Package ‘dtangle’

December 1, 2019

Title  Cell Type Deconvolution from Gene Expressions
Version 2.0.9
Description  Deconvolving cell types from high-throughput gene profiling data. For more information on dtangle see Hunt et al. (2019) <doi:10.1093/bioinformatics/bty926>.

Date 2019-11-29
LazyData true
RoxygenNote 6.1.1
Imports DEoptimR,
Suggests knitr, rmarkdown, testthat
VignetteBuilder knitr
License GPL-3
Depends R(>= 3.5.0)
NeedsCompilation no
Author Gregory Hunt [aut, cre],
    Johann Gagnon-Bartsch [aut]
Maintainer Gregory Hunt <ghunt@wm.edu>
Repository CRAN
Date/Publication 2019-12-01 17:30:05 UTC

R topics documented:

baseline_exprs .......................... 2
combine_Y_refs ................................ 3
dtangle ........................................ 4
dtangle2 ....................................... 6
est_phats ..................................... 10
find_markers .................................. 11
get_gamma ..................................... 13
process_markers ................................ 13
shen_orr_ex .................................. 15

Index 16
baseline_exprs

*Estimate the offset terms.*

**Description**

Estimate the offset terms.

**Usage**

```
baseline_exprs(Y, pure_samples, markers, summary_fn = mean)
```

**Arguments**

- **Y**
  - Expression matrix.
  - (Required) Two-dimensional numeric. Must implement `as.matrix`.
  - Each row contains expression measurements for a particular sample. Each column contains the measurements of the same gene over all individuals. Can either contain just the mixture samples to be deconvolved or both the mixture samples and the reference samples. See `pure_samples` and `references` for more details.

- **pure_samples**
  - The pure sample indices.
  - (Optional) List of one-dimensional integer. Must implement `as.list`.
  - The i-th element of the top-level list is a vector of indicies (rows of `Y` or `references`) that are pure samples of type i. If `references` is not specified then this argument identifies which rows of `Y` correspond to pure reference samples of which cell-types. If `references` is specified then this makes same identification but for the `references` matrix instead.

- **markers**
  - Marker gene indices.
  - (Optional) List of one-dimensional integer.
  - Top-level list should be same length as `pure_samples`, i.e. one element for each cell type. Each element of the top-level list is a vector of indicies (columns of `Y`) that will be considered markers of that particular type. If not supplied then `dtangle` finds markers internally using `find_markers`. Alternatively, one can supply the output of `find_markers` to the `markers` argument.

- **summary_fn**
  - What summary statistic to use when aggregating expression measurements.
  - (Optional) Function that takes a one-dimensional vector of numeric and returns a single numeric.
  - Defaults to the mean. Other good options include the median.

**Value**

List of vectors. Each vector is estimated estimated basline in pure samples of markers for each group, resp.
**Examples**

```r
truth = shen_orr_ex$annotation$mixture
pure_samples <- lapply(1:3, function(i) {
    which(truth[, i] == 1)
})
Y <- shen_orr_ex$data$log
markers = find_markers(Y=Y, pure_samples=pure_samples, data_type='microarray-gene', marker_method='ratio')$L
K = length(pure_samples)
n_markers = rep(20, K)
mrkrs <- lapply(1:K, function(i) {
    markers[[i]][1:n_markers[i]]
})
dtangle:::baseline_exprs(Y, pure_samples, mrkrs)
```

---

**Description**

Row-binds Y with references and generates pure_samples.

**Usage**

```r
combine_Y_refs(Y, references, pure_samples)
```

**Arguments**

- **Y**
  - Expression matrix.
  - (Required) Two-dimensional numeric. Must implement `as.matrix`.
  - Each row contains expression measurements for a particular sample. Each column contains the measurements of the same gene over all individuals. Can either contain just the mixture samples to be deconvolved or both the mixture samples and the reference samples. See `pure_samples` and `references` for more details.

- **references**
  - Cell-type reference expression matrix.
  - (Optional) Two-dimensional numeric. Must implement `as.matrix`. Must have same number of columns as Y. Columns must correspond to columns of Y.
  - Each row contains expression measurements for a reference profile of a particular cell type. Columns contain measurements of reference profiles of a gene. Optionally may merge this matrix with Y and use `pure_samples` to indicate which rows of Y are pure samples. If `pure_samples` is not specified `references` must be specified. In this case each row of `references` is assumed to be a distinct cell-type. If both `pure_samples` and `references` are specified then multiple rows of `references` may refer be the same cell type, and `pure_samples` specifies to which cell-type each row of `references` corresponds.
dtangle

Deconvolve cell type mixing proportions from gene expression data.

Description

Deconvolve cell type mixing proportions from gene expression data.

Usage

dtangle(Y, references = NULL, pure_samples = NULL, n_markers = NULL, data_type = NULL, gamma = NULL, markers = NULL, marker_method = "ratio", summary_fn = mean)

Arguments

Y
(Required) Two-dimensional numeric. Must implement as.matrix.
Each row contains expression measurements for a particular sample. Each column
contains the measurements of the same gene over all individuals. Can either con-
tain just the mixture samples to be deconvolved or both the mixture samples and
the reference samples. See pure_samples and references for more details.

references
(Optional) Two-dimensional numeric. Must implement as.matrix. Must have
same number of columns as Y. Columns must correspond to columns of Y.
Each row contains expression measurements for a reference profile of a particu-
lar cell type. Columns contain measurements of reference profiles of a gene. Op-
tionally may merge this matrix with Y and use pure_samples to indicate which
rows of Y are pure samples. If pure_samples is not specified references must
be specified. In this case each row of references is assumed to be a distinct
cell-type. If both pure_samples and references are specified then multiple
rows of references may refer be the same cell type, and pure_samples speci-
fies to which cell-type each row of references corresponds.

pure_samples
(Optional) List of one-dimensional integer. Must implement as.list.
The i-th element of the top-level list is a vector of indicies (rows of Y or references)
that are pure samples of type i. If references is not specified then this argu-
ment identifies which rows of Y correspond to pure reference samples of which
cell-types. If references is specified then this makes same identification but for
the references matrix instead.
n_markers  Number of marker genes.
            (Optional) One-dimensional numeric.
How many markers genes to use for deconvolution. Can either be a single integer, vector of integers (one for each cell type), or single or vector of percentages (numeric in 0 to 1). If a single integer then all cell types use that number of markers. If a vector then the i-th element determines how many marker genes are used for the i-th cell type. If single percentage (in 0 to 1) then that percentage of markers are used for all types. If vector of percentages then that percentage used for each type, respectively. If not specified then top 10% of genes are used.

data_type  Type of expression measurements.
            (Optional) One-dimensional string.
An optional string indicating the type of the expression measurements. This is used to set gamma to a pre-determined value based upon the data type. Valid values are for probe-level microarray as "microarray-probe", gene-level microarray as "microarray-gene" or rna-seq as "rna-seq". Alternatively can set gamma directly.

gamma  Expression adjustment term.
            (Optional) One-dimensional positive numeric.
If provided as a single positive number then that value will be used for gamma and over-ride the value of gamma chosen by the data_type argument. If neither gamma nor data_type are specified then gamma will be set to one.

markers  Marker gene indices.
            (Optional) List of one-dimensional integer.
Top-level list should be same length as pure_samples, i.e. one element for each cell type. Each element of the top-level list is a vector of indicies (columns of Y) that will be considered markers of that particular type. If not supplied then dtangle finds markers internally using find_markers. Alternatively, one can supply the output of find_markers to the markers argument.

marker_method  Method used to rank marker genes.
            (Optional) One-dimensional string.
The method used to rank genes as markers. If not supplied defaults to “ratio”. Only used if markers are not provided to argument “markers”. Options are

• ’ratio’ selects and ranks markers by the ratio of the mean expression of each gene in each cell type to the mean of that gene in all other cell types.
• ’regression’ selects and ranks markers by estimated regression coefficients in a series of regressions with single covariate that is indicator of each type.
• ’diff’ selects and ranks markers based upon the difference, for each cell type, between the median expression of a gene by each cell type and the median expression of that gene by the second most highly expressed cell type.
• ’p.value’ selects and ranks markers based upon the p-value of a t-test between the median expression of a gene by each cell type and the median expression of that gene by the second most highly expressed cell type.

summary_fn  What summary statistic to use when aggregating expression measurements.
(Optional) Function that takes a one-dimensional vector of numeric and returns a single numeric. Defaults to the mean. Other good options include the median.

Value

List.

- ‘estimates’ a matrix estimated mixing proportions. One row for each sample, one column for each cell type.
- ‘markers’ list of vectors of marker used for each cell type. Each element of list is vector of columns of Y used as a marker for the i-th cell type.
- ‘n_markers’ vector of number of markers used for each cell type.
- ‘gamma’ value of the sensitivity parameter gamma used by dtangle.

See Also

find_markers

Examples

```r
truth = shen_orr_ex$annotation$mixture
pure_samples <- lapply(1:3, function(i) {
  which(truth[, i] == 1)
})
Y <- shen_orr_ex$data$log
n_markers = 20

dtangle(Y, pure_samples = pure_samples, 
n_markers=n_markers,data_type=’Var microarray-gene’,marker_method = ’ratio’)

n_markers = c(10,11,12)
dtangle(Y, pure_samples=pure_samples, 
n_markers=n_markers,marker_method = ’regression’)
```

---

**dtangle2**  
*Deconvolve cell type mixing proportions from gene expression data.*

Description

Deconvolve cell type mixing proportions from gene expression data.

Usage

dtangle2(Y, references = NULL, pure_samples = NULL, n_markers = NULL, 
markers = NULL, marker_method = ”ratio”, weights = NULL, 
sto = TRUE, inv_scale = function(x) 2^x, fit_scale = log, 
loss_smry = ”var”, dtangle_init = TRUE, seed = NULL, 
verbose = FALSE, optim_opts = NULL)
Arguments

Y

Expression matrix. 
(Required) Two-dimensional numeric. Must implement as.matrix.
Each row contains expression measurements for a particular sample. Each column contains the measurements of the same gene over all individuals. Can either contain just the mixture samples to be deconvolved or both the mixture samples and the reference samples. See pure_samples and references for more details.

references

Cell-type reference expression matrix. 
(Optional) Two-dimensional numeric. Must implement as.matrix. Must have same number of columns as Y. Columns must correspond to columns of Y.
Each row contains expression measurements for a reference profile of a particular cell type. Columns contain measurements of reference profiles of a gene. Optionally may merge this matrix with Y and use pure_samples to indicate which rows of Y are pure samples. If pure_samples is not specified references must be specified. In this case each row of references is assumed to be a distinct cell-type. If both pure_samples and references are specified then pure_samples specifies to which cell-type each row of references corresponds.

pure_samples

The pure sample indicies. 
(Optional) List of one-dimensional integer. Must implement as.list.
The i-th element of the top-level list is a vector of indicies (rows of Y or references) that are pure samples of type i. If references is not specified then this argument identifies which rows of Y correspond to pure reference samples of which cell-types. If references is specified then this makes same identification but for the references matrix instead.

n_markers

Number of marker genes. 
(Optional) One-dimensional numeric. 
How many markers genes to use for deconvolution. Can either be a single integer, vector of integers (one for each cell type), or single or vector of percentages (numeric in 0 to 1). If a single integer then all cell types use that number of markers. If a vector then the i-th element determines how many marker genes are used for the i-th cell type. If single percentage (in 0 to 1) then that percentage of markers are used for all types. If vector of percentages then that percentage used for each type, respectively. If not specified then top 10% of genes are used.

markers

Marker gene indices. 
(Optional) List of one-dimensional integer. 
Top-level list should be same length as pure_samples, i.e. one element for each cell type. Each element of the top-level list is a vector of indicies (columns of Y) that will be considered markers of that particular type. If not supplied then dtangle finds markers internally using find_markers. Alternatively, one can supply the output of find_markers to the markers argument.

marker_method

Method used to rank marker genes. 
(Optional) One-dimensional string. 
The method used to rank genes as markers. If not supplied defaults to “ratio”. Only used if markers are not provided to argument “markers”. Options are
dtangle2

- 'ratio' selects and ranks markers by the ratio of the mean expression of each gene in each cell type to the mean of that gene in all other cell types.
- 'regression' selects and ranks markers by estimated regression coefficients in a series of regressions with single covariate that is indicator of each type.
- 'diff' selects and ranks markers based upon the difference, for each cell type, between the median expression of a gene by each cell type and the median expression of that gene by the second most highly expressed cell type.
- 'p.value' selects and ranks markers based upon the p-value of a t-test between the median expression of a gene by each cell type and the median expression of that gene by the second most highly expressed cell type.

weights
Weights for the genes.
(Optional) String or one-dimensional numeric vector.
Weights for the genes in the optimization. If NULL (default) then does not weight genes differently. If 'variance' then inversely weights with the variance of the references. This only works if there is more than one reference per cell type so that the variance can be estimated. If a numeric then this uses whatever is specified as weights. They must be non-negative.

sto
Sum-to-one constraint.
(Optional) Boolean.
Re-normalize the estimates so that the cell-type proportions sum to one.

inv_scale
Inverse scale transformation.
(Optional) Function.
Defaults to 2^x. This is equivalent to assuming that the data has been log2-transformed. If another transformation has been applied to the data then this function should be used to specify the inverse of that transformation needed to put gene expressions on the linear scale.

fit_scale
Transformation to used as part of optimization.
(Optional) Function.
Function to apply to gene expressions as part of optimization. Defaults to log.

loss_smry
Loss summary function minimized to find estimated proportions.
(Optional) String.
Either 'var' (default) to minimize the (weighted) variance of the residuals or 'L2' to minimize the (weighted) sums of squares of the residuals.

dtangle_init
Optimization initialization.
(Optional) Boolean.
Boolean controlling if dtangle2 optimization should be initialized using dtangle1 estimates.

seed
(Optional) Integer.
Value at which to seed the random seed before estimating. Optimization initialization might change if this value is not specified.

verbose
(Optional) Boolean.
Controls if optimization output is printed or not.
Optimization options passed to DEoptimR controlling optimization. Options that may be set are

- `constr` constraint to enforce. Either 'box' for 0-1 box constraints that proportions are between zero and one, 'ineq' for constraints that proportions sum to less than one, 'eq' for equality constraints that proportions sum to one, or 'eq_solve' to solve for one of the parameters in terms of the other and enforce equality constraints using inequality on remaining parameters. Default and recommended is 'box'.
- `ninit` number of randomly initialized points as part of the DEoptimR initial population.
- `tritter` how often to print results if `verbose=TRUE`.
- `maxiter` maximum number of optimization iterations to use before exiting.
- `convtol` tolerance for convergence tolerance stopping criterion.
- `constrtol` tolerance for constraint enforcement.

Value

List.

- 'estimates' a matrix estimated mixing proportions. One row for each sample, one column for each cell type.
- 'markers' list of vectors of marker used for each cell type. Each element of list is vector of columns of `Y` used as a marker for the i-th cell type.
- 'n_markers' vector of number of markers used for each cell type.
- 'weights' the weights used as part of the optimization.
- 'diag' diagnostic values for the estimated proportions. `resids_hat, loss_hat, and p_hat` are the residuals, loss, and estimates for the proportions returned by dtangle2. Similarly, `resids_opt, loss_opt` and `p_opt` are these values for the optimized value not re-scaled to enforce the STO constraint.

See Also

- `find_markers`

Examples

```r
truth = shen_orr_ex$annotation$mixture
pure_samples <- lapply(1:3, function(i) { which(truth[, i] == 1) })
Y <- shen_orr_ex$data$log
n_markers = 20

dtangle2(Y, pure_samples = pure_samples, n_markers=n_markers)
```
Estimate the gene type proportions.

**Description**

Estimate the gene type proportions.

**Usage**

```r
est_phats(Y, markers, baseline_ests, gamma, summary_fn = mean, inv_scale = function(x) 2^x)
```

**Arguments**

- **Y**
  - Expression matrix.
  - (Required) Two-dimensional numeric. Must implement `as.matrix`.
  - Each row contains expression measurements for a particular sample. Each column contains the measurements of the same gene over all individuals. Can either contain just the mixture samples to be deconvolved or both the mixture samples and the reference samples. See `pure_samples` and `references` for more details.

- **markers**
  - Marker gene indices.
  - (Optional) List of one-dimensional integer.
  - Top-level list should be same length as `pure_samples`, i.e. one element for each cell type. Each element of the top-level list is a vector of indicies (columns of `Y`) that will be considered markers of that particular type. If not supplied then `dtangle` finds markers internally using `find_markers`. Alternatively, one can supply the output of `find_markers` to the `markers` argument.

- **baseline_ests**
  - List of vectors (same structure as `markers`). One list entry for each cell type.
  - Each list element is a vector of estimated offset for each marker of the respective type (output from `baseline_exprs`).

- **gamma**
  - Expression adjustment term.
  - (Optional) One-dimensional positive numeric.
  - If provided as a single positive number then that value will be used for `gamma` and over-ride the value of `gamma` chosen by the `data_type` argument. If neither `gamma` nor `data_type` are specified then `gamma` will be set to one.

- **summary_fn**
  - What summary statistic to use when aggregating expression measurements.
  - (Optional) Function that takes a one-dimensional vector of numeric and returns a single numeric.
  - Defaults to the mean. Other good options include the median.

- **inv_scale**
  - Inverse scale transformation. Default to exponential as `dtangle` assumes data has been logarithmically transformed.

**Value**

Estimated matrix of mixing proportions.
Examples

```r
truth = shen_orr_ex$annotation$mixture
pure_samples <- lapply(1:3, function(i) {
    which(truth[, i] == 1)
})
Y <- shen_orr_ex$data$log
markers = find_markers(Y=Y, pure_samples = pure_samples, data_type='microarray-gene', marker_method='ratio')$L
K = length(pure_samples)
n_markers = rep(20,K)
mrkrs <- lapply(1:K, function(i) {
    markers[[i]][1:n_markers[i]]
})
baseline = dtangle:::baseline_exprs(Y, pure_samples, mrkrs)
phats <- dtangle:::est_phats(Y, mrkrs, baseline, gamma=.8)
```

Description

Find marker genes for each cell type.

Usage

```r
find_markers(Y, references = NULL, pure_samples = NULL, data_type = NULL, gamma = NULL, marker_method = "ratio")
```

Arguments

- **Y**: Expression matrix.
  - (Required) Two-dimensional numeric. Must implement `as.matrix`.
  - Each row contains expression measurements for a particular sample. Each column contains the measurements of the same gene over all individuals. Can either contain just the mixture samples to be deconvolved or both the mixture samples and the reference samples. See `pure_samples` and `references` for more details.

- **references**: Cell-type reference expression matrix.
  - (Optional) Two-dimensional numeric. Must implement `as.matrix`. Must have same number of columns as `Y`. Columns must correspond to columns of `Y`.
  - Each row contains expression measurements for a reference profile of a particular cell type. Columns contain measurements of reference profiles of a gene. Optionally may merge this matrix with `Y` and use `pure_samples` to indicate which rows of `Y` are pure samples. If `pure_samples` is not specified `references` must be specified. In this case each row of `references` is assumed to be a distinct cell-type. If both `pure_samples` and `references` are specified then multiple rows of `references` may refer be the same cell type, and `pure_samples` specifies to which cell-type each row of `references` corresponds.
find_markers

pure_samples  The pure sample indicies.
(Optional) List of one-dimensional integer. Must implement as.list.
The i-th element of the top-level list is a vector of indicies (rows of Y or references) that are pure samples of type i. If references is not specified then this argument identifies which rows of Y correspond to pure reference samples of which cell-types. If references is specified then this makes same identification but for the references matrix instead.

data_type  Type of expression measurements.
(Optional) One-dimensional string.
An optional string indicating the type of the expression measurements. This is used to set gamma to a pre-determined value based upon the data type. Valid values are for probe-level microarray as “microarray-probe”, gene-level microarray as “microarray-gene” or rna-seq as “rna-seq”. Alternatively can set gamma directly.

gamma  Expression adjustment term.
(Optional) One-dimensional positive numeric.
If provided as a single positive number then that value will be used for gamma and over-ride the value of gamma chosen by the data_type argument. If neither gamma nor data_type are specified then gamma will be set to one.

marker_method  Method used to rank marker genes.
(Optional) One-dimensional string.
The method used to rank genes as markers. If not supplied defaults to “ratio”. Only used if markers are not provided to argument “markers”. Options are
• ‘ratio’ selects and ranks markers by the ratio of the mean expression of each gene in each cell type to the mean of that gene in all other cell types.
• ‘regression’ selects and ranks markers by estimated regression coefficients in a series of regressions with single covariate that is indicator of each type.
• ‘diff’ selects and ranks markers based upon the difference, for each cell type, between the median expression of a gene by each cell type and the median expression of that gene by the second most highly expressed cell type.
• ‘p.value’ selects and ranks markers based upon the p-value of a t-test between the median expression of a gene by each cell type and the median expression of that gene by the second most highly expressed cell type.

Value
List with four elements. “L” is respective ranked markers for each cell type and “V” is the corresponding values of the ranking method (higher are better) used to determine markers and sort them, “M” is the matrix used to create the other two arguments after sorting and subsetting, and “sM” is a sorted version of M.

Examples

truth = shen_orr_ex$annotation$mixture
pure_samples <- lapply(1:3, function(i) {
```r
get_gamma

    which(truth[, i] == 1)
})
Y <- shen_orr_ex$data$log
find_markers(Y=Y, pure_samples=pure_samples,
    data_type='microarray-gene', marker_method='ratio')
```

---

**get_gamma**

*Determine gamma value by data type.*

**Description**

Determine gamma value by data type.

**Usage**

```r
get_gamma(data_type)
```

**Arguments**

- `data_type`  
  Type of expression measurements.  
  (Optional) One-dimensional string.  
  An optional string indicating the type of the expression measurements. This is used to set gamma to a pre-determined value based upon the data type. Valid values are for probe-level microarray as “microarray-probe”, gene-level microarray as “microarray-gene” or rna-seq as “rna-seq”. Alternatively can set gamma directly.

---

**process_markers**

*Determines number of markers n_markers, marker list mrkrs, and gamma.*

**Description**

Determines number of markers `n_markers`, marker list `mrkrs`, and `gamma`.

**Usage**

```r
process_markers(Y, pure_samples, n_markers, data_type, gamma, markers, 
    marker_method)
```
Arguments

**Y**
Expression matrix.
(Required) Two-dimensional numeric. Must implement `as.matrix`.
Each row contains expression measurements for a particular sample. Each column contains the measurements of the same gene over all individuals. Can either contain just the mixture samples to be deconvolved or both the mixture samples and the reference samples. See `pure_samples` and `references` for more details.

**pure_samples**
The pure sample indices.
(Optional) List of one-dimensional integer. Must implement `as.list`.
The i-th element of the top-level list is a vector of indices (rows of `Y` or `references`) that are pure samples of type i. If `references` is not specified then this argument identifies which rows of `Y` correspond to pure reference samples of which cell-types. If `references` is specified then this makes same identification but for the `references` matrix instead.

**n_markers**
Number of marker genes.
(Optional) One-dimensional numeric.
How many markers genes to use for deconvolution. Can either be a single integer, vector of integers (one for each cell type), or single or vector of percentages (numeric in 0 to 1). If a single integer then all cell types use that number of markers. If a vector then the i-th element determines how many marker genes are used for the i-th cell type. If single percentage (in 0 to 1) then that percentage of markers are used for all types. If vector of percentages then that percentage used for each type, respectively. If not specified then top 10% of genes are used.

**data_type**
Type of expression measurements.
(Optional) One-dimensional string.
An optional string indicating the type of the expression measurements. This is used to set gamma to a pre-determined value based upon the data type. Valid values are for probe-level microarray as “microarray-probe”, gene-level microarray as “microarray-gene” or rna-seq as “rna-seq”. Alternatively can set `gamma` directly.

**gamma**
Expression adjustment term.
(Optional) One-dimensional positive numeric.
If provided as a single positive number then that value will be used for gamma and over-ride the value of gamma chosen by the `data_type` argument. If neither `gamma` nor `data_type` are specified then `gamma` will be set to one.

**markers**
Marker gene indices.
(Optional) List of one-dimensional integer.
Top-level list should be same length as `pure_samples`, i.e. one element for each cell type. Each element of the top-level list is a vector of indices (columns of `Y`) that will be considered markers of that particular type. If not supplied then `dtangle` finds markers internally using `find_markers`. Alternatively, one can supply the output of `find_markers` to the `markers` argument.

**marker_method**
Method used to rank marker genes.
(Optional) One-dimensional string.
The method used to rank genes as markers. If not supplied defaults to “ratio”. Only used if markers are not provided to argument “markers”. Options are

- ‘ratio’ selects and ranks markers by the ratio of the mean expression of each gene in each cell type to the mean of that gene in all other cell types.
- ‘regression ‘ selects and ranks markers by estimated regression coefficients in a series of regressions with single covariate that is indicator of each type.
- ‘diff’ selects and ranks markers based upon the difference, for each cell type, between the median expression of a gene by each cell type and the median expression of that gene by the second most highly expressed cell type.
- ‘p.value’ selects and ranks markers based upon the p-value of a t-test between the median expression of a gene by each cell type and the median expression of that gene by the second most highly expressed cell type.

### Description

A subset of data from Shen-Orr et al. Triplicate samples of liver, brain and lung tissue were extracted from rats. RNA was extracted and mixed in known quantities. Gene expressions were measured using the Affymetrix Rat Genome 230 2.0 Array. True mixture proportions were known from experimental design. Gene expression measurements were summarized by RMA at the log2 level. Cell types reported are Liver, Brain and Lung. Data set introduced in ‘Cell type-specific gene expression differences in complex tissues’ by Shen-Orr et al.

### Usage

```r
shen_orr_ex
```

### Format

List of lists.

- **data**: list of data sets
- **annotation**: annotation for the data set

### Source

Index

*Topic datasets
  shen_orr_ex, 15

baseline_exprs, 2
combine_Y.refs, 3
dtangle, 4
dtangle2, 6
est_phats, 10
find_markers, 6, 9, 11
get_gamma, 13
process_markers, 13
shen_orr_ex, 15