Package 'cellmig'

January 12, 2026

Type Package

Title Uncertainty-aware quantitative analysis of high-throughput live cell migration data

Version 1.0.0

Description High-throughput cell imaging facilitates the analysis of cell migration across many wells treated under different biological conditions. These workflows generate considerable technical noise and biological variability, and therefore technical and biological replicates are necessary, leading to large, hierarchically structured datasets, i.e., cells are nested within technical replicates that are nested within biological replicates. Current statistical analyses of such data usually ignore the hierarchical structure of the data and fail to explicitly quantify uncertainty arising from technical or biological variability. To address this gap, we present cellmig, an R package implementing Bayesian hierarchical models for migration analysis. cellmig quantifies condition-specific velocity changes (e.g., drug effects) while modeling nested data structures and technical artifacts. It further enables synthetic data generation for experimental design optimization.

License GPL-3 + file LICENSE

Depends R (>= 4.5.0)

Imports base, ggplot2, ggforce, ggtree, patchwork, ape, methods, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), reshape2, rstan (>= 2.18.1), rstantools (>= 2.4.0), stats, utils, scales

Suggests BiocStyle, knitr, testthat

Encoding UTF-8

NeedsCompilation yes

biocViews SingleCell, CellBiology, Bayesian, ExperimentalDesign, Software, BatchEffect, Regression, Clustering

BugReports https://github.com/snaketron/cellmig/issues

URL https://github.com/snaketron/cellmig

RoxygenNote 7.3.1

VignetteBuilder knitr

Biarch true

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

SystemRequirements GNU make

git_url https://git.bioconductor.org/packages/cellmig

git_branch RELEASE_3_22

git_last_commit e77030e

git_last_commit_date 2025-10-29

Repository Bioconductor 3.22

Date/Publication 2026-01-12

Author Simo Kitanovski [aut, cre] (ORCID: <https://orcid.org/0000-0003-2909-5376>)

Maintainer Simo Kitanovski <simokitanovski@gmail.com>

Contents

`cellmig-package` *cellmig: quantifying cell migration with hierarchical Bayesian models*

Description

`cellmig` implements Bayesian hierarchical models for migration analysis and quantifies condition-specific (drug effects) effects on cell velocity while modeling nested data structures and technical artifacts, providing uncertainty-aware estimates through credible intervals.

Details

This package contains functions for modeling, clustering, and visualization of cell migration velocity from high-throughput migration assays.

Author(s)

Authors and maintainers:

- Simo Kitanovski <simokitanovski@gmail.com> (ORCID)

See Also

Useful links:

- https://github.com/snaketron/cellmig
- Report bugs at https://github.com/snaketron/cellmig/issues

Examples

```
# load package input data
data(d, package = "cellmig")
o <- cellmig(x = d,
             control = list(mcmc_warmup = 200,
                            mcmc_steps = 700,
                            mcmc_chains = 2,
                            mcmc_cores = 2,
                            mcmc_algorithm = "NUTS",
                            adapt_delta = 0.8,
                            max_treedepth = 10))
str(o)
```

`cellmig` *Model-based quantification of cell migration speed*

Description

The `cellmig` function estimates cell migration speed using a Bayesian model implemented in `rstan`. It takes a data.frame as input and allows users to configure the Markov Chain Monte Carlo (MCMC) sampling procedure via the `control` parameter. The function returns a list containing:

- `fit` - the fitted model as an `rstan` object.
- `data` - the input/processed input data
- `posteriors` - a summary of model parameters: means, medians, and 95% credible intervals.
- `control` - the list of input controls

Usage

```
cellmig(x, control = NULL)
```

Arguments

`x` A data.frame containing the following columns. Each row represents the features of a cell:

- well = character. Well ID (w1, w2, w3, etc.).
- plate = character. plate ID (p1, p2, p3, etc.). A plate contains multiple wells.
- compound = character. Compound name (c1, c2, c3, etc.)
- dose = character. Treatment dose (0, 1, 5, 10, low, mid, high, etc.)
- v = numeric. Observed cell migration speed.

- offset = binary (0 or 1). Indicates whether a treatment should be used for batch correction across plates. Default: 0 (no correction). Set to 1 for specific treatment groups used as offsets, ensuring they appear on each plate.

`control` A list configuring the MCMC sampling algorithm, and parameter priors with the following default values:

- mcmc_warmup = 500 (Number of warmup iterations).
- mcmc_steps = 1500 (Number of sampling iterations).
- mcmc_chains = 4 (Number of Markov chains).
- mcmc_cores = 1 (Number of CPU cores used for sampling).
- mcmc_algorithm = "NUTS" (MCMC algorithm).
- adapt_delta = 0.8 (Target acceptance probability for NUTS sampler).
- max_treedepth = 10 (Maximum depth of tree exploration in NUTS sampler).
- prior_alpha_p_M = 2, prior_alpha_p_SD = 1 (prior on alpha_p: normal distribution mean (M) and standard deviation (SD))
- prior_sigma_bio_M = 0, prior_sigma_bio_SD = 0.5 (prior on sigma_bio: normal distribution mean (M) and standard deviation (SD))
- prior_sigma_tech_M = 0, prior_sigma_tech_SD = 0.5 (prior on sigma_tech: normal distribution mean (M) and standard deviation (SD))
- prior_kappa_mu_M = 2, prior_kappa_mu_SD = 1 (prior on kappa_mu: normal distribution mean (M) and standard deviation (SD))
- prior_kappa_sigma_M = 0, prior_kappa_sigma_SD = 1 (prior on kappa_sigma: normal distribution mean (M) and standard deviation (SD))
- prior_mu_group_M = 0, prior_mu_group_SD = 1 (prior on mu_group: normal distribution mean (M) and standard deviation (SD))

Value

A list containing:

- `fit` = The fitted model as an rstan object.
- `data` = The raw and processed input data.
- `posteriors` = A summary of model parameters, including means and 95% credible intervals.
  - alpha_p: batch effect on plate p
  - delta_t: overall treatment effects relative to the selected control
  - delta_pt: plate-specific treatment effects relative to the selected control
  - well_mu: mean of cell velocity distribution per well
  - well_kappa: shape of cell velocity distribution per well
  - kappa_mu, kappa_sigma: mean/standard deviation parameters of the population of well_kappas
  - sigma_bio: variability between biological replicates
  - sigma_tech: variability between technical replicates
  - yhat: posterior predictions
- `control` = The list of input controls

Examples

```
data(d, package = "cellmig")
o <- cellmig(x = d,
             control = list(mcmc_warmup = 200,
                            mcmc_steps = 700,
                            mcmc_chains = 2,
                            mcmc_cores = 2,
                            mcmc_algorithm = "NUTS",
                            adapt_delta = 0.8,
                            max_treedepth = 10))
str(o)
```

---

d *Example dataset* d

Description

This dataset d contains simulated cell migration speed data from an imaginery experiment. It includes migration speed values (column v) for different samples (column sample) treated with chemical compounds (column compound) administered at a dost (column dose) on experimental plate with identifier (column plate). The dataset d_mini contains a subset of d with only four of the compounds (C1 to C4) and three doses (D3, D5, and D5).

Usage

```
data("d", package = "cellmig")
```

Format

A data frame with 6,300 (cells) observations with the following features.

v a numeric vector with cell migration speeds

well a character vector

compound a character vector

dose a character vector

plate a character vector

offset indicator: 1 = control, 0 = non-control. Used for batch correction

Details

The code used to generate this data is in inst/script/sim_s.R

Source

The code used to generate this data is in inst/script/sim_s.R

References

The code used to generate this data is in inst/script/sim_s.R

Examples

```
data(d, package = "cellmig")
data(d_mini, package = "cellmig")
str(d)
str(d_mini)
```

---

gen_full *Simulating data from fully generative Bayesian model*

Description

Simulates cell velocities based on a hierarchical model using a Stan model, where some of the model parameters are drawn from their prior distributions.

Usage

```
gen_full(control = list(N_biorep = 3,
                        N_techrep = 3,
                        N_cell = 50,
                        N_group = 5,
                        prior_alpha_p_M = 1.7,
                        prior_alpha_p_SD = 1,
                        prior_kappa_mu_M = 1.7,
                        prior_kappa_mu_SD = 1,
                        prior_kappa_sigma_M = 0,
                        prior_kappa_sigma_SD = 1,
                        prior_sigma_bio_M = 0,
                        prior_sigma_bio_SD = 1,
                        prior_sigma_tech_M = 0,
                        prior_sigma_tech_SD = 1,
                        prior_mu_group_M = 0,
                        prior_mu_group_SD = 1))
```

Arguments

`control` A list configuring the MCMC sampling algorithm, and parameter priors with the following default values:

- N_biorep: Number of plates (biological replicates) in the simulation.
- N_techrep: Number of technical replicates: wells on a plate treated with the same treatment.
- N_cell: Number of cells per well.
- prior_alpha_p_M, prior_alpha_p_SD: Mean (M) and standard deviation (SD) of a normal distribution describing plate-specific batch effects (on log-scale).
- prior_kappa_mu_M, prior_kappa_mu_SD: Mean (M) and standard deviation (SD) of a normal distribution describing the mean of the population of well-specific shape (kappa) parameters.
- prior_kappa_sigma_M, prior_kappa_sigma_SD: Mean (M) and standard deviation (SD) of a normal distribution describing the standard deviation of the population of well-specific shape (kappa) parameters.

- prior_sigma_bio_M, prior_sigma_bio_SD: Mean (M) and standard deviation (SD) of a normal distribution describing the variability of overall treatment effects between biological replicates.
- prior_sigma_tech_M, prior_sigma_tech_SD: Mean (M) and standard deviation (SD) of a normal distribution describing the variability of treatment effects between technical replicates.
- prior_mu_group_M, prior_mu_group_SD: Mean (M) and standard deviation (SD) of a normal distribution describing the population of overall treatment effects.

Details

This function constructs a hierarchical dataset by simulating values using a partially generative Stan model. It generates metadata for each well and simulates data using the 'sampling' function from the 'rstan' package.

Value

A data.frame containing cells as rows with:

- v = Simulated cell velocity.
- well_id = Unique well identifier.
- group_id = Experimental group identifier.
- plate_id = Biological replicate (plate) identifier.
- trep_id = Technical replicate identifier.

Examples

```
f <- gen_full(control = list(N_biorep = 3,
                             N_techrep = 3,
                             N_cell = 50,
                             N_group = 5,
                             prior_alpha_p_M = 1.7,
                             prior_alpha_p_SD = 1.0,
                             prior_kappa_mu_M = 1.7,
                             prior_kappa_mu_SD = 1.0,
                             prior_kappa_sigma_M = 0,
                             prior_kappa_sigma_SD = 1.0,
                             prior_sigma_bio_M = 0.0,
                             prior_sigma_bio_SD = 1.0,
                             prior_sigma_tech_M = 0.0,
                             prior_sigma_tech_SD = 1.0,
                             prior_mu_group_M = 0.0,
                             prior_mu_group_SD = 1.0))
str(f)
```

gen_partial *Simulating data from partially generative Bayesian model*

Description

Simulates cell velocities based on a hierarchical model using a Stan model, where some of the model parameter values are defined (fixed) by the user, while the rest are drawn from the prior distributions.

Usage

```
gen_partial(control = list(N_biorep = 3,
                           N_techrep = 3,
                           N_cell = 50,
                           delta,
                           sigma_bio = 0.1,
                           sigma_tech = 0.05,
                           offset = 1,
                           prior_alpha_p_M = 1.7,
                           prior_alpha_p_SD = 0.5,
                           prior_kappa_mu_M = 1.7,
                           prior_kappa_mu_SD = 0.5,
                           prior_kappa_sigma_M = 0,
                           prior_kappa_sigma_SD = 0.3))
```

Arguments

`control` A list configuring the MCMC sampling algorithm, and parameter priors with the following default values:

- N_biorep: Number of plates (biological replicates) in the simulation.
- N_techrep: Number of technical replicates: wells on a plate treated with the same treatment.
- N_cell: Number of cells per well.
- delta: Effects (on log-scale) on cell velocity for each treatment.
- sigma_bio: Variability between biological replicates.
- sigma_tech: Variability between technical replicates.
- offset: Index of the control treatment (which treatment should be used for batch correction between plates).
- prior_alpha_p_M, prior_alpha_p_SD: Mean (M) and standard deviation (SD) of a normal distribution describing plate-specific batch effects (on log-scale).
- prior_kappa_mu_M, prior_kappa_mu_SD: Mean (M) and standard deviation (SD) of a normal distribution describing the mean of the population of well-specific shape (kappa) parameters.
- prior_kappa_sigma_M, prior_kappa_sigma_SD: Mean (M) and standard deviation (SD) of a normal distribution describing the standard deviation of the population of well-specific shape (kappa) parameters.

Details

This function constructs a hierarchical dataset by simulating values using a partially generative Stan model. It generates metadata for each well and simulates data using the 'sampling' function from the 'rstan' package.

Value

A data frame containing:

- iteration = Simulation iteration index.
- well_id = Unique well identifier.
- y = Simulated response value.
- group_id = Experimental group identifier.
- plate_id = Plate identifier.

Examples

```
g <- gen_partial(control = list(N_biorep = 3,
                                N_techrep = 3,
                                N_cell = 50,
                                delta=c(0, -0.4, -0.2, -0.1, 0, 0.1, 0.2, 0.4),
                                sigma_bio = 0.2,
                                sigma_tech = 0.05,
                                offset = 1,
                                prior_alpha_p_M = 1.7,
                                prior_alpha_p_SD = 0.5,
                                prior_kappa_mu_M = 1.7,
                                prior_kappa_mu_SD = 0.5,
                                prior_kappa_sigma_M = 0,
                                prior_kappa_sigma_SD = 0.3))
str(g)
```

---

get_dose_response_profile

Visualization of dose-response profiles

Description

The function takes as its main input (x) the output of the cellmig function. Users can select a subset of treatment groups (compounds + dose combinations) using groups. To construct the hierarchical dendrogram, one has to select a distance metric, hc_dist, (e.g. hc_dist = "euclidean"); and a linkage function, hc_link, (e.g. hc_link = "average").

Usage

```
get_dose_response_profile(x,
                          hc_link = "average",
                          hc_dist = "euclidean",
                          groups,
                          B = 1000)
```

Arguments

x	An object generated by cellmig
hc_link	hierarchical clustering linkage function
hc_dist	hierarchical clustering distance metric
groups	use only selected treatment groups
B	number of samples to draw from posterior

Details

The function get_dose_response_profile visualizes compound effect profiles, identifying compounds with similar effects on cell migration across different doses. These compounds are clustered together in the hierarchical dendrogram.

Value

A patchwork plot containing:

- **A**: Hierarchical dendrogram comparing compound effect curves.
- **B**: Mean effects of compounds and doses on cell migration with 95% credible intervals.
- **C**: Mean effects of compounds and doses on cell migration in each replicate with 95% credible intervals.

Examples

```
data(d, package = "cellmig")
o <- cellmig(x = d,
             control = list(mcmc_warmup = 200,
                            mcmc_steps = 500,
                            mcmc_chains = 2,
                            mcmc_cores = 2,
                            mcmc_algorithm = "NUTS",
                            adapt_delta = 0.8,
                            max_treedepth = 10))

p <- get_dose_response_profile(x = o,
                               hc_link = "average",
                               hc_dist = "euclidean",
                               B = 100)
p
```

---

get_groups *Extract group labels*

Description

This function extracts treatment group metadata from an object generated by cellmig.

Usage

```
get_groups(x)
```

Arguments

x	An object generated by cellmig, containing treatment group names and IDs, compound names and dose.

Details

The function retrieves group metadata such as group identifiers, names, compound information, and dose levels.

Value

A data frame containing the following columns:

group_id	Unique identifier for each treatment group.
group	Name of the treatment group.
compound	Compound associated with the treatment group.
dose	Dose level of the compound.

See Also

cellmig, get_pairs, get_groups, get_violins

Examples

```
data(d_mini, package = "cellmig")
o <- cellmig(x = d_mini,
             control = list(mcmc_warmup = 200,
                            mcmc_steps = 500,
                            mcmc_chains = 2,
                            mcmc_cores = 2,
                            mcmc_algorithm = "NUTS",
                            adapt_delta = 0.8,
                            max_treedepth = 10))
u <- get_groups(x = o)
```

---

get_pairs *Compare the cell migration effects of treatment groups (cellvel results)*

Description

The function get_pairs compares the overall treatment group effects with each other, i.e. computes differences in their posterior distributions.

Usage

```
get_pairs(x, groups, exponentiate)
```

Arguments

x	an object generated by cellvel
groups	vector of treatment groups to select, see groups by calling get_groups
exponentiate	logical, should the effect be exponentiating turning log-fold-changes into more interpretable fold-changes

Details

The main input, `x`, of `get_pairs` is the output object from the function `cellmig`.

For a pair of treatment groups, `get_pairs` compares extracts the posterior distributions of their cell migration effects, and then compute an absolute difference between the posteriors. The result is another posterior distribution (ρ), with L95 and H95 as the lower and upper bounds of the 95% highest density interval of ρ.

π is estimated as: $\pi = 2 \cdot \max\left(\int_{-\infty}^{0} \rho, \int_{0}^{+\infty} \rho\right) - 1$.

For treatment groups, i and j, where $\rho < 0$ and the 95% HDIs of ρ lie mostly or completely below 0 (0 = null effect), we have strong evidence of lower cell migration effect in treatment i compared to j. On the other hand, for treatment groups, i and j, where ρ and the 95% HDI of ρ lie mostly or completely above 0, we have strong evidence of higher cell migration effect in treatment i compared to j. Distributions with the 95% HDIs more or less centered around 0 indicate that there is no evidence for a clear difference in the cell migration effects between the treatment groups. Note that unclear evidence is not equivalent to no change, because for a treatment group with $\rho \approx 0$ we may also have a wide 95% HDI, including possibilities for positive or negative change.

Value

A data.frame with the following data in each row (comparison):

- treatment group x
- treatment group y
- ρ_M = mean rho
- ρ_M, ρ_{L95}, ρ_{H95} = mean, lower bound and upper bound of the difference between the migration effects of treatments x and y
- π = probability of differential effects

Two heatmaps (ggplot2 objects) showing ρ (plot_rho) and π (plot_pi) between treatments (interpretation: rows vs. columns).

Examples

```
data(d_mini, package = "cellmig")
o <- cellmig(x = d_mini,
             control = list(mcmc_warmup = 200,
                            mcmc_steps = 500,
                            mcmc_chains = 2,
                            mcmc_cores = 2,
                            mcmc_algorithm = "NUTS",
                            adapt_delta = 0.8,
                            max_treedepth = 10))
u <- get_pairs(x = o, exponentiate = FALSE)
head(u)
```

`get_ppc_means` *Observed vs. predicted cell migration means in each well*

Description

The `get_ppc_means` function visualizes the comparison between observed and posterior predicted migration means. The function aggregates observed migration values by well, merges them with posterior predictive means, and generates a scatter plot with error bars representing the 95% HDI.

Usage

```
get_ppc_means(x)
```

Arguments

`x`	The function takes as its main input (`x`) the output of the `cellmig` function, containing posterior samples from a Bayesian model. The posterior predictive means are stored in `x$s$yhat`, while the observed migration values are extracted from `x$x$d`.

Details

The function follows these steps:

1. Compute mean observed (scaled) cell migration in each well
2. Compute mean predicted (scaled) cell migration in each well + 95% credible intervals
3. Generate a `ggplot2` visualization with:
   - A dashed diagonal indicating perfect agreement.
   - Scatter points representing observed vs. predicted means.
   - Error bars (dark gray) showing the 95% HDI for predictions.

Value

A `ggplot2` object displaying:

- A scatter plot comparing observed vs. predicted migration means.
- 95% highest density interval (HDI) error bars for the predicted means.
- A dashed reference line (`y = x`) for ideal agreement.

Examples

```
# Load example dataset
data(d_mini, package = "cellmig")

# Run cellmig with MCMC sampling
o <- cellmig(x = d_mini,
             control = list(mcmc_warmup = 200,
                            mcmc_steps = 500,
                            mcmc_chains = 2,
                            mcmc_cores = 2,
                            mcmc_algorithm = "NUTS",
```

```
                          adapt_delta = 0.8,
                          max_treedepth = 10))

# Generate PPC means plot
p <- get_ppc_means(x = o)
print(p)
```

`get_ppc_violins` *Posterior Predictive Checks (PPC) Visualization*

Description

The `get_ppc_violins` function visualizes posterior predictive checks (PPC) by comparing observed data with posterior predictive distributions. The function extracts simulated data from the fitted Stan model and overlays them with observed values, grouped by compound and plate.

Usage

```
get_ppc_violins(x, wrap = FALSE, ncol = 4)
```

Arguments

`x`	The function takes as its main input (`x`) the output of the `cellmig` function, containing posterior samples from a Bayesian model, specifically the output of a fitted Stan model with posterior predictive samples stored in `x$f`.
`wrap`	logical, if wrap = FALSE (default) we will use facet_grid, and if wrap = TRUE we will use facet_wrap to split data into compound x plate pairs.
`ncol`	In case wrap = TRUE, how many columns should the plot have?

Details

The function extracts posterior predictive samples using `rstan::extract`, reshapes them into long format using `reshape2::melt`, and merges them with metadata from the input object. It then generates a `ggplot2` visualization, where:

- Violin plots (dashed red) represent posterior predictive distributions.
- Overlaid sina plots (black) represent observed data points.
- Facets are arranged by compound and plate for better comparison.

Value

A `ggplot2` object displaying:

- Violin plots of posterior predictive distributions for each compound and plate.
- Overlaid scatter plots (sina plot) of observed values.

Examples

```
data(d_mini, package = "cellmig")
o <- cellmig(x = d_mini,
             control = list(mcmc_warmup = 200,
                            mcmc_steps = 500,
                            mcmc_chains = 2,
                            mcmc_cores = 2,
                            mcmc_algorithm = "NUTS",
                            adapt_delta = 0.8,
                            max_treedepth = 10))

p <- get_ppc_violins(x = o, wrap = TRUE, ncol = 4)
print(p)
```

---

get_treatment_profile *Visualization of treatment effect profiles*

Description

The function takes as its main input (x) the output of the cellmig function. Users can select a subset of treatment groups (compounds + dose combinations) using groups. To construct the hierarchical dendrogram, one has to select a distance metric, hc_dist, (e.g. hc_dist = "euclidean"); and a linkage function, hc_link, (e.g. hc_link = "average").

Usage

```
get_treatment_profile(x,
                      hc_link = "average",
                      hc_dist = "euclidean",
                      groups,
                      B = 1000)
```

Arguments

x	An object generated by cellmig
hc_link	hierarchical clustering linkage function
hc_dist	hierarchical clustering distance metric
groups	use only selected treatment groups
B	number of samples to draw from posterior

Details

The function get_treatment_profile visualizes treatment group effect profiles: these treatment groups are clustered together in a hierarchical dendrogram.

Value

Hierarchical dendrogram comparing treatment group effects.

Examples

```
data(d, package = "cellmig")
o <- cellmig(x = d,
             control = list(mcmc_warmup = 200,
                            mcmc_steps = 500,
                            mcmc_chains = 2,
                            mcmc_cores = 2,
                            mcmc_algorithm = "NUTS",
                            adapt_delta = 0.8,
                            max_treedepth = 10))

p <- get_treatment_profile(x = o,
                           hc_link = "average",
                           hc_dist = "euclidean",
                           B = 100)
p
```

---

get_violins *Generate violin plots for treatment group comparisons*

Description

This function generates violin plots for comparing posterior distributions of treatment group effects on cell migration.

Usage

```
get_violins(x, from_groups, to_group, exponentiate)
```

Arguments

x	An object generated by cellmig.
from_groups	A vector of group names to compare against the target group.
to_group	A single group name serving as the reference for comparisons.
exponentiate	logical, should the effect be exponentiating turning log-fold-changes into more interpretable fold-changes

Details

The function extracts posterior samples of treatment group-level means and computes differences between specified groups. It generates a violin plot illustrating these differences and returns both the computed data and the plot object.

Value

A list with two elements:

ds	A data frame containing computed differences, treatment group identifiers, and associated statistics.
plot	A ggplot2 object representing the violin plot of the computed treatment differences.

See Also

cellmig, get_pairs, get_groups

Examples

```
data(d_mini, package = "cellmig")
o <- cellmig(x = d_mini,
             control = list(mcmc_warmup = 200,
                            mcmc_steps = 500,
                            mcmc_chains = 2,
                            mcmc_cores = 2,
                            mcmc_algorithm = "NUTS",
                            adapt_delta = 0.8,
                            max_treedepth = 10))

str(get_groups(x = o))
u <- get_violins(x = o,
                 from_groups = c("C2|D3", "C2|D4"),
                 to_group = "C3|D3",
                 exponentiate = FALSE)
```

Index