The pepStat user guide

Raphael Gottardo[*], Greg Imholte[†], Mike Jiang[‡]and Renan Sauteraud[§]

April 23, 2013

A step-by-step guide in the analysis of peptide microarray antibody binding

Contents

[*] rgottard@fhcrc.org

[†] gimholte@uw.edu

[‡] wjiang2@fhcrc.org

[§] rsautera@fhcrc.org

1 Introduction

The pepStat package offers a complete analytical framework for the analysis of peptide microarray data. It includes a novel normalization method to remove non-specific peptide binding activity of antibodies, a data smoothing reducing the background noise and subject-specific positivity calls.

1.1 Loading the package

As with any R package, it should first be loaded in the session.

```
> library(pepStat)
```

2 Generating a peptideSet

The reading function takes a path as its argument and parse all the files in the given directory.

```
> mapFile <- system.file("extdata/mapping.csv", package = "pepStat")
> dirToParse <- system.file("extdata/RVV", package = "pepStat")
> pSet <- makePeptideSet(files = NULL, path = dirToParse, mapping.file = mapFile,
+     rm.control.list = c("empty", "none", "JPT-", "Ig", "Cy",
+         "landmark"), log = TRUE)
```

makePeptideSet returns a new object of class peptideSet. This is the base structure used in pepStat. It contains only the sequence of the peptides and the intensities measured. Any additional information must be passed using a known collection of peptides. In this example, we use pep_hxb2 available in the PEP.db package.

```
> library(PEP.db)
> data(pep_hxb2)
> psSet <- summarizePeptides(pSet, summary = "mean", position = pep_hxb2)
```

pep_hxb2 gives information regarding the position of each peptide, their z-scores, the clades they belong to and the alignment with the reference sequence HXB2.

3 Normalizing the peptideSet

The primary goal of the data normalization step is to remove non-biological source of bias and increase the comparability of true positive signal intensities across slides. The method developped for this package uses physiochemical properties of individual peptides to model non-specific antibody binding to arrays.

```
> pnSet <- NormalizeArray(psSet)
```

An object of class `peptideSet` containing the corrected peptides intensities is returned.

4 Data smoothing

The data smoothing step takes advantage of the overlapping nature of the peptides on the array to remove background noise caused by experimental variation. It is expected that the binding effect of two overlapping peptides will be positively correlated. Therefore, `pepStat` use a sliding mean technique technique to borrow strength across neighboring peptides and to reduce signal variability. Even though this is a loss of resolution, this statistic increases detection of binding *hotspots* that noisy signals might otherwise obscure.

```
> Vnorm <- subset(pnSet, treatment != "PLACEBO")
> V_smooth <- slidingMean(Vnorm, width = 9)
```

The returned `peptideSet` intensities distribution is centered around zero.

5 Making calls

The final step is to make the positivity calls.

```
> cutoff <- 1.1
> V_calls <- makeCalls(V_smooth, cutoff = cutoff, method = "absolute")
```

`makeCalls` returns a `numeric` vector with a single value for each peptide. If the score for a given peptide is superior to 0. The peptide is positive for antibody binding.