

Package ‘StatFingerprints’

April 22, 2009

Title StatFingerprints

Version 1.3

Date 2009-03-19

Author Rory Michelland <rory.michelland@gmail.com>, Laurent Cauquil
<laurent.cauquil@toulouse.inra.fr>

Maintainer Laurent Cauquil <laurent.cauquil@toulouse.inra.fr>

Description processing and statistical analysis of molecular fingerprint profiles

Depends rgl,car,MASS,ffmanova,akima,labdsv,RColorBrewer,vegan,tcltk2,tcltk,maptools,sciplot

License GPL-2

Repository CRAN

Date/Publication 2009-04-21 07:11:22

R topics documented:

Internal functions	1
StatFingerprints	2

Index	6
--------------	----------

Internal functions *Internal functions for the StatFingerprints package*

Description

Internal functions for the StatFingerprints package

Details

These are not to be called by the user

Author(s)

Rory Michelland <rory.michelland@gmail.com>
Laurent Cauquil (laurent.cauquil@toulouse.inra.fr)

See Also

[StatFingerprints](#)

StatFingerprints *Graphical user interface of StatFingerprints package*

Description

A Tcl/Tk GUI to use functions of the StatFingerprints package.

Menu bar

The menu bar at the top of the window is used to launch all the functions of the package.

The code `Files` menu is used to create, save and load a new project, and to quit R:

- `New project...` is composed of two submenus: the first `Import fingerprint profiles` to import fingerprint profiles, and the second `Import variables` (quantitative or qualitative) to import tables of external variables. The fingerprint profiles can be imported in three different ways: i) from a folder containing FSA files obtained with an ABI Prism 310 or 3100 sequencer (choose `Convert FSA files and import`), ii) from a folder containing ASCII files (choose `Import ASCII files`), iii) from an ecological table with fingerprint profiles in rows or columns (choose `Import an ecological table`). The last two of these can read all ASCII formats.

- `Load project` is used to import a StatFingerprints project previously saved as an Rdata file.

- `Save project as...` is used to save a StatFingerprints project in a specific directory. The project is saved as an Rdata file.

- `Save` is used to save the current StatFingerprints project in its current directory.

- `Quit R` opens a dialog window asking whether the environment should be saved before quitting R.

The `Edit` menu is used to rename, add or delete fingerprint profiles of the project:

- `Change names of fingerprint profiles` renames the name of the fingerprint profiles in the current project.

- `Add fingerprint profiles to the project` adds new fingerprint profiles within the current project. The new fingerprint profiles must be first processed in another new project and then this new project is merged with the current project.

- `Delete fingerprint profiles in the project` deletes fingerprint profiles in the current project.

-Select fingerprint profiles using levels of factors offers the possibility of selecting several fingerprint profiles in the current project corresponding to one or more levels of a factor.

The Fingerprint profiles processing menu includes all functionalities required to process the fingerprint profiles. The fingerprint profiles must be processed before statistical analyses to make them comparable.

- Step 1: define standard is needed to define the peaks of the reference standard on which fingerprint profiles will be aligned together using the Align fingerprint profiles one by one function. The x-values of each peak of the reference standard can be either manually entered using Define peaks of your own standard or automatically selected if the ROX HD400 is used with the Use peaks of ROX HD400 Applied Biosystems function.
- Step 2: align fingerprint profiles to the standard firstly allows the fingerprint profiles to be aligned one by one using the Align fingerprint profiles one by one function and secondly to check the alignment using Option: Check quality of alignment.
- Step 3: define a common baseline for all fingerprint profiles is used to homogenise and reorientate the baseline between fingerprint profiles on a horizontal axis.
- Option: Delete backgrounds under fingerprint profiles can be used to delete background under fingerprint profile.
- Step 4: define the range of the fingerprint profiles allows just the section where the community appears to be selected by deleting extremities of the fingerprint profiles.
- Option: Rebuild peaks of fingerprint profiles presenting defects is used to redesign a peak presenting a defect into a fingerprint profile.
- Step 5: normalise area under fingerprint profiles is used to normalise the area under each fingerprint profile so that the area under a signal is equal to one. Three different algorithms are available to deal with minimum and negative values.
- Option: Transform profiles into presence/absence profiles allows quantitative fingerprint profiles to be transformed into binary fingerprint profiles so that areas with and without peaks are respectively equal to one and zero.

The Plot menu includes all functionalities to plot the fingerprint profiles of the project and the ordination graphs:

- Plot fingerprint profiles is used to plot one or several fingerprint profiles in two dimensions or the entire set of fingerprint profiles in three dimensions for any processing step of the fingerprint profiles.
- Plot saved nMDS/PCA: advanced tools plots previously calculated non-Metric Multidimensional Scaling (nMDS) and Principal Components Analysis (PCA) in a 2- or 3-dimensional display. PCA and nMDS are plotted with coloured points according to a factor and/or with contour lines coming from a quantitative variable.

The Univariate statistic: diversity index menu can be used both to estimate diversity indexes and to compute basic univariate statistics. It illustrates the effect of factors or external parameters on the diversity of the community.

- Compute diversity index allows various diversity indexes to be estimated.
- Descriptive statistic calculates the arithmetic mean and standard deviation of diversity indexes or imported quantitative variables.
- Multifactor ANOVA displays summary results of multifactor ANOVA using a diversity index or imported quantitative variables as the variable to be explained and one or more imported qualitative variables as explanatory variables.
- Simple correlation computes a simple correlation using a Pearson algorithm between two quantitative variables (diversity index or imported quantitative variable). This function returns the Pearson R-squared, its associated p value and the first degree regression coefficient.

The `Multivariate statistic: structure` menu is composed of exploratory statistical methods and multivariate statistical tests. It allows the similarity between fingerprint profiles to be explored, to test the effect of a factor along the fingerprint profiles and to assess the relationship between the structure of the community and external parameters.

- Non-Metric multiDimensional Scaling (nMDS) computes iterative nMDS and keeps the best one, i.e. that with the lowest value of the Kruskal stress.
- Principal Components Analysis (PCA) computes PCA, optionally centred and or scaled. - Compare PCA vs nMDS compares PCA and nMDS by calculating the Pearson R squared between the initial distances and the new distances computed by the two ordination methods.
- Hierarchical clustering displays a dendrogram. Several proximity indexes and dendrogram algorithms are proposed.
- Heatmap displays a coloured heatmap based on hierarchical clustering.
- Multivariate ANOVA (50 50 F test and rotation) computes general linear modelling of one or more qualitative variables with the fingerprint profile set. - ANalysis Of SIMilarity (ANOSIM) tests whether there is a significant difference between two or more groups of fingerprint profiles. Groups of fingerprint profiles are designed according to the levels of a qualitative variable with the `Global ANOSIM: test effect of factor` function or to each pair of levels within a factor with the `Pairwise ANOSIM: test effect of levels within factor` function.
- Within group variability calculates within each group the variability between all fingerprint profiles and tests whether these variabilities are significantly different. Groups are designed according to the levels of a factor.
- SIMilarity PERcentages procedure (SIMPER) calculates the relative contribution of each variable of the fingerprint profiles to the total dissimilarity (measure by a proximity index) between two groups of fingerprint profiles. The two groups are designed according to the levels of a factor. It is a simple method for assessing which areas along the fingerprint profiles are primarily responsible for an observed difference between the two groups. - Iterative tests (ANOVA, Mann Whitney, Fisher s exact) tests the significant differences of each variable along the fingerprint profile between two groups. The test used is the t test, the Mann Whitney test or the Fisher s exact test. The two groups are designed according to the levels of a factor. Consequently it allows assessing which areas of the fingerprint profiles are significantly responsible for the difference observed between the two groups.
- Multivariate correlation (50 50 F test and rotation) is used to test the relationship between the structure of the fingerprint profile set and an imported quantitative vari-

able.

The `Help` menu contains the user guide manual, an on line help, the most recent version of the present program and the bugs report.

Author(s)

Rory Michelland <rory.michelland@gmail.com>
Laurent Cauquil <laurent.cauquil@toulouse.inra.fr>

Examples

```
## Not run:  
## One of these lines allows the StatFingerprints program to be started  
StatFingerprints()  
StatFingerprint()  
statfingerprints()  
statfingerprint()  
SF()  
sf()  
## End(Not run)
```

Index

*Topic **multivariate**

Internal functions, 1

addGUI (*Internal functions*), 1

align (*Internal functions*), 1

alignGUI (*Internal functions*), 1

alignGUI2 (*Internal functions*), 1

anovaGUI (*Internal functions*), 1

askquitGUI (*Internal functions*), 1

baseline (*Internal functions*), 1

baselineGUI (*Internal functions*), 1

best.nmnds (*Internal functions*), 1

best.nmndsGUI (*Internal functions*), 1

best.ordination (*Internal functions*), 1

best.ordinationGUI (*Internal functions*), 1

binary (*Internal functions*), 1

binaryGUI (*Internal functions*), 1

change.nameGUI (*Internal functions*), 1

checkfact (*Internal functions*), 1

checkparam (*Internal functions*), 1

checkprofile (*Internal functions*), 1

convertGUI (*Internal functions*), 1

correlationGUI (*Internal functions*), 1

delete.background (*Internal functions*), 1

delete.backgroundGUI (*Internal functions*), 1

deletefactGUI (*Internal functions*), 1

deleteGUI (*Internal functions*), 1

dendoGUI (*Internal functions*), 1

des.univGUI (*Internal functions*), 1

diversities (*Internal functions*), 1

diversitiesGUI (*Internal functions*), 1

ffcorGUI (*Internal functions*), 1

globalanosimGUI (*Internal functions*), 1

heat.map (*Internal functions*), 1

heat.mapGUI (*Internal functions*), 1

importGUI1 (*Internal functions*), 1

importGUI2 (*Internal functions*), 1

importGUI3 (*Internal functions*), 1

Internal functions, 1

iterative.test (*Internal functions*), 1

iterative.testGUI (*Internal functions*), 1

iterative.testGUI1 (*Internal functions*), 1

loadGUI (*Internal functions*), 1

make.peak (*Internal functions*), 1

make.peakGUI (*Internal functions*), 1

newdist (*Internal functions*), 1

newrdata (*Internal functions*), 1

nmnds.contour (*Internal functions*), 1

normalisation (*Internal functions*), 1

normalisationGUI (*Internal functions*), 1

oppdf (*Internal functions*), 1

pcaGUI (*Internal functions*), 1

peakparameters (*Internal functions*), 1

permanovaGUI (*Internal functions*), 1

plot2dGUI (*Internal functions*), 1

plot3dim (*Internal functions*), 1

plot3dimGUI (*Internal functions*), 1

plotord3dGUI (*Internal functions*), 1

plotordGUI (*Internal functions*), 1

pwanosim (*Internal functions*), 1

pwanosimGUI (*Internal functions*), 1

rollball (*Internal functions*), 1

roxdefault (*Internal functions*), 1

roxnewGUI (*Internal functions*), 1

saveasGUI (*Internal functions*), 1

saveGUI (*Internal functions*), 1

SF (*StatFingerprints*), 2

sf (*StatFingerprints*), 2

simper (*Internal functions*), 1

simperGUI (*Internal functions*), 1

simperGUI1 (*Internal functions*), 1

StatFingerprint (*StatFingerprints*), 2

statfingerprint (*StatFingerprints*), 2

StatFingerprints, 2, 2

statfingerprints (*StatFingerprints*), 2

trunck (*Internal functions*), 1

trunckGUI (*Internal functions*), 1

withingroupvariability (*Internal functions*), 1

withingroupvariabilityGUI (*Internal functions*), 1