Package 'genphen'

October 12, 2016

Type Package

Title A tool for computing genotype-phenotype associations using statistical learning techniques

Version 1.0.0

Date 2016-03-20

Author Simo Kitanovski

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

Description Given a set of genetic polymorphisms in the form of single nucleotide poylmorphisms or single amino acid polymorphisms and a corresponding phenotype data, often we are interested to quantify their association such that we can identify the causal polymorphisms. Using statistical learning techniques such as random forests and support vector machines, this tool provides the means to estimate genotype-phenotype associations. It also provides visualization functions which enable the user to visually inspect the results of such genetic association study and conveniently select the genotypes which have the highest strenght ofassociation with the phenotype.

License GPL (>= 2)

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

LazyLoad yes

biocViews GenomeWideAssociation, Regression, Classification, SupportVectorMachine, Genetics, SequenceMatching

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 vecotr 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 vecotr 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)
```

`plotGenphenResults` *Visualizing GWAS results*

Description

This procedure plots the results obtained using runGenphenSnp or runGenphenSaap.

Usage

```
plotGenphenResults(genphen.results)
```

Arguments

`genphen.results`	Data.frame resulting from runGenphenSnp or runGenphenSaap.

Details

This procedure plots the results obtained using runGenphenSnp or runGenphenSaap. Each result entry is plotted as a point with respect to its effect size and classification accuracy attributes, whereby the color of the points is directly proportional to the classification accuracy. The region in the top-right corner of the plot is where the genotypes which have the strongest association with the phenotype are found.

Value

`plot`	ggplot plot object.

Author(s)

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

See Also

runGenphenSaap, runGenphenSnp, plotGenphenResults, 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.results <- runGenphenSnp(genotype = genotype.snp[, 1:5],
phenotype = phenotype.snp, technique = "svm", fold.cv = 0.66, boots = 100)
genphen.plot <- plotGenphenResults(genphen.results = genphen.results)
```

```
#Example 2:
data(genotype.saap)
#or data(genotype.saap.msa) in this case you cannot subset genotype.saap[, 1:5]
data(phenotype.saap)
genphen.results <- runGenphenSaap(genotype = genotype.saap[, 1:5],
phenotype = phenotype.saap, technique = "svm", fold.cv = 0.66, boots = 100)
genphen.plot <- plotGenphenResults(genphen.results = genphen.results)
```

`plotManhattan` *Visualizing genphen results with Manhattan plots*

Description

This procedure plots the results obtained using runGenphenSnp or runGenphenSaap.

Usage

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

Arguments

`genphen.results`
: Data.frame resulting from runGenphenSnp or runGenphenSaap.

Details

This procedure plots the results obtained using runGenphenSnp or runGenphenSaap. Each result entry is plotted as a point with respect to its effect size and classification accuracy attributes, whereby the color of the points is directly proportional to the classification accuracy. The region in the top-right corner of the plot is where the genotypes which have the strongest association with the phenotype are found.

Value

`plot` ggplot plot object.

Author(s)

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

See Also

runGenphenSaap, runGenphenSnp, plotGenphenResults, 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.results <- runGenphenSnp(genotype = genotype.snp[, 1:5],
phenotype = phenotype.snp, technique = "svm", fold.cv = 0.66, boots = 100)
manhattan.plot <- plotManhattan(genphen.results = genphen.results)

#Example 2:
data(genotype.saap)
#or data(genotype.saap.msa) in this case you cannot subset genotype.saap[, 1:5]
data(phenotype.saap)
genphen.results <- runGenphenSaap(genotype = genotype.saap[, 1:5],
phenotype = phenotype.saap, technique = "svm", fold.cv = 0.66, boots = 100)
manhattan.plot <- plotManhattan(genphen.results = genphen.results)
```

---

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

runGenphenSaap, runGenphenSnp, 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 ata(genotype.saap.msa)
data(phenotype.saap)
specific.genotype.plot <- plotSpecificGenotype(genotype = genotype.saap,
phenotype = phenotype.saap, index = 3)
```

`runGenphenSaap` *Performing genetic association analysis between SAAPs and phenotypes*

Description

This procedure quantifies the association between single amino acid polymorphisms (SAAPs) and phenotypes.

Usage

```
runGenphenSaap(genotype, phenotype, technique, fold.cv, boots)
```

Arguments

`genotype`	Character matrix or data frame, containing SAAPs as columns or alternatively as AAMultipleAlignment Biostrings object
`phenotype`	Numerical vector, where each element is a measured phenotype corresponding to the observations of the genotype data.
`technique`	Two techniques are provided: random forests (rf) or linear support vector machines (svm) (recommended = svm).
`fold.cv`	The cross-validation fraction (0, 1) of the data which is used to train the classifier (recommended = 0.66). The ramaining fraction (1-fold.cv) of the data is used to test the classifier.
`boots`	Number of bootstraps to be performed to estimate the classification accuracy and the corresponding confidence intervals (recommended >= 100).

Details

This procedure takes two types of data as input: first a genotype data composed of a set of 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.

Using these two data types, it quantifies the association between each SAAP and the phenotype. SAAPs are more complex than SNPs because they may be composed of more than two genetic states, i.e. more than two types of amino acid states, whereas SNPs are exclusively composed of only two genetics states, i.e. two nucleotide states. To compute the association between a SAAP and a phenotype, the SAAP is first deconstructed into its amino acid substitution pairs. Following the deconstruction of a SAAP, the procedure computes the association between each amino acid substitution pair and the phenotype with respect to the two metics "effect size" and "classification accuracy".

The effect size of an amino acid substitution pair is estimated by computing the Cohen's d statistics (Cohen 1988). The 95% confidence intervals are computed as well. The effect size quantifies the phenotypic effect of substituting one amino acid state for the other at the specific SAAP site. Substitution pairs characterized with a substantial effect size and tight confidence intervals which do not include the null effect are to be prioritized.

Classification accuracy is the second metric which is computed using statistical learning techniques. This is the metric which is used to quantify the strength of the association between an amino acid substitution pair and a phenotype. The idea is to use either linear suppport vector machines or random forests to build a classification model between the phenotype vector and the substitution pair vector. The more accurate the model, the easier we can predict the two states of the substitution pair from the phenotype and hence the stronger is the mutual association between the two vectors. In order to obtain a robust classification accuracy measure, the classification analysis is done in a bootstrapping fashion. First a subset of the substitution- phenotype vectors is randomly selected to train a classifier, while the remaining data is used to test the classifier. This step is repeated multiple times after which the classification accuracies of all the classifiers are averaged into a single classification accuracy measure and the corresponding confidence intervals are computed.

In order 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.

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 95% CI.

`ca, ca.CI.low, ca.CI.high, ca.CI.length`
: Mean classification accuracy and its 95% CI.

`kappa, kappa.CI.low, kappa.CI.high, kappa.CI.length`
: Cohen's kappa statistics and its 95% CI.

`site, aa1, aa2, count.aa1, count.aa1`
: General information about the genotype.

`anova.score`
: P-value score from an ANOVA test.

Author(s)

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

References

Cohen, J. (1988) Statistical power analysis for the behavioral sciences (2nd ed.). New York:Academic Press.

Cohen, J. (1960) A coefficient of agreement for nominal scales.

See Also

runGenphenSnp, plotGenphenResults, plotSpecificGenotype, plotManhattan

Examples

```
data(genotype.saap)
#or data(genotype.saap.msa) in this case you cannot subset genotype.saap[, 1:3]
data(phenotype.saap)
genphen.results <- runGenphenSaap(genotype = genotype.saap[, 1:3],
phenotype = phenotype.saap, technique = "svm", fold.cv = 0.66, boots = 100)
```

---

runGenphenSnp — *Performing genetic association analysis between SNPs and phenotypes*

Description

This procedure computes the association between single nucleotide polymorphisms (SNPs) and phenotypes.

Usage

```
runGenphenSnp(genotype, phenotype, technique, fold.cv, boots)
```

Arguments

genotype	Character matrix or data frame, containing SNPs as columns or alternatively a DNAMultipleAlignment Biostrings object
phenotype	Numerical vector, where each element is a measured phenotype corresponding to the observations of the genotype data.
technique	Two techniques are provided: random forests (rf) or linear support vector machines (svm) (recommended = svm).
fold.cv	The cross-validation fraction (0, 1) of the data which is used to train the classifier (recommended = 0.66). The ramaining fraction (1-fold.cv) of the data is used to test the classifier.
boots	Number of bootstraps to be performed to estimate the classification accuracy and the corresponding confidence intervals (recommended >= 100).

Details

This procedure takes as an input two types of data: first a genotype data composed of single nucleotide polymorphism (SNP) sites, each of which is represented by a column of alleles, whereby at most two types of alleles should exist in each column; second a numerical phenotype vector, where the elements sorted to correspond to the rows of the genotype data.

Using these two data types, it computes the association between each SNP and the phenotype. For each SNP two metrics are computed, called "effect size" and "classification accuracy".

The effect size of a given SNP is obtained by computing the Cohen's d statistics (Cohen 1988). The 95% confidence intervals are computed as well.

Classification accuracy is the second metric which is computed using statistical learning techniques. This is the metric which is used to quantify the strength of the association between a SNP and a phenotype. The idea is to use either linear suppport vector machines or random forests to build a classification model between the phenotype vector and the SNP vector. The more accurate the model, the easier we can predict the two allele states of the SNP from the phenotype and hence the stronger is the mutual association between the two vectors. In order to obtain a robust classification accuracy measure, the classification analysis is done in a bootstrapping fashion. First a subset of the SNP-phenotype vectors is randomly selected to train a classifier, while the remaining data is used to test the classifier. This step is repeated multiple times after which the classification accuracies of all the classifiers are averaged into a single classification accuracy measure and the corresponding confidence intervals are computed.

In order 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.

Value

Five classes of results are computed for each SNP 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 95% CI.

`ca, ca.CI.low, ca.CI.high, ca.CI.length`
: Mean classification accuracy and its 95% CI.

`kappa, kappa.CI.low, kappa.CI.high, kappa.CI.length`
: Cohen's kappa statistics and its 95% CI.

`site, allele1, allele2, count.allele1, count.allele2`
: General information about the genotype.

`anova.score`
: P-value score from an ANOVA test.

Author(s)

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

References

Cohen, J. (1988) Statistical power analysis for the behavioral sciences (2nd ed.). New York:Academic Press.

Cohen, J. (1960) A coefficient of agreement for nominal scales.

See Also

runGenphenSaap, plotGenphenResults, plotSpecificGenotype, plotManhattan

Examples

```
data(genotype.snp)
#or data(genotype.snp.msa) in this case you cannot subset genotype.snp[, 1:3]
data(phenotype.snp)
genphen.results <- runGenphenSnp(genotype = genotype.snp[, 1:3],
phenotype = phenotype.snp, technique = "svm", fold.cv = 0.66, boots = 100)
```

Index