Package 'genphen'

October 7, 2018

Type Package

Title A tool for quantification of associations between genotypes and phenotypes with Bayesian inference and statistical learning techniques

Version 1.8.0

Date 2018-04-15

Author Simo Kitanovski

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

Description Genetic association studies are an essential tool for studying the relationship between genotypes and phenotypes and thus for the discovery of disease-causing genetic variants. With genphen we propose a new method for conducting genetic association studies, using a combined approach which is based on both Bayesian inference and statistical learning techniques such as random forest and support vector machines.

License GPL (>= 2)

Depends R(>= 3.4), stats, graphics, e1071, Biostrings, rstan, ranger, parallel, foreach, doParallel

LazyLoad yes

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

NeedsCompilation no

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

git_branch RELEASE_3_7

git_last_commit fdf75eb

git_last_commit_date 2018-04-30

Date/Publication 2018-10-06

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

Running genphen for hundreds of thousands of predictors (e.g. SNPs) can be computationally costly. Motivated by the biological fact that most SNPs have no or weak associations with the phenotype, genphen allows the user to run a light-weight diagnostic procedure which allows the user to discard large portion of the non-informative SNPs before running the main association analysis. The data reduction step proceeds as follows: 1) using random forest and their measures of variable importance, we obtain one importance value for each SNP. We can use the spectrum of importances as a 'rough' guide to determine the importance level at which the non-informative SNPs are dominant; 2) the user can then select different points along the importance spectrum (e.g. ranks 1:5, 100:105, 1000:1005, ...) for which genphen is run using its standard procedure. We can then use the association scores produced by genphen to determine the importance rank at which the SNPs are no longer informative and thereby achieve data reduction.

Usage

```
runDiagnostics(genotype, phenotype, phenotype.type, rf.importance.trees,
               with.anchor.points, mcmc.chains, mcmc.iterations, mcmc.warmup,
               mcmc.cores, hdi.level, anchor.points)
```

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 continuous-phenotype analysis, numerical or character vector for dichotonous-phenotype analysis.
`phenotype.type`	'continuous' or 'dichotomous' based on phenotype type.
`rf.importance.trees`	Number of random forest trees to use for the variable importance analysis (default = 50,000).
`with.anchor.points`	Boolean whether to run the complete diagnostics procedure (TRUE), or only the random forest based importance estimation (FALSE)
`mcmc.chains`	Number of MCMC chains used to test each association test. We recomend mcmc.chains >= 2.
`mcmc.iterations`	Length of MCMC chains (default = 1,000).
`mcmc.warmup`	Length of adaptive MCMC chains (default = 500).

mcmc.cores	Number of cores used for the MCMC (default = 1). The same parameter is for multicore execution of the statistical learning procedures.
hdi.level	Highest density interval (HDI) (default = 0.95).
anchor.points	Vector of ranks (based on the importance measure) at which to select the genotypes, for which the diagnostics will be run.

Details

Procedure: 1) Run random forest on the complete genotype-phenotype data and infer variable importance for each genotype. 2) Sort genotypes by importance, and sample few genotypes at different points along the importance spectrum, performing for each genotype the procedure explained in runGenphen. 3) Visualize results which can help the user to determine whether a sensible data reduction can be done, i.e. to select X number of most important genotypes for the main analysis.

Value

General parameters:

site	id of the site (e.g. position in the provided sequence alignment)
mutation	type of polymorphism (e.g. T->A)
data	number of data points for each allele (e.g. A:10, T:20)

Association score parameters:

cohens.d or absolute.d	Cohen's d effect size (continuous phenotype analysis) or absolute effect size (dichotomous phenotype analysis) point estimate
cohens.d.L/cohens.d.H or absolute.d.L/absolute.d.H	The highest density interval (HDI) of the estimated effect size
bc	Bhattacharyya coefficient, degree of overlap between the posterior predicted distributions of the phenotype in the two alleles of a SNP (or two amino acid states of an SAAP.
anchor.point	Indicator of selected anchor.point

Ranked variable importance scores:

site	id of the site (e.g. position in the provided sequence alignment)
importance	magnitude of importance of the site
importance.rank	rank based on the importance

Author(s)

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

See Also

runGenphen, runPhyloBiasCheck

Examples

```
# I: Continuous diagnostics
# genotype inputs:
data(genotype.saap)
# phenotype inputs:
data(phenotype.saap)

# run genphen
continuous.diagnostics <- runDiagnostics(genotype = genotype.saap,
                                         phenotype = phenotype.saap,
                                         phenotype.type = "continuous",
                                         rf.importance.trees = 50000,
                                         with.anchor.points = TRUE,
                                         mcmc.chains = 2,
                                         mcmc.iterations = 1500,
                                         mcmc.warmup = 500,
                                         mcmc.cores = 2,
                                         hdi.level = 0.95,
                                         anchor.points = c(1:10))
```

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

Description

Given a set of genotypes such as single nucleotide polymorphisms (SNPs) or single amino acid polymorphisms (SAAPs) for a set of N individuals, and the corresponding N phenotypes, genphen computes severaly genotype-phenotypes association scores using Bayesian inference and statistical learning.

Usage

```
runGenphen(genotype, phenotype, phenotype.type,
           mcmc.chains, mcmc.iterations, mcmc.warmup,
           mcmc.cores, hdi.level, stat.learn.method,
           cv.iterations, with.rpa, rpa.iterations,
           rpa.rope)
```

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 continuous-phenotype analysis, numerical or character vector for dichotonous-phenotype analysis.
`phenotype.type`	'continuous' or 'dichotomous' based on phenotype type.
`mcmc.chains`	Number of MCMC chains used to test each association test. We recomend mcmc.chains >= 2.
`mcmc.iterations`	Length of MCMC chains (default = 1,000).

mcmc.warmup	Length of adaptive MCMC chains (default = 500).
mcmc.cores	Number of cores used for the MCMC (default = 1). The same parameter is for multicore execution of the statistical learning procedures.
hdi.level	Highest density interval (HDI) (default = 0.95).
stat.learn.method	Character 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.iterations	cross-validation iterations (default = 1,000).
with.rpa	with retrospective power analysis (RPA).
rpa.iterations	Number of simulation used to compute RPA scores.
rpa.rope	Region of practical equivalence (ROPE) for the RPA.

Details

Input:

- genotype P genotypes of N individuals in the form of NxP character matrix/data frame or vector (if P = 1).
- phenotype phenotypes of N individuals in the form of a N-sized vector. The type of the phenotype can either be continuous or dichotomous. Therefore, genphen has two analysis modes for each situation. The main difference between them is the design of the Bayesian hierarchical model which are used.

Goal: To quantify the association between each genotype and phenotype. With genphen, we provide several measures of association:

- Classification accuracy (CA): it is a metric obtained with a statistical learning technique such as random forest (RF) or support vector machine (SVM), and measures the degree of accuracy with which one can classify (predict) the alleles of an SNP from the phenotype measurements. If there exists a strong association between a particular SNP and the phenotype, one should be able to build a statistical model which accurately classifies the two alleles of that SNP solely from the phenotype data (CA approx. 1). Otherwise, the classification accuracy of statistical model should be approximately similar to that of simple guessing (CA approx. 0.5). Promising SNPs thus have a high CA close to 1.

  For each CA estimate, we also compute a corresponding Cohen's kappa statistic. With Cohen's kappa we compare the observed CA with the classification accuracy which is expected simply by chance (CA_exp). This is in particular useful when the genetic states of the genotype are not evenly represented, i.e. allele A of a given SNP may be represented in 80% of the individ-uals, while the other allele T may be represented in only 20% of the individuals. Such an uneven composition of the genotype can affect the classification analysis, potentially resulting in high CA simply because the classifier only predicts the dominant allele. Promising SNPs have high Cohen's kappa close to 1.
- Effect size: for each SNP we compute the effect size, i.e. the size of the difference in phenotype observed between its alleles (amino acids in the case of SAAP). Depending on the type of the phenotype, we either compute the Cohen's d effect size for continuous phenotypes, or the absolute effect size for dichotomous phenotypes. We first use Byesian inference to infer the parameters of the distribution of the phenotype in each allele, and then plug the posterior distribution of these parameters into the corresponding equations for computing the effect size. In addition to the point estimates of the effect sizes, we also estimate the corresponding highest density intervals (HDIs).

- Bhattacharyya coefficient (BC): the infered parameters needed to compute the effect size are used to perform posterior predictive checks and generate simulated distributions of the phenotype for each allele of a given SNP (amino acid states in the case of SAAP). Using the Bhattacharyya coefficient, we can measure the overlap between any two distributions, i.e. for no or low overlap (weak strength of association), BC is close to 0, and BC is close to 1 for a complete overlap (strong strength of association).

  The association scores can be correlated, i.e. when CA is close to 1, the effect size is large and (BC = 0). This is not always the case, i.e. we can have a small but significant effect size, yet a perfect CA. This is an interesting information which could be lost if a single association metric was used. Moreover, RF and SVM allow us to capture non-linear associations.

Value

General parameters:

`site`	id of the site (e.g. position in the provided sequence alignment)
`mutation`	type of polymorphism (e.g. T->A)
`data`	number of data points for each allele (e.g. A:10, T:20)

Association score parameters:

`cohens.d or absolute.d`
: Cohen's d effect size (continuous phenotype analysis) or absolute effect size (dichotomous phenotype analysis). point estimate

`cohens.d.L/cohens.d.H or absolute.d.L/absolute.d.H`
: The highest density interval (HDI) of the estimated effect size.

`ca, ca.L, ca.H`
: Classification accuracy (CA) estimate and its HDI

`kappa, kappa.L, kappa.H`
: Cohen's kappa estimate and its HDI

`bc`
: Bhattacharyya coefficient, degree of overlap between the posterior predicted distributions of the phenotype in the two alleles of a SNP (or two amino acid states of an SAAP.

`sd.d, sd.d.L, sd.d.H`
: (only for continuous phenotypes) Difference in standard deviations its HDI

MCMC convergence parameters:

`s, g, n`
: s=site, g=genotype, n=number of observations

`mu.rhat, sigma.rhat`
: Potential scale reduction factor from the MCMC simulation for each parameter

`mu.ess, sigma.ess`
: Effective sampling size from the MCMC simulation for each parameter

`divergence`
: Indicator of occuring divergences during the MCMC simulation

`treedepth`
: Indicator of treedepth exceptions during the MCMC simulation

RPA parameters:

`s, g, n`	s=site, g=genotype, n=number of observations
`rpa.count`	Number of RPA runs in which a significant effect was found

Posterior predictive check parameters:

`s, g, n`
: s=site, g=genotype, n=number of observations

`predicted.mu.i, predicted.mu.j, real.mu.i, real.mu.j`
: Predicted and real means in each group of a SNP

Author(s)

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

See Also

runDiagnostics, runPhyloBiasCheck

Examples

```
# I: Continuous phenotype analysis

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

# run genphen
continuous.analysis <- runGenphen(genotype = genotype.saap[, 1:3],
                                  phenotype = phenotype.saap,
                                  phenotype.type = "continuous",
                                  mcmc.chains = 2,
                                  mcmc.iterations = 2000,
                                  mcmc.warmup = 500,
                                  mcmc.cores = 2,
                                  hdi.level = 0.95,
                                  stat.learn.method = "rf",
                                  cv.iterations = 500,
                                  with.rpa = FALSE,
                                  rpa.iterations = 100,
                                  rpa.rope = 0)

# II: Dichotomous phenotype analysis

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

# run genphen
dichotomous.analysis <- runGenphen(genotype = genotype.saap[, 1:3],
                                   phenotype = dichotomous.phenotype.saap,
                                   phenotype.type = "dichotomous",
                                   mcmc.chains = 2,
                                   mcmc.iterations = 2000,
                                   mcmc.warmup = 500,
                                   mcmc.cores = 2,
                                   hdi.level = 0.95,
                                   stat.learn.method = "rf",
                                   cv.iterations = 500,
                                   with.rpa = FALSE,
                                   rpa.iterations = 100,
                                   rpa.rope = 0)
```

`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