R Package `CellPlot`
User Manual

Sven E. Templer
`sven.templer@gmail.com`

September 15, 2015

Contents

1 Introduction 2

2 Installation & Development 2

3 Example Dataset 2

3.1 Data `golubstat` 2

4 Plots 3

4.1 `cell.plot` 3

4.2 `sym.plot` 3

1 Introduction

CellPlot

A GNU R package for comparted bar plots.

Introduction text...

2 Installation & Development

You can install the package to use it in any `R` session. The stable versions are published on the comprehensive R archive network (CRAN) servers. The web description of this and previous versions can be found on the page cran.r-project.org/web/packages/CellPlot. The installation takes place in an `R` session or script. Run the code:

```
> install.packages("CellPlot")
```

A more administrative way is to use the command line in a shell environment and install the package from a downloaded `.tar.gz` archive. Invoking `R` from the shell provides the syntax `R CMD INSTALL /path/to/package.tar.gz` therefore.

The most recent version can be found on github.com/dieterich-lab/CellPlot. With the `devtools` package, it is easy to install it:

```
> install.packages("devtools")
> library(devtools)
> install_github(repo='dieterich-lab/CellPlot')
```

Once installed, you need to load the package into the current session to access the functions and help pages. With the functions `library` or `require` you perform this step:

```
> library(cellplot)
```

In the following sections the usage of the functions is shown with example data and the output. Each function is accomanied by a help page, which is typically invoked on the `R` shell with `?` in front of the function's name.

3 Example Dataset

3.1 Data `golubstat`

This data set contains microarray gene expression data from leukemia study of Golub et al. (1999). Processed as described in Dudoit et al. (2002). Adapted from package multtest. Differential gene expression was performed using a Student t-test to compare the two groups. GO annotation was done with the annotate and `hu6800.db` packages from Bioconductor. The `topGO` package was utilized to perform a GO enrichment test based on Fisher's exact test. Data of the DEG and GO enrichment was merged with the provided go.enrich function. The code to create the data object is shown in the examples section.

```
> data("golubstat")
```

See the help page of the data set on how it was created from the `golub` dataset using differential gene expression, GO annotation and GO enrichment.

4 Plots

4.1 cell.plot

Plots a horizontal barchart of strictly positive values in x. For each entry, a vector of additional values needs to be provided in a list. The additional values are plotted as cells subdividing the length of the corresponding bar. A typical usage scenario is plotting the enrichment of Gene Ontology terms, with individual cells reflecting the differential regulation of the constituent genes.

```
> x <- subset(golubstat, p<=.05 & significant>4 & !duplicated(genes))
> x <- head(x, 10)
> cell.plot( x = setNames(x$loge, x$term), cells = x$deg.log2fc,
+            main = "Golub et al. (1999) - Gene Ontology Enrichment" )
```

4.2 sym.plot

Plots a split barchart, showing the proportions of two mutually exclusive sets in relation to a set containing them both. E.g., Gene Ontology terms, showing the proportions of differentially down-regulated and up-regulated annotated genes from a perturbation experiment. The color of the central column elements maps to the value provided in x (e.g. GO term enrichment). Associated genes may be provided as a list of vectors of expression values, same as for cell.plot(), or as separate vectors x.up and x.down, providing the numbers of up- and down-regulated genes in the same order as x.