Sample Size and Power Analysis for Microarray Studies

Maarten van Iterson and Renée de Menezes
Center for Human and Clinical Genetics,
Leiden University Medical Center, The Netherlands
Package `SSPA`, version 1.5.5

April 14, 2011

Contents

1 Introduction

This document shows the functionality of the R-package `SSPA`. The package performs power and sample size analysis using a method described by [1, 2]. Our implementation allows for fast and realistic estimates of power and sample size for microarray experiments, given pilot data. By means of two simple commands (`pilotData(), sampleSize()`), a researcher can read their data in and compute the desired estimates. Other functions are provided to facilitate interpretation of results.

Given a set of test statistics from the pilot data, the knowledge of their distribution under the null hypothesis and the sample size used to compute them, the method estimates the power for a given false discovery rate. The multiple

testing problem is controlled through the adaptive version of the Benjamini and Hochberg method [3]. For more details about the implementation and method we refer to [4, 1, 2]. In [5] we describe two biological case studies using the packageSSPA.

For comparison we implement the power and sample size estimation method proposed by Ruppert [6] which uses a different estimation approach. Two additional packages need to be installed for using this method namely `quadprog` and `splines`.

2 Basic Example

We demonstrate the functionality of this package by using the preprocessed gene expression data from the leukemia ALL/AML study of [7] from the `multtest` package. To load the leukemia dataset, use `data(golub)`, and to view a description of the experiment and data, type `?golub`. The number of samples per group are shown by `table(golub.cl)` where ALL is class 0 and AML class 1.

```
> library(multtest)
> data(golub)
> table(golub.cl)
```

```
golub.cl
 0  1
27 11
```

The required input for the sample size and power analysis is a vector of test-statistics and the sample sizes used to compute them. The test-statistics are obtained by a differential gene expression analysis using one of the available packages like `limma, maanova, multtest` (it is also possible to import the vector of test-statistics in `R` if they are calculated using another software package). Here we will use the function `mt.teststat` from the `multtest` package to obtain a vector of test-statistics from the leukemia data.

```
> tst <- mt.teststat(golub, golub.cl)
```

The first step in doing the sample size and power analysis is creating a object of class `PilotData` which will contain all the necessary information needed for the following power and sample size analysis; a vector of test-statistics and the sample sizes used to compute them. A user-friendly interface for creating an object of `PilotData` is available as `pilotData()`.

```
> library(SSPA)
> pd <- pilotData(name = "ALL/AML", testStatistics = tst, sampleSizeA = 11,
     sampleSizeB = 27)
```

Several ways of viewing the content of the `PilotData` object are possible either graphically or using a `show`-method by just typing the name of the created object of `PilotData`:

```
> par(mar = c(4.1, 4.1, 2.1, 2.1))
> layout(matrix(c(1, 2), nrow = 2))
> hist(pd, cex.main = 1)
> plot(pd, cex.main = 1)
```

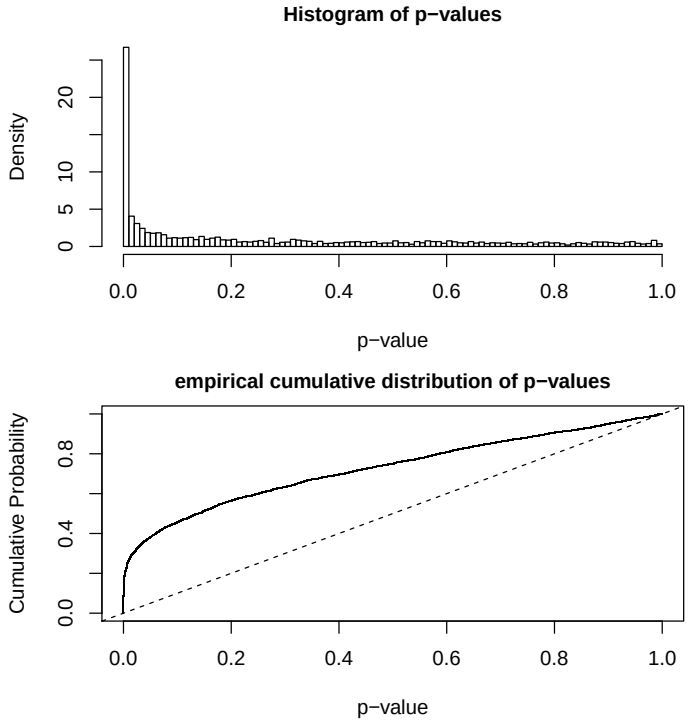

Figure 1: Histogram and ECDF plot of p-values.

```
> pd

An object of class "PilotData"
Experiment name:            ALL/AML
Number of test-statistics:  3051
Effective sample size:      7.82
Null distribution:      normal
```

A histogram of p-values is obtained by just calling `hist(pd)` and an empirical cumulative distribution of p-values by `plot(pd)`. The accumulation of p-values near zero indicates that some genes are differentially expressed, as the deviation from the uniform distribution in the lower panel.

Now we can create an object of class `SampleSize` which will perform the estimation of the proportion of non-differentially expressed genes and the density of effect sizes. Several options are available see `?sampleSize`. The default

method for estimation of the proportion of non-differentially expressed genes is the method proposed by [8] as implemented by `convest()` function from the package `limma`. Additionally the method by [9] and [2] are available, also a user-defined proportion is allowed. Again a generic `show`-method is implemented.

```
> ss <- sampleSize(pd)
> ss
```

```
An object of class "SampleSize"
Distribution of effect sizes is estimated from -6 to 6 using 1024 points.
Method for estimation of the proportion of non-differentially expressed: "Langaas
Fraction of non-differentially expressed genes: 0.4702 (adjusted=0.4719).
Kernel used in the deconvolution is "fan" with bandwidth 0.353.
```

The density of effect size can be shown by a call to `plotEffectSize()`. When there are both up- and down-regulated genes a bimodal density is observed.

```
> plotEffectSize(ss, type = "l")
```

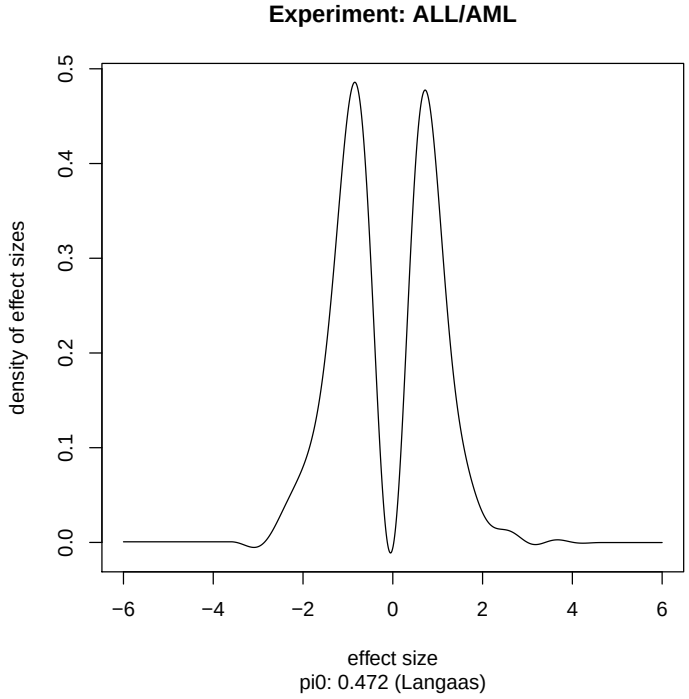

Figure 2: Density of effect-sizes.

Estimating the average power for other sample sizes then that of the pilot-data can be performed with the `Power()`-function. The user can also give the

desired false discovery rate level or possible multiple false discovery rate levels as a vector.

```
> layout(matrix(c(1:2), nrow = 2))
> par(mar = c(4.1, 4.1, 2.1, 2.1))
> pwr <- Power(ss, plot = FALSE, samplesizes = c(5, 10, 15, 20),
    fdr = 0.01)
> plot(c(5, 10, 15, 20), pwr, ylim = c(0, 1), type = "b", ylab = "Power",
    xlab = "Sample size per group")
> legend("bottomright", colnames(pwr), col = c(1:ncol(pwr)), pch = 1,
    lty = 1)
> pwr <- Power(ss, plot = FALSE, samplesizes = c(5, 10, 15, 20),
    fdr = c(0.01, 0.05))
> matplot(c(5, 10, 15, 20), pwr, ylim = c(0, 1), type = "b", pch = 1,
    ylab = "Power", xlab = "Sample size per group")
> legend("bottomright", colnames(pwr), col = c(1:ncol(pwr)), pch = 1,
    lty = 1)
```

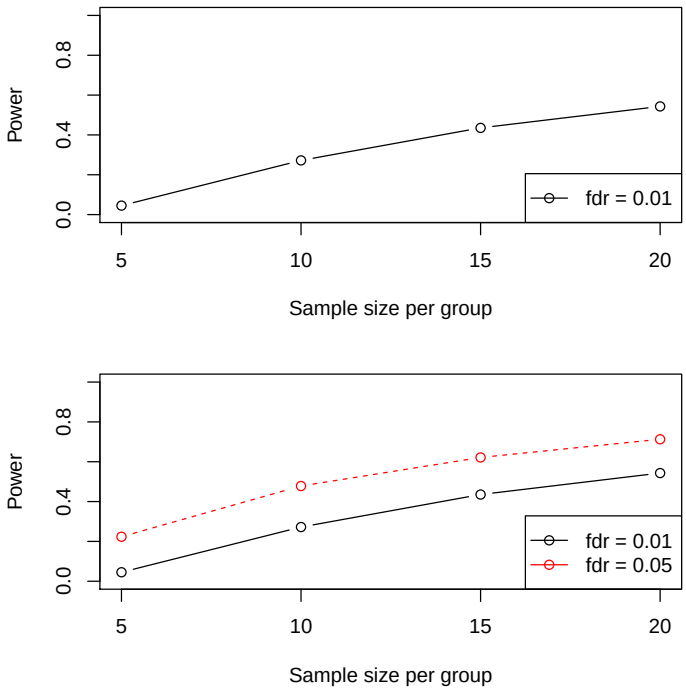

Figure 3: Estimated average power for different sample sizes and FDR-levels.

3 *SSPA* adapted for the *moderated* t-statistics of *limma*

3.1 Approximated t-distribution

Sofar the estimations obtained are based on assumption that the test-statistics follow a normal distribution. Here we descibe how we implemented an approximated t-distribution. Assume the t-statistics for each gene is calculated by:

$$T = \frac{\bar{X} - \bar{Y}}{\hat{\sigma}/\sqrt{N}} = \frac{\bar{X} - \bar{Y} - \Delta}{\hat{\sigma}/\sqrt{N}} + \frac{\Delta}{\hat{\sigma}/\sqrt{N}}, \tag{1}$$

where $N = (1/n_X + 1/n_Y)^{-1}$, Δ is the difference between the means of the two groups and $\hat{\sigma}^2$ the pooled sample variance,

$$\hat{\sigma}^2 = \frac{n_X - 1}{n_X + n_Y - 2} S_X^2 + \frac{n_Y - 1}{n_X + n_Y - 2} S_Y^2. \tag{2}$$

If $\Delta = 0$, then T has exactly a Student t-distribution with $\nu = n_X + n_Y - 2$ degrees of freedom. If $\Delta \neq 0$, then T is the sum of Student random variable with $n_X + n_Y - 2$ degrees of freedom plus

$$\frac{\Delta}{\hat{\sigma}/\sqrt{N}} = \frac{\Delta}{\sigma}\sqrt{N}\frac{\sigma}{\hat{\sigma}} \tag{3}$$

In other words, T is a Student random variable plus the square root of the effective sample size N times an effect size given by

$$\theta = \frac{\Delta}{\sigma}\frac{\sigma}{\hat{\sigma}} \tag{4}$$

This effect size is random, but since $\sigma/\hat{\sigma}$ converges a.s. to 1 it seems allright to view θ as a slightly perturbed effect size. The random variable T becomes $T + \sqrt{N}\theta$, when $\Delta \neq 0$. We can still use the model of Ferreira and Zwinderman (2006) but where the standard normal is replaced by the Student t-distribution.

3.2 moderated t-statistics

The moderated t-statistics calculated by *limma* follows a t-distribution with moderated degrees of freedom $\tilde{\nu}$. So we only have to plug-in the moderated degrees of freedom and we can use the moderated t-statistics in our power and sample size calculations. Because we use moderated degrees of freedom we also have a moderated sample size for each group.

The moderated degrees of freedom $\tilde{\nu}$ constists of two parts: ν_0 is the *prior* degrees of freedom estimated by *limma* using a empirical Bayesian approach and the residual degrees of freedom; for a two-group comparison this is given by $\nu = n_X + n_Y - 2$. Now we can define a moderated sample size for each group:

$$\tilde{\nu} = n_X + \nu_0/2 + n_Y + \nu_0/2 - 2 = \tilde{n_X} + \tilde{n_Y} - 2, \tag{5}$$

where the $\nu_0/2$ is added to the original sample sizes of each group.

Like before, first we created an object of class *pilotData* but now with `testStatistics` the moderated t-statistics of the contrast of interest, `sampleSizeA` the moderated sample size of group A plus half the *prior* degrees of freedom (from a *limma* `eBayes`-fit object using `fit$df.prior`). Internally the degrees of freedom are defined as `fit$df.prior + fit$df.residual`. By default the standard normal distribution is assumed so we need specifically the argument `nullDist = "student"`.

3.3 Example using moderated t-statistics from *limma*

For illustration we will use the Swirl example from the *limma*-userguide but the Swirl data if obtained from the *marray*-package and using *convert* converted to a *limma* `RGList`. In the following bit of code the data is loaded, normalized and an empirical Bayesian linear model is fitted.

```
> library(marray)
> library(convert)
> library(limma)
> data(swirl)
> swirl <- as(swirl, "RGList")
> MA <- normalizeWithinArrays(swirl)
> design <- c(-1, 1, -1, 1)
> fit <- lmFit(MA, design)
> ordinary.t <- fit$coef/fit$stdev.unscaled/fit$sigma
> fitModerated <- eBayes(fit)
```

As described by the *limma* userguide; we can inspect the difference between moderated and orginal t test-statistics using qq-plots:

Now we load the *SSPA*-package and create two *pilotData*-objects, one for the ordinary t test-statistics and one for the moderated t test-statistics with the appropriate degrees of freedom and sample sizes.

```
> library(SSPA)
> nu <- fit$df.residual[1]
> nu0 <- fitModerated$df.prior
> pd <- pilotData(name = "Swirl", testStatistics = ordinary.t[,
     1], sampleSizeA = 2, sampleSizeB = 2, dof = nu, nullDist = "student")
> pd

An object of class "PilotData"
Experiment name:            Swirl
Number of test-statistics:  8448
Effective sample size:      1
Null distribution:      student
Degree of Freedom:      3
```

```
> par(mfcol = c(1, 2))
> qqt(fitModerated$t, df = fitModerated$df.prior + fitModerated$df.residual,
     pch = 16, cex = 0.5, main = "Moderated t")
> abline(0, 1)
> qqt(ordinary.t, df = fit$df.residual, pch = 16, cex = 0.5, main = "Ordinary t")
> abline(0, 1)
```

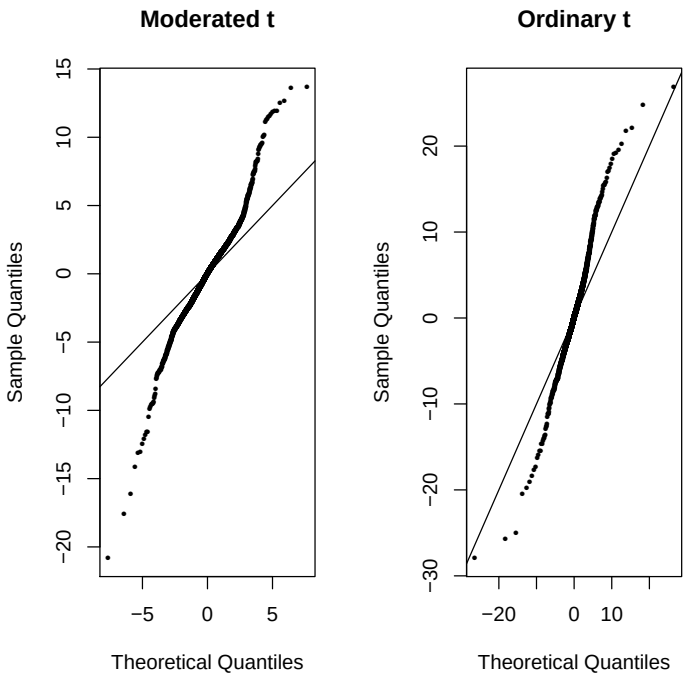

Figure 4: qq-plots: showing the difference between moderated and orginal t test-statistics.

```
> pdMod <- pilotData(name = "Swirl", testStatistics = fitModerated$t[,
     1], sampleSizeA = 2 + nu0/2, sampleSizeB = 2 + nu0/2, dof = nu +
     nu0, nullDist = "student")
> pdMod

An object of class "PilotData"
Experiment name:            Swirl
Number of test-statistics:  8448
Effective sample size:      2.01
Null distribution:      student
Degree of Freedom:      7.024394
```

The data comes from a dye-swap experiment with four microarrays, thus for each RNA source we have two biological and two technical replicates. Because this is an dye-swap experiment we can estimate the difference between the biological sources, leaving three degrees of freedom for estimation of the error.

The `pdMod`-object has a moderated sample sizes and moderated degrees of freedom, the moderated Student t-distribution will be used for the analysis. The effective sample size was defined by: $N = (1/n_X + 1/n_Y)^{-1}$, thus with both four samples gives two. The moderated sample size is 2.01.

The next figure show the difference between the ordinary t test-statistics and moderated t test-statistics (left-panel). The right-panel show the difference in p-values where one is calculated using the ordinary degrees of freedom and the other using the moderated degrees of freedom.

```
> ss <- sampleSize(pd)
> ss

An object of class "SampleSize"
Distribution of effect sizes is estimated from -6 to 6 using 1024 points.
Method for estimation of the proportion of non-differentially expressed: "Langaas
Fraction of non-differentially expressed genes: 0.5957 (adjusted=0.6151).
Kernel used in the deconvolution is "fan" with bandwidth 0.3326.

> ssMod <- sampleSize(pdMod)
> ssMod

An object of class "SampleSize"
Distribution of effect sizes is estimated from -6 to 6 using 1024 points.
Method for estimation of the proportion of non-differentially expressed: "Langaas
Fraction of non-differentially expressed genes: 0.6117 (adjusted=0.6256).
Kernel used in the deconvolution is "fan" with bandwidth 0.3326.
```

Once the proportion of non-differentially expressed genes is estimated the density of effect sizes can be estimated.

The dip below zero for the density of effect sizes based on the moderated test-statistics is unwanted. We implemented the suggestion by Efron *et al.* [10]

```
> par(mfcol = c(1, 2))
> plot(abs(pd@testStatistics), abs(pdMod@testStatistics), xlab = "|ordinary t test statistic
     ylab = "|moderated t test statistics|", pch = 16, cex = 0.5)
> abline(a = 0, b = 1, col = "red", lwd = 2, lty = 2)
> plot(-log10(pd@pValues), -log10(pdMod@pValues), xaxt = "n", yaxt = "n",
     xlab = expression(paste(-log[10], "(ordinary p-values)")),
     ylab = expression(paste(-log[10], "(moderated p-values)")),
     pch = 16, cex = 0.5)
> at <- axTicks(2)
> axis(2, at = at, tcl = -1, labels = parse(text = c("10^0", paste("10^-",
     at[-1], sep = ""))))
> at <- axTicks(1)
> axis(1, at = at, tcl = -1, labels = parse(text = c("10^0", paste("10^-",
     at[-1], sep = ""))))
> abline(a = 0, b = 1, col = "red", lwd = 2, lty = 2)
```

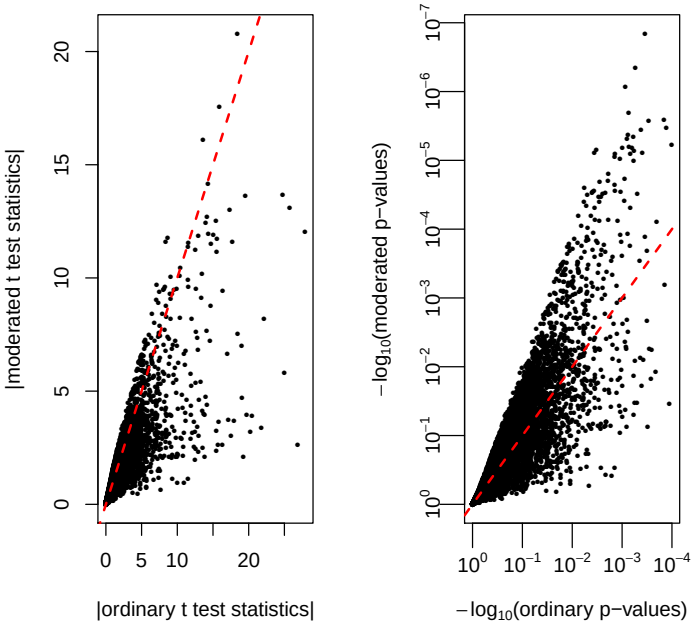

Figure 5: scatterplots: showing the difference between moderated and orginal t test-statistics.

```
> plotEffectSize(ssMod, type = "l", lwd = 2, sub = NULL)
> lines(ss@theta, ss@lambda, col = "red", lwd = 2)
> pi0Mod <- paste("pi0: ", signif(ssMod@pi0[[2]], 3))
> pi0 <- paste("pi0: ", signif(ss@pi0[[2]], 3))
> labels <- c(paste("moderated ", pi0Mod), paste("ordinary ", pi0))
> legend("topleft", labels, col = c("black", "red"), pch = 1)
```

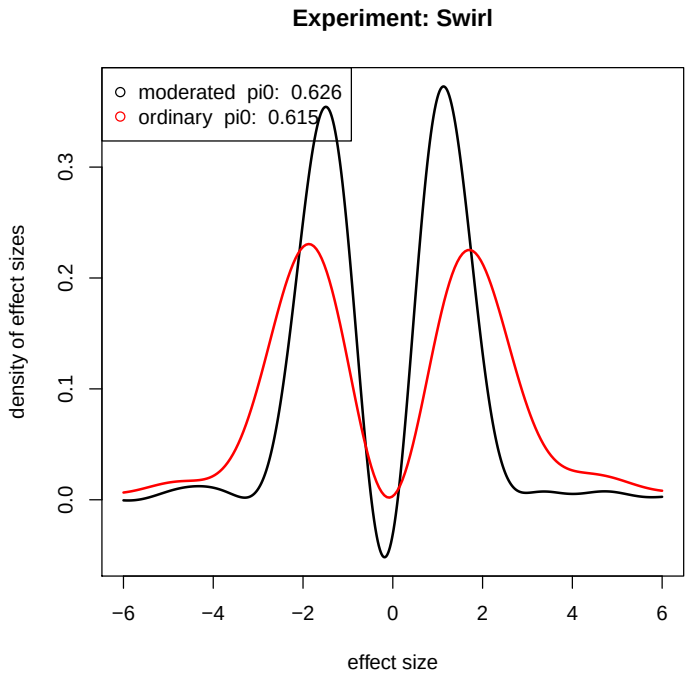

Figure 6: Density of effect size for moderated and orginal t test-statistics.

for better upper bound estimates of π_0. The default value 0.5 this value could be tuned to get a valid density.

Again estimation of the average power for other sample sizes then that of the pilot-data can be performed:

```
> samplesizes <- c(2, 4, 6, 8, 10)
> pwr <- Power(ss, plot = FALSE, samplesizes = samplesizes, fdr = c(0.05,
    0.1))
> pwrMod <- Power(ssMod, plot = FALSE, samplesizes = samplesizes +
    nu0/2, fdr = c(0.05, 0.1))
> matplot(samplesizes, pwr, ylim = c(0, 1), type = "b", col = 1,
    pch = 1, ylab = "Power", xlab = "Sample size per group",
    main = "Power Curves")
> matlines(samplesizes, pwrMod, col = 2, type = "b", pch = 1)
> legend("topleft", colnames(pwr), col = 1, lty = c(1, 2))
> legend("bottomright", c("ordinary", "moderated"), col = 1:2,
    lty = 1)
```

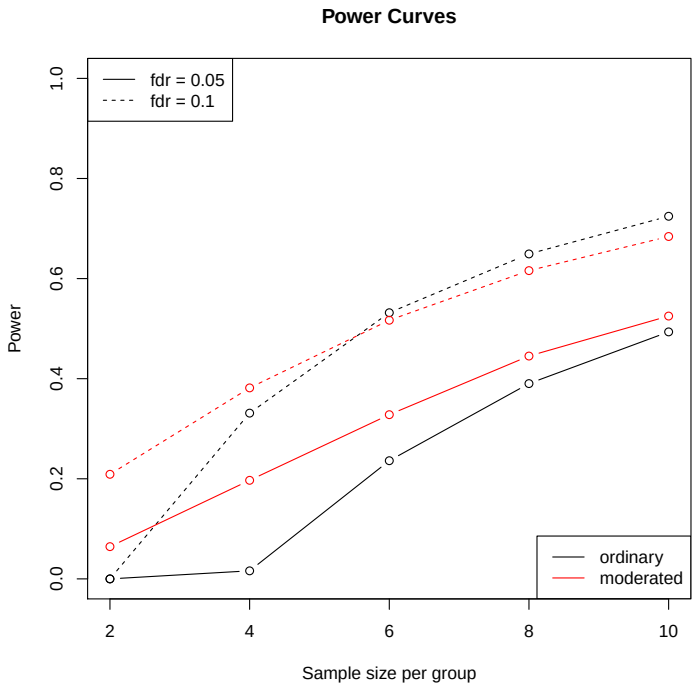

Figure 7: Estimated average power for other sample sizes then that of the pilot-data for both moderated and orginal t test-statistics and for different false discovery rate thresholds.

For small sample sizes the moderated test-statistics has higher power for larger sample sizes both statistics converge to eachother, as expected.

4 Additional functionality

4.1 Ferreira's π_0 estimate

Ferreira *et al.* propose a semi-parametric method to estimate the proportion of non-differentially expressed genes[1, 2]. Use `sampleSize(pd, method="Ferreira", pi0=seq(0.05, 0.5, 0.05), doplot=TRUE)`.

4.2 Rupperts estimation method

For using the method proposed by Ruppert [6] two additional packages need to be installed for using this method namely `quadprog` and `splines`. Using `sampleSize(pd, method="Ruppert", nKnots = 11, bDegree = 3)` both π_0 and the density of effect sizes is estimated by Ruppert's method.

5 Details

This document was written using:

- R version 2.13.0 (2011-04-13), `i386-pc-mingw32`
- Locale: `LC_COLLATE=C`, `LC_CTYPE=English_United States.1252`, `LC_MONETARY=English_United States.1252`, `LC_NUMERIC=C`, `LC_TIME=English_United States.1252`
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: Biobase 2.12.0, SSPA 1.8.0, convert 1.28.0, limma 3.8.0, marray 1.30.0, multtest 2.8.0, qvalue 1.26.0
- Loaded via a namespace (and not attached): MASS 7.3-12, splines 2.13.0, survival 2.36-5, tcltk 2.13.0, tools 2.13.0

References

[1] J.A. Ferreira and A. Zwinderman. Approximate Power and Sample Size Calculations with the Benjamini-Hochberg Method. *International Journal of Biostatistics*, 2(1), 2006.

[2] J.A. Ferreira and A. Zwinderman. Approximate Sample Size Calculations with Microarray Data: An Illustration. *Statistical Applications in Genetics and Molecular Biology*, 5(1), 2006.

[3] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B*, 57:289–300, 1995.

[4] van Iterson M., J.A. Ferreira, and R.X. de Menezes. SSPA: Power and Sample Size Analysis package. 2010. in preparation.

[5] M. van Iterson, P.A.C. 't Hoen, P. Pedotti, G.J.E.J. Hooiveld, J.T. den Dunnen, G.J.B. van Ommen, J.M. Boer, and R.X. Menezes. Relative power and sample size analysis on gene expression profiling data. *BMC Genomics*, 10:439–, 2009.

[6] D. Ruppert, D. Nettleton, and J.T.G. Hwang. Exploring the information in p-values for the analysis and planning of multiple-test experiments. *Biometrics*, 63(2):483–95, 2007.

[7] T R Golub, D K Slonim, P Tamayo, C Huard, M Gaasenbeek, J P Mesirov, H Coller, M L Loh, J R Downing, M A Caligiuri, C D Bloomfield, and E S Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–7, 1999.

[8] M. Langaas, B.H. Lindqvist, and E. Ferkingstad. Estimating the proportion of true null hypotheses, with application to DNA microarray data. *Journal of the Royal Statistical Society: Series B*, 67(4):555–572, 2005.

[9] J.D. Storey. A direct approach to false discovery rates. *Journal of the Royal Statistical Society: Series B*, 64:479–498, 2002.

[10] B. Efron, R. Tibshirani, J.D. Storey, and V. Tusher. Empirical bayes analysis of a microarray experiment. *Journal of the American Statistical Association*, 96(456):1151–1160, 2001.