# Package ‘cellmig’

February 1, 2026

**Type** Package

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

**Version** 1.1.6

**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** devel

**git_last_commit** 6d4aa19

**git_last_commit_date** 2026-01-29

**Repository** Bioconductor 3.23

**Date/Publication** 2026-02-01

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

**Maintainer** Simo Kitanovski <`simokitanovski@gmail.com`>

# Contents

cellmig-package . . . . . . . . . . 2
cellmig . . . . . . . . . . 3
d . . . . . . . . . . 5
gen_full . . . . . . . . . . 6
gen_partial . . . . . . . . . . 8
get_dose_response_profile . . . . . . . . . . 10
get_groups . . . . . . . . . . 11
get_pairs . . . . . . . . . . 12
get_ppc_means . . . . . . . . . . 14
get_ppc_violins . . . . . . . . . . 15
get_violins . . . . . . . . . . 16

**Index** **18**

---

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

---

## Description

The `cellmig` package implements Bayesian hierarchical models for the analysis of cell migration speed. It quantifies condition-specific effects (e.g., drug treatments) on cell migration speed while explicitly modeling nested data structures and technical artifacts, and provides uncertainty-aware estimates via posterior credible intervals.

## Details

This package contains functions for modeling, clustering, and visualization of cell migration speed data derived 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 (in micrometers/min).
- offset = binary (0 or 1). Indicates whether a treatment should be used as control 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 the NUTS sampler.
- `max_treedepth` = `10`: maximum tree depth for the NUTS sampler.
- `prior_alpha_p_M` = `-0.5`, `prior_alpha_p_SD` = `1`: prior mean and standard deviation for `alpha_p`, representing the plate-specific mean (log-scale) cell speed.
- `prior_sigma_bio_M` = `0`, `prior_sigma_bio_SD` = `0.5`: prior for `sigma_bio`, representing variability between biological replicates.
- `prior_sigma_tech_M` = `0`, `prior_sigma_tech_SD` = `0.5`: prior for `sigma_tech`, representing variability between technical replicates.
- `prior_kappa_mu_M` = `1.5`, `prior_kappa_mu_SD` = `1`: prior for `kappa_mu`, the mean of the population distribution of Gamma shape parameters.
- `prior_kappa_sigma_M` = `0`, `prior_kappa_sigma_SD` = `1`: prior for `kappa_sigma`, the standard deviation of the population distribution of Gamma shape parameters.
- `prior_delta_t_M` = `0`, `prior_delta_t_SD` = `1`: prior for `delta_t`, representing overall treatment effects relative to a selected control.

## 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 speed distribution per well
  - well_kappa: shape of cell speed 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

The dataset `d` contains simulated cell migration speed data from an imaginary experiment. It includes cell migration speed measurements (column `v`) for individual cells measured in different wells, treated with chemical compounds (`compound`) at varying doses (`dose`) and measured on experimental plates (`plate`).

The dataset `d_mini` is a reduced subset of `d`, containing data for four compounds (C1–C4) and three doses.

## Usage

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

## Format

A data frame with cells (rows) having with the following features.

`v` Numeric. Cell migration speed (micrometers per minute).

`well` Character. Well identifier.

`compound` Character. Compound identifier.

`dose` Character. Treatment dose.

`plate` Character. Plate identifier.

`offset` Integer (0 or 1). Indicates control samples used for batch correction across plates (1 = control, 0 = non-control).

## Details

Both datasets were generated using a simulation script located at `inst/scripts/sim_s.R`.

## Source

Simulated data generated using `inst/scripts/sim_s.R`.

## References

Simulated data generated using `inst/scripts/sim_s.R`.

## Examples

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

---

`gen_full` *Simulate data from a fully generative hierarchical Bayesian model*

---

## Description

Simulates cell migration speed data from a hierarchical Bayesian model implemented in Stan, where 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 = -0.5,
                          prior_alpha_p_SD = 1,
                          prior_kappa_mu_M = 1.5,
                          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_delta_t_M = 0,
                          prior_delta_t_SD = 1))
```

## Arguments

| | |
|---|---|
| `control` | A named list specifying simulation settings and prior distributions. Default values are:<br>• `N_biorep`: Number of biological replicates (plates).<br>• `N_techrep`: Number of technical replicates (wells per plate receiving the same treatment).<br>• `N_cell`: Number of cells per well.<br>• `N_group`: Number of treatment groups.<br>• `prior_alpha_p_M`, `prior_alpha_p_SD`: Mean and standard deviation of the normal prior for plate-specific batch effects (log-scale).<br>• `prior_kappa_mu_M`, `prior_kappa_mu_SD`: Mean and standard deviation of the normal prior for the population mean of well-specific Gamma shape parameters.<br>• `prior_kappa_sigma_M`, `prior_kappa_sigma_SD`: Mean and standard deviation of the normal prior for the population standard deviation of well-specific Gamma shape parameters.<br>• `prior_sigma_bio_M`, `prior_sigma_bio_SD`: Mean and standard deviation of the normal prior describing variability in treatment effects between biological replicates.<br>• `prior_sigma_tech_M`, `prior_sigma_tech_SD`: Mean and standard deviation of the normal prior describing variability in treatment effects between technical replicates.<br>• `prior_delta_t_M`, `prior_delta_t_SD`: Mean and standard deviation of the normal prior for overall treatment effects. |

## Details

The function constructs a hierarchical synthetic dataset by simulating cell migration speed values from a fully generative Stan model. Metadata for plates, wells, treatment groups, and replicates are generated, and cell-level observations are simulated using the `sampling` function from the **rstan** package.

## Value

A data frame with one row per cell and the following columns:

`v` Simulated cell migration speed.

`well_id` Unique well identifier.

`group_id` Treatment 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 = -0.5,
                             prior_alpha_p_SD = 1.0,
                             prior_kappa_mu_M = 1.5,
                             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_delta_t_M = 0.0,
                             prior_delta_t_SD = 1.0))
str(f)
```

---

`gen_partial` *Simulate data from a partially generative hierarchical Bayesian model*

---

## Description

Simulates cell migration speed data from a hierarchical Bayesian model implemented in Stan, where some model parameters are fixed by the user and the remaining parameters are drawn from their 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 = -0.5,
                           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 named list specifying simulation settings, fixed parameter values, and prior distributions. Default values are: |

- `N_biorep`: Number of biological replicates (plates).
- `N_techrep`: Number of technical replicates (wells per plate receiving the same treatment).
- `N_cell`: Number of cells per well.
- `delta`: Numeric vector of treatment effects on cell migration speed (log-scale), one value per treatment group.
- `sigma_bio`: Variability in treatment effects between biological replicates.
- `sigma_tech`: Variability in treatment effects between technical replicates.
- `offset`: Index of the control treatment used for batch correction across plates.
- `prior_alpha_p_M`, `prior_alpha_p_SD`: Mean and standard deviation of the normal prior for plate-specific batch effects (log-scale).
- `prior_kappa_mu_M`, `prior_kappa_mu_SD`: Mean and standard deviation of the normal prior for the population mean (log-scale) of well-specific Gamma shape parameters.
- `prior_kappa_sigma_M`, `prior_kappa_sigma_SD`: Mean and standard deviation of the normal prior for the population standard deviation of well-specific Gamma shape parameters.

## Details

The function constructs a hierarchical synthetic dataset by simulating cell migration speed values from a partially generative Stan model. User-specified parameters (e.g., treatment effects and variability terms) are held fixed, while remaining parameters are sampled from their prior distributions. Cell-level observations are simulated using the `sampling` function from the **rstan** package.

## Value

A data frame with one row per simulated observation and the following columns:

| | |
|---|---|
| `iteration` | Simulation iteration index. |
| `well_id` | Unique well identifier. |
| `y` | Simulated cell migration speed. |
| `group_id` | Treatment group identifier. |
| `plate_id` | Biological replicate (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 = -0.5,
                                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,
                          exponentiate)
```

## Arguments

| | |
|---|---|
| `x` | An object generated by `cellmig` |
| `hc_link` | Character. Linkage method used for hierarchical clustering (e.g., `"average"`, `"complete"`, `"single"`). |
| `hc_dist` | Character. Distance metric used for hierarchical clustering (e.g., `"euclidean"`, `"manhattan"`). |
| `groups` | Optional character vector specifying a subset of treatment groups (compound–dose combinations) to include in the analysis. |
| `B` | Integer. Number of posterior samples drawn for visualization. |
| `exponentiate` | Logical. If `TRUE`, treatment effects are exponentiated, transforming log-scale effects into fold changes. |

## Details

The function `get_dose_response_profile` visualizes compound effect profiles, identifying compounds with similar effects on cell migration speed 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 speed with 95% credible intervals.
- **C**: Mean effects of compounds and doses on cell migration speed 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,
                               exponentiate = TRUE)
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` *Pairwise comparison of treatment effects on cell migration speed*

---

## Description

The function `get_pairs` performs pairwise comparisons of overall treatment effects by computing differences between their posterior distributions.

## Usage

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

## Arguments

| | |
|---|---|
| `x` | An object generated by `cellmig` |
| `groups` | Optional character vector specifying treatment groups to include. Available groups can be obtained using `get_groups`. |
| `exponentiate` | Logical. If `TRUE`, treatment effects are exponentiated, transforming log-scale effects into fold changes. |

## Details

The main input, `x`, is the output of the `cellmig` function. For each pair of treatment groups, `get_pairs` extracts the posterior distributions of their cell migration speed effects and computes the difference between them. This difference defines a posterior distribution denoted by $\rho$.

The 95% credible interval of $\rho$ is summarized by its lower ($\rho_{L95}$) and upper ($\rho_{H95}$) bounds.

The posterior probability of differential effects, $\pi$, is defined as:

$$\pi = 2 \cdot \max\left(\int_{-\infty}^{0} \rho, \int_{0}^{+\infty} \rho\right) - 1$$

If $\rho$ and its 95% credible interval lie mostly or entirely below zero, there is strong evidence for a lower cell migration speed effect in treatment group $i$ compared to $j$. Conversely, if $\rho$ and its 95% credible interval lie mostly or entirely above zero, there is strong evidence for a higher effect in treatment group $i$ compared to $j$.

If the 95% credible interval of $\rho$ overlaps zero substantially, the data do not provide clear evidence for a difference between treatment groups. Note that lack of clear evidence does not imply absence of an effect, as wide credible intervals may still include biologically relevant differences.

## Value

A list containing:

- `comparisons`: A data frame with one row per pairwise comparison, containing:

| | |
|---|---|
| `group_x` | First treatment group. |
| `group_y` | Second treatment group. |
| $\rho_M$ | Posterior mean of the difference in treatment effects. |
| $\rho_{L95}$ | Lower bound of the 95% credible interval. |
| $\rho_{H95}$ | Upper bound of the 95% credible interval. |
| $\pi$ | Posterior probability of differential effects. |

- `plot_rho`: A heatmap (ggplot2 object) of $\rho$ values for all treatment group pairs.
- `plot_pi`: A heatmap (ggplot2 object) of $\pi$ values for all treatment group pairs.

## 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_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

∗ **datasets**
    d, 5

cellmig, 3
cellmig-package, 2

d, 5
d_mini *(*d*)*, 5

gen_full, 6
gen_partial, 8
get_dose_response_profile, 10
get_groups, 11, *12*
get_pairs, 12
get_ppc_means, 14
get_ppc_violins, 15
get_violins, 16