pplr User Guide

Xuejun Liu, Magnus Rattray, Marta Milo, Neil D. Lawrence

July 13, 2006

1 Introduction

The *pplr* package is to detect differential gene expression (1) given the estimated gene expression levels and uncertainties of these measurements from probe-level analysis models, like mgMOS (2) and multi-mgMOS (3) (available in R-pacakge *mmgmos*). This package makes use of probe-level measurement error in deteting differentially expressed genes. When there are no replicate chips for each condition, *pplr* uses directly the probe-level variance. When each condition has several replicates, *pplr* combines probe-level variance with between-replicate variance before finding differential gene expression. The optimisation of parameters in *pplr* is done by donlp2 (4).

2 Step 1: Loading gene expression data

There are two ways to load expression data. If the expression data is stored in an instance of exprReslt class, eset, calculated from *mmgmos*, the following codes shows how to extract gene expression data from it.

```
R> e<-exprs(eset) ##extract the mean of expression value into a matrix
R> se<-se.exprs(eset)  ##extrac the standard deviation of expression
                       ##value into a matrix
```

If the results from *pplr* has already been saved in CSV files, data should be read from these files as the following.

```
R> e<-read.csv("filename_of_exprs.csv",check.names=FALSE,row.names=1)
                          ##read the mean of expression value
R> se<-read.csv("filename_of_se.csv",check.names=FALSE,row.names=1)
                    ##read the standard deviation of expression value
```

Make sure that gene expression values should be in log2 scale.

3 Step 2a: sing chip for each condition

When there are no replicates, use the loaded data directly as the folowing,

```
R> p<-pplr(data.frame(list(e,se)),1,2) ##1 is the column index of control,
                                       ##2 is the column index of experiment
R> write.csv(p,"filename.csv") ##save results into a CSV file
```

Refer to the help of pplr for the details of results from function pplr.

4 Step 2b: replicates for each condition

When there are replicates needed to be combined, use the function bcomb to combine replicate signal first.

```
R> r<-bcomb(e,se,replicates=c(1,1,1,2,2,2,3,3,3),method="em")
                          ## combining replicate signal
                          ## 3 replicates for each of 3 conditions
R> p<-pplr(r,1,2) ##1 is the column index of control,
                  ##2 is the column index of experiment
R> write.csv(p,"filename.csv") ##save results into a CSV file
```

The parameter "replicates" in bcomb should be a vector indicating which chip belongs to which condition. The length of the vector is the number of chips. The combination method can be either "map" or "em". Refer to the help of bcomb for more details by typing

```
R>?bcomb
```

References

[1] Liu,X., Milo,M., Lawrence,N.D. and Rattray,M. (2005) Probe-level variances improve accuracy in detecting differential gene expression. technical report available upon request

[2] Milo,M., Niranjan,M., Holley,M.C., Rattray,M. and Lawrence,N.D. (2004) A probabilistic approach for summarising oligonucleotide gene expression data. Technical report available upon request.

[3] Liu,X., Milo,M., Lawrence,N.D. and Rattray,M. (2005) A tractable probabilistic model for Affymetrix probe-level analysis across multiple chips. Bioinformatics, 21(18):3637-3644.

[4] Peter Spellucci. DONLP2 code and accompanying documentation. Electronically available via http://plato.la.asu.edu/donlp2.html.