]  (Adaptation)
## Iteration: 100 / 250 [ 40%]  (Adaptation)
## Iteration: 150 / 250 [ 60%]  (Adaptation)
## Iteration: 200 / 250 [ 80%]  (Adaptation)
## Success! Found best value [eta = 1] earlier than expected.
##
## Begin stochastic gradient ascent.
##   iter       ELBO   delta_ELBO_mean   delta_ELBO_med   notes
##    100     -2e+04             1.000            1.000
##    200     -6e+02            16.736           32.473
##    300      6e+03            11.520            1.086
##    400      8e+03             8.696            1.086
##    500      8e+03             6.961            1.000
##    600      9e+03             5.809            1.000
##    700      9e+03             4.979            0.224
##    800      9e+03             4.357            0.224
##    900      9e+03             3.874            0.045
##   1000      9e+03             3.487            0.045
##   1100      9e+03             3.387            0.023   MAY BE DIVERGING... INSPECT ELBO
##   1200      9e+03             0.140            0.006
##   1300      9e+03             0.031            0.005
##   1400      9e+03             0.009            0.004   MEDIAN ELBO CONVERGED
##
```

```
## Drawing a sample of size 1000 from the approximate posterior...
## COMPLETED.
```

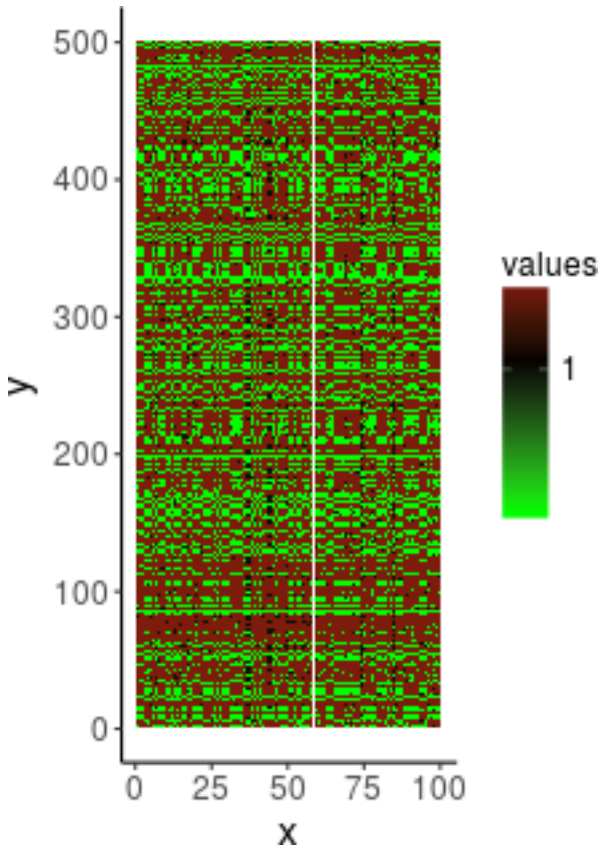
```
budsParams <- (rstan::extract(budsFit$fit_buds))
tau_df <- get_tau_df(budsParams, prob = 0.95)
```

```
set.seed(1)
idx <- sample(1:nrow(X), nfeatures)
(plt <- plot_ordered_matrix(X, tau_df$tau,
                            log_trans = TRUE,
                            keep_fatures = idx) +
    coord_fixed(0.5))
```

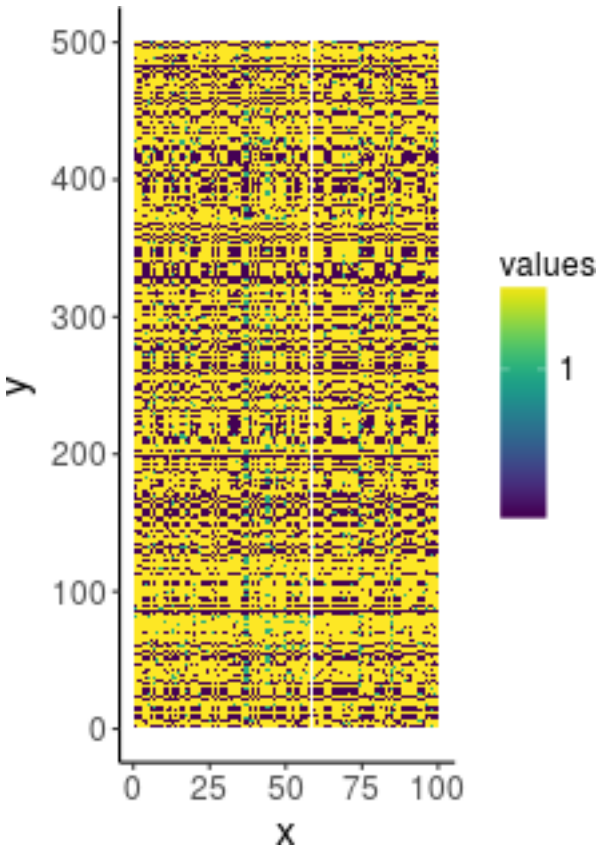



```
Xsmall <- as.matrix(X[idx, ])
NeatMap::heatmap1(Xsmall + 1) + coord_fixed(0.45) +
  scale_fill_gradient2(low = "green", high = "red", mid = "black",
                       midpoint = mean(log10(Xsmall + 1), na.rm = TRUE),
                       trans = "log10")
```

```
## Scale for 'fill' is already present. Adding another scale for 'fill',
## which will replace the existing scale.
```




```
Xsmall <- X[idx, ]
NeatMap::heatmap1(Xsmall+1) + coord_fixed(0.45) +
  scale_fill_viridis(trans = "log10")
```


500
400
300
200
100
0
y
0
25
50
75
100
x
values
1