Package 'TTMap'

April 6, 2026

Type Package

Title Two-Tier Mapper: a clustering tool based on topological data analysis

Version 1.33.0

Date 2017-07-12

Author Rachel Jeitziner

Maintainer Rachel Jeitziner <rachel.jeitziner@epfl.ch>

Description
TTMap is a clustering method that groups together samples with the same deviation in comparison to a control group. It is specially useful when the data is small. It is parameter free.

License GPL-2

Suggests BiocStyle, airway

Depends rgl, colorRamps

Imports grDevices,graphics,stats,utils, methods, SummarizedExperiment, Biobase

biocViews Software, Microarray, DifferentialExpression, MultipleComparison, Clustering, Classification

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

git_branch devel

git_last_commit a4bf3ca

git_last_commit_date 2025-10-29

Repository Bioconductor 3.23

Date/Publication 2026-04-05

Contents

TTMap-package 2
calcul_e 3
control_adjustment 4
generate_correlation 6

hyperrectangle_deviation_assessment 7
make_matrices 9
make_matrices-methods 11
ttmap 11
ttmap_sgn_genes 13
write_pcl 15

Index **18**

`TTMap-package` *Two-Tier Mapper: a clustering tool based on topological data analysis*

Description

TTMap is a clustering method that groups together samples with the same deviation in comparison to a control group. It is specially useful when the data is small. It is parameter free.

Details

The DESCRIPTION file: TTMap/DESCRIPTION Version 1.0

Author(s)

Rachel Jeitziner Maintainer: Rachel Jeitziner <rachel.jeitziner@epfl.ch>

References

R. Jeitziner et al., TTMap, 2018, DOI:arXiv:1801.01841

See Also

rgl, colorRamps

Examples

```
#to be found in \code{\link[TTMap]{ttmap_sgn_genes}}
```

calcul_e *Calculation of the value of epsilon*

Description

Calculation of the value of epsilon

Usage

```
calcul_e(dd5, pvalcutoff = 0.95, tt1, alpha = 1, S =
colnames(tt1$Normal.mat))
calcul_e_single(dd5, pvalcutoff = 0.95, tt1, alpha = 1, S =
colnames(tt1$Normal.mat))
```

Arguments

dd5	distance matrix as created by generate_mismatch_distance
pvalcutoff	cutoff of 0.05 percent (default) or less
tt1	output of control_adjustment
alpha	a cutoff value for the FC between the group of control and the disease group
S	subset of columns to be considered

Value

al	number representing the cutoff to choose for the relatedness with dd5

Author(s)

Rachel Jeitziner

See Also

control_adjustment, hyperrectangle_deviation_assessment, ttmap_sgn_genes, generate_mismatch_distance

Examples

```
##--
library(airway)
data(airway)
airway <- airway[rowSums(assay(airway))>80,]
assay(airway) <- log(assay(airway)+1,2)
ALPHA <- 1
the_experiment <- TTMap::make_matrices(airway,
seq_len(4), seq_len(4) + 4,
rownames(airway), rownames(airway))
TTMAP_part1prime <-TTMap::control_adjustment(
normal.pcl = the_experiment$CTRL,
```

```
tumor.pcl = the_experiment$TEST,
normalname = "The_healthy_controls",
dataname = "Effect_of_cancer",
org.directory = tempdir(), e = 0, P = 1.1, B = 0);
Kprime <- 4;
TTMAP_part1_hda <-
TTMap::hyperrectangle_deviation_assessment(x =
TTMAP_part1prime,
k = Kprime,dataname = "Effect_of_cancer",
normalname = "The_healthy_controls");
annot <- c(paste(colnames(
the_experiment$TEST[,-(seq_len(3))]), "Dis", sep = "."),
paste(colnames(the_experiment$CTRL[,
-seq_len(3)]), "Dis", sep = "."))
dd5_sgn_only <-TTMap::generate_mismatch_distance(
TTMAP_part1_hda,
select=rownames(TTMAP_part1_hda$Dc.Dmat), alpha = ALPHA)
e <- TTMap::calcul_e(dd5_sgn_only, 0.95, TTMAP_part1prime, 1)
```

`control_adjustment` *Calculates a corrected control group, discovers outliers in it.*

Description

`control_adjustment` function finds outliers in the control group and removes them

Usage

```
control_adjustment(normal.pcl, tumor.pcl, normalname, dataname,
org.directory = "", A = 1, e = 0, meth = 0, P = 1.1, B = 0)
```

Arguments

`normal.pcl`	the control matrix with annotation as obtained by $CTRL from `make_matrices`
`tumor.pcl`	the disease/test data matrix with annotation as obtained by $TEST from `make_matrices`
`normalname`	A name for the corrected control files
`dataname`	the name of the project
`org.directory`	where the outputs should be saved
`A`	integer if A=0 then the difference to the median is calculated otherwise the difference to the mean.
`e`	integer giving how far to the median an outlier is at least
`meth`	value or method that defines how to replace outliers, default is set to replace by the median
`P`	if more than P percent of features are outliers the feature is removed, by default all are kept
`B`	Batch vector a vector for normal and test samples with a same number corresponding to a same batch

Details

control_adjustment calculates a corrected control group, discovers outliers in it.

Value

Several files are created

`paste(org.directory, normalname, ".normMesh", sep = "")`
: The normal matrix with only common features with the test matrix. This file is only created if the two have different rows

`paste(org.directory, dataname, ".normMesh", sep = "")`
: The test matrix with only common features with the normal matrix. This file is only created if the two have different rows.

`mean_vs_variance.pdf`
: A pdf showing a plot of the mean (X axis) against the variances (Y axis) of each feature

`mean_vs_variance_after_correction.pdf`
: A pdf showing a plot of the mean (X axis) against the variances (Y axis) of each feature after correction of the control group

`na_numbers_per_row.txt`
: number of outliers per row

`na_numbers_per_col.txt`
: number of outliers per column

And values of ttmap_part1_ctrl_adj

`e`	Selected criteria for what is an outlier
`tag.pcl`	Annotation of features, ID of features and weight
`Normal.mat`	The control matrix without annotation and only with the common rows with Disease.mat
`Disease.mat`	The test/disease matrix without annotation and only with the common rows with Disease.mat
`flat.Nmat`	A list $mat being the corrected control matrix $m a record of the different numbers of removed genes per sample
`record`	numbers recording the number of columns in Disease.mat and Normal.mat
`B`	The batch vector B introduced in the begining
`U1`	The different batches in Normal.mat
`U2`	The different batches in Disease.mat

Author(s)

Rachel Jeitziner

See Also

hyperrectangle_deviation_assessment, ttmap ttmap_sgn_genes

Examples

```
##--
library(airway)
data(airway)
airway <- airway[rowSums(assay(airway))>80,]
assay(airway) <- log(assay(airway)+1,2)
ALPHA <- 1
the_experiment <- TTMap::make_matrices(airway,
seq_len(4), seq_len(4) + 4,
rownames(airway), rownames(airway))
TTMAP_part1prime <-TTMap::control_adjustment(
normal.pcl = the_experiment$CTRL,
tumor.pcl = the_experiment$TEST,
normalname = "The_healthy_controls",
dataname = "Effect_of_cancer",
org.directory = tempdir(), e = 0, P = 1.1, B = 0);
```

`generate_correlation` *Generates different distance matrices*

Description

Single cell complete mismatch distance, single cell complete mismatch distance with a parameter of cutoff, mismatch distance, correlation distance, p-value of correlation test distance and euclidean distance.

Usage

```
generate_single_cell_complete_mismatch(ttmap_part1_hda,
select, alpha = 1)
generate_single_cell_mismatch_with_parameter(ttmap_part1_hda,
select, alpha = 1)
generate_correlation(ttmap_part1_hda, select)
generate_euclidean(ttmap_part1_hda, select)
generate_mismatch_distance(ttmap_part1_hda, select, alpha = 1)
generate_p_val_correlation(ttmap_part1_hda, select)
```

Arguments

`ttmap_part1_hda`	an object given back by `hyperrectangle_deviation_assessment`
`select`	A sublist of rownames of ttmap_part1_hda$Dc.Dmat
`alpha`	A real number corresponding to a cutoff

Details

If one is interested only in clustering samples according to a list of genes belonging to a certain pathway, then this list is provided to the parameter select. Alpha is a cutoff for deviations that should be considered as noise, for gene expression data such as normalised RNA-seq or microarrays for instance a cutoff of 1, corresponding to a two fold change is being chosen.

Value

Distance matrix

Author(s)

Rachel Jeitziner

Examples

```
ttmap_part1_hda <- list()
ttmap_part1_hda$Dc.Dmat <- matrix(c(-1, 2, 0, -4, 5, 6), nrow = 2)
rownames(ttmap_part1_hda$Dc.Dmat) <- c("Gene1", "Gene2")
colnames(ttmap_part1_hda$Dc.Dmat) <- c("A", "B", "C")
dd <- TTMap::generate_mismatch_distance(ttmap_part1_hda, select =
rownames(ttmap_part1_hda$Dc.Dmat))
dd <- TTMap::generate_euclidean(ttmap_part1_hda, select =
rownames(ttmap_part1_hda$Dc.Dmat))
```

---

hyperrectangle_deviation_assessment

Calculation of deviation components

Description

`hyperrectangle_deviation_assessment` function calculates the hyperrectangle deviation assessment (HDA) that calculates the deviation components using `normal_hda2` which calculates the normal component of the test sample and `deviation_hda2` which calculates the deviation component.

Usage

```
hyperrectangle_deviation_assessment(x,
k = dim(x$Normal.mat)[2], dataname,
normalname,Org.directory = getwd())
```

Arguments

`x`	output object given back by `control_adjustment`, list
`k`	A factor if not all the lines in the control group should be kept
`dataname`	the name of the project
`normalname`	A name for the corrected control files
`Org.directory`	where the outputs should be saved

Details

The function performs the hyperrectangle deviation assessment (HDA)

Value

Outputs

`Tdis.pcl`	The matrix of the deviation components for each test sample
`Tnorm.pcl`	The matrix of the normal components for each test sample
`NormalModel.pcl`	The normal model used

Values

`Dc.Dmat`	the deviation component matrix composed of the deviation components of all the samples in the test group
`m`	the values of the filter function per sample in the test group

Author(s)

Rachel Jeitziner

See Also

`control_adjustment`, `hyperrectangle_deviation_assessment`, `ttmap_sgn_genes`

Examples

```
##a full example can be found in ttmap_sgn_genes
##--
library(airway)
data(airway)
airway <- airway[rowSums(assay(airway))>80,]
assay(airway) <- log(assay(airway)+1,2)
ALPHA <- 1
the_experiment <- TTMap::make_matrices(airway,
seq_len(4), seq_len(4) + 4,
rownames(airway), rownames(airway))
TTMAP_part1prime <-TTMap::control_adjustment(
normal.pcl = the_experiment$CTRL,
tumor.pcl = the_experiment$TEST,
normalname = "The_healthy_controls",
dataname = "Effect_of_cancer",
org.directory = tempdir(), e = 0, P = 1.1, B = 0);
Kprime <- 4;
TTMAP_part1_hda <-
TTMap::hyperrectangle_deviation_assessment(x =
TTMAP_part1prime,
k = Kprime, dataname = "Effect_of_cancer",
normalname = "The_healthy_controls");
```

make_matrices *Prepares the matrices for* control_adjustment

Description

make_matrices generates the control and the test matrice in the right format

Usage

```
make_matrices(mat, col_ctrl, col_test, NAME, CLID,
GWEIGHT = rep(1, dim(mat)[1]), EWEIGHT = 0)
```

Arguments

mat	the gene expressions can be **matrix**, **data.frame**, " RangedSummarizedExperiment", " ExpressionSet" format
col_ctrl	the columns in the matrix "mat" of the control samples
col_test	the columns in the matrix "mat" of the test samples
NAME	Name of genes,or annotation, e.g. WNT4
CLID	Identities of genes,e.g. ENSMUSG00000000001
GWEIGHT	the weight for each gene
EWEIGHT	the weight for each experiment

Details

make_matrices generates the test matrix and the control matrix in the format accepted by control_adjustment from a matrix object

Value

junk	A list containing $CTRL and $TEST the matrices to impute in control_adjustment

Author(s)

Rachel Jeitziner

See Also

control_adjustment, hyperrectangle_deviation_assessment, ttmap_sgn_genes, " RangedSummarizedExperiment"

Examples

```
##--
##--
Aa = 6
B1 = 3
B2 = 3
C0 = 100
D0 = 10000
a0 = 4
b0 = 0.1
a1 = 6
b1 = 0.1
a2 = 2
b2 = 0.5
ALPHA = 1
E = 1
Pw = 1.1
Bw = 0
RA <- matrix(rep(0, Aa * D0), nrow = D0)
RB1 <- matrix(rep(0, B1 * D0), nrow = D0)
RB2 <- matrix(rep(0, B2 * D0), nrow = D0)
RA <- lapply(seq_len(D0 - C0), function(i) rnorm(Aa,
mean = a0, sd = sqrt(b0)))
RA<-do.call(rbind, RA)
RB1<- lapply(seq_len(D0 - C0), function(i) rnorm(B1,
mean = a0, sd = sqrt(b0)))
RB1 <- do.call(rbind, RB1)
RB2 <- lapply(seq_len(D0 - C0), function(i) rnorm(B2,
mean = a0, sd = sqrt(b0)))
RB2 <- do.call(rbind, RB2)
RA_c <- lapply(seq_len(C0), function(i) rnorm(Aa,
mean = a0, sd = sqrt(b0)))
RA_c <- do.call(rbind, RA_c)
RB1_c <- lapply(seq_len(C0), function(i) rnorm(B1,
mean = a1, sd = sqrt(b1)))
RB1_c <- do.call(rbind, RB1_c)
RB2_c <- lapply(seq_len(C0), function(i) rnorm(B2,
mean = a2, sd = sqrt(b2)))
RB2_c <- do.call(rbind, RB2_c)
norm1 <- rbind(RA, RA_c)
dis <- cbind(rbind(RB1, RB1_c), rbind(RB2, RB2_c))
colnames(norm1) <- paste("N", seq_len(Aa), sep = "")
rownames(norm1) <- c(paste("norm", seq_len(D0 - C0), sep = ""),
paste("diff", seq_len(C0), sep = ""))
colnames(dis) <- c(paste("B1", seq_len(B1), sep=""),
paste("B2", seq_len(B2), sep  =""))
rownames(dis)<-c(paste("norm",
seq_len(D0 - C0), sep = ""),
paste("diff", seq_len(C0), sep = ""))
the_experiment <- TTMap::make_matrices(cbind(norm1, dis),
col_ctrl = colnames(norm1),
col_test = colnames(dis), NAME = rownames(norm1),
```

```
CLID = rownames(norm1))
###other example using SummarizedExperiment
library(airway)
data(airway)
airway <- airway[rowSums(assay(airway))>80,]
assay(airway) <- log(assay(airway)+1,2)
the_experiment <- TTMap::make_matrices(airway,
seq_len(4), seq_len(4) + 4,
rownames(airway), rownames(airway))
```

`make_matrices-methods` *Prepares the matrices for* `control_adjustment`

Description

make_matrices generates the control (output $CTRL) and the test (output $TEST) matrice in the right format for `control_adjustment`

Methods

`signature(mat = "data.frame")` Method make_matrice for `data.frame` object.

`signature(mat = "matrix")` Method make_matrice for `matrix` object.

`signature(mat = "SummarizedExperiment")` Method make_matrice for `SummarizedExperiment` object.

`signature(mat = "RangedSummarizedExperiment")` Method make_matrice for `RangedSummarizedExperiment` object.

`signature(mat = "ExpressionSet")` Method make_matrice for `ExpressionSet` object.

`ttmap` *Visualisation of the clustering*

Description

Enables a quick view on the groups in the dataset (globally) and how locally they differ.

Usage

```
ttmap(ttmap_part1_hda, m1,
select = row.names(ttmap_part1_hda$Dc.Dmat),
ddd, e, filename = "TEST", n = 3, ad = 0, bd = 0, piq = 1,
dd = generate_mismatch_distance(ttmap_part1_hda = ttmap_part1_hda,
select = select), mean_value_m1 = "N", ni = 2)
```

Arguments

ttmap_part1_hda	list output of hyperrectangle_deviation_assessment
m1	either a user imputed vector whose names are the names of the samples with addition of .Dis. or by default it is the amount of deviation
select	Should all the features (default) or only a sublist be considered to calculate the distance
ddd	Annotation matrix with rownames the different sample names with addition of .Dis. There can be as many columns as wanted, but only the column n will be selected to annotated the clusters
e	integer parameter defining under which value two samples are considered to be close
filename	Name for the description file annotating the clusters
n	The column to be considered to annotate the clusters
ad	if ad!=0 then the clusters on the output picture will not be annotated
bd	if different than 0 (default), the output will be without outliers of the test data set (clusters composed of only "piq" element)
piq	parameter used to determine what small clusters are, see bd
dd	the distance matrix to be used
mean_value_m1	if == "N" the average of the values in m1 divided by the number of the samples are put into the legend (by default represents the average of the samples in a cluster of the mean-deviation of the features) otherwise it will show the average value of the values in m1 (is useful for instance if m1 represents the age of the samples)
ni	The column to consider to annotate the samples (is put into parenthesis) for the description file

Details

Is the Two-tiers Mapper function. The output is an interactive image of the clusters in the different layers.

Value

all	the clusters in the overall group
low	the clusters in the lower quartile group
mid1	the clusters in the first middle quartile group
mid2	the clusters in the second middle quartile group
high	the clusters in the higher quartile group

Author(s)

Rachel Jeitziner

See Also

control_adjustment, hyperrectangle_deviation_assessment, ttmap_sgn_genes

Examples

```
##--
library(airway)
data(airway)
airway <- airway[rowSums(assay(airway))>80,]
assay(airway) <- log(assay(airway)+1,2)
ALPHA <- 1
the_experiment <- TTMap::make_matrices(airway,
seq_len(4), seq_len(4) + 4,
rownames(airway), rownames(airway))
TTMAP_part1prime <-TTMap::control_adjustment(
normal.pcl = the_experiment$CTRL,
tumor.pcl = the_experiment$TEST,
normalname = "The_healthy_controls",
dataname = "Effect_of_cancer",
org.directory = tempdir(), e = 0, P = 1.1, B = 0);
Kprime <- 4;
TTMAP_part1_hda <-
TTMap::hyperrectangle_deviation_assessment(x =
TTMAP_part1prime,
k = Kprime,dataname = "Effect_of_cancer",
normalname = "The_healthy_controls");
annot <- c(paste(colnames(
the_experiment$TEST[,-(seq_len(3))]),"Dis", sep = "."),
paste(colnames(the_experiment$CTRL[,
-seq_len(3)]), "Dis", sep = "."))
annot <- cbind(annot, annot)
rownames(annot)<-annot[, 1]
dd5_sgn_only <-TTMap::generate_mismatch_distance(
TTMAP_part1_hda,
select=rownames(TTMAP_part1_hda$Dc.Dmat), alpha = ALPHA)
TTMAP_part2 <-
TTMap::ttmap(TTMAP_part1_hda, TTMAP_part1_hda$m,
select = rownames(TTMAP_part1_hda$Dc.Dmat), annot,
e = TTMap::calcul_e(dd5_sgn_only, 0.95, TTMAP_part1prime, 1),
filename = "first_comparison", n =  1, dd = dd5_sgn_only)
```

---

ttmap_sgn_genes *Gives a list of associated genes per cluster*

Description

ttmap_sgn_genes function

Usage

```
ttmap_sgn_genes(ttmap_part2_gtlmap, ttmap_part1_hda,
ttmap_part1_ctrl_adj, c, n = 2, a = 0,
filename = "TEST2", annot = ttmap_part1_ctrl_adj$tag.pcl,
col = "NAME", path = getwd(), Relaxed = 1)
ttmap_sgn_genes_inter2(q, ttmap_part1_hda, alpha = 0)
ttmap_sgn_genes_inter(q, ttmap_part1_hda, alpha = 0)
```

Arguments

ttmap_part2_gtlmap	output of ttmap
ttmap_part1_hda	output of hyperrectangle_deviation_assessment
ttmap_part1_ctrl_adj	output of control_adjustment
c	annotation file of the samples
n	column to give the name to the cluster
a	cutoff to be considered different than noise
filename	Name of the files
annot	annotation file
col	which column should be considered to annotate the features
path	where to put the output files
Relaxed	If Relaxed then one allows sample to be as the control and for all the others in one cluster to be going in the same direction (more than alpha) otherwise all the features must be deviating to be considered a significant feature
q	The sample in one cluster
alpha	cutoff to be considered different than noise inherited by a

Details

Is giving per cluster the features that vary in the same direction

Value

generates a file per cluster of significant features with an annotation

Author(s)

Rachel Jeitziner

Examples

```
##--
library(airway)
data(airway)
airway <- airway[rowSums(assay(airway))>80,]
assay(airway) <- log(assay(airway)+1,2)
ALPHA <- 1
the_experiment <- TTMap::make_matrices(airway,
seq_len(4), seq_len(4) + 4,
rownames(airway), rownames(airway))
TTMAP_part1prime <-TTMap::control_adjustment(
normal.pcl = the_experiment$CTRL,
tumor.pcl = the_experiment$TEST,
normalname = "The_healthy_controls",
dataname = "Effect_of_cancer",
org.directory = tempdir(), e = 0, P = 1.1, B = 0);
Kprime <- 4;
TTMAP_part1_hda <-
TTMap::hyperrectangle_deviation_assessment(x =
TTMAP_part1prime,
k = Kprime,dataname = "Effect_of_cancer",
normalname = "The_healthy_controls");
annot <- c(paste(colnames(
the_experiment$TEST[,-(seq_len(3))]),"Dis", sep = "."),
paste(colnames(the_experiment$CTRL[,
-seq_len(3)]), "Dis", sep = "."))
annot <- cbind(annot, annot)
rownames(annot)<-annot[, 1]
dd5_sgn_only <-TTMap::generate_mismatch_distance(
TTMAP_part1_hda,
select=rownames(TTMAP_part1_hda$Dc.Dmat), alpha = ALPHA)
TTMAP_part2 <-
TTMap::ttmap(TTMAP_part1_hda, TTMAP_part1_hda$m,
select = rownames(TTMAP_part1_hda$Dc.Dmat), annot,
e = TTMap::calcul_e(dd5_sgn_only, 0.95, TTMAP_part1prime, 1),
filename = "first_comparison", n =  1, dd = dd5_sgn_only)
TTMap::ttmap_sgn_genes(TTMAP_part2, TTMAP_part1_hda,
TTMAP_part1prime, annot,
n = 2, a = 1, filename = "first_list_of_genes",
annot = TTMAP_part1prime$tag.pcl, col = "NAME",
path = getwd(), Relaxed = 1)
```

`write_pcl` *Reading, writing and annotation files*

Description

Reading (`read_pcl`), writing (`write_pcl`) files and annotating matrices (mat2pcl)

Usage

```
mat2pcl(mat, tag)
write_pcl(df, dataname, fileaddress = "")
read_pcl(filename, na.type = "", Nrows = -1,
Comment.char = "", ...)
```

Arguments

df	PCL object to be saved
dataname	Name of the file
fileaddress	Where to save the file
filename	File name to be loaded on R
na.type	feels the parameter na.strings of read.table
Nrows	Number of rows to be ignored (nrows of read.table)
Comment.char	comment.char of read.table
...	other read.table arguments
mat	matrix to be changed in annotated
tag	annotation

Details

The file (called filename) MUST contain 3 columns before the actual values, which are called CLID, NAME and GWEIGHT, described bellow. The first row must be the header of the columns (starting with CLID,NAME and GWEIGHT) and the second row must be EWEIGHT. Representing how much weight each column has: if some columns are n replicates they can have each a weight of 1/n.

Value

Data frame composed of

CLID	Column called CLID which is the ID of the features, which will then be the rownames of the dataframe
NAME	A possibly longer name, more meaningfull than CLID, text format
GWEIGHT	A weight for each gene or feature. If some genes are less important than others or only a pathway should be selected than the file (called filename) should have this information
Matrix	The matrix with numbers of the different observations

Author(s)

Rachel Jeitziner

See Also

control_adjustment

Examples

```
library(airway)
data(airway)
airway <- airway[rowSums(assay(airway))>80,]
assay(airway) <- log(assay(airway)+1,2)
ALPHA <- 1
to_be_saved <- TTMap::make_matrices(airway,
seq_len(4), seq_len(4) + 4,
rownames(airway), rownames(airway))
TTMap::write_pcl(to_be_saved, "tempfile()", getwd())
```

Index

∗ **TTMap**
    TTMap-package, 2
∗ **methods**
    make_matrices-methods, 11

calcul_e, 3
calcul_e_single *(*calcul_e*)*, 3
control_adjustment, *3, 4*, 4, *5*, *7–9*, *11*, *13, 14*, *16*

generate_correlation, 6
generate_euclidean *(*generate_correlation*)*, 6
generate_mismatch_distance, *3*
generate_mismatch_distance *(*generate_correlation*)*, 6
generate_p_val_correlation *(*generate_correlation*)*, 6
generate_single_cell_complete_mismatch *(*generate_correlation*)*, 6
generate_single_cell_mismatch_with_parameter *(*generate_correlation*)*, 6

hyperrectangle_deviation_assessment, *3*, *5–7*, 7, *8, 9*, *12–14*

make_matrices, *4*, *9*, 9
make_matrices,data.frame-method *(*make_matrices-methods*)*, 11
make_matrices,ExpressionSet-method *(*make_matrices-methods*)*, 11
make_matrices,matrix-method *(*make_matrices-methods*)*, 11
make_matrices,RangedSummarizedExperiment-method *(*make_matrices-methods*)*, 11
make_matrices,SummarizedExperiment-method *(*make_matrices-methods*)*, 11
make_matrices-methods, 11
mat2pcl *(*write_pcl*)*, 15

read_pcl, *15*
read_pcl *(*write_pcl*)*, 15

TTMap *(*TTMap-package*)*, 2
ttmap, *5*, 11, *14*
TTMap-package, 2
ttmap_sgn_genes, *3*, *5*, *8, 9*, *13*, 13
ttmap_sgn_genes_inter *(*ttmap_sgn_genes*)*, 13
ttmap_sgn_genes_inter2 *(*ttmap_sgn_genes*)*, 13

write_pcl, *15*, 15