Package ‘genphen’

October 17, 2017

Type Package

Title A tool for quantification of associations between genotypes and phenotypes with statistical learning techniques such as random forests and support vector machines as well as with Bayesian inference using hierarchical models

Version 1.4.0

Date 2016-09-28

Author Simo Kitanovski

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

Description Genetic association studies have become an essential tool for studying the relationship between genotypes and phenotypes. They are necessary for the discovery of disease-causing genetic variants. Here we provide a tool for conducting genetic association studies, which uses statistical learning techniques such as random forests and support vector machines, as well as using Bayesian inference with Bayesian hierarchical models. These techniques are superior to the commonly used (frequentist) statistical approaches, alleviating the multiple hypothesis problems and the need for P value corrections, which often lead to massive numbers of false negatives. Thus, with genphen we provide a framework to compare the results obtained using frequentist methods with those obtained using the more sophisticated methods provided by this tool. The tool also provides a few visualization functions which enable the user to inspect the results of such genetic association study and conveniently select the genotypes which have the highest strength of association with the phenotype.

License GPL (>= 2)

Depends R(>= 3.3), randomForest, e1071, ggplot2, effsize, Biostrings, rjags

LazyLoad yes

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

NeedsCompilation no

R topics documented:

`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` *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 51 metric elements, representing phenotypes measured for 120 organisms.

Examples

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

`phenotype.snp` *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)
```

plotGenphenBayes *Visualizing the genphen results of runGenphenBayes*

Description

This procedure visualizes the results obtained using the function runGenphenBayes.

Usage

```
plotGenphenBayes(genphen.results, hdi)
```

Arguments

`genphen.results`	data.frame resulting from runGenphenBayes.
`hdi`	single HDI level (e.g. 0.95) which has previously been estimated with runGenphenBayes

Details

This procedure plots the results of the function runGenphenBayes. Each association is shown as a point with respect to its Bayesian effect size (mu.effect), including the corresponding HDI.

Value

`plot`	ggplot plot object.

Author(s)

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

See Also

runGenphenRf, runGenphenSvm, runGenphenBayes, plotGenphenRfSvm, plotSpecificGenotype

Examples

```
#Example:
data(genotype.saap)
# or data(genotype.saap.msa) in this case you cannot subset genotype.saap[, 1:5]
data(phenotype.saap)

# run analysis
genphen.bayes <- runGenphenBayes(genotype = genotype.saap[, 1:5],
                                 phenotype = phenotype.saap,
                                 mcmc.iter = 1000,
                                 chain.nr = 2,
                                 model = "tdist",
                                 hdi.levels = c(0.95, 0.99))
plotGenphenBayes(genphen.results = genphen.bayes, hdi = 0.95)
```

plotGenphenRfSvm *Visualizing the genphen results of runGenphenRf or runGenphenSvm*

Description

This procedure visualizes the results obtained using the functions runGenphenRf or runGenphenSvm.

Usage

```
plotGenphenRfSvm(genphen.results)
```

Arguments

genphen.results	data.frame resulting from runGenphenRf/runGenphenSvm.

Details

This procedure plots the results of the functions runGenphenRf or runGenphenSvm. Each association is shown as a point with respect to its effect size (Cohen's d) and the classification accuracy (CA). The colors of the points correspond to the Cohen's kappas. The confidence intervals estimated for the CA point estimates are provided here as well.

Value

plot	ggplot plot object.

Author(s)

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

See Also

runGenphenRf, runGenphenSvm, plotGenphenBayes, plotSpecificGenotype, plotManhattan

Examples

```
#Example 1 (RF results):
data(genotype.snp)
#or data(genotype.snp.msa) in this case you cannot subset genotype.snp[, 1:5]
data(phenotype.snp)
genphen.rf <- runGenphenRf(genotype = genotype.snp[, 1:5],
                           phenotype = phenotype.snp,
                           cv.fold = 0.66,
                           cv.steps = 100,
                           hdi.level = 0.99,
                           ntree = 1000)
plotGenphenRfSvm(genphen.results = genphen.rf)
```

`plotManhattan` *Visualizing genphen results with Manhattan plots*

Description

This procedure plots the complementary T-test results obtained using runGenphenRf, runGenphenSvm or runGenphenBayes.

Usage

```
plotManhattan(genphen.results)
```

Arguments

genphen.results	Data.frame resulting from runGenphenRf, runGenphenSvm or runGenphenBayesian.

Details

This procedure plots the results from a two-sample T-test analysis, conducted in each of the three functions: runGenphenRf, runGenphenSvm or runGenphenBayesian. The P-values resulting from the T-test are first corrected by the Benjamini -Hochberg correction procedure (FDR), followed by a -log10 transformation. These values are then plotted as points, where each point corresponds to a single genotype-phenotype association as seen in a classical Manhattan plot.

Value

plot	ggplot plot object.

Author(s)

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

See Also

runGenphenRf, runGenphenSvm, plotGenphenBayes, plotGenphenRfSvm, plotSpecificGenotype

Examples

```
#Example 1:
data(genotype.snp)
#or data(genotype.snp.msa) in this case you cannot subset genotype.snp[, 1:5]
data(phenotype.snp)
genphen.rf <- runGenphenRf(genotype = genotype.snp[, 1:5],
                           phenotype = phenotype.snp,
                           cv.fold = 0.66,
                           cv.steps = 100,
                           hdi.level = 0.99,
                           ntree = 1000)
plotManhattan(genphen.results = genphen.rf)
```

plotSpecificGenotype *Visualizing specific genotypes*

Description

This procedure visualizes the phenotypic distribution linked to each of the genetic states of a specific genotype.

Usage

```
plotSpecificGenotype(genotype, phenotype, index)
```

Arguments

`genotype`	Character matrix or data frame, containing SNPs/SAAPs as columns or alternatively as a DNAMultipleAlignment/AAMultipleAlignment Biostrings object.
`phenotype`	Numerical vector whose elements correspond to the genotype.
`index`	Index (number) of the specific genotype column within the genotype data which is to be plotted.

Details

This procedure allows the user to inspect a specific genotype with respect to the the phenotype. It uses a boxplot notation to plot the phenotypes as a function of the states of that genotype. The resulting boxplot will visualize whether the different states of the specific genotype are linked to different and disjoint phenotypic distributions, which is a signature of a strong association between the genotype and the phenotype.

Value

`plot`	ggplot object

Author(s)

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

See Also

runGenphenRf, runGenphenSvm, runGenphenBayesian, plotGenphenResults, plotManhattan

Examples

```
#Example 1:
data(genotype.snp) #or data(genotype.snp.msa)
data(phenotype.snp)
specific.genotype.plot <- plotSpecificGenotype(genotype = genotype.snp,
phenotype = phenotype.snp, index = 1)

#Example 2:
data(genotype.saap) #or data(genotype.saap.msa)
data(phenotype.saap)
specific.genotype.plot <- plotSpecificGenotype(genotype = genotype.saap,
phenotype = phenotype.saap, index = 3)
```

runGenphenBayes *Conducting genetic association analysis with Bayesian inference, using hierarchical models.*

Description

This procedure performs a Bayesian two-sample t-test, to quantify the association between genotypes (SNPs or SAAPs) and phenotypes.

Usage

```
runGenphenBayes(genotype, phenotype, chain.nr, mcmc.iter, model, hdi.levels)
```

Arguments

`genotype`	Character matrix or data frame, containing SNPs/SAAPs as columns or alternatively as DNAMultipleAlignment or AAMultipleAlignment Biostrings object.
`phenotype`	Numerical vector, where each element is a measured phenotype corresponding to the observations of the genotype data.
`chain.nr`	Number of MCMC chains to simulate (recommended > 1).
`mcmc.iter`	Number of MCMC iterations used in the Bayesian inference (recommended >= 1000).
`model`	The hierarchical model used for the Bayesian inference (currently only two models are available). In case of a normally distributed phenotype, the model 'ndist' should be used, and 'tdist' if it is t-distributed (more robust to outliers).
`hdi.levels`	Highest density intervals (HDI) (default = c(0.95, 0.99)).

Details

This procedure takes two types of data as input: first a genotype data (SNPs or SAAPs), each represented as a categorical vector; second a phenotype represented by a vector of continuous numbers, whose elements correspond to the elements of the genotype data. It then performs a Bayesian two-sample t-test analysis, computing the contrast between the phenotype means in the two groups (mu.1-mu.2). The high density intervals (HDIs) of the posteriors are also provided to quantify the reliability of the contrast.

In the classical two-sample t-test two assumptions are made: first, the phenotypes in each group should be normally distributed, and second they should have equal variances. Here, we have designed hierarchial models such that these assumptions are relaxed. Thus, we allow for the phenotype in either group to either be normally (model = 'ndist') or t-distributed (model = 'tdist'), while allowing heterogeneous variances.

Value

Five classes of results are computed for each genotype-phenotype association, stored as a row in a data frame:

`site, g.1, g.1, count.1, count.1`
General information about the genotype.

`mu.1, mu.2, sd.1, sd.2`
: Mean posterior estimates for the mean phenotype associated with each genotype (mu.1 and mu.2), as well as the mean standard deviation posteriors sd.1 and sd.2.

`mu.effect, mu.effect.95.L, mu.effect.95.H`
: The bayesian effect size mu.effect = mu.1 - mu.2, including the high density interval (HDI).

`ess.mu.1, ess.mu.2`
: Effective sampling size estimated for mu.1 and mu.2

`t.test.pvalue`
: P-value score from a classical two-sample T-test.

Author(s)

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

References

- Kruschke, John. Doing Bayesian data analysis: A tutorial with R, JAGS, and Stan. Academic Press, 2014.

See Also

runGenpenRf, runGenpenSvm, plotGenphenBayes, plotSpecificGenotype, plotManhattan

Examples

```
data(genotype.saap)
# or data(genotype.saap.msa) in this case you cannot subset genotype.saap[, 1:5]
data(phenotype.saap)

# run analysis
genphen.bayes <- runGenphenBayes(genotype = genotype.saap[, 1:5],
                                 phenotype = phenotype.saap,
                                 mcmc.iter = 1000,
                                 chain.nr = 2,
                                 model = "tdist",
                                 hdi.levels = c(0.95, 0.99))
```

runGenphenRf *Conducting genetic association analysis with random forest*

Description

This procedure quantifies the accuracy with which one can predict a given genotypes (SNPs or SAAPs) from the corresponding phenotypes using random forest.

Usage

```
runGenphenRf(genotype, phenotype, cv.fold, cv.steps, hdi.level, ntree)
```

Arguments

`genotype` Character matrix or data frame, containing SNPs/SAAPs as columns or alternatively as DNAMultipleAlignment or AAMultipleAlignment Biostrings object.

`phenotype` Numerical vector, where each element is a measured phenotype corresponding to the observations of the genotype data.

`cv.fold` The cross-validation fraction (0, 1) of the data which is used to train the classifier (default = 0.66). The remaining fraction (1-cv.fold) of the data is used to test the classifier.

`cv.steps` Number of steps in the cross-validation to be performed to estimate the classification accuracy and the corresponding highest density intervals(recommended >= 100).

`hdi.level` Highest density level (default = 0.99).

`ntree` Number of trees to grow in the random forest model.

Details

This procedure takes two types of data as input: first a genotype data (SNPs or SAAPs), each represented as a categorical vector; second a phenotype represented by a vector of continuous numbers, whose elements correspond to the elements of the genotype data. It then performs a random forest classification analysis, computing the classification accuracy (CA) with which one can predict the categories of a given genotype from the numerical values of the phenotype (predictor). If there is a significant association between a given genotype and phenotype, then one should be able to accurately predict the states of the genotype from that phenotype (CA = 1 for a perfect classification). To obtain a robust CA, this procedure applies cross-validation, training the models on a subset of the genotype-phenotype data, and then testing the models on the remaining data. Thus, in addition to the CA point estimate, one also estimates the corresponding confidence intervals.

In some instances the two states of the genotype are not evenly represented, i.e. one allele of a SNP might be represented in 90% of the individuals, while the other allele might only be represented in 10% of the individuals. Such data compositions can lead to skewed CA estimates. Therefore, to verify the CA, the procedure also computes the Cohen's kappa statistic (Cohen 1960), which compares the observed classification accuracy (CA) with the classification accuracy expected by chance (CAexp). If CA > CAexp, the procedure will yield high kappas (kappa = 1 in an ideal case), otherwise low kappas are to be expected (kappa ~ 0).

The function runGenphenRf also computes statistics such as Cohen's d (effect size) and the P-value resulting from a two-sample T-test, allowing the user to compare the random forest based results with those computed with simpler techniques which are frequently used for genetic association studies.

Value

Five classes of results are computed for each genotype-phenotype association, stored as a row in a data frame:

`site, g.1, g.1, count.1, count.1`
General information about the genotype.

`ca, ca.hdi.low, ca.hdi.high, ca.hdi.length`
Mean classification accuracy and its HDI.

`kappa, kappa.hdi.low, kappa.hdi.high, kappa.hdi.length`
Cohen's kappa statistics and its HDI.

`effect.size, effect.CI.low, effect.CI.high`
Cohen's effect size and CI.

`t.test.pvalue` P-value score from a classical two-sample T-test.

Author(s)

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

References

- Breiman, Leo. Random forests. Machine learning 45.1 (2001): 5-32.
- Cohen, Jacob. Statistical power analysis for the behavior science. Lawrance Eribaum Association (1988).
- Cohen, Jacob. A coefficient of agreement for nominal scales (1960).

See Also

runGenpenSvm, runGenpenBayesian, plotGenphenResults, plotSpecificGenotype, plotManhattan

Examples

```
data(genotype.saap)
#or data(genotype.saap.msa) in this case you cannot subset genotype.saap[, 1:5]
data(phenotype.saap)
genphen.rf <- runGenphenRf(genotype = genotype.saap[, 1:5],
                           phenotype = phenotype.saap,
                           cv.fold = 0.66,
                           cv.steps = 100,
                           hdi.level = 0.99,
                           ntree = 1000)
```

`runGenphenSvm` *Conducting genetic association analysis with linear support vector machines (LSVM)*

Description

This procedure quantifies the accuracy with which one can predict a given genotypes (SNPs or SAAPs) from the corresponding phenotypes using linear support vector machines (LSVM).

Usage

```
runGenphenSvm(genotype, phenotype, cv.fold, cv.steps, hdi.level)
```

Arguments

`genotype` Character matrix or data frame, containing SNPs/SAAPs as columns or alternatively as DNAMultipleAlignment or AAMultipleAlignment Biostrings object.

`phenotype` Numerical vector, where each element is a measured phenotype corresponding to the observations of the genotype data.

`cv.fold` The cross-validation fraction (0, 1) of the data which is used to train the classifier (recommended = 0.66). The remaining fraction (1-cv.fold) of the data is used to test the classifier.

`cv.steps` Number of steps in the cross-validation to be performed to estimate the classification accuracy and the corresponding highest density intervals(recommended >= 100).

`hdi.level` Highest density interval (default = 0.99).

Details

This procedure takes two types of data as input: first a genotype data composed of a set of single nucleotide polymorphisms (SNPs) or alternatively single amino acid polymorphisms (SAAPs), each of which is represented by a column of character amino acids; second a numerical phenotype vector, where the elements sorted to correspond to the rows of the genotype data. This method quantifies the association between the polymorphic site (SNP or SAAP) and the phenotype via a classification analysis using linear support vector machines. The analysis results in a classification accuracy score between 0 and 1, where 1 indicates a perfect association between the genotype and the phenotype. To validate the classification accuracy, the tool also computes the Cohen's kappa statistics (Cohen 1960) which compares the observed classification accuracy with the expected classification accuracy. If the expected and observed classification accuracies are in concordance, the computed association can be taken seriously, otherwise it can be discarded as noise.

The function runGenphenSvm also computes statistics such as Cohen's d (effect size) and the P-value resulting from a two-sample T-test, allowing the user to compare the linear support vector based results with those computed with simpler techniques which are frequently used for genetic association studies.

Value

Five classes of results are computed for each SAAP with respect to the phenotype, resulting in a 18 element vector which is stored as a row in the final data frame:

`effect.size, effect.CI.low, effect.CI.high`
Cohen's effect size and CI.

`ca, ca.hdi.low, ca.hdi.high, ca.hdi.length`
Mean classification accuracy and its HDI.

`kappa, kappa.hdi.low, kappa.hdi.high, kappa.hdi.length`
Cohen's kappa statistics and its HDI.

`site, g.1, g.2, count.1, count.2`
General information about the genotype.

`t.test.pvalue` P-value score from an two-sample T-test.

Author(s)

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

References

- Cortes, Corinna, and Vladimir Vapnik. Support-vector networks. Machine learning 20.3 (1995): 273-297.
- Cohen, Jacob. Statistical power analysis for the behavior science. Lawrance Eribaum Association (1988).
- Cohen, Jacob. A coefficient of agreement for nominal scales (1960).

See Also

runGenpenRf, runGenpenBayes, plotGenphenRfSvm, plotGenphenBayes, plotSpecificGenotype, plotManhattan

Examples

```
data(genotype.saap)
#or data(genotype.saap.msa) in this case you cannot subset genotype.saap[, 1:5]
data(phenotype.saap)
genphen.svm <- runGenphenSvm(genotype = genotype.saap[, 1:5],
                             phenotype = phenotype.saap,
                             cv.fold = 0.66,
                             cv.steps = 100,
                             hdi.level = 0.99)
```

Index