Package 'genphen'

October 17, 2020

Type Package

Title A tool for quantification of associations between genotypes and phenotypes in genome wide association studies (GWAS) with Bayesian inference and statistical learning

Version 1.16.0

Date 2019-08-02

Description Genetic association studies are an essential tool for studying the relationship between genotypes and phenotypes. With genphen we can jointly study multiple phenotypes of different types, by quantifying the association between different genotypes and each phenotype using a hybrid method which uses statistical learning techniques such as random forest and support vector machines, and Bayesian inference using hierarchical models.

License GPL (>= 2)

Depends R (>= 3.5.0), Rcpp (>= 0.12.17), methods, stats, graphics

Imports rstan (>= 2.17.3), ranger, parallel, foreach, doParallel, e1071, Biostrings, rPref

Suggests testthat, ggplot2, gridExtra, ape, ggrepel, knitr, reshape, xtable

LazyLoad yes

ByteCompile true

NeedsCompilation no

Encoding UTF-8

BugReports https://github.com/snaketron/genphen/issues

biocViews GenomeWideAssociation, Regression, Classification, SupportVectorMachine, Genetics, SequenceMatching, Bayesian, FeatureExtraction, Sequencing

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

git_branch RELEASE_3_11

git_last_commit a348d23

git_last_commit_date 2020-04-27

Date/Publication 2020-10-16

Author Simo Kitanovski [aut, cre]

Maintainer Simo Kitanovski <simo.kitanovski@uni-due.de>

R topics documented:

`dichotomous.phenotype.saap`
Dichotomous phenotype dataset

Description

The phenotype data is a numerical vector of length 120. It represents 120 dichotomous measured phenotypes for 120 organisms. We used it as a dependent variable in combination with the genotype.saap data, and quantified the association between each of the SAAP and the phenotype.

Usage

```
data(dichotomous.phenotype.saap)
```

Format

A numerical vector with 120 elements (organisms) which correspond to the rows of he genotype data.

Value

Vector of 120 metric elements, representing phenotypes measured for 120 organisms.

Examples

```
data(dichotomous.phenotype.saap)
```

genotype.saap *SAAP genotype dataset*

Description

The genotype.saap data is a character matrix with dimensions 120x154. It contains 154 amino acid protein sites across 120 organisms. The data is used in combination with the phenotype.aa data to quantify the association between each amino acid substitution pair and the phenotype vector.

Usage

```
data(genotype.saap)
```

Format

A matrix with 120 observations and 154 columns (some of which qualify as single amino acid polymorphisms).

Value

Matrix with 120 rows and 154 columns, whereby each row is a protein sequence and the elements represent an amino acids.

Source

http://www.ncbi.nlm.nih.gov/genbank/

Examples

```
data(genotype.saap)
```

genotype.saap.msa *SAAP genotype dataset (msa)*

Description

The genotype.saap.msa data is a multiple sequence alignment in Biostrings AAMultipleAlignment format. It contains 120 protein sequences, each with 154 sites (SAAPs). The data is used in combination with the phenotype.aa data to quantify the association between each amino acid substitution pair and the phenotype vector.

Usage

```
data("genotype.saap.msa")
```

Format

AAMultipleAlignment object with 120 sequences each made of 154 amino acid sites (SNPs), some of which qualify as single amino acid polymorphisms.

Value

AAMultipleAlignment object with 120 sequences each made of 154 amino acid sites (SNPs), some of which qualify as single amino acid polymorphisms.

Source

http://www.ncbi.nlm.nih.gov/genbank/

Examples

```
data("genotype.saap.msa")
```

---

genotype.snp *SNP genotype dataset*

Description

The genotype.snp data is a character matrix with dimensions 51x100. It contains 100 SNPs across 51 mouse strains, taken from the publicly available Mouse Hapmap data. We used it in combination with the phenotype.snp data to compute the association between each SNP and the phenotype data.

Usage

```
data(genotype.snp)
```

Format

A matrix with 51 observations (laboratory mouse strains) and 100 variables (SNPs).

Value

Matrix with 51 rows and 100 columns, whereby each column is a SNP, and the elements represent an alleles (nucleotides).

Source

http://mouse.cs.ucla.edu/mousehapmap/emma.html

Examples

```
data(genotype.snp)
```

genotype.snp.msa *SNP genotype dataset (msa)*

Description

The genotype.snp.msa data is a multiple sequence alignment in Biostrings DNAMultipleAlignment format. It contains 51 DNA sequences, each with 100 sites (SNPs), taken from the publicly available Mouse Hapmap data. We used it in combination with the phenotype.snp data to compute the association between each SNP and the phenotype data.

Usage

```
data("genotype.snp.msa")
```

Format

DNAMultipleAlignment object with 51 sequences each made of 100 nucleotides (SNPs).

Value

DNAMultipleAlignment object with 51 sequences each made of 100 nucleotides (SNPs).

Source

http://mouse.cs.ucla.edu/mousehapmap/emma.html

Examples

```
data("genotype.snp.msa")
```

phenotype.saap *Continuous phenotype dataset*

Description

The phenotype data is a numerical vector of length 120. It represents 120 measured phenotypes for 120 organisms. We used it as a dependent variable in combination with the genotype.saap data, and quantified the association between each of the SAAP and the phenotype.

Usage

```
data(phenotype.saap)
```

Format

A numerical vector with 120 elements (organisms) which correspond to the rows of he genotype data.

Value

Vector of 120 metric elements, representing phenotypes measured for 120 organisms.

Examples

```
data(phenotype.saap)
```

`phenotype.snp` *Continuous phenotype dataset*

Description

The phenotype data is a numerical vector of length 51. It represents 51 measured phenotypes for 51 laboratory mouse strains. It is to be used as a dependent variable in combination with the SNP genotype data, in order to compute the association between each of the SNPs and the phenotype.

Usage

```
data(phenotype.snp)
```

Format

A numerical vector with 51 elements (laboratory mice) which correspond to the rows of he genotype data.

Value

Vector of 51 metric elements, representing phenotypes measured for 51 laboratory mice.

Examples

```
data(phenotype.snp)
```

`runDiagnostics` *Data reduction procedure*

Description

The methods implemented in genphen are statistically superior to the ones implemented by most classical (frequentist) tools for GWAS. A major challenge, however, of our method is the substantially increased computational cost when analyzing thousands of SNPs. Inspired by the biological assumption that the major fraction of the studied SNPs are non-informative (genetic noise) with respect to the selected phenotype, various data reduction techniques can be implemented to quickly scan the SNP and discard a substantial portion of the the SNPs deemed to be clearly non-informative.

Usage

```
runDiagnostics(genotype, phenotype, phenotype.type, rf.trees)
```

Arguments

genotype	Character matrix/data frame or a vector, containing SNPs/SAAPs as columns or alternatively as DNAMultipleAlignment or AAMultipleAlignment Biostrings object.
phenotype	Numerical vector.
phenotype.type	Character indicator of the type of the phenotype, with 'Q' for a quantitative, or 'D' for a dichotomous phenotype.
rf.trees	Number of random forest trees (default = 5,000).

Details

The data reduction procedure includes the following steps:

1. The complete data (genotypes and a single phenotype) is used to train a random forest (RF) model, which will quantify the importance of each SNP/SAAP in explaining the phenotypeassociation between each SNP and the phenotype.
2. We can then plot the distribution of variable importances, to get an insight into the structure of the importances values and potentially disect the signal from the noise.
3. The main analysis can then be performed with runGenphen using a subset (based on their importance) of SNPs

Value

site	id of the site (e.g. position in the provided sequence alignment)
importance	Magnitude of importance (impurity) of the site, estimated with random forest implemented in R package ranger

Author(s)

Simo Kitanovski <simo.kitanovski@uni-due.de>

See Also

runGenphen, runPhyloBiasCheck

Examples

```
# genotypes:
data(genotype.saap)
# quantitative phenotype:
data(phenotype.saap)

# run diagnostics
diag <- runDiagnostics(genotype = genotype.saap,
                       phenotype = phenotype.saap,
                       phenotype.type = "Q",
                       rf.trees = 5000)
```

runGenphen *Genetic association analysis using Bayesian inference and statistical learning methods*

Description

Given a set of genotypes (single nucleotide polymorphisms - SNPs; or single amino acid polymorphisms - SAAPs) for a set of individuals, and a corresponding set of phenotypes, genphen quantifies the association between each genotype and phenotype using Bayesian inference and statistical learning.

Usage

```
runGenphen(genotype, phenotype, phenotype.type, model.type,
           mcmc.chains, mcmc.steps, mcmc.warmup, cores,
           hdi.level, stat.learn.method, cv.steps, ...)
```

Arguments

genotype	Character matrix/data frame or a vector, containing SNPs/SAAPs as columns or alternatively as DNAMultipleAlignment or AAMultipleAlignment Biostrings object.
phenotype	Numerical vector (for a single phenotype) or matrix with multiple phenotypes stored as columns.
phenotype.type	Vector representing the type of each phenotype (of the phenotype input), with 'Q' identifier for quantitative, or 'D' for dichotomous phenotypes.
model.type	Type of Bayesian model: 'univariate' or 'hierarchical'
mcmc.chains	Number of MCMC chains (default = 2).
mcmc.steps	Length of MCMC chains (default = 1,000).
mcmc.warmup	Length of adaptive part of MCMC chains (default = 500).
cores	Number of cores to use (default = 1).
hdi.level	Highest density interval (HDI) (default = 0.95).
stat.learn.method	Parameter used to specify the statistical learning method used in the analysis. Currently two methods are available: random forest ('rf') and support vector machine ('svm'). For no statistical learning select 'none'.
cv.steps	cross-validation steps (default = 1,000).
...	Optional parameters include adapt_delta: STAN configuration (default = 0.9); max_treedepth: STAN configuration (default = 10); ntree: Number of random forest trees to grow, only in case stat.learn.method = 'rf' (default = 1000); cv.fold: Cross-validation fold (default = 0.66).

Details

Input:

- genotype genotype data (e.g. set of 1,000 SNPs found along the aligned genomes of 10 individuals) - provided in one of three possible input types:

  - character vector of length N (if only a single SNP/SAAP is provided), containing the genotypes of N individuals.
  - character matrix with dimensions NxS (N = individuals, S = SNPs/SAAPs).
  - AAMultipleAlignment or DNAMultipleAlignment object; if the genotype data is a multiple sequence alignment composed of N sequences.
- phenotype phenotype data (dichotomous or quantitative phenotypes allowed)
  - numerical vector of length N if only a single phenotype is analyzed
  - numerical matrix NxP, if P phenotypes are provided.
- phenotype.type Vector with identifiers specifying the type of the phenotypes with 'Q' (for quantitative) or 'D' (for dichotomous) for each column in the phenotype dataset.
- model.type Specifies the structure of Bayesian model used to estimate the effect size of each genotype. Options allow for either 'univariate' (each SNP/SAAP treated as completely independent) or 'hierarchical' (SNP/SAAP effects share information through partial pooling).

Metrics: To quantify the association between each genotype and phenotype genphen computes multiple measures of association:

- Effect size (beta): for each SNP we compute beta (effect) with Bayesian inference). beta quantifies the strength of the association between the genotypes and the phenotype. We report for each beta its mean and 95% (for instance) highest density interval (HDI) of beta, which is defined as the interval that covers a 95% of the posterior distribution, with every point inside the interval having a higher credibility than any point outside it.
- Classification accuracy (CA): CA measures the degree of accuracy with which one can classify (predict) the alleles of a SNP from the phenotype. If there exists a strong association between a particular SNP and the phenotype, one should be able to train a statistical model (using RF or SVM) which accurately classifies the two alleles of that SNP solely from the phenotype data (CA close to 1). Otherwise, the model should perform poorly, with the classification accuracy of the model being approximately similar to that of simple guessing (CA close to 0.5)
- Cohen's kappa statistic: There is one pitfall where the CA estimate can be misleading, and this is the case when the analyzed SNP is composed of unevenly represented genetic states (alleles). For instance, the allele A of a given SNP is found in 90% of the individuals, while the other allele T in only 10%. Such an uneven composition of the alleles can lead to misleading results, i.e. even without proper learning the algorithm can produce a high $CA close to 0.9 simply by always predicting the dominant label. The kappa statistics is a quality metric, which is to be used together with CA. Cohen defines the following meaningful kappa intervals: [kappa<0]: "no agreement", [0.0-0.2]: "slight agreement", [0.2-0.4]: "fair agreement", [0.4-0.6]: "moderate agreement", [0.6-0.8]: "substantial agreement" and [0.8-1.0]: "almost perfect agreement".

Value

General parameters:

`site`	id of the site (e.g. position in the provided sequence alignment)
`ref, alt`	reference and alternative genotypey
`refN, altN`	count of ref and alt genotypes
`phenotype.id`	Identifier of the studied phenotype

Association scores:

`beta.mean`, `beta.se`, `beta.sd`, `beta.hdi.low/beta.hdi.high`
: Estimates of the mean, standard error, standard deviation and HDI of the slope coefficient

`ca.mean`, `ca.hdi.low/ca.hdi.high`
: CA estimate and HDI

`kappa.mean`, `kappa.hdi.low/kappa.hdi.high`
: Cohen's kappa and HDI

`rank`
: Pareto optimiazion based front (rank) of SNP/SAAP estimated by maximizing metrics beta.mean and kappa.mean

MCMC convergence parameters:

`Neff`
: Effective sampling size

`Rhat`
: Potential scale reduction factor

Posterior predictions:

`ppc`
: Posterior prediction check and real data summary for each genotype.

Posterior summary:

`complete.posterior`
: Complete stan object containing the posterior of each parameter estimated during the Bayesian inference. The data can be used for model debugging, posterior predictive checks, etc.

Author(s)

Simo Kitanovski <simo.kitanovski@uni-due.de>

See Also

runDiagnostics, runPhyloBiasCheck

Examples

```
# genotypes:
data(genotype.saap)
# quantitative phenotype:
data(phenotype.saap)
# dichotomous phenotype:
data(dichotomous.phenotype.saap)
# make phenotype matrix (column = phenotype)
phenotypes <- cbind(phenotype.saap, dichotomous.phenotype.saap)

# run genphen
out <- runGenphen(genotype = genotype.saap[, 80:82],
                  phenotype = phenotypes,
                  phenotype.type = c("Q", "D"),
                  model.type = "univariate",
                  mcmc.chains = 4,
                  mcmc.steps = 1500,
                  mcmc.warmup = 500,
                  cores = 2,
                  hdi.level = 0.95,
                  stat.learn.method = "rf",
                  cv.steps = 200)
```

`runPhyloBiasCheck` *Check for phylogenetic bias*

Description

Given a set of genotypes such as single nucleotide polymorphisms (SNPs) or single amino acid polymorphisms (SAAPs) for a set of N individuals, the procedure can operate in two modes:

1) it computes a NxN kinship matrix (matrix populated with pairwise distances (Hamming) between each two individuals computed using all the genotypes). Based on the kinship matrix it then estimates the degree of phlyogenetic bias related to each genotype as 1-mean.phylo.dist(allele)/mean.phylo.dist(all)

2) it uses a precomputed kinship matrix and then estimates the degree of phlyogenetic bias related to each genotype using the same procedure.

Usage

```
runPhyloBiasCheck(input.kinship.matrix, genotype)
```

Arguments

`genotype`	Character matrix/data frame or a vector, containing SNPs/SAAPs as columns or alternatively as DNAMultipleAlignment or AAMultipleAlignment Biostrings object.
`input.kinship.matrix`	precomputed kinship matrix provided by the user.

Details

Input:

- genotype P genotypes of N individuals in the form of NxP character matrix/data frame or vector (if P = 1).
- input.kinship.matrix precomputed NxN matrix (row/column for each individual)

Value

Genotype parameters:

`site`	id of the site (e.g. position in the provided sequence alignment)
`genotype`	allele of a SNP or amino acid of SAAP
`bias`	number between 0 (no bias) or 1 (complete bias)

Mutation bias:

`site`	id of the site (e.g. position in the provided sequence alignment)
`mutation`	allele of a SNP or amino acid of SAAP
`bias`	number between 0 (no bias) or 1 (complete bias) for the mutation computed as max(bias in genotype 1, bias in genotype 2)

Kinship matrix:

`kinship.matrix`	NxN matrix

Author(s)

Simo Kitanovski <simo.kitanovski@uni-due.de>

See Also

runDiagnostics, runGenphen

Examples

```
# genotype inputs:
data(genotype.saap)
# phenotype inputs:
data(phenotype.saap)

# phylogenetic bias analysis
bias <- runPhyloBiasCheck(input.kinship.matrix = NULL,
                          genotype = genotype.saap)
```

Index