prot2D : Statistical Tools for volume data from 2D Gel Electrophoresis

Sébastien Artigaud *

October 30, 2017

Contents

1 Introduction 1

2 Input data for prot2D 2

3 Getting started 2

4 prot2D workflow 2
   4.1 Vizualize and Normalize volume data . . . . . . . . . . . . . . . . 2
   4.2 Coerce data into an ExpressionSet . . . . . . . . . . . . . . . . . . 3
   4.3 Find differentially expressed proteins . . . . . . . . . . . . . . . . 4

5 Simulation of 2D Volume data 5

6 Session Info 7

7 References 8

1 Introduction

This document briefly describes how to use the prot2D package. An R package designed to analyze (i.e. Normalize and select significant spots) data issued from 2D SDS PAGE experiments. prot2D provides a simple interface for analysing data from 2D gel experiments. Functions for normalization as well as selecting significant spots are provided. Furthermore, a function to simulates realistic 2D Gel volume data is also provided.If you use this package please cite :


*Laboratoire des Sciences de l’Environnement Marin, LEMAR UMR 6539, Université de Bretagne Occidentale, Institut Universitaire Européen de la Mer, 29280 Plouzané, France
Table 1: Example of input data for prot2D with 3 replicates gels in each condition

<table>
<thead>
<tr>
<th>Spot_1</th>
<th>Replicates Condition 1</th>
<th>Replicates Condition 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gel1</td>
<td>Gel2</td>
</tr>
<tr>
<td>X_{1,1}</td>
<td>X_{2,1}</td>
<td>X_{3,1}</td>
</tr>
<tr>
<td>X_{1,2}</td>
<td>X_{2,2}</td>
<td>X_{3,2}</td>
</tr>
<tr>
<td>X_{1,i}</td>
<td>X_{2,i}</td>
<td>X_{3,i}</td>
</tr>
</tbody>
</table>

2 Input data for prot2D

prot2D uses raw 2D volume data intensities as input, datasets must be exported from specialized Image Software in the form of a dataframe of volume data $X_{j,i}$ with gels $j$ as columns and spots $i$ as rows. Note that the name of columns should therefore correspond to the names of the gels and the names of the rows to the name of the spots. The replicates for each condition should be ordered in the following columns (see Table 1). Furthermore, another dataframe is needed to describe the experiment with the names of gels as rownames and a single column giving the two level of condition for data.

3 Getting started

To load the prot2D package into your R environment type:

```r
> library(prot2D)
```

In this tutorial we will be using the pecten dataset obtained from a 2-DE experiment performed on proteins from the gills of *Pecten maximus* subjected to a temperature challenge. 766 spots were identified with 6 replicates per condition, therefore the dataset is a dataframe of 766 rows and 12 columns (for details, see Artigaud et al., 2013). pecten.fac describe the data by giving the names of the gels (as rownames) and the condition for the temperature challenge (15C = control vs 25C) in the "Condition" column, load by typing:

```r
> data(pecten)
> data(pecten.fac)
```

4 prot2D workflow

4.1 Vizualize and Normalize volume data

Vizualization

Dudoit et al. (2002) proposed a method for the visualization of artifacts in microarray datasets, called the MA-plot, which was transposed for proteomics data as the Ratio-Intensity plot (Meunier et al., 2005; R-I plot).
It consists in plotting the intensity $\log_2$-ratio ($R$) against mean $\log_{10}$ intensity ($I$):

$$ R = \log_2 \frac{\text{mean}(V_{\text{Cond}2})}{\text{mean}(V_{\text{Cond}1})} $$  \hspace{1cm} (1)

The intensity ($I$) is the $\log_{10}$ of the mean of volume data in condition 2 by the mean of volume data in condition 1:

$$ I = \log_{10}(\text{mean}(V_{\text{Cond}2}) \times \text{mean}(V_{\text{Cond}1})) $$  \hspace{1cm} (2)

Where $V_{\text{Cond}1}$ and $V_{\text{Cond}2}$ are spot volumes for conditions 1 and 2, respectively. In prot2D, R-I plot can be easily displayed with RIplot:

```r
> RIplot(pecten, n1=6, n2=6)
```

RIplot requires data supplied as a dataframe or a matrix as well as the number of replicates for each conditions.

**Normalization**

2D Gel Volume data must be normalized in order to remove systemic variation prior to data analysis. Two widely used methods are provided, the "Variance Stabilizing Normalization" (vsn) and the "Quantiles, Normalization". The principle of the "quantiles normalization" is to set each quantile of each column (i.e. the spots volume data of each gels) to the mean of that quantile across gels. The intention is to make all the normalized columns have the same empirical distribution. Whereas the vsn methods relies on a transformation $h$, of the parametric form $h(x)=\arcsinh(a+bx)$ (Huber et al., 2002). The parameters of $h$ together with those of the calibration between experiments are estimated with a robust variant of maximum-likelihood estimation. Both methods recentered the data around a zero log ratio, nevertheless for low values of intensities the vsn normalized data seems to be less efficient in order to recentered the cloud of points. Users are thus advised to use the quantiles normalization via a call to Norm.qt.

```r
> pecten.norm <- Norm.qt(pecten, n1=6, n2=6, plot=TRUE)
```

**4.2 Coerce data into an ExpressionSet**

Prior to analysis for finding differentially expressed proteins, data must be coerced into an ExpressionSet. This can be done easily with ES.prot, which requires a matrix of normalized volume data, the number of replicates in each condition and a dataframe giving the condition for the experiment.

```r
> ES.p <- ES.prot(pecten.norm, n1=6, n2=6, f=pecten.fac)
```

The matrix of spots intensities (i.e. Volume) is log2 transformed and stored in the assayData slot of the ExpressionSet. Furthermore, the log2-ratio is computed and stored in the featureData slot.
4.3 Find differentially expressed proteins

As described in Artigaud et al (2013) prot2D provide functions adapted from microarray analysis (from the st, samr, limma, fdrtool package). 2-DE experiments analysis require a variant of the t-statistic that is suitable for high-dimensional data and large-scale multiple testing. For this purpose, in the last few years, various test procedures have been suggested. prot2D provides:

- the classical Student’s t-test (in ttest.Prot function)
- two tests especially modified for micro-array analysis: Efron’s t-test (2001; efronT.Prot function) and the modified t-test used in Significance Analysis for Microarray (Tusher et al, 2001; samT.Prot function).
- two methods that take advantage of hierarchical Bayes methods for estimation of the variance across genes: the “moderate t-test” from Smyth (2004; modT.Prot function) and the ”Shrinkage t” statistic test from Opgen-Rhein and Strimmer (2007; shrinkT.Prot function).

As statistical tests allowing the identification of differentially expressed proteins must take into account a correction for multiple tests in order to avoid false conclusions. prot2D also provide different methods to estimate the False Discovery Rate:

- the classical FDR estimator of Benjamini and Hochberg (1995).
– the local FDR estimator of Strimmer (2008).

ttest.Prot function, modT.Prot function, samT.Prot function, efronT.Prot function or shrinkT.Prot function can be used to find differentially expressed proteins and the different FDR mode of calculation are implemented with the method.fdr argument. However, the moderate t-test with the FDR correction of Benjamini and Hochberg was found to be the best combination in terms of sensitivity and specificity. Thus, users are advised to use this combination by typing:

```r
> ES.diff <- modT.Prot(ES.p, plot=TRUE, fdr.thr=0.1,
  + method.fdr="BH")
```

Number of up-regulated spots in Condition 2

```
[1] 0
```

Number of down-regulated spots in Condition 2

```
[1] 1
```

The function returns an ExpressionSet containing only the spots declared as significant. A plot can also be generated to visualize the FDR cut-off. Additionally, it can be useful to select only the spots with an absolute ratio greater than 2, as they are often considered as the most biologically relevant proteins, this can be done by adding the command Fold2=T. The names of the selected spots can be retrieved with:

```r
> featureNames(ES.diff)
```

```
[1] "2607"
```

Displaying fold change (as log2(ratio)) for selected spots

```r
> head(fData(ES.diff))
```

```
   ratio
2607  -1.912657
```

Volume normalized data for selected spots

```r
> head(exprs(ES.diff))
```

```
  Br_23865  Br_23883  Br_23884  Br_23728  Br_23729  Br_23730  Br_23731  Br_23732
   Br_23733  Br_23875  Br_23876  Br_23877
```

5 Simulation of 2D Volume data

In order to compare FDR and the responses of the different tests as well as the influence of the number of replicates, simulated data can be used. Sim.Prot.2D simulates realistic 2D Gel volume data, based on parameters estimate from real dataset. Volume data are computed following these steps (see Smyth, 2004 and Artigaud et al., 2013 for details) :
Log2 mean volumes from data are computed for each spot.
- Means are used as input parameters in order to simulate a normal
distribution (with no differential expression between conditions) for
each spot with standard deviations computed as described by Smyth
(2004).
- A define proportion \( p_0 \) of the spots are randomly picked for introducing
differential expression in both conditions \((p_0/2 \text{ in each condition})\).

The `Sim.Prot.2D` returns an `ExpressionSet` of simulated volume data
(log2 transformed) with 2 conditions ("Cond1" and "Cond2" in `pheno-
Data`) slot of the `ExpressionSet`. The spots differentially generated can
be retrieve with `notes`. Simulate data based on "pecten"

```r
> Sim.data <- Sim.Prot.2D(data=pecten, nsp=700,
+          nr=10, p0=0.1, s2_0=0.2, d0=3)
```

Compare different methods for finding differentially expressed proteins

```r
> res.stud <- ttest.Prot(Sim.data, fdr.thr=0.1, plot=FALSE)
> res.mo <- modT.Prot(Sim.data, fdr.thr=0.1, plot=FALSE)
```

Number of up-regulated spots in Condition 2
```
[1] 23
```
Number of down-regulated spots in Condition 2
```
[1] 25
```

Names of the spots selected by student’s t-test with an FDR of 0.1
```
> featureNames(res.stud)
```
```
[1] "14"  "17"  "30"  "36"  "37"  "52"  "54"  "69"  "70"  "122"  "129"  "130"
[13] "150"  "153"  "158"  "163"  "173"  "186"  "213"  "225"  "236"  "239"  "249"  "250"
[25] "253"  "258"  "269"  "304"  "307"  "308"  "313"  "358"  "380"  "433"  "444"  "447"
[37] "457"  "494"  "502"  "543"  "554"  "564"  "575"  "579"  "621"  "639"  "659"  "676"
```

Names of the spots selected by modT-test with an FDR of 0.1
```
> featureNames(res.mo)
```
```
[1] "10"  "14"  "27"  "30"  "36"  "37"  "44"  "51"  "52"  "54"  "69"  "70"
[13] "92"  "120"  "121"  "122"  "129"  "130"  "150"  "154"  "158"  "163"  "166"  "167"
[25] "173"  "186"  "225"  "236"  "252"  "253"  "257"  "279"  "284"  "302"  "304"  "308"
[37] "313"  "334"  "358"  "380"  "401"  "417"  "429"  "433"  "444"  "457"  "494"  "502"
[49] "543"  "554"  "564"  "575"  "579"  "602"  "621"  "639"  "659"  "662"  "676"  "677"
```

Names of the differentially generated spots
> notes(Sim.data)$SpotSig

[1] "14" "30" "54" "88" "120" "121" "129" "130" "154" "186" "225"
[13] "236" "252" "253" "263" "300" "313" "334" "380" "433" "460" "463" "543"
[25] "575" "602" "610" "621" "644" "676" "10" "27" "36" "37" "51" "52"
[37] "62" "69" "70" "92" "119" "150" "163" "164" "257" "272" "279"
[49] "284" "308" "358" "398" "401" "417" "444" "457" "460" "494" "502"
[61] "554" "564" "579" "610" "659" "662" "677"

6 Session Info

> sessionInfo()

R version 3.4.2 (2017-09-28)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 16.04.3 LTS

Matrix products: default
BLAS: /home/biocbuild/bbs-3.6-bioc/R/lib/libRblas.so
LAPACK: /home/biocbuild/bbs-3.6-bioc/R/lib/libRlapack.so

locale:
[1] LC_CTYPE=en_US.UTF-8 LC_NUMERIC=C
[3] LC_TIME=en_US.UTF-8 LC_COLLATE=C
[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

attached base packages:
[1] grid parallel stats graphics grDevices utils datasets
[8] methods base

other attached packages:
[1] prot2D_1.16.0 qvalue_2.10.0 MASS_7.3-47
[4] Mulcom_1.28.0 fields_9.0 maps_3.2.0
[7] spam_2.1-1 dotCall64_0.9-04 limma_3.34.0
[10] Biobase_2.38.0 BiocGenerics_0.24.0 samr_2.0
[13] matrixStats_0.52.2 impute_1.52.0 st_1.2.5
[16] sda_1.3.7 corpor_1.6.9 entropy_1.2.1
[19] fdrtool_1.2.15

loaded via a namespace (and not attached):
[1] Rcpp_0.12.13 magrittr_1.5 splines_3.4.2 statmod_1.4.30
[5] munsell_0.4.3 colorspace_1.3-2 rlang_0.1.2 stringr_1.2.0
[9] plyr_1.8.4 tools_3.4.2 gtable_0.2.0 lazyeval_0.2.1
[13] tibble_1.3.4 reshape2_1.4.2 ggplot2_2.2.1 stringi_1.1.5
[17] compiler_3.4.2 scales_0.5.0

7
7 References


