RPS:Reproducibility Probability Score for Microarray

Guixian Lin <glin2@uiuc.edu>
Sheng Zhong <szhong@ad.uiuc.edu>

2006-11-01

Contents

1 Introduction

This software, written in the R language, computes reproducibility probability score to select differentially expressed genes. The Reproducibility Probability Score (RPS), takes into consideration both the replicated data in a particular lab and the measurement variability across labs. The measurement variability is assessed by utilizing the reference gene expression data generated in the Microarray Quality Control (MAQC) project. Specifically, we applied the data generated across replicate gene expression analysis that was conducted in multiple facilities as part of this effort. A larger RPS means a gene is more likely to be differentially expressed; and if similar transcription profiling measurements are made in other laboratories, it is highly likely to be confirmed.

2 Installation

2.1. Installing RPS package in R

1. Install a recent version of the R statistical package from the R Project for windows version.
2. For windows, download RPS zip file into your local machine, then click on the R icon on your desktop, pull down the `Packages` menu item and select `Install package(s) from local zip files`. In addition, the "lme4" package is required to be installed for using "RPS" package.

3. Download the pre-computed correlation data file across labs into your local machine and unzip the data file into a directory(for example, called “rps”); then pull down the `File` menu item and select Change dir… and set the working directory as the directory where you unzip your correlation data file.(for example, “rps”).

Till now you download all related files and can use RPS.

2.2 Using RPS

In Windows, while in R pull down the `Packages` menu item and select `Load package`. Select `RPS`.

Type ?RPS in R, if it shows RPS help window, that means you are successful to install RPS and ready to use it now.

2.3 How does RPS work?

1. Load the pre-computed correlation across labs for particular platform. For example, **load("ABI")** will load correlation for "ABI" platform.
2. Read the actual Microarray data for the corresponding platform. For example, if you load correlation for platform "ABI" in first step,that means your microarray data are from "ABI" platform. that is,
   RPS.readnew("ABI_12091.txt",transformed=T)
3. Use **RPS.select()** to compute the RPS for each probe set

3 Data and function list

3.1 Preparing the input files

In our precomputed correlation data(correlation.zip), we use the data from the MAQC.

3.2 A quick tour of the functions

The following table lists the functions available in RPS package. More details refer to RPS package.

Help pages for package `RPS' version 1.0

RPS-package	Reproducibility Probability Score(RPS) for Microarray Data
RPS.learn	Compute correlation rho across labs
RPS.load	Load correlation between labs for the pre-processed MAQC data
RPS.m.foldchange	Compute fold changes for microarray data

RPS.m.t.test	Do t-test
RPS.readnew	Read in the actual microarray data to be analyzed
RPS.select	Compute RPS

4. Examples

In the following, we will use ABI platform to show how to use RPS Software to compute Reproducibility Probability Score(RPS).

4.1 First load precomputed gene-specific correlations for different platforms using MAQC microarray dataset.

```
>library(RPS)  #load package 'RPS'
>load("ABI.RData") # load the correlations
##The probe set names can be obtained from the precomputed gene-
##specific correlations.
> probenames=names(corr$corr.Labs)
```

4.2 Specify sample information. For example, if the data include two samples with 5 replicated respectively, first 5 replicates are from sample A while the next 5 replicates are from sample B.

```
>n.replicate=5   # Specify the replicates for each sample
>test.sample.info=c(rep(1, n.replicate),rep(2, n.replicate))
```

4.3 Specify the number of labs to be simulated, which is recommended between 30 and 50.The larger the number, the longer it will take to compute RPS

```
>K=10
```

4.4 Specify p-value threshold and fold change for computing RPS

```
>p.value=1e-03 # p-value threshold for computing RPS
>FC=2.5        # Fold change's threshold for computing RPS
```

4.5 Read the actural microarray data to compute RPS. Here the data set include 3 labs,4 samples , and 5 replicates. If transformed=F, then the actural data need log2-tranformation.For example, the following data set only contain 12091 probe set from the platform ABI.

```
>MAQC.test=RPS.readnew("ABI_12091.txt",transformed=F)
```

4.6 Compute RPS using the function RPS.select(), whose return value is a list that includes RPS and K simulation data

```
>subcorr= RPS.subcorr(corr,geneId=MAQC.test[,1])
>results=RPS.select(MAQC.test[,2:11],test.sample.info,subcorr,n.replica
te,test="t",p.value,FC,K,flag=T)
```

4.7 Look at the results

```
>probeSetName=MAQC.test[,1] ##First column is the probe set names

#combine the probe set name with the corresponding computed RPS;
#Similarly, we can also combine the genes names into the results if we
#have
>rps=data.frame(probeSetName,results$RPS)
>summary(rps)

>j=2;#make sure that j<=K

##The simulated data saved in results$simu.data are in log2 scales.
>simu.dataOneLab=results$simu.data[,1:(n.replicate*2)+(j-
1)*(n.replicate*2)]

##The first n.replicate called “A” are from sample A while the next
##n.replicate called “B” are from sample B

>colnames( simu.dataOneLab)<-
c(rep("A",n.replicate),rep("B",n.replicate))
>summary(2^simu.dataOneLab)##
```