The package ABAEnrichment is designed to test for enrichment of user defined candidate genes in the set of expressed genes in different human brain regions. The core function `aba_enrich` integrates the expression of the candidate gene set (averaged across donors) and the structural information of the brain using an ontology, both provided by the Allen Brain Atlas project. `aba_enrich` interfaces the ontology enrichment software FUNC to perform the statistical analyses. Additional functions provided in this package like `get_expression` and `plot_expression` facilitate exploring the expression data.

License  GPL (>= 2)
Imports  Rcpp (>= 0.11.5), gplots (>= 2.14.2), ABAData (>= 0.99.2)
Depends  R (>= 3.2)
LinkingTo  Rcpp
Suggests  BiocStyle, knitr, testthat
VignetteBuilder  knitr
biocViews  GeneSetEnrichment, GeneExpression
NeedsCompilation  yes

R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABAEnrichment-package</td>
<td>2</td>
</tr>
<tr>
<td>aba_enrich</td>
<td>3</td>
</tr>
<tr>
<td>get_expression</td>
<td>6</td>
</tr>
<tr>
<td>get_name</td>
<td>8</td>
</tr>
<tr>
<td>get_sampled_substructures</td>
<td>9</td>
</tr>
<tr>
<td>get_superstructures</td>
<td>10</td>
</tr>
<tr>
<td>plot_expression</td>
<td>11</td>
</tr>
</tbody>
</table>

Index 14
Description

The package ABAEnrichment is designed to test for enrichment of user defined candidate genes in the set of expressed genes in different human brain regions. The package integrates the expression of the candidate gene set (averaged across donors) and the structural information of the brain using an ontology, both provided by the Allen Brain Atlas project [1-4]. The statistical analysis is performed by the core function `aba_enrich` which interfaces the ontology enrichment software FUNC [5]. Additional functions provided in this package like `get_expression` and `plot_expression` facilitate exploring the expression data.

Details

Package: ABAEnrichment  
Type: Package  
Version: 1.3.5  
Date: 2016-10-13  
License: GPL (>= 2)

For details see vignette("ABAEnrichment",package="ABAEnrichment")

Author(s)

Steffi Grote  
Maintainer: Steffi Grote <steffi_grote@eva.mpg.de>

References


See Also

vignette("ABAEnrichment",package="ABAEnrichment")  
vignette("ABAData",package="ABAData")  
aba_enrich  
get_expression  
plot_expression
aba_enrich

**Description**

Tests for enrichment of user defined candidate genes in the set of expressed protein coding genes in different human brain regions. It integrates the expression of the candidate gene set (averaged across donors) and the structural information of the brain using an ontology, both provided by the Allen Brain Atlas project [1-4]. The statistical analysis is performed by interfacing the ontology enrichment software FUNC [5].

**Usage**

```r
aba_enrich(genes, dataset = 'adult', test = 'hyper',
            cutoff_quantiles = seq(0.1, 0.9, 0.1), n_randsets = 1000, gene_len = FALSE, circ_chrom = FALSE)
```

**Arguments**

- **genes**: If `test = 'wilcoxon'` a numeric vector of scores. If `test = 'hyper'` (default) a binary vector with 1 for candidate genes and 0 for background genes. If no background genes are defined, all remaining protein coding genes are used as background. The names of the vector are the gene identifiers: either Entrez-ID, Ensembl-ID or HGNC-symbol. For `test = 'hyper'` the names of the vector can also describe chromosomal regions (‘chr:start-stop’).

- **dataset**: ‘adult’ for the microarray dataset of adult human brains; ‘5_stages’ for RNA-seq expression data for different stages of the developing human brain, grouped into 5 developmental stages; ‘dev_effect’ for a developmental effect score. For details see vignette(“ABAData”,package="ABAData”).

- **test**: ‘hyper’ (default) for the hypergeometric test or ‘wilcoxon’ for the Wilcoxon rank test.

- **cutoff_quantiles**: the FUNC enrichment analyses will be performed for the sets of expressed genes at given expression quantiles defined in this vector [0,1].

- **n_randsets**: integer defining the number of random sets created to compute the FWER.

- **gene_len**: logical. If `test = 'hyper'` the probability of a background gene to be chosen as a candidate gene in a random set is dependent on the gene length.

- **circ_chrom**: logical. When `genes` defines chromosomal regions, `circ_chrom = TRUE` uses background regions from the same chromosome and allows randomly chosen blocks to overlap multiple background regions. Only if `test = 'hyper'`. 
aba_enrich

Details

The function aba_enrich performs enrichment analyses of candidate genes within expressed protein coding genes in human brain regions. The brain regions are categorized using an ontology. Enrichment of candidate genes is tested using the hypergeometric or the Wilcoxon rank test of the ontology enrichment software FUNC [5].

The hypergeometric test evaluates the enrichment of expressed candidate genes compared to a set of expressed background genes for each brain region. The background genes can be defined explicitly like the candidate genes or, as default, consist of all protein coding genes from the dataset, which are not candidate genes.

To account for multiple testing the FWER is computed using random permutations of candidate and background genes (see package vignette for details). By default each gene is chosen with the same probability as a random candidate gene. If gene_len = TRUE the probability is dependent on the gene length, i.e. a gene that is twice as long as another gene is also twice as likely to be chosen as a random candidate gene.

Instead of defining candidate and background genes explicitly in the genes input vector, it is also possible to define entire chromosomal regions as candidate and background regions. The expression enrichment is then tested for all protein coding genes located in or overlapping the candidate region on the plus or the minus strand. The gene coordinates used to identify those genes were obtained from http://grch37.ensembl.org/biomart/martview/. For the random permutations used to compute the FWER, blocks as long as candidate regions are chosen from the background regions and genes contained in these blocks are considered candidate genes. The output of aba_enrich is identical to the one that is produced for single genes.

To define chromosomal regions in the input vector, the names of the 1/0 vector have to be of the form chr:start-stop, where ‘start’ always has to be smaller than ‘stop’. Note that this option requires the input of background regions. If multiple candidate regions are provided, in the randomsets they are placed randomly, but non-overlapping into the background regions. If the background regions are relatively small, it can happen that the remaining background regions available (after a candidate region has been placed there) are too small for the next candidate region to fit entirely and non-overlapping. In this case the random selection of candidate regions inside the background regions is restarted. If this fails 10 times aba_enrich quits.

An alternative method to choose random blocks from the background regions can be used with the option circ_chrom=TRUE. Every candidate region is then compared to background regions on the same chromosome. And in contrast to the default circ_chrom=FALSE, randomly chosen blocks do not have to be located inside a single background region, but are allowed to overlap multiple background regions. This means that a randomly chose block can consist of the end of the last background region and the beginning of the first background region on a given chromosome.

The Wilcoxon rank test does not compare candidate and background genes, but the user defined scores associated with the candidate genes, i.e. it compares the ranks of the scores of expressed genes in a given brain region to the ranks of all candidate genes that are expressed somewhere in the brain.

In addition to gene expression the enrichment may refer to a developmental effect score, which describes how much a gene’s expression changes over time. Three different datasets can be used with aba_enrich: first, the developmental effect score, second, microarray data from adult donors and third, RNA-seq data from donors of five different developmental stages (prenatal, infant, child, adolescent, adult). In the latter case the analyses are performed independently for each developmental stage.

The expression definition for genes is variable. Different quantiles of expression over all genes are used (e.g. the lowest 40% of gene expression are ‘not expressed’ and the upper 60% are ‘expressed’ for a quantile of 0.4). These cutoffs are set with the parameter cutoff_quantiles and an analysis is run for every cutoff separately.
**Value**

A list with components

- **results** a dataframe with the FWERs from the enrichment analyses per brain region and age category, ordered by 'age_category', 'times_FWER_under_0.05', 'mean_FWER' and 'min_FWER'; with 'min_FWER' for example denoting the minimum FWER for expression enrichment of the candidate genes in this brain region across all expression cutoffs. 'FWERs' is a semicolon separated string with the single FWERs for all cutoffs. 'equivalent_structures' is a semicolon separated string that lists structures with identical expression data due to lack of independent expression measurements in all regions.

- **genes** a vector of the requested genes, excluding those genes for which no expression data is available and which therefore were not included in the enrichment analysis.

- **cutoffs** a dataframe with the expression values that correspond to the requested cutoff quantiles.

**Author(s)**

Steffi Grote

**References**


**See Also**

vignette("ABAEnrichment",package="ABAEnrichment")

vignette("ABAData",package="ABAData")

get_expression

plot_expression

get_name

get_sampled_substructures

get_superstructures

**Examples**

#### Note that arguments 'cutoff_quantiles' and 'n_randsets' are reduced to lower computational time in the examples.

Perform gene expression enrichment analysis on 13 candidate genes in five developmental stages of the human brain using the hypergeometric test implemented in FUNC[S]

create input vector with candidate genes (HGNC-symbols)
```r
genes = rep(1, 13)
names(genes) = c("NCAPG", "APOL4", "NGFR", "NXPH4", "C21orf59", "CACNG2", "AGTR1", "ANO1", "BTBD3", "MTUS1", "CALB1", "GYG1", "PAX2")

## run enrichment analysis
res = aba_enrich(genes, dataset = "5_stages", cutoff_quantiles = c(0.5, 0.7, 0.9), n_randsets = 100)

## get FWERs for enrichment of candidate genes among expressed genes
fwers = res[[1]]

## see results for the brain regions with highest enrichment for children (age_category = 3)
head(fwers[fwers[,1] == 3,])

## see the input genes vector (only genes with expression data available)
res[2]

## see the expression values that correspond to the requested cutoff quantiles
res[3]

#### Perform the same analysis, but with random sets dependent on gene length
res = aba_enrich(genes, dataset = "5_stages", cutoff_quantiles = c(0.5, 0.7, 0.9), n_randsets = 100, gene_len = TRUE)

#### Perform gene expression enrichment analysis for a chromosomal region
#### for the adult human brain using the hypergeometric test implemented in FUNC[5]
## create input vector with a candidate regions on chromosome 3 and background regions on chromosome 3, 4 and 5
genes = c(1, rep(0, 6))

## run enrichment analysis for the 'adult' dataset
res = aba_enrich(genes, dataset = "adult", cutoff_quantiles = c(0.5, 0.7, 0.9), n_randsets = 100)

## look at he results from the enrichment analysis
head(res[[1]])

## see which genes are located in the candidate regions
input_genes = res[[2]]
candidate_genes = input_genes[input_genes == 1]
candidate_genes

#### Perform gene expression enrichment analysis on 15 candidate genes in the
#### adult human brain using the wilcoxon rank test implemented in FUNC[5]
## create input vector with random scores associated with the candidate genes (Entrez-Ids)
genes = sample(1:50, 15)
names(genes) = c(324, 8312, 673, 1029, 3417, 3418, 8085, 3845, 9968, 5290, 5727, 5728)

## run enrichment analysis
res = aba_enrich(genes, dataset = "adult", test = "wilcoxon", cutoff_quantiles = c(0.2, 0.5, 0.8), n_randsets = 100)

## see results for the brain regions with highest enrichment
head(res[[1]])

## see the input genes vector and the expression values that correspond
## to the requested cutoff quantiles
res[2:3]
```

---

**get_expression**

Get expression data for given genes and brain structure ids

---

**Description**

Expression data obtained from the Allen Brain Atlas project [1-4].

**Usage**

```r
get_expression(structure_ids, gene_ids = NA, dataset = NA, background = FALSE)
```
get_expression

Arguments

structure_ids vector of brain structure ids, e.g. 'Allen:10208'.
gene_ids vector of gene identifiers, either Entrez-ID, Ensembl-ID or HGNC-symbol. If not defined, genes from previous enrichment analysis with aba_enrich are used.
dataset 'adult' for the microarray dataset of adult human brains; '5_stages' for RNA-seq expression data of the developing human brain, grouped into 5 developmental stages; 'dev_effect' for a developmental effect score. If not defined, dataset from last enrichment analysis with aba_enrich are used.
background logical indicating whether expression from background genes should be included. Only used when gene_ids and dataset are NA and test from preceding aba_enrich call is 'hyper'.

Details

Get gene expression in defined brain regions from adult or developing humans, or a developmental effect score for the developing human brain. Expression data is obtained from the Allen Brain Atlas project [1-4], averaged across donors, and for the developing human brain divided into five major age categories. The developmental effect score is based on expression data of the developing human brain. If gene_ids and dataset are not specified, the genes and dataset from the last enrichment analysis with aba_enrich are used, since it may be a common case to first run the enrichment analysis and then look at the expression data. If a requested brain region has no expression data annotated, data from sampled substructures of this region is returned. Please refer to the ABAData package vignette for details on the datasets.

Value

A matrix with expression values or developmental effect scores per brain region (rows) and gene (columns).
For expression data from the developing human brain ('5_stages') it is a list with an expression matrix for each of the 5 developmental stages.

Author(s)

Steffi Grote

References

get_name

See Also
vignette("ABAEnrichment",package="ABAEnrichment")
vignette("ABAData",package="ABAData")
plot_expression
aba_enrich
get_name
get_sampled_substructures

Examples

## get expression data for six genes in the brain structure 'Allen:4010'
get_expression(structure_ids=c('Allen:4010'),gene_ids=c(324,8312,673,1029,64764,1499),
               dataset='adult')
## get expression data of six genes in two brain regions from developing human brain,
## each of the five list elements corresponds to an age category
get_expression(structure_ids=c('Allen:10657','Allen:10208'), gene_ids=c('ENSG00000168036',
                      'ENSG00000157764', 'ENSG00000182158', 'ENSG00000147889'),dataset='5_stages')

get_name

Get the full name of a brain region given structure ids

Description

Returns the full name of brain regions given the structure IDs, e.g. 'Allen:10657' as used throughout
the ABAEnrichment package. The full name is composed of an acronym and the name as used by
the Allen Brain Atlas project [1-2].

Usage

get_name(structure_ids)

Arguments

structure_ids a vector of brain structure IDs, e.g. c('Allen:10657','Allen:10173')

Value

vector of the full names of the brain structures; composed of acronym, underscore and name.

Note

The acronym is added because the names alone are not unique.

Author(s)

Steffi Grote
**get_sampled_substructures**

**References**


**See Also**

*get_sampled_substructures*
*get_superstructures*

**Examples**

```r
## get the full names of the brain structures 'Allen:10657' and 'Allen:10225'
get_name(c('Allen:10657', 'Allen:10225'))
```

---

**get_sampled_substructures**

*Return sampled substructures of a given brain region*

**Description**

The function returns for a given brain structure ID all its substructures with available expression data, potentially including the structure itself.

**Usage**

```r
get_sampled_substructures(structure_id)
```

**Arguments**

- `structure_id` a brain structure ID, e.g. 'Allen:10657'

**Details**

The ontology enrichment analysis in `aba_enrich` tests all brain regions for which data is available, although the region might not have been sampled directly. In this case the region inherits the expression data from its substructures with available expression data. The function `get_sampled_substructures` helps to explore where the expression data for a brain region came from.

**Value**

vector of brain structure IDs that contains all substructures of the requested brain region that were sampled.

**Author(s)**

Steffi Grote


References


See Also

vignette("ABAEnrichment", package="ABAEnrichment")
vignette("ABAData", package="ABAData")
aba_enrich
get_name
get_superstructures

Examples

```r
## get the brain structures from which the brain structures 'Allen:4010' and 'Allen:10208'
## inherit their expression data
get_sampled_substructures('Allen:4010')
get_sampled_substructures('Allen:10208')
```

get_superstructures  Returns all superstructures of a brain region using the Allen Brain Atlas ontology

Description

Returns all superstructures of a brain region and the brain region itself given a structure ID, e.g. 'Allen:10657' as used throughout the ABAEnrichment package. The output vector contains the superstructures according to the hierarchy provided by the Allen Brain Atlas ontology [1,2] beginning with the root ('brain' or 'neural plate') and ending with the requested brain region.

Usage

get_superstructures(structure_id)

Arguments

structure_id a brain structure ID, e.g. 'Allen:10657'

Value

vector of brain structure IDs that contains all superstructures of the requested brain region and the brain region itself. The order of the brain regions follows the hierarchical organization of the brain.

Note

The ontologies for the adult and the developing human brain are different.
plot_expression

Author(s)

Steffi Grote

References


See Also

get_name
get_sampled_substructures

Examples

```r
## Get the ids of the superstructures of the precentral gyrus (adult brain ontology)
get_superstructures('Allen:4010')
## Get the ids and the names of the superstructures of the dorsolateral prefrontal cortex
## (developing brain ontology)
data.frame(hierarchy=get_name(get_superstructures("Allen:10173")))
```

plot_expression  Plot expression data for given genes and brain structure ids

Description

The function produces a heatmap (heatmap.2 from package gplots) of gene expression in defined brain regions from adult or developing humans, or a developmental effect score for the developing human brain. Expression data is obtained from the Allen Brain Atlas project [1-4], averaged across donors, and for the developing human brain divided into five major age categories. If gene_ids and dataset are not specified, the genes and dataset from the last enrichment analysis with aba_enrich are used, since it may be a common case to first run the enrichment analysis and then look at the expression data. If a requested brain region has no expression data annotated, data from sampled substructures of this region is returned.

Usage

```r
plot_expression(structure_ids, gene_ids = NA, dataset = NA, background = FALSE, dendro = TRUE, age_category = 1)
```

Arguments

- `structure_ids`: vector of brain structure ids, e.g. "Allen:10208".
- `gene_ids`: vector of gene identifiers, either Entrez-ID, Ensembl-ID or HGNC-symbol. If not defined, genes from previous enrichment analysis with aba_enrich are used.
plot_expression

dataset 'adult' for the microarray dataset of adult human brains; '5_stages' for RNA-seq expression data of the developing human brain, grouped into 5 developmental stages; 'dev_effect' for a developmental effect score. If not defined, dataset from last enrichment analysis with `aba_enrich` are used.

background logical indicating whether expression from background genes should be included. Only used when `gene_ids` and `dataset` are NA so that genes from the last enrichment analysis with `aba_enrich` are used and when this analysis was performed using the hypergeometric test.

dendro logical indicating whether rows and columns should be rearranged with a dendrogram based on row/column means (using `hclust`). If `FALSE` and if `gene_ids` and `dataset` are NA so that genes from the last enrichment analysis with `aba_enrich` are used, the genes are arranged according to the last `aba_enrich` execution: for a hypergeometric test the genes are grouped into candidate and background genes (indicated by a coloured side-bar with red and black, respectively) and for a Wilcoxon rank test the genes are ordered by the scores which they were given for the Wilcoxon rank test, which are also indicated by a side-bar.

age_category an integer between 1 and 5 indicating the age category if `dataset = '5_stages'`.

Value

Invisibly, a list with components

- `rowInd` row index permutation vector as returned by `order.dendrogram`
- `colInd` column index permutation vector.
- `call` the matched call
- `carpet` reordered 'x' values used to generate the main 'carpet'
- `rowDendrogram` row dendrogram, if present
- `colDendrogram` column dendrogram, if present
- `breaks` values used for color break points
- `col` colors used
- `colorTable` A three-column data frame providing the lower and upper bound and color for each bin

Author(s)

Steffi Grote

References

**plot_expression**

**See Also**

vignette("ABAEnrichment", package="ABAEnrichment")
vignette("ABAData", package="ABAData")
get_expression
aba_enrich
get_name
get_sampled_substructures
heatmap.2
hclust

**Examples**

```r
## plot expression data for six genes in the brain structure 'Allen:4010' with dendrogram
plot_expression(structure_ids=c("Allen:4010"), gene_ids=c(324, 8312, 673, 1029, 64764, 1499), dataset="adult")

## plot expression data of six genes in two brain regions from children (age_category 3) without dendrogram
plot_expression(structure_ids=c("Allen:10657", "Allen:10208"), gene_ids=c("ENSG00000168036", "ENSG00000157764", "ENSG00000182158", "ENSG00000147889"), dataset="5_stages", dendro=FALSE, age_category=3)
```
Index

*Topic htest
  aba_enrich, 3
  ABAEnrichment-package, 2

*Topic package
  ABAEnrichment-package, 2

aba_enrich, 2, 3, 7–13
ABAData, 7
ABAEEnrichment (ABAEEnrichment-package), 2
ABAEEnrichment-package, 2

get_expression, 2, 5, 6, 13
get_name, 3, 5, 8, 8, 10, 11, 13
get_sampled_substructures, 3, 5, 8, 9, 9, 11, 13
get_superstructures, 3, 5, 9, 10, 10

hclust, 12, 13
heatmap, 2, 11, 13

order.dendrogram, 12

plot_expression, 2, 5, 8, 11