Vignette for Package survGenesInterim (Version 1.0)

Andreas Leha <andreas.leha@med.uni-goettingen.de>

February 15, 2013

1 Overview

The identification of biomarkers that are correlated to therapy response is an important goal of medical research on severe diseases. For example, the gene expression levels of pretherapeutic tissue samples can be measured by means of microarrays and correlated to survival times.

```
> library(survGenesInterim)
```

The `survGenesInterim` package provides a simulation framework that allows to easily simulate such survival studies by providing functions for data generation and visualization, study simulation and result visualization. It provides the possibility to simulate just one trial as well as to do a full (parallelized) simulation of many trials, which is useful e.g. for sample size planning.

2 Data Generation

First, let us generate the data that shall be the basis for the simulation. There are three parts: The patient data, the expression level data and the trial data.

2.1 Patient Data

The N patients are simulated to arrive uniformly distributed within $[0, l'_1$, where l'_1 denotes the anticipated length of the recruitment period. The survival times are drawn from an exponential distribution with the mean survival time set to λ. The function for the generation of the patient data is `generatePatientData`.

```
> N <- 50 # sample size: 50 patients
> l.1.tick <- 60 # [months]
> lambda <- 60 # [months]
> patdat <- generatePatientData(N, l.1.tick, lambda)
```

The return value is a list, that contains as main values the simulated arrival times and survival times.

```
> patdat$arrivalTimes
```

```
 [1] 50.7248864 35.5919687 12.9999996  3.8928564 42.8799779
 [6] 11.4013529 39.9175999 11.2961346  5.4911543 11.5991757
[11] 54.7686456 55.2179362 47.6731665  4.3983028  9.7841603
[16] 21.5175510 23.7436477 36.5374912  2.0791437  2.3228769
[21] 57.0371142 20.5113471 12.5913088 39.1272653  2.7990028
[26] 55.2202273  0.1377889 54.9497253 54.3213212  8.8872640
[31] 18.9682217  6.1522891 23.5518263  6.3574890 40.7831920
[36] 41.1959425 15.4344342 11.6262770 18.1024841 18.1683477
[41]  9.2354279 35.2766896 32.2950979 55.2767895 43.2391559
[46] 56.8118888 19.6370247  1.7206503 18.4865901 50.1193405
```

```
> patdat$survivalTimes
```

```
 [1]  90.7400490  41.3869277 179.0622009  48.1588296
 [5]  20.9857568   2.5644056  25.0832760  49.4098411
 [9]  43.8926074   1.6229520  20.7692243  98.9267775
[13] 180.7168208  28.2159976   5.6774933  14.1858782
[17]   9.5722912  78.4103230  89.0176090   5.7378384
[21] 122.0062320  23.2685989   8.0426583  60.7440147
[25]  74.9437003   6.7701772   4.9140254  73.3634248
[29] 125.7471684   6.3444268 114.9170064  22.3317817
[33]  40.1698352   4.0795502 209.4361208  38.5412267
[37]  30.0254862  92.8769536  95.1197667  49.7363188
[41]  31.3137357   0.9535248  79.1976265   1.2831293
[45]  18.9531247  17.6276327 174.5043021 203.0868119
[49] 115.2605241  23.8189878
```

Furthermore it contains the numbers N_1 and N_2 that indicate how many patient had simulated survival times less then the given mean λ and how many patients had simulated survival times greater to the given mean, respectively.

```
> patdat$N.1
```

```
[1] 31
```

```
> patdat$N.2
```

```
[1] 19
```

Figure˜2 show a visualization of this data produced with the function `plot-patdat`:

```
> plotpatdat(patdat)
```

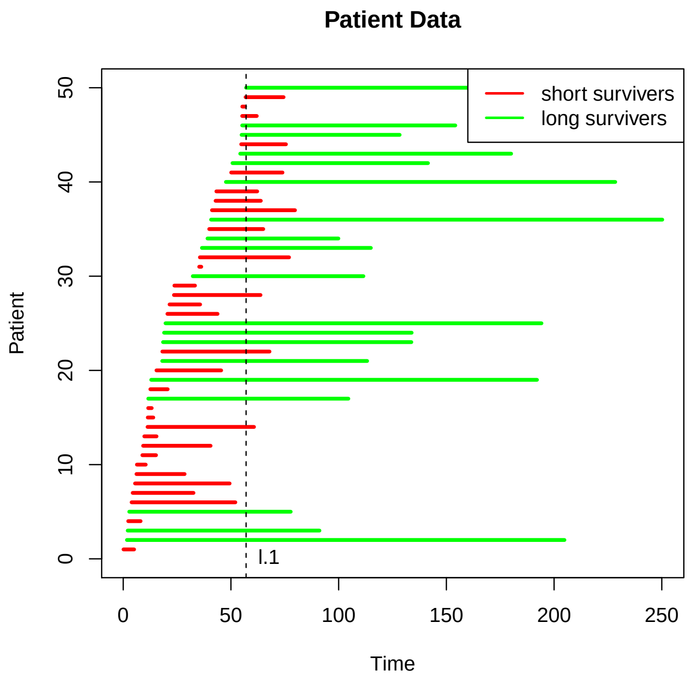

Figure 1: The visualized patient data. The arrival time of a patient is given as the left end the line and his death by the right end. The dashed line marks the arrival of the last patient included in the study and, thus, the end of the recruitment period.

2.2 Expression Level Data

In this package the effect of expression levels to survival times are modelled as a group effect, meaning that the available samples are split in two groups: the short survivors and the long survivors. Therefore, the effect is modelled as the fold change (fc) between the expression levels of these two groups. The group sizes N_1 and N_2 of these two groups can be set freely (e.g. to equal sized groups). The package offers an easy way to split the patients at the mean survival time that was used to generate the data.

```
> N.1 <- patdat$N.1
> N.2 <- patdat$N.2
```

The expression level data are drawn from a multivariate normal distribution. Thus, the paramters needed include next to the fold change fc also the covariance matrices Σ_1 and Σ_2 to use in either group. If no covariance Σ_2 is specified, the package uses Σ_1 for both groups. In the example below we set the two covariance matrices explicitly, though we could also omit the setting of Sigma.2. The data generation is done in the function `generateExpressionData`.

```
> d <- 100 # we simulate 100 genes
> fc <- c(rep(0,d/2), ceiling(rnorm(d/2, 0, 1)-0.5))
> Sigma.1 <- diag(100) # uncorrelated
> gendat <- generateExpressionData(fc=fc, Sigma.1=Sigma.1, Sigma.2=Sigma.1, N.1=N.1, N.2=N.2
```

The return value is again a list. The main items are the expression levels for both groups $\mathbb{X}_1$ and $\mathbb{X}_2$:

```
> head(gendat$X.1[,1:5])

           [,1]       [,2]       [,3]       [,4]       [,5]
[1,]  0.6557028 -2.0061074  0.4710592  0.9866261 -1.0114955
[2,]  0.5515358 -1.0635621  1.2601722  0.3192593 -0.6586597
[3,] -0.7641809 -0.5239073  0.5779373  0.8035640  1.4012880
[4,]  0.2647438 -0.1632233 -1.3850477  0.5624364 -0.3292252
[5,]  0.5124090 -0.3265811  0.4955312 -0.3171171  1.2486731
[6,] -0.4432861  0.1707516 -0.3286416 -1.7885215  1.4539944
```

```
> head(gendat$X.2[,1:5])

           [,1]       [,2]       [,3]        [,4]
[1,] -0.4594198  0.7232580 -0.4579017 -0.31327173
[2,] -0.2074153 -1.1964085 -1.1458616 -0.04106106
[3,] -0.5834974  1.5285991  2.2853292 -0.58740235
[4,] -0.1032242 -0.3994239 -0.2376984 -1.11509886
[5,]  0.1553986  0.6257748 -2.9617283  0.15169939
[6,] -0.7721300 -1.1291102 -0.4955861  0.06390160
           [,5]
```

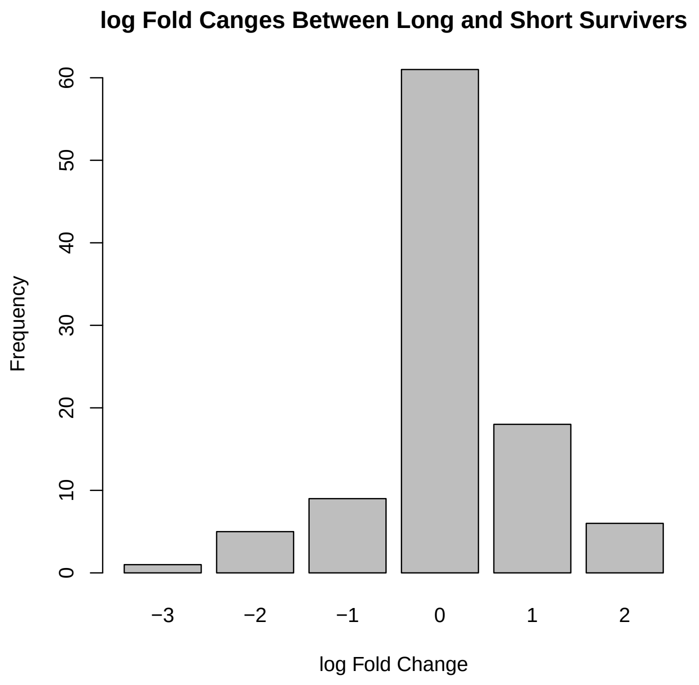

Figure 2: The visualized fold change vector used to genereate the expression level data.

```
[1,] -0.2524903
[2,] -1.5806345
[3,] -0.3974319
[4,] -0.2858272
[5,] -0.1049531
[6,]  0.3755842
```

The fold changes can be easily visualized (see **??**) with the help of the function `plotgendat`:

```
> plotgendat(gendat)
```

2.3 Trial Data

The parameter that control the trial desing are the length of the follow-up preriod l_2 and the number of (interim) analyses to perform during the recruitment period (M_1) and during the follow-up period ($M.2$).

```
> l.2 <- 60 # [months]
> M.1 <- 2
> M.2 <- 2
```

Finally there are the more tecnical parameters to set. As the trial simulation internally uses the Cox-regression and performs one test per gene, there is need for a p-value adjustment. The method can be chosen from `c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none")`. Also the tolerable type I error level α is a paramter.

```
> adjustment <- "BH"
> alpha <- 0.05
```

The trial is aborted as soon as the estimated power rate exceeds a given threshold, which is the last parameter to set:

```
> powerThreshold <- 0.8
```

3 Simulation of One Trial

Once all parameters are set and the data have been generated, the simulation of one trial is a simple function call:

```
> oneTrial <- interimTrial(patdat, gendat, l.2, M.1, M.2, alpha, powerThreshold, adjustment)
```

The result is a list, which has as main part a resultTable, that shows (row by row) at which analysis the trial was stopped, the error rate (FDR), the power rate (APR), the estimated error rate (eFDR), and the estimated power rate (eAPR):

```
> oneTrial$resultTable

                           Analysis 1 Analysis 2
Fraction of Stopped Studies          0  0.0000000
FDR                                  0  0.0000000
APR                                  0  0.2051282
Estimated FDR                        0  0.0350000
Estimated APR                        0  0.2573333
                           Analysis 3 Analysis 4
Fraction of Stopped Studies 1.00000000          0
FDR                         0.05882353         NA
APR                         0.82051282         NA
Estimated FDR               0.03900000         NA
Estimated APR               1.48518182         NA
```

The error rate at each analysis can also be barplotted as shown in Figure˜3 with

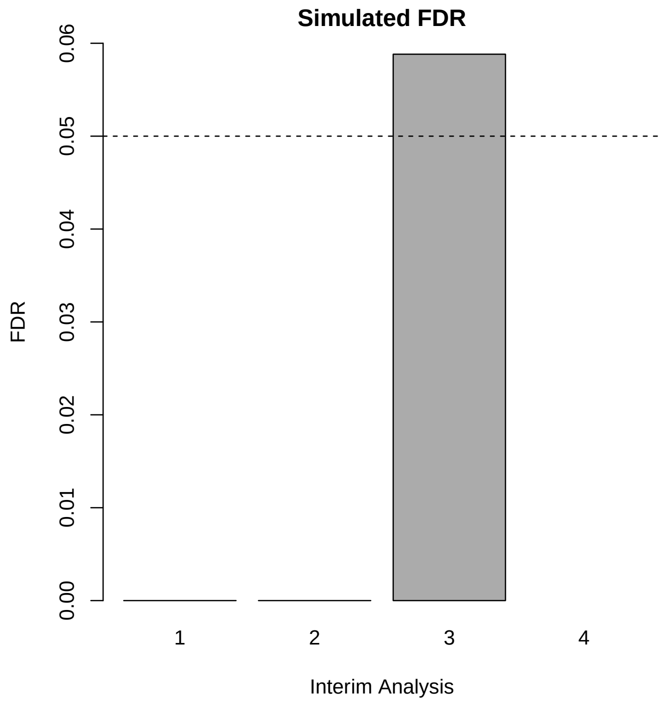

Figure 3: The visual presentation of the achieved error rates at each analysis.

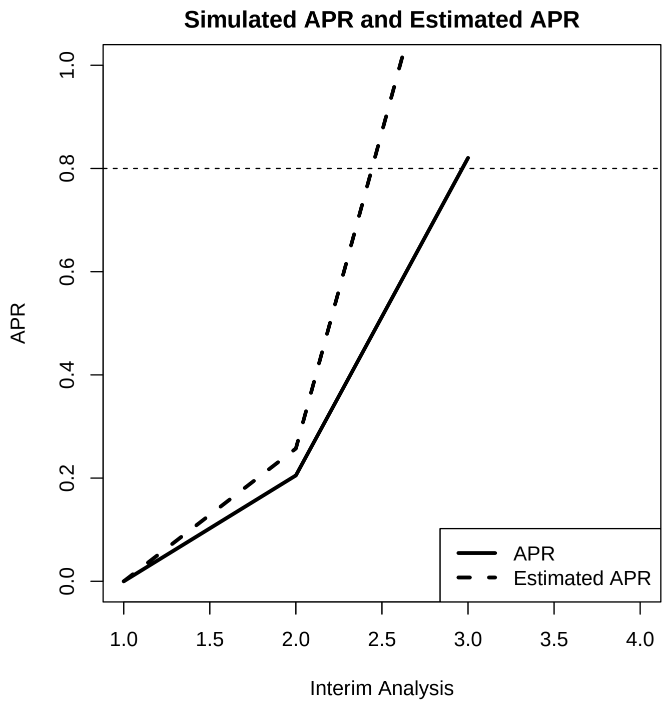

Figure 4: The visual presentation of the achieved power rates and their estimations at each analysis.

```
> plotinterimFDR(oneTrial)
```

The package containes also a function, that plots the power rates. This function also includes the estimated power rates in the plot to make a comparison possible (see Figure˜5):

```
> plotinterimAPR(oneTrial)
```

4 Simulation of more Trials

It is not hard to write a simulation loop that utilizes the above shown functions. The package, however, includes such simluation function as well, which makes use of the cholesky decomposition of the covariance matrices to speed up the generation of expression level data and is even available in a parallel version using the `multicore`.

For the use of the simulation function shipped with this package, two more parameters can be set: The number of trials to simulate and whether to use the parallel version.

```
> numSimRuns <- 2 # for the sake of time neede to build this vignette only 2
> useMulticore <- FALSE # just in case 'multicore' is not available
```

The simulation is invoked by this function call:

```
> moreTrials <- interimTrialSimulation(fc, Sigma.1, NULL,
+                                      N, l.1.tick, l.2, lambda,
+                                      M.1, M.2, alpha, powerThreshold, adjustment,
+                                      numSimRuns, useMulticore)
```

The result is structured in the same way as in the case of only one trial being simulated. The only difference is, that `resultTable` now contains average values.

The same plot functions are available as well. There is one more plot function (`plotinterimStops`) that is interesting in this case of more trials having been conducted. This function shows how many of the simulated trials were stopped at each analysis. **??** show the plot generated by:

```
> plotinterimStops(moreTrials)
```

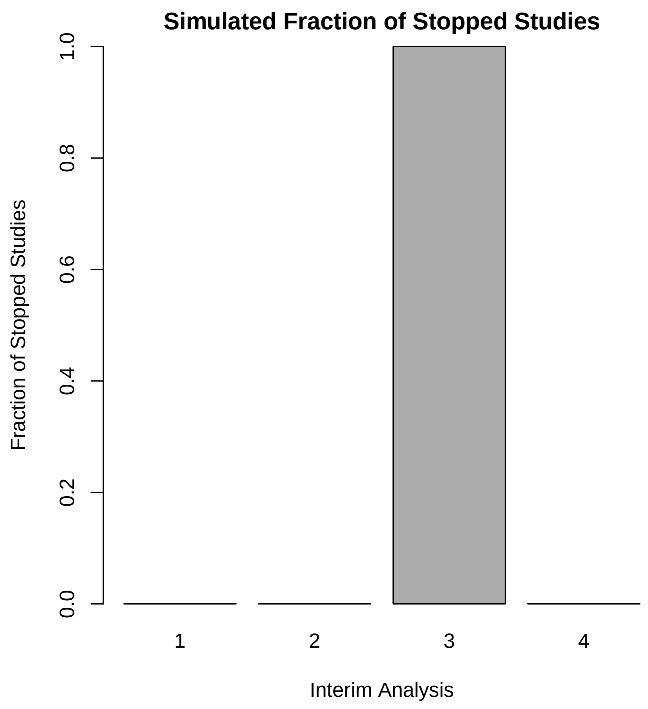

Figure 5: The visual presentation of the fraction of simulated trial stopped at the analyses.