Package ‘ComICS’

May 13, 2018

Title Computational Methods for Immune Cell-Type Subsets

Description Provided are Computational methods for Immune Cell-type Subsets, including:(1) DCQ (Digital Cell Quantifier) to infer global dynamic changes in immune cell quantities within a complex tissue; and (2) VoCAL (Variation of Cell-type Abundance Loci) a deconvolution-based method that utilizes transcriptome data to infer the quantities of immune-cell types, and then uses these quantitative traits to uncover the underlying DNA loci.

Version 1.0.4

Imports glmnet, stats

Depends R (>= 3.1.1)

License GPL-2

LazyData true

URL `http://dcq.tau.ac.il/` , `http://csgi.tau.ac.il/VoCAL/`

RoxygenNote 5.0.1

NeedsCompilation no

Author Yael Steuerman [aut, cre],
Irit Gat-Viks [aut]

Maintainer Yael Steuerman <`yaelsteu@mail.tau.ac.il`>

Repository CRAN

Date/Publication 2018-05-13 08:19:17 UTC

R topics documented:

ComICS *Computational methods for Immune Cell-type Subsets*

Description

Computational methods for Immune Cell-type Subsets.

Author(s)

Yael Steuerman and Irit Gat-Viks

commons *Shared Immunological datasets*

Description

Example datasets (Reference data and marker set):

`immgen_dat`: An immune cell compendium, consisting of transcriptional profiles of isolated immune cell subsets, taken from various tissues, stimulations and time points (adapted from Heng et al., 2008). The full immgen dataset is available for download at `http://dcq.tau.ac.il/` or `http://csgi.tau.ac.il/VoCAL/` .

`DCQ_mar`: Preselected group of genes that likely discriminate well between the immune-cell types given in the reference data (adapted from Altboum et al., 2014).

Usage

```
data(commons)
```

dcq *DCQ - Digital Cell Quantifier*

Description

DCQ combines genome-wide gene expression data with an immune cell-type reference data to infer changes in the quantities immune cell subpopulations.

Usage

```
dcq(reference_data, mix_data, marker_set, alpha_used=0.05,
 lambda_min=0.2, number_of_repeats=3, precent_of_data=1.0)
```

Arguments

`reference_data` a data frame representing immune cell expression profiles. Each row represents an expression of a gene, and each column represents a different immune cell type. `colnames` contains the name of each immune cell type and the `rownames` includes the genes' symbol. The names of each immune cell type and the symbol of each gene should be unique. Any gene with missing expression values must be excluded.

`mix_data` a data frame representing RNA-seq or microarray gene-expression profiles of a given complex tissue. Each row represents an expression of a gene, and each column represents a different experimental sample. `colnames` contain the name of each sample and `rownames` includes the genes' symbol. The name of each individual sample and the symbol of each gene should be unique. Any gene with missing expression values should be excluded.

`marker_set` data frames of one column, that includes a preselected list of genes that likely discriminate well between the immune-cell types given in the reference data.

`alpha_used, lambda_min`
parameters of the L1 and L2 regularization. It is generally recommended to leave the default value. For more information about this parameter, see the glmnet package.

`number_of_repeats`
using one repeat will generate only one output model. Using many repeats, DCQ calculates a collection of models, and outputs the average and standard deviation for each predicted relative cell quantity.

`precent_of_data`
in order to run the analysis over all the cell types use 1.0. For bootstrap purposes, you can use part of the data (e.g, 0.5).

Value

a list that contains two matrices

`average` a matrix that contains the average relative quantities for each cell type in every-test sample.

`stdev` a matrix that contains the standard deviations over all repeats for each cell types in each test sample.

References

Altboum Z, Steuerman Y, David E, Barnett-Itzhaki Z, Valadarsky L, Keren-Shaul H, et al. Digital cell quantification identifies global immune cell dynamics during influenza infection. Mol Syst Biol. 2014;10: 720. doi:10.1002/msb.134947

Examples

```
data(commons)
data(dcqEx)
results <- dcq(reference_data=immgen_dat, mix_data=lung_time_series_dat, marker_set=DCQ_mar)
```

dcqEx *Example datasets for runnning dcq*

Description

Example datasets for runnning dcq (mix data):

`lung_time_series_dat`: RNA-seq or microarray differential gene expression profiles of a test sample compared to a reference sample (adapted from Altboum et al., 2014). The full dataset is available for download at `http://dcq.tau.ac.il/` or `http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE49934` .

uses `DCQ_mar` and `immgen_dat` from commons.RData .

Usage

```
data(dcqEx)
```

vocal *Variation in Cell Abundance Loci*

Description

Probing immune system genetics via gene expression. VoCAL is a deconvolution-based method that utilizes transcriptome data to infer the quantities of immune-cell types, and then uses these quantitative traits to uncover the underlying DNA loci (iQTLs) assuming homozygosity (such as in the case of recombinent inbred strains).

Usage

```
vocal(...,reference_data,expression_data,genotyping_data,normalize_data,
 T.i=5,T.e=10,eqtl_association_scores=NULL)
```

Arguments

`...` one or more data frames of one column, each one represents a preselected marker set that likely discriminate well between the immune-cell types given in the reference data. The number of data frames defines the number of association scores that would be combined to generate the final iQTL association score.

`reference_data` a data frame representing immune cell expression profiles. Each row represents an expression of a gene, and each column represents a different immune cell type. `colnames` contains the name of each immune cell type and the `rownames` includes the genes' symbol. The names of each immune cell type and the symbol of each gene should be unique. Any gene with missing expression values must be excluded.

`expression_data`
a data frame representing RNA-seq or microarray gene-expression profiles of a given complex tissue across a population of genetically distinct (genotyped) individuals. Each row represents an expression of a gene, and each column represents a genetically distinct individual. `colnames` contain the name of each individual, as written in the `genotyping_data`, and `rownames` includes the genes' symbol. The name of each individual sample and the symbol of each gene should be unique. Any gene with missing expression values should be excluded.

`genotyping_data`
a data frame where each row represents a different locus, and each column represents a genetically distinct individual. The genotype should be taken from homozygous individuals only. Where the genotype is unknown `NA` should be used. The first six columns contain the following information: (1) The sequential identifier of the locus; (2) The name of each locus Chr; (3) Chromosome position; (4) Start genome position; (5) End genome position; (6) position in cM.

`normalize_data` normalization type. The data will be normalized by either: (1) "All" - subtraction of the mean expression of all strains; (2) "None" - data is already normalized, do nothing; (3) name of individual included in `colnames` of `expression_data`;

`T.i` numerical. significant iQTL association score (`-log10(Pvalue)`) cutoff for the refinement step of the VoCAL algorithm.

`T.e` numerical. significant eQTL association score (`-log10(Pvalue)`) cutoff for the refinement step of the VoCAL algorithm.

`eqtl_association_scores`
(optional) a data frame where each entry represents an association score for a gene given the genotype of all the individuals that appear in the expression_data data frame, in a specific locus. This eQTL analysis should be peformed over the normalized expression_data. `colnames` contain the UID (as written in the genotyping_data) and `rownames` includes the genes' symbol (as written in the expression_data). The symbol of each gene should be unique. These scores should be in -log10(P value). Default is NULL, meaning that eQTL analysis will be performed.

Value

a list of two martices

`final_association_score`
a matrix that contains the output iQTL association score after applying the iterative filteration procedure. Each row represents the genome wide-association result for a specific immune trait over a range of DNA loci. `rownames` provides the identifier of the locus and `colnames` contains the immune-cell type names. Each entry provides the `-log10(P value)` of an iQTL association score.

`marker_info` the names of all the markers removed from the different marker sets provided

References

Steuerman Y and Gat-Viks I. Exploiting Gene-Expression Deconvolution to Probe the Genetics of the Immune System (2015), Submitted.

Examples

```
data(commons)
data(vocalEx)
## Not run:
 results <- vocal(DCQ_mar, reference_data=immgen_dat, expression_data=lung_dat,
 genotyping_data=gBXD, normalize_data="B6", eqtl_association_scores=eQTL_res)

## End(Not run)
```

---

vocalEx *Example datasets for runnning vocal*

Description

Example datasets for runnning vocal (Expression data, genotype data and eQTL results data):

lung_dat: RNA-seq or microarray gene-expression profiles of a given complex tissue across a population of genetically distinct (genotyped) individuals (adapted from E-MTAB-848).

gBXD: Genotyping of the different individuals under study (adapted from GeneNetworks).

eQTL_res: eQTL analysis results of the different genes in the expression data (specifically the genes that appear in the marker set(s) selected).

uses DCQ_mar and immgen_dat from commons.RData .

Usage

```
data(vocalEx)
```

Index