Package 'Unico'

February 26, 2024

Type Package

Title Unified Cross-Omics Deconvolution

Version 0.1.0

Description UNIfied Cross-Omics deconvolution (Unico) deconvolves standard 2-dimensional bulk matrices of samples by features into a 3-dimensional tensors representing samples by features by cell types. Unico stands out as the first principled model-based deconvolution method that is theoretically justified for any heterogeneous genomic data. For more details see Chen and Rahmani et al. (2024) <doi:10.1101/2024.01.27.577588>.

License GPL-3

URL https://cozygene.github.io/Unico/

BugReports https://github.com/cozygene/Unico/issues

Depends R (>= 4.0.0)

Imports compositions, config, data.table, futile.logger, MASS, matrixcalc, matrixStats, mgcv, nloptr, parallel, pbapply, pracma, stats, testit, utils

Suggests egg, hexbin, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr, rmarkdown

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Author Zeyuan Chen [aut, cre], Elior Rahmani [aut]

Maintainer Zeyuan Chen <johnsonchen@cs.ucla.edu>

Repository CRAN

Date/Publication 2024-02-26 16:50:06 UTC

R topics documented:

`association_asymptotic` *Performs asymptotic statistical testing under no distribution assumption*

Description

Performs asymptotic statistical testing on (1) the marginal effect of each covariate in `C1` at source-specific level (2) non-source-specific effect for each covariate in `C2`. In the context of bulk genomic data containing a mixture of cell types, these correspond to the marginal effect of each covariate in `C1` (potentially including the phenotype of interest) at each cell type and tissue-level effect for each covariate in `C2`.

Usage

```
association_asymptotic(
  X,
  Unico.mdl,
  slot_name = "asymptotic",
  diag_only = FALSE,
  intercept = TRUE,
  X_max_stds = 2,
  Q_max_stds = Inf,
  V_min_qlt = 0.05,
  parallel = TRUE,
  num_cores = NULL,
  log_file = "Unico.log",
  verbose = FALSE,
  debug = FALSE
)
```

Arguments

`X`	An `m` by `n` matrix of measurements of `m` features for `n` observations. Each column in `X` is assumed to be a mixture of `k` sources. Note that `X` must include row names and column names and that NA values are currently not supported. `X` should not include features that are constant across all observations. Note that `X` must be the same `X` used to learn `Unico.mdl` (i.e. the original observed 2D mixture used to fit the model).

Unico.mdl	The entire set of model parameters estimated by Unico on the 2D mixture matrix (i.e. the list returned by applying function Unico to X).
slot_name	A string indicating the key for storing the results under Unico.mdl
diag_only	A logical value indicating whether to only use the estimated source-level variances (and thus ignoring the estimate covariance) for controlling the heterogeneity in the observed mixture. if set to FALSE, Unico instead estimates the observation- and feature-specific variance in the mixture by leveraging the entire k by k variance-covariance matrix.
intercept	A logical value indicating whether to fit the intercept term when performing the statistical testing.
X_max_stds	A non-negative numeric value indicating, for each feature, the portions of data that are considered as outliers due to the observed mixture value. Only samples whose observed mixture value fall within X_max_stds standard deviations from the mean will be used for the statistical testing of a given feature.
Q_max_stds	A non-negative numeric value indicating, for each feature, the portions of data that are considered as outliers due to the estimated mixture variance. Only samples whose estimated mixture variance fall within Q_max_stds standard deviations from the mean will be used for the statistical testing of a given feature.
V_min_qlt	A non-negative numeric value indicating, for each feature, the portions of data that are considered as outliers due to the estimated moment condition variance. This value should be between 0 and 1. Only samples whose estimated moment condition variance fall outside the bottom V_min_qlt quantile will be used for the statistical testing of a given feature.
parallel	A logical value indicating whether to use parallel computing (possible when using a multi-core machine).
num_cores	A numeric value indicating the number of cores to use (activated only if parallel == TRUE). If num_cores == NULL then all available cores except for one will be used.
log_file	A path to an output log file. Note that if the file log_file already exists then logs will be appended to the end of the file. Set log_file to NULL to prevent output from being saved into a file; note that if verbose == FALSE then no output file will be generated regardless of the value of log_file.
verbose	A logical value indicating whether to print logs.
debug	A logical value indicating whether to set the logger to a more detailed debug level; set debug to TRUE before reporting issues.

Details

Under no distribution assumption, we can solve for the following weighted least square problem, which is similar to the heteroskedastic regression view described in association_parametric.

$$\hat{\phi_j}^{\text{asym}} = \text{argmin}_{\phi_j}(x_j - S\phi_j)^T Q_j (x_j - S\phi_j)$$

S denotes the design matrix formed by stacking samples in the rows and dependent variables $\{\{w_i\}, \{w_i c_i^{(1)}\}, \{c_i^{(2)}\}\}$ on the columns. ϕ_j denotes the corresponding effect sizes on the dependent variables. Q_j denotes the feature-specific weighting scheme. Similar to the parametric

counterpart, $Q_j = \text{diag}(q_{1j}^2, ..., q_{nj}^2)$, where for each sample i, its corresponding weight will be the inverse of the estimated variance in the mixture: $q_{ij}^2 = \frac{1}{sum(w_i w_i^T \odot \hat{\Sigma}_j)}$. Marginal testing can thus be carried out on each dependent variable via the asymptotic distribution of the estimator $\hat{\phi}_j^{\text{asym}}$.

Value

An updated `Unico.mdl` object with the the following list of effect size and p-value estimates stored in an additional key specified by `slot_name`

`gammas_hat` An `m` by `k*p1` matrix of the estimated effects of the `p1` covariates in `C1` on each of the `m` features in `X`, where the first `p1` columns are the source-specific effects of the `p1` covariates on the first source, the following `p1` columns are the source-specific effects on the second source and so on.

`betas_hat` An `m` by `p2` matrix of the estimated effects of the `p2` covariates in `C2` on the mixture values of each of the `m` features in `X`.

`gammas_hat_pvals` An `m` by `k*p1` matrix of p-values for the estimates in `gammas_hat` (based on a T-test).

`betas_hat_pvals` An `m` by `p2` matrix of p-values for the estimates in `betas_hat` (based on a T-test).

`Q` An `m` by `n` matrix of weights used for controlling the heterogeneity of each observation at each feature (activated only if `debug == TRUE`).

`masks` An `m` by `n` matrix of logical values indicating whether observation participated in statistical testing at each feature (activated only if `debug == TRUE`).

`fphi_hat` An `m` by `n` matrix containing the entire estimated moment condition variance for each feature. Note that observations who are considered as outliers due to any of the criteria will be marked as -1 in the estimated moment condition variance (activated only if `debug == TRUE`).

`phi_hat` An `m` by `k+p1*k+p2` matrix containing the entire estimated effect sizes (including those on source weights) for each feature (activated only if `debug == TRUE`).

`phi_se` An `m` by `k+p1*k+p2` matrix containing the estimated standard errors associated with `phi_hat` for each feature (activated only if `debug == TRUE`).

`phi_hat_pvals` An `m` by `k+p1*k+p2` matrix containing the p-values associated with `phi_hat` for each feature (activated only if `debug == TRUE`).

Examples

```
data = simulate_data(n=100, m=2, k=3, p1=1, p2=1, taus_std=0, log_file=NULL)
res = list()
res$params.hat = Unico(data$X, data$W, data$C1, data$C2, parallel=FALSE, log_file=NULL)
res$params.hat = association_asymptotic(data$X, res$params.hat, parallel=FALSE, log_file=NULL)
```

association_parametric

Performs parametric statistical testing

Description

Performs parametric statistical testing (T-test) on (1) the marginal effect of each covariate in C1 at source-specific level (2) the joint effect across all sources for each covariate in C1 (3) non-source-specific effect for each covariate in C2. In the context of bulk genomic data containing a mixture of cell types, these correspond to the marginal effect of each covariate in C1 (potentially including the phenotype of interest) at each cell type, joint tissue-level effect for each covariate in C1, and tissue-level effect for each covariate in C2.

Usage

```
association_parametric(
  X,
  Unico.mdl,
  slot_name = "parametric",
  diag_only = FALSE,
  intercept = TRUE,
  X_max_stds = 2,
  Q_max_stds = Inf,
  XQ_max_stds = Inf,
  parallel = TRUE,
  num_cores = NULL,
  log_file = "Unico.log",
  verbose = FALSE,
  debug = FALSE
)
```

Arguments

X	An m by n matrix of measurements of m features for n observations. Each column in X is assumed to be a mixture of k sources. Note that X must include row names and column names and that NA values are currently not supported. X should not include features that are constant across all observations. Note that X must be the same X used to learn Unico.mdl (i.e. the original observed 2D mixture used to fit the model).
Unico.mdl	The entire set of model parameters estimated by Unico on the 2D mixture matrix (i.e. the list returned by applying function Unico to X).
slot_name	A string indicating the key for storing the results under Unico.mdl
diag_only	A logical value indicating whether to only use the estimated source-level variances (and thus ignoring the estimate covariance) for controlling the heterogeneity in the observed mixture. if set to FALSE, Unico instead estimates the observation- and feature-specific variance in the mixture by leveraging the entire k by k variance-covariance matrix.

intercept	A logical value indicating whether to fit the intercept term when performing the statistical testing.
X_max_stds	A non-negative numeric value indicating, for each feature, the portions of data that are considered as outliers due to the observed mixture value. Only samples whose observed mixture value fall within X_max_stds standard deviations from the mean will be used for the statistical testing of a given feature.
Q_max_stds	A non-negative numeric value indicating, for each feature, the portions of data that are considered as outliers due to the estimated mixture variance. Only samples whose estimated mixture variance fall within Q_max_stds standard deviations from the mean will be used for the statistical testing of a given feature.
XQ_max_stds	A non-negative numeric value indicating, for each feature, the portions of data that are considered as outliers due to the weighted mixture value. Only samples whose weighted mixture value fall within XQ_max_stds standard deviations from the mean will be used for the statistical testing of a given feature.
parallel	A logical value indicating whether to use parallel computing (possible when using a multi-core machine).
num_cores	A numeric value indicating the number of cores to use (activated only if parallel == TRUE). If num_cores == NULL then all available cores except for one will be used.
log_file	A path to an output log file. Note that if the file log_file already exists then logs will be appended to the end of the file. Set log_file to NULL to prevent output from being saved into a file; note that if verbose == FALSE then no output file will be generated regardless of the value of log_file.
verbose	A logical value indicating whether to print logs.
debug	A logical value indicating whether to set the logger to a more detailed debug level; set debug to TRUE before reporting issues.

Details

If we assume that source-specific values Z_{ijh} are normally distributed, under the Unico model, we have the following:

$$Z_{ij} \sim \mathcal{N}\left(\mu_j + (c_i^{(1)})^T \gamma_{jh}, \sigma_{jh}^2\right)$$

$$X_{ij} \sim \mathcal{N}\left(w_i^T(\mu_j + (c_i^{(1)})^T \gamma_{jh}) + (c_i^{(2)})^T \beta_j, \text{Sum}\left((w_i w_i^T) \odot \Sigma_j\right) + \tau_j^2\right)$$

For a given feature j under test, the above equation corresponds to a heteroskedastic regression problem with X_{ij} as the dependent variable and $\{\{w_i\}, \{w_i c_i^{(1)}\}, \{c_i^{(2)}\}\}$ as the set of independent variables. This view allows us to perform parametric statistical testing (T-test for marginal effects and partial F-test for joint effects) by solving a generalized least squares problem with sample i scaled by the inverse of its estimated standard deviation.

Value

An updated Unico.mdl object with the the following list of effect size and p-value estimates stored in an additional key specified by slot_name

gammas_hat — An m by k*p1 matrix of the estimated effects of the p1 covariates in C1 on each of the m features in X, where the first p1 columns are the source-specific effects of the p1 covariates on the first source, the following p1 columns are the source-specific effects on the second source and so on.

betas_hat — An m by p2 matrix of the estimated effects of the p2 covariates in C2 on the mixture values of each of the m features in X.

gammas_hat_pvals — An m by k*p1 matrix of p-values for the estimates in gammas_hat (based on a T-test).

betas_hat_pvals — An m by p2 matrix of p-values for the estimates in betas_hat (based on a T-test).

gammas_hat_pvals.joint — An m by p1 matrix of p-values for the joint effects (i.e. across all k sources) of each of the p1 covariates in C1 on each of the m features in X (based on a partial F-test). In other words, these are p-values for the combined statistical effects (across all sources) of each one of the p1 covariates on each of the m features under the Unico model.

Q — An m by n matrix of weights used for controlling the heterogeneity of each observation at each feature (activated only if debug == TRUE).

masks — An m by n matrix of logical values indicating whether observation participated in statistical testing at each feature (activated only if debug == TRUE).

phi_hat — An m by k+p1*k+p2 matrix containing the entire estimated effect sizes (including those on source weights) for each feature (activated only if debug == TRUE).

phi_se — An m by k+p1*k+p2 matrix containing the estimated standard errors associated with phi_hat for each feature (activated only if debug == TRUE).

phi_hat_pvals — An m by k+p1*k+p2 matrix containing the p-values associated with phi_hat for each feature (activated only if debug == TRUE).

Examples

```
data = simulate_data(n=100, m=2, k=3, p1=1, p2=1, taus_std=0, log_file=NULL)
res = list()
res$params.hat = Unico(data$X, data$W, data$C1, data$C2, parallel=FALSE, log_file=NULL)
res$params.hat = association_parametric(data$X, res$params.hat, parallel=FALSE, log_file=NULL)
```

---

simulate_data *Simulate data under Unico model assumption*

Description

Simulate all model parameters and sample source specific data from multivariate gaussian with full covariance structure.

Usage

```
simulate_data(
  n,
  m,
  k,
  p1,
  p2,
  mus_mean = 10,
  mus_std = 2,
  gammas_mean = 1,
  gammas_std = 0.1,
  betas_mean = 1,
  betas_std = 0.1,
  sigmas_lb = 0,
  sigmas_ub = 1,
  taus_std = 0.1,
  log_file = "Unico.log",
  verbose = FALSE
)
```

Arguments

n	A positive integer indicating the number of observations to simulate.
m	A positive integer indicating the number of features to simulate.
k	A positive integer indicating the number of sources to simulate.
p1	A non-negative integer indicating the number of source-specific covariates to simulate.
p2	A non-negative indicating the number of non-source-specific covariates to simulate.
mus_mean	A numerical value indicating the average of the source specific means.
mus_std	A positive value indicating the variation of the source specific means across difference sources.
gammas_mean	A numerical value indicating the average effect sizes of the source-specific covariates.
gammas_std	A non-negative numerical value indicating the variation of the effect sizes of the source-specific covariates.
betas_mean	A numerical value indicating the average effect sizes of the non-source-specific covariates.
betas_std	A non-negative numerical value indicating the variation of the effect sizes of the non-source-specific covariates.
sigmas_lb	A numerical value indicating the lower bound of a uniform distribution from which we sample entries of matrix A used to construct the feature specific k by k variance-covariance matrix.

sigmas_ub	A numerical value indicating the upper bound of a uniform distribution from which we sample entries of matrix A used to construct the feature specific k by k variance-covariance matrix.
taus_std	non-negative numerical value indicating the variation of the measurement noise across difference features.
log_file	A path to an output log file. Note that if the file log_file already exists then logs will be appended to the end of the file. Set log_file to NULL to prevent output from being saved into a file; note that if verbose == FALSE then no output file will be generated regardless of the value of log_file.
verbose	A logical value indicating whether to print logs.

Details

Simulate data based on the generative model described in function Unico.

Value

A list of simulated model parameters, covariates, observed mixture, and source-specific data.

X	An m by n matrix of the simulated mixture for m features and n observations.
W	An n by k matrix of the weights/proportions of k source for each of the n observations.
C1	An n by p1 matrix of the simulated covariates that affect the source-specific values.
C2	An n by p2 matrix of the simulated covariates that affect the mixture values.
Z	A k by m by n tensor of the source specific values for each of the k sources
mus	An m by k matrix of the mean of each of the m features for each of the k sources.
gammas	An m by k*p1 matrix of the effect sizes of the p1 covariates in C1 on each of the m features in X, where the first p1 columns are the source-specific effects of the p1 covariates on the first source, the following p1 columns are the source-specific effects on the second source and so on.
betas	An m by p2 matrix of the effect sizes of the p2 covariates in C2 on the mixture values of each of the m features.
sigmas	An m by k by k tensor of the variance-covariance matrix of each of the m features.
taus	An m by 1 matrix of the feature specific variance of the measurement noise for all m features.

Examples

```
data = simulate_data(n=100, m=2, k=3, p1=1, p2=1, taus_std=0, log_file=NULL)
```

`tensor` *Inferring the underlying source-specific 3D tensor*

Description

Infers the underlying (sources by features by observations) 3D tensor from the observed (features by observations) 2D mixture, under the assumption of the Unico model that each observation is a mixture of unique source-specific values (in each feature in the data). In the context of bulk genomics containing a mixture of cell types (i.e. the input could be CpG sites by individuals for DNA methylation and genes by individuals for RNA expression), `tensor` allows to estimate the cell-type-specific levels for each individual in each CpG site/gene (i.e. a tensor of CpG sites/genes by individuals by cell types).

Usage

```
tensor(
  X,
  W,
  C1,
  C2,
  Unico.mdl,
  parallel = TRUE,
  num_cores = NULL,
  log_file = "Unico.log",
  verbose = FALSE,
  debug = FALSE
)
```

Arguments

`X`	An `m` by `n` matrix of measurements of `m` features for `n` observations. Each column in `X` is assumed to be a mixture of `k` sources. Note that `X` must include row names and column names and that NA values are currently not supported. `X` should not include features that are constant across all observations. Note that `X` could potentially be different from the `X` used to learn `Unico.mdl` (i.e. the original observed 2D mixture used to fit the model).
`W`	An `n` by `k` matrix of weights - the weights of `k` sources for each of the `n` mixtures (observations). All the weights must be positive and each row - corresponding to the weights of a single observation - must sum up to 1. Note that `W` must include row names and column names and that NA values are currently not supported.
`C1`	An `n` by `p1` design matrix of covariates that may affect the hidden source-specific values (possibly a different effect size in each source). Note that `C1` must include row names and column names and should not include an intercept term. NA values are currently not supported. Note that all covariates in `C1` must be present and match the order of the set of covariates in `C1` stored in `Unico.mdl` (i.e. the original set of source-specific covariates available when initially fitting the model).

C2	An n by p2 design matrix of covariates that may affect the mixture (i.e. rather than directly the sources of the mixture; for example, variables that capture biases in the collection of the measurements). Note that C2 must include row names and column names and should not include an intercept term. NA values are currently not supported. Note that all covariates in C2 must be present and match the order of the set of covariates in C2 stored in Unico.mdl (i.e. the original set of not source-specific covariates available when initially fitting the model).
Unico.mdl	The entire set of model parameters estimated by Unico on the 2D mixture matrix (i.e. the list returned by applying function Unico to X).
parallel	A logical value indicating whether to use parallel computing (possible when using a multi-core machine).
num_cores	A numeric value indicating the number of cores to use (activated only if parallel == TRUE). If num_cores == NULL then all available cores except for one will be used.
log_file	A path to an output log file. Note that if the file log_file already exists then logs will be appended to the end of the file. Set log_file to NULL to prevent output from being saved into a file; note that if verbose == FALSE then no output file will be generated regardless of the value of log_file.
verbose	A logical value indicating whether to print logs.
debug	A logical value indicating whether to set the logger to a more detailed debug level; set debug to TRUE before reporting issues.

Details

After obtaining all the estimated parameters in the Unico model (by calling Unico), tensor uses the conditional distribution $Z_{jh}^{i}|X_{ij} = x_{ij}$ for estimating the k source-specific levels of each sample i at each feature j.

Value

A k by m by n array with the estimated source-specific values. The first axis/dimension in the array corresponds to the different sources.

Examples

```
data = simulate_data(n=100, m=2, k=3, p1=1, p2=1, taus_std=0, log_file=NULL)
res = list()
res$params.hat = Unico(data$X, data$W, data$C1, data$C2, parallel=FALSE, log_file=NULL)
res$Z = tensor(data$X, data$W, data$C1, data$C2, res$params.hat, parallel=FALSE, log_file=NULL)
```

Unico *Fitting the Unico model*

Description

Fits the Unico model for an input matrix of features by observations that are coming from a mixture of k sources, under the assumption that each observation is a mixture of unique (unobserved) source-specific values (in each feature in the data). Specifically, for each feature, it standardizes the data and learns the source-specific mean and full k by k variance-covariance matrix.

Usage

```
Unico(
  X,
  W,
  C1,
  C2,
  fit_tau = FALSE,
  mean_penalty = 0,
  var_penalty = 0.01,
  covar_penalty = 0.01,
  mean_max_iterations = 2,
  var_max_iterations = 3,
  nloptr_opts_algorithm = "NLOPT_LN_COBYLA",
  max_stds = 2,
  init_weight = "default",
  max_u = 1,
  max_v = 1,
  parallel = TRUE,
  num_cores = NULL,
  log_file = "Unico.log",
  verbose = FALSE,
  debug = FALSE
)
```

Arguments

X	An m by n matrix of measurements of m features for n observations. Each column in X is assumed to be a mixture of k sources. Note that X must include row names and column names and that NA values are currently not supported. X should not include features that are constant across all observations.
W	An n by k matrix of weights - the weights of k sources for each of the n mixtures (observations). All the weights must be positive and each row - corresponding to the weights of a single observation - must sum up to 1. Note that W must include row names and column names and that NA values are currently not supported.

`C1` An `n` by `p1` design matrix of covariates that may affect the hidden source-specific values (possibly a different effect size in each source). Note that `C1` must include row names and column names and should not include an intercept term. NA values are currently not supported. Note that each covariate in `C1` results in `k` additional parameters in the model of each feature, therefore, in order to alleviate the possibility of model overfitting, it is advised to be mindful of the balance between the size of `C1` and the sample size in `X`.

`C2` An `n` by `p2` design matrix of covariates that may affect the mixture (i.e. rather than directly the sources of the mixture; for example, variables that capture biases in the collection of the measurements). Note that `C2` must include row names and column names and should not include an intercept term. NA values are currently not supported.

`fit_tau` A logical value indicating whether to fit the standard deviation of the measurement noise (i.e. the i.i.d. component of variation in the model denoted as τ).

`mean_penalty` A non-negative numeric value indicating the regularization strength on the source-specific mean estimates.

`var_penalty` A non-negative numeric value indicating the regularization strength on the diagonal entries of the full `k` by `k` variance-covariance matrix.

`covar_penalty` A non-negative numeric value indicating the regularization strength on the off diagonal entries of the full `k` by `k` variance-covariance matrix.

`mean_max_iterations`
A non-negative numeric value indicating the number of iterative updates performed on the mean estimates.

`var_max_iterations`
A non-negative numeric value indicating the number of iterative updates performed on the variance-covariance matrix.

`nloptr_opts_algorithm`
A string indicating the optimization algorithm to use.

`max_stds` A non-negative numeric value indicating, for each feature, the portions of data that are considered as outliers. Only samples within `max_stds` standard deviations from the mean will be used for the moments estimation of a given feature.

`init_weight` A string indicating the initial weights on the samples to start the iterative optimization.

`max_u` A non-negative numeric value indicating the maximum weights/influence a sample can have on mean estimates.

`max_v` A non-negative numeric value indicating the maximum weights/influence a sample can have on variance-covariance estimates.

`parallel` A logical value indicating whether to use parallel computing (possible when using a multi-core machine).

`num_cores` A numeric value indicating the number of cores to use (activated only if `parallel == TRUE`). If `num_cores == NULL` then all available cores except for one will be used.

`log_file` A path to an output log file. Note that if the file `log_file` already exists then logs will be appended to the end of the file. Set `log_file` to `NULL` to prevent output from being saved into a file; note that if `verbose == FALSE` then no output file will be generated regardless of the value of `log_file`.

verbose	A logical value indicating whether to print logs.
debug	A logical value indicating whether to set the logger to a more detailed debug level; set debug to TRUE before reporting issues.

Details

Unico assumes the following model:

$$X_{ij} = w_i^T Z_{ij} + (c_i^{(2)})^T \beta_j + e_{ij}$$

The mixture value at sample i feature j: X_{ij} is modeled as a weighted linear combination, specified by weights $w_i = (w_{i1}, ..., w_{ik})$, of a total of k source-specific levels, specified by $Z_{ij} = (Z_{ij1}, ..., Z_{ijk})$. In addition, we also consider global-level covariates $c_i^{(2)}$ that systematically affect the observed mixture values and their effect sizes β_j. e_{ij} denotes the i.i.d measurement noise with variance τ across all samples. Weights have be to non-negative and sum up to 1 across all sources for each sample. In practice, we assume that the weights are fixed and estimated by external methods.

Source specific profiles are further modeled as:

$$Z_{ijh} = \mu_{jh} + (c_i^{(1)})^T \gamma_{jh} + \epsilon_{ijh}$$

μ_{jh} denotes the population level mean of feature j at source h. We also consider covariates $c_i^{(1)}$ that systematically affect the source-specific values and their effect sizes γ_{jh} on each source. Finally, we actively model the k by k covariance structure of a given feature j across all k sources $Var[\vec{\epsilon_{ij}}] = \Sigma_j \in \mathbf{R}^{k \times k}$.

Value

A list with the estimated parameters of the model. This list can be then used as the input to other functions such as tensor.

W	An n by k matrix of weights. This is the same as W from input.
C1	An n by p1 design matrix of source-specific covariates. This is the same as C1 from input.
C2	An n by p2 design matrix of not source-specific covariates. This is the same as C2 from input.
mus_hat	An m by k matrix of estimates for the mean of each source in each feature.
gammas_hat	An m by k*p1 matrix of the estimated effects of the p1 covariates in C1 on each of the m features in X, where the first p1 columns are the source-specific effects of the p1 covariates on the first source, the following p1 columns are the source-specific effects on the second source and so on.
betas_hat	An m by p2 matrix of the estimated effects of the p2 covariates in C2 on the mixture values of each of the m features in X.
sigmas_hat	An m by k by k tensor of estimates for the cross source k by k variance-covariance matrix in each feature.
taus_hat	An m by 1 matrix of estimates for the variance of the measurement noise.

`scale.factor` An `m` by `1` matrix of scaling factors for standardizing each feature.

`config` A list with hyper-parameters used for fitting the model and configurations for in the optimization algorithm.

`Us_hat_list` A list tracking, for each feature, the sample weights used for each iteration of the mean optimization (activated only if `debug == TRUE`).

`Vs_hat_list` A list tracking, for each feature, the sample weights used for each iteration of the variance-covariance optimization (activated only if `debug == TRUE`).

`Ls_hat_list` A list tracking, for each feature, the computed estimates of the upper triangular cholesky decomposition of variance-covariance matrix at each iteration of the variance-covariance optimization (activated only if `debug == TRUE`).

`sigmas_hat_list`
A list tracking, for each feature, the computed estimates of the variance-covariance matrix at each iteration of the variance-covariance optimization (activated only if `debug == TRUE`).

Examples

```
data = simulate_data(n=100, m=2, k=3, p1=1, p2=1, taus_std=0, log_file=NULL)
res = list()
res$params.hat = Unico(data$X, data$W, data$C1, data$C2, parallel=FALSE, log_file=NULL)
```

Index