Using pre-genereated report for the analysis of peptide microarray data

Renan Sauteraud*

October 10, 2013

Contents

*rsautera@fhcrc.org

1 Introduction

This vignette presents how to generate and use an automated report to easily analyse peptide microarray data. Reports aim at making the analysis accessible and reproducible. However, this set of function does not replace the use of all packages involved, in an interactive session. The options available for the customization of the analysis are a subset of what these packages can do. For a more customized analysis and a detailed description of the options offered by these packages, please refer to the other vignettes and the man pages.

As with any other R package, `pepStat` has to be loaded first in order to use the report creation functions.

```
library(pepStat)
```

2 Creating a new project

The following instruction creates new folder 'project' in the current path. The directory will contain all the required subfolders as well as the first report and a configuration file.

```
mk_project(path = "./", project_name = "project", report_name = "report1")

## Creating a new project in .//project
```

3 Importing data

The 'data' directory contains two subfolders 'input' and 'output'. All the .gpr files as well as the mapping file should be copied into the 'input' folder.

All the files in this folder will be used by the functions in the reports. If two reports use different sets of data (even a subset of the other), a new project must be created.

`pepStat`'s user guide describe the required input in details in the **Generating a peptideSet** section.

4 Running the analysis

The report itself is the document that contains the R code used for the analysis. It is an R markdown document located in the 'reports' folder.

`pepStat` makes use of the`knitr` package to read and execute the report file.

The chunk of code below executes the report 'report1' if it exists in the project.

```
run_report(path = "./", project_name = "project", report_name = "report1")
```

`run_report` is essentially a wrapper around `knitr`'s `knit2pdf` function. A single project can have multiple reports and each report can have a different set of options. However, all the reports will use the same data file.

5 Retrieving the results

All results are placed in the 'data/output/' folder. The output of the report can be found here as well as anything that gets written on disk during the execution of the report, such as `peptidSets` saved as csv files or extra figures.

6 Settings

There are a few options related to the analysis that can be modified by the user. Each report has an associated configuration file located in the 'settings' subfolder of the project. The options are described using the yaml format.

6.1 pepStat options

Here we list the options available for the functions related to `pepStat`.

- mapping: mapping.csv
  Changing this filename changes the mapping file associated with the report. The file should be a valid csv file located in the the 'data/input/' folder.

- collection: pep_hxb2
  This is the peptide collection used to get the coordinates of the peptides on the array. It should be one of the valid datasets of the `PEP.db` package.

- call.cutoff: 0.1
  The cutoff used in the `makeCalls` function.

6.2 HIV.db options

The following options are available for customizing the annotations associated with the analysis.

- genome: hxb2 The reference genome for the annotations. Available: hxb2, mac239.

- features:
  - V1
  - V2
  - TM
  - ...
  The list of features that should be annotated in the figures. Add elements on new lines to add more features. The features should be available in `HIV.db` for the selected genome. To get a list of available features for the selected genome, refer to `HIV.db` vignette and `?HIV.db::getFeature`.

In order to draw custom plots, or use more advanced options, use the packages in an interactive session.

7 Removing a project

A project can be removed manually by removing the top level folder. For convenience and safety purpose, `pepStat` also offer a function to delete the project from R.

```
rm_project(path = "./", project_name = "project")

## Removing the project .//project
```

This function make some checks and removes the main folder if it is indeed a project or send an error message if it is not.

8 Session info

```
sessionInfo()

## R Under development (unstable) (2013-10-02 r64018)
## Platform: x86_64-unknown-linux-gnu (64-bit)
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] PEP.db_0.99.5        pepStat_0.99.6       IRanges_1.19.38
## [4] Biobase_2.21.7       BiocGenerics_0.7.5   knitr_1.5.4
## [7] BiocInstaller_1.11.4
##
## loaded via a namespace (and not attached):
##  [1] colorspace_1.2-4   dichromat_2.0-0    digest_0.6.3
##  [4] evaluate_0.5.1     fields_6.8         formatR_0.9
##  [7] ggplot2_0.9.3.1    grid_3.1.0         gtable_0.1.2
## [10] highr_0.2.1        labeling_0.2       limma_3.17.25
## [13] MASS_7.3-29        munsell_0.4.2      plyr_1.8
## [16] proto_0.3-10       RColorBrewer_1.0-5 reshape2_1.2.2
## [19] scales_0.2.3       spam_0.40-0        stats4_3.1.0
## [22] stringr_0.6.2      tools_3.1.0
```