Package ‘TSGS’

January 20, 2025

Title Trait Specific Gene Selection using SVM and GA

Version 1.0

Description

Obtaining relevant set of trait specific genes from gene expression data is important for clinical diagnosis of disease and discovery of disease mechanisms in plants and animals. This process involves identification of relevant genes and removal of redundant genes as much as possible from a whole gene set. This package returns the trait specific gene set from the high dimensional RNA-seq count data by applying combination of two conventional machine learning algorithms, support vector machine (SVM) and genetic algorithm (GA). GA is used to control and optimize the subset of genes sent to the SVM for classification and evaluation. Genetic algorithm uses repeated learning steps and cross validation over number of possible solution and selects the best. The algorithm selects the set of genes based on a fitness function that is obtained via support vector machines. Using SVM as the classifier performance and the genetic algorithm for feature selection, a set of trait specific gene set is obtained.

License GPL-2 | GPL-3

Encoding UTF-8

Imports caret, edgeR, fastmatch, genalg, kernlab, e1071

URL https://github.com/SudhirSrivastava/TSGS

BugReports https://github.com/SudhirSrivastava/TSGS/issues

RoxygenNote 7.1.1.9001

NeedsCompilation no

Author Md. Samir Farooqi [aut],
K.K. Chaturvedi [aut],
D.C. Mishra [aut],
Sudhir Srivastava [cre, aut]

Maintainer Sudhir Srivastava <Sudhir.Srivastava@icar.gov.in>

Repository CRAN

Date/Publication 2021-09-08 09:40:04 UTC

Contents

featureSelect *Trait specific gene selection using SVM and GA*

Description

This function gives the optimal set of informative genes based on RNA-Seq count data

Usage

```
featureSelect(X, y, p = 5, n.iter = 1, alpha = 0.05, p.adj.method = "bonferroni")
```

Arguments

`X` X is a G x N data frame of gene expression values (raw count data) where rows represent genes and columns represent samples. Each cell entry represents the read counts of of a gene in a sample (row names of X as gene names or gene ids)

`y` y is a N x 1 numeric vector with entries 0 or 1 representing sample labels, where, 0/1 represents the sample label of samples for two conditions, e.g., 0 for Control and 1 for Case

`p` Population size, by default 5

`n.iter` The number of iterations, by default 1

`alpha` The level of significance, by default 0.05

`p.adj.method` Method of adjusting p-values, by default "bonferroni". The other methods available are "BH", "holm", "hochberg", "hommel", "BY".

Value

`InformativeGenes`
List of informative genes selected

`LogCPM` Log cpm data of informative genes

`DEA_Result` Differential Expression Analysis Result of informative genes

Author(s)

c(person("Md. Samir", "Farooqi", email = "ms.Farooqi@icar.gov.in", role = "aut"), person("K.K.", "Chaturvedi", email = "kk.Chaturvedi@icar.gov.in", role = "aut"), person("D.C.", "Mishra", email = "Dwijesh.Mishra@icar.gov.in", role = "aut"), person("Sudhir", "Srivastava", email = "Sudhir.Srivastava@icar.gov.in", role = c("cre","aut")))

Examples

```
filename <- system.file("extdata", "exampleData.csv", package = "TSGS")
cdata <- read.csv(filename, header = TRUE, row.names = 1, stringsAsFactors = FALSE)
X <- as.data.frame(cdata[-1,])
y <- as.numeric(cdata[1,])
set.seed(100)
result <- featureSelect(X, y, 5, 1, 0.05, "bonferroni")
gene_list <- result$InformativeGenes
logcpm_data <- result$LogCPM
dea_result <- result$DEA_Result
```

Index