MethylAid: Visual and interactive quality control of large Illumina 450k data sets

Maarten van Iterson, Elmar Tobi, Roderick Slieker, Wouter den Hollander, Rene Luijk,
Eline Slagboom and Bas Heijmans
Department of Molecular Epidemiology,
Leiden University Medical Center, Leiden, The Netherlands

May 8, 2015

Contents

1 Introduction

MethylAid is specially designed for quality control of large sets of DNA methylation data e.g., epigenomewide association studies (EWAS). Extracting intensities from IDAT files can be done in batches and/or in parallel to reduce memory load and/or overcome long run-times. It requires two function calls in going from IDAT files to launch the interactive web application; `summarize` and `visualize`. For more information see van Iterson *et al.* [1].

To show the utility of *MethylAid*, we first show a quick example using a small set of idat files taken from the *minfiData* package. A second example uses presummarized data on 500 samples which can be used directly to launch the web application. A third example shows how level 1 data downloaded from The Cancer Genome Atlas can be used. Similar, in the fourth example we show how data downloaded from GEO[2] can be used. For example, Liu *et al.* [3] studied genome-wide DNA methylation levels to determine whether Rheumatoid arthritis patients has methylation differences comparing to normal controls in peripheral blood leukocytes (PBLs) and deposit raw data on GEO. Furthermore, we show how the summarization can be performed in batches or in parallel using the *BiocParallel* package which provides a uniform idiom for parallel computing resources.

MethylAid identifies poorly performing samples that are to be removed prior to processing and further analysis of the data. Other *R*/*Bioconductor*-packages such as, *wateRmelon, minfi, methylumi, lumi, COHCAP, ChAMP*, are available for processing, i.e. background, dye-bias and probe correction or for further analysis such as the detection of differentially methylated regions. Many of these packages include quality control as well. The package shinyMethyl allows quality control assessment and interactive exploration of 450k array data in a similar way as *MethylAid*.

We have a demo running at http://shiny.bioexp.nl/MethylAid using the example-data of the package.

2 Quick start

This example shows how to summarize a small set of idat files e.g. the summarized data fits into the available RAM. The *minfiData*-package provides such a set. The package contains 450k DNA methylation data on 6 samples across 2 groups. Since, *MethylAid* uses internally the function `read.450k.exp` from the *minfi*-package[4], target information should be provided in a similar way as is done when using the *minfi*-package.

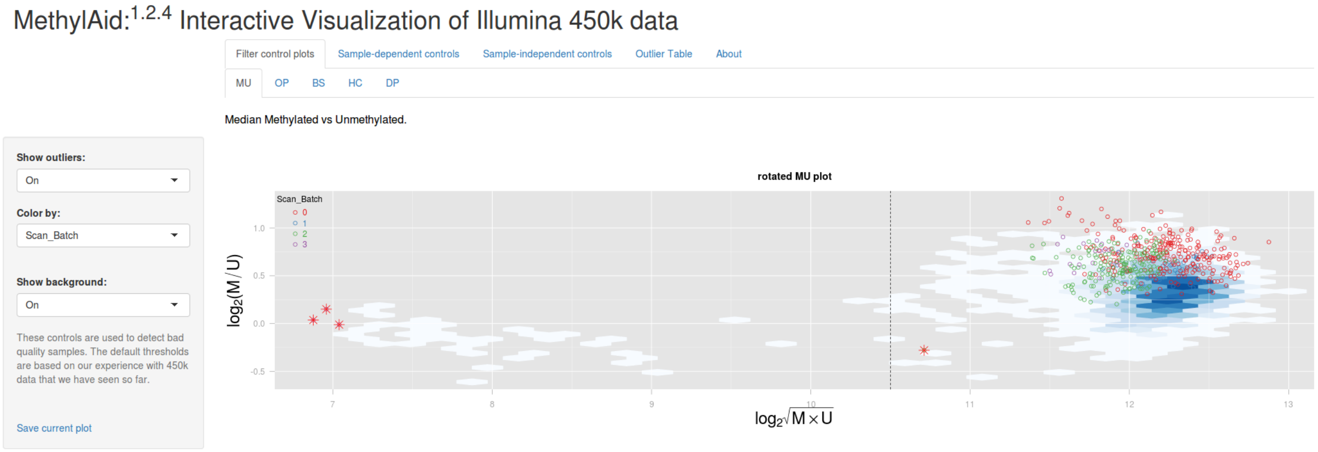
```
library(MethylAid)
library(minfiData)
baseDir <- system.file("extdata", package = "minfiData")
targets <- read.450k.sheet(baseDir)

## [read.450k.sheet] Found the following CSV files:
## [1] "/home/biocbuild/bbs-3.1-bioc/R/library/minfiData/extdata/SampleSheet.csv"
```

The function `summarize` performs the summarization of the the control probes present on the array. The output produced by `summarize` is an object of class `summarizedData` which should be passed on to the function `visualize` which in turn will launch the interactive web application.

```
data <- summarize(targets)
visualize(data)
```

comment: `visualize` can only be called interactively from within a R session.



MethylAid:$^{1.2.4}$ Interactive Visualization of Illumina 450k data

Median Methylated vs Unmethylated.

Show outliers: On

Color by: Scan_Batch

Show background: On

These controls are used to detect bad quality samples. The default thresholds are based on our experience with 450k data that we have seen so far.

Figure 1: Screen shot *MethylAid* interactive web application: The web interface contains three parts; a panel with widgets to control the appearance of the quality control plots, tab-panels for chosing the quality control plot of interest and the interactive plotting area. When the interactive web application is launched the rotated M versus U plot is shown. Here using example data from the package on 500 450k human methylation samples with selected "scan_batches" using the input selector widget "Color by". The different "scan_batches" are indicated with different colors. The dashed-vertical line indicated the filter threshold, samples below the threshold should be removed. In this example also a background dataset is loaded represented by the blue hexagonal bins, see *MethylAidData*.

3 Example using presummarized 450k data

For those who directly want to explore the web application we have presummarized 450k data on 500 samples.

```
library(MethylAid)
data(exampleData)
visualize(exampleData)
```

comment: `visualize` can only be called interactively from within a R session.

When your webbrowser opens you should see something like Figure 1. In the 'About' panel a description is given of what you see and can do.

4 Example using 450k data downloaded from The Cancer Genome Atlas

The Cancer Genome Atlas (`http://cancergenome.nih.gov/`) provides a valuable source of genomic data of various types of different cancers. Here all Breast invasive carcinoma (BRCA) 450k level 1 DNA methylation data were download from the TCGA Data Portal. A targets file can be constructed from the sdrf file but some preprocessing is necessary. The following code chunk show how to make a minimal targets file from a sdrf file. There is also a check to see if all files are present otherwise these will be removed from the targets file. *fixme: In the future this might be added as an internal check.*

```
sdrfFile <- list.files(pattern="sdrf", full.name=TRUE)
targets <- read.table(sdrfFile, header=TRUE, sep="\t")
path <- "path_to_idat_files"
targets <- targets[file.exists(file.path(path, targets$Array.Data.File)),]
targets <- targets[grepl("Red", targets$Array.Data.File),]
targets$Basename <- gsub("_Red.*$", "", file.path(path, targets$Array.Data.File))
rownames(targets) <- basename(targets$Basename)
head(targets)
```

Now the 2×137 idat files can be summarized. This could be too much to read in at once therefore the option `batchSize = 15` is used for summarization of the data in batches of size 15. Furthermore, the summarized data is stored as an `RData`-object for later use, using the option `file = "tcgaBRCA"`. After summarization the data can be loaded into *R* and passed to `visualize` for interactive exploration of the data.

```
summarize(targets, batchSize = 15, file = "tcgaBRCA")
load("tcgaBRCA.RData")
visualize(tcgaBRCA)
```

comment: `visualize` can only be called interactive from within a R session.

5 Example using 450k data downloaded from GEO

Several 450k data sets are available from GEO some include raw idat files e.g. the study of Liu *et al.* [3] with GEO series number: GSE42861. The idat files of this study are available under GSE42861_RAW.tar. Target information was extracted using the *GEOquery* package. *comment: To run the following code chunk a considerable amount of RAM should be available.*

```
library(GEOquery)
gse <- getGEO("GSE42861")
targets <- pData(phenoData(gse[[1]]))
path <- "path_to_idat_files"
targets$Basename <- file.path(path,
gsub("_Grn.*$", "", basename(targets$supplementary_file)))
rownames(targets) <- basename(targets$Basename)
```

Again we store the summarized data for later use and summarize the data in batches of size 15.

```
summarize(targets, batchSize = 15, file="RA")
load("RA.RData")
visualize(RA)
```

comment: `visualize` can only be called interactive from within a R session.

6 Parallel summarization

MethylAid was specially designed for quality control of large set of DNA methylation data e.g. EWAS studies. Summarization can be performed in batches to overcome memory problems e.g. when too many idat files are read at once. Just provide the option `batchSize` to the `summarize` function as we did in the previous example. Too overcome long run-times summarization can be performed in parallel as well using various computing resource facilitated by the *BiocParallel* package.

The `summarize` accepts a `bpparam` argument e.g. `MulticoreParam` (*comment: unfortunately this is not available on Windows*). For example, perform summarization on a multiple core machine with 8 cores requested is easy as this:

```
library(BiocParallel)
tcga <- summarize(targets, batchSize = 15, BPPARAM = MulticoreParam(workers = 8))
```

The *BiocParallel* also allows parallelization on a cluster of computers using different job schedulers. For example, using the appropriate configuration file a `BatchJobsParam`-object can be constructed and passed on to the `summarize`-function.

```
library(BiocParallel)
conffile <- system.file("scripts/config.R", package="MethylAid")
BPPARAM <- BatchJobsParam(workers = 10,
progressbar = FALSE,
conffile = conffile)
summarize(targets, batchSize = 50, BPPARAM = BPPARAM)
```

The script folder of the package contains example files for parallel summarization on a cluster computer using the Sun Grid Engine job scheduler. For more information on how to setup a configuration file for other job schedulers see the description of *BatchJobs* `https://github.com/tudo-r/BatchJobs` (*BiocParallel* relies on the cran *R* package *BatchJobs*[5]). Probaly, you should at least set your email address in `scripts/summarize.sh` and use the proper *R* executable e.g change this in `scripts/summarize.sh` and `scripts/sge.tmpl`.

7 Customize MethylAid

7.1 User-defined thresholds

MethylAid uses pre-defined thresholds in the filter control plots to determine outlying samples. These thresholds are based on experience with several 450k data sets that were analysed in the Department of Molecular Epidemiology (Leiden University Medical Center). The values of these thresholds are further supported by a large reference set available from the *MethylAidData*-package. However, if one wished to use customised thresholds, e.g. for hydroxymethylation data, these can be given as arguments the the `visualize`-function.

```
visualize(exampleData,
          thresholds = list(MU = 10.5, OP = 11.75,
              BS = 12.75, HC = 13.25, DP = 0.95))
```

7.2 Generating your own reference dataset

The *MethylAidData*-package provides a reference dataset on 2800 samples. A reference data set can be used to compare with another data set. For example, call `visualize` with the argument `background` as shown below.

```
library(MethylAid)
data(exampleData) ##500 samples
library(MethylAidData)
data(exampleDataLarge) ##2800 samples
outliers <- visualize(exampleData, background=exampleDataLarge)
head(outliers)
```

Every dataset that has been summarized by *MethylAid* can be used as a reference data set. Furthermore, different summarized data sets can be combined to one bigger reference data set using the `combine`.

```
library(MethylAid)
data(exampleData)
exampleData

## summarizedData object with 500 samples.
## Containing: median Methylated and Unmethylation values,
##             detection P-values
##             and all quality control probe intensities.

combine(exampleData, exampleData)

## combining summarizedData objects

## List of 2
##  $ <S4 object of class "summarizedData">:Formal class 'summarizedData' [package "MethylA
##   .. ..@ targets  :'data.frame': 500 obs. of  7 variables:
##   .. .. ..$ Sample_Well     : Factor w/ 96 levels "A01","A02","A04",..: 1 18 38 55 5 22
##   .. .. ..$ Sample_Plate    : Factor w/ 6 levels " 0"," 1"," 8",..: 3 3 3 3 3 3 3 3 3 3
##   .. .. ..$ Sentrix_Barcode : Factor w/ 82 levels "7310440014","7310440029",..: 52 52 43
##   .. .. ..$ Sentrix_Position: Factor w/ 12 levels "R01C01","R01C02",..: 1 10 6 2 9 5 1
##   .. .. ..$ Bisulfite_Batch : Factor w/ 11 levels " 1"," 2"," 3",..: 8 8 8 8 8 8 8 8 8
##   .. .. ..$ Scan_Batch      : Factor w/ 4 levels "0","1","2","3": 3 3 3 3 3 3 3 3 3 4 .
##   .. .. ..$ None            : Factor w/ 1 level "1": 1 1 1 1 1 1 1 1 1 1 ...
##   .. ..@ controls :'data.frame': 848 obs. of  4 variables:
##   .. .. ..$ Address      : int [1:848] 10627500 10673427 10714330 10721502 10731326 1073
##   .. .. ..$ Type         : chr [1:848] "NEGATIVE" "SPECIFICITY I" "NORM_T" "NEGATIVE" .
##   .. .. ..$ Color_Channel: chr [1:848] "Purple" "Lime" "Purple" "BlueViolet" ...
##   .. .. ..$ Name         : chr [1:848] "Negative 265" "GT Mismatch 3 (PM)" "Norm_T46" "N
##   .. ..@ Rcontrols: int [1:848, 1:500] 124 298 3808 170 206 113 131 153 177 489 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:848] "10627500" "10673427" "10714330" "10721502" ...
##   .. .. .. ..$ : chr [1:500] "9374341033_R01C01" "9374341033_R05C02" "9340996084_R03C02"
##   .. ..@ Gcontrols: int [1:848, 1:500] 126 8076 195 84 145 77 106 80 47 7079 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:848] "10627500" "10673427" "10714330" "10721502" ...
##   .. .. .. ..$ : chr [1:500] "9374341033_R01C01" "9374341033_R05C02" "9340996084_R03C02"
##   .. ..@ DPfreq   : Named num [1:500] 0.999 1 0.999 1 0.999 ...
##   .. .. ..- attr(*, "names")= chr [1:500] "9374341033_R01C01" "9374341033_R05C02" "93409
##   .. ..@ MU       : num [1:2, 1:500] 4622 2953 4978 3242 5144 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:2] "Methylated" "Unmethylated"
##   .. .. .. ..$ : chr [1:500] "9374341033_R01C01" "9374341033_R05C02" "9340996084_R03C02"
##   .. ..@ plotdata :'data.frame': 331000 obs. of  7 variables:
##   .. .. ..$ Address      : int [1:331000] 10627500 10627500 10627500 10627500 10627500 1
##   .. .. ..$ Type         : chr [1:331000] "NEGATIVE" "NEGATIVE" "NEGATIVE" "NEGATIVE" .
##   .. .. ..$ Color_Channel: chr [1:331000] "Purple" "Purple" "Purple" "Purple" ...
##   .. .. ..$ Name         : chr [1:331000] "Negative 265" "Negative 265" "Negative 265" "
##   .. .. ..$ Samples      : Factor w/ 962 levels "7310440014_R01C01",..: 754 169 258 576
##   .. .. ..$ IntRed       : num [1:331000] 7.29 8.69 7.27 8.5 7.85 ...
##   .. .. ..$ IntGrn       : num [1:331000] 6.66 7.79 6.64 7.83 8.17 ...
##  $ NA                                   :Formal class 'summarizedData' [package "MethylA
##   .. ..@ targets  :'data.frame': 500 obs. of  7 variables:
##   .. .. ..$ Sample_Well     : Factor w/ 96 levels "A01","A02","A04",..: 1 18 38 55 5 22
##   .. .. ..$ Sample_Plate    : Factor w/ 6 levels " 0"," 1"," 8",..: 3 3 3 3 3 3 3 3 3 3
##   .. .. ..$ Sentrix_Barcode : Factor w/ 82 levels "7310440014","7310440029",..: 52 52 43
##   .. .. ..$ Sentrix_Position: Factor w/ 12 levels "R01C01","R01C02",..: 1 10 6 2 9 5 1 1
##   .. .. ..$ Bisulfite_Batch : Factor w/ 11 levels " 1"," 2"," 3",..: 8 8 8 8 8 8 8 8 8 8
##   .. .. ..$ Scan_Batch      : Factor w/ 4 levels "0","1","2","3": 3 3 3 3 3 3 3 3 3 4 .
##   .. .. ..$ None            : Factor w/ 1 level "1": 1 1 1 1 1 1 1 1 1 1 ...
##   .. ..@ controls :'data.frame': 848 obs. of  4 variables:
##   .. .. ..$ Address      : int [1:848] 10627500 10673427 10714330 10721502 10731326 1073
##   .. .. ..$ Type         : chr [1:848] "NEGATIVE" "SPECIFICITY I" "NORM_T" "NEGATIVE" .
##   .. .. ..$ Color_Channel: chr [1:848] "Purple" "Lime" "Purple" "BlueViolet" ...
##   .. .. ..$ Name         : chr [1:848] "Negative 265" "GT Mismatch 3 (PM)" "Norm_T46" "N
##   .. ..@ Rcontrols: int [1:848, 1:500] 124 298 3808 170 206 113 131 153 177 489 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:848] "10627500" "10673427" "10714330" "10721502" ...
##   .. .. .. ..$ : chr [1:500] "9374341033_R01C01" "9374341033_R05C02" "9340996084_R03C02"
##   .. ..@ Gcontrols: int [1:848, 1:500] 126 8076 195 84 145 77 106 80 47 7079 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:848] "10627500" "10673427" "10714330" "10721502" ...
##   .. .. .. ..$ : chr [1:500] "9374341033_R01C01" "9374341033_R05C02" "9340996084_R03C02"
##   .. ..@ DPfreq   : Named num [1:500] 0.999 1 0.999 1 0.999 ...
##   .. .. ..- attr(*, "names")= chr [1:500] "9374341033_R01C01" "9374341033_R05C02" "93409
##   .. ..@ MU       : num [1:2, 1:500] 4622 2953 4978 3242 5144 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:2] "Methylated" "Unmethylated"
##   .. .. .. ..$ : chr [1:500] "9374341033_R01C01" "9374341033_R05C02" "9340996084_R03C02"
##   .. ..@ plotdata :'data.frame': 331000 obs. of  7 variables:
##   .. .. ..$ Address      : int [1:331000] 10627500 10627500 10627500 10627500 10627500 1
##   .. .. ..$ Type         : chr [1:331000] "NEGATIVE" "NEGATIVE" "NEGATIVE" "NEGATIVE" .
##   .. .. ..$ Color_Channel: chr [1:331000] "Purple" "Purple" "Purple" "Purple" ...
##   .. .. ..$ Name         : chr [1:331000] "Negative 265" "Negative 265" "Negative 265" "
##   .. .. ..$ Samples      : Factor w/ 962 levels "7310440014_R01C01",..: 754 169 258 576
##   .. .. ..$ IntRed       : num [1:331000] 7.29 8.69 7.27 8.5 7.85 ...
##   .. .. ..$ IntGrn       : num [1:331000] 6.66 7.79 6.64 7.83 8.17 ...

##

## summarizedData object with 1000 samples.
## Containing: median Methylated and Unmethylation values,
##             detection P-values
##             and all quality control probe intensities.
```

8 Session info

Here is the output of `sessionInfo` on the system on which this document was compiled:

- R version 3.2.0 (2015-04-16), `x86_64-unknown-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_US.UTF-8`, `LC_COLLATE=C`, `LC_MONETARY=en_US.UTF-8`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=en_US.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.UTF-8`, `LC_IDENTIFICATION=C`
- Base packages: base, datasets, grDevices, graphics, methods, parallel, stats, stats4, utils
- Other packages: Biobase 2.28.0, BiocGenerics 0.14.0, Biostrings 2.36.1, DBI 0.3.1, GenomeInfoDb 1.4.0, GenomicRanges 1.20.3, IRanges 2.2.1, IlluminaHumanMethylation450kanno.ilmn12.hg19 0.2.1, IlluminaHumanMethylation450kmanifest 0.4.0, MethylAid 1.2.5, RSQLite 1.0.0, S4Vectors 0.6.0, XVector 0.8.0, bumphunter 1.8.0, foreach 1.4.2, iterators 1.0.7, lattice 0.20-31, locfit 1.5-9.1, minfi 1.14.0, minfiData 0.10.0
- Loaded via a namespace (and not attached): AnnotationDbi 1.30.1, BiocParallel 1.2.1, BiocStyle 1.6.0, FDb.InfiniumMethylation.hg19 2.2.0, GEOquery 2.34.0, GenomicAlignments 1.4.1, GenomicFeatures 1.20.1, MASS 7.3-40, R6 2.0.1, RColorBrewer 1.1-2, RCurl 1.95-4.6, RJSONIO 1.3-0, Rcpp 0.11.6, Rsamtools 1.20.1,

TxDb.Hsapiens.UCSC.hg19.knownGene 3.1.2, XML 3.98-1.1, annotate 1.46.0, base64 1.1, beanplot 1.2, biomaRt 2.24.0, bitops 1.0-6, codetools 0.2-11, colorspace 1.2-6, digest 0.6.8, doRNG 1.6, evaluate 0.7, formatR 1.2, futile.logger 1.4.1, futile.options 1.0.0, genefilter 1.50.0, ggplot2 1.0.1, grid 3.2.0, gridBase 0.4-7, gtable 0.1.2, hexbin 1.27.0, highr 0.5, htmltools 0.2.6, httpuv 1.3.2, illuminaio 0.10.0, knitr 1.10.5, lambda.r 1.1.7, limma 3.24.3, magrittr 1.5, matrixStats 0.14.0, mclust 5.0.1, mime 0.3, multtest 2.24.0, munsell 0.4.2, nlme 3.1-120, nor1mix 1.2-0, org.Hs.eg.db 3.1.2, pkgmaker 0.22, plyr 1.8.2, preprocessCore 1.30.0, proto 0.3-10, quadprog 1.5-5, registry 0.2, reshape 0.8.5, reshape2 1.4.1, rngtools 1.2.4, rtracklayer 1.28.2, scales 0.2.4, shiny 0.11.1, siggenes 1.42.0, splines 3.2.0, stringi 0.4-1, stringr 1.0.0, survival 2.38-1, tools 3.2.0, xtable 1.7-4, zlibbioc 1.14.0

References

[1] M. van Iterson, E. W. Tobi, R. C. Slieker, W. den Hollander, R. Luijk, P. E. Slagboom, and B. T. Heijmans. MethylAid: visual and interactive quality control of large Illumina 450k datasets. *Bioinformatics*, 30(23):3435–3437, 2014.

[2] R. Edgar, M. Domrachev, and A. E. Lash. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res.*, 30(1):207–210, Jan 2002.

[3] Y. Liu, M. J. Aryee, L. Padyukov, M. D. Fallin, E. Hesselberg, A. Runarsson, L. Reinius, N. Acevedo, M. Taub, M. Ronninger, K. Shchetynsky, A. Scheynius, J. Kere, L. Alfredsson, L. Klareskog, T. J. Ekstrom, and A. P. Feinberg. Epigenome-wide association data implicate DNA methylation as an intermediary of genetic risk in rheumatoid arthritis. *Nat. Biotechnol.*, 31(2):142–147, Feb 2013.

[4] M. J. Aryee, A. E. Jaffe, H. Corrada-Bravo, C. Ladd-Acosta, A. P. Feinberg, K. D. Hansen, and R. A. Irizarry. Minfi: a flexible and comprehensive Bioconductor package for the analysis of Infinium DNA methylation microarrays. *Bioinformatics*, 2014.

[5] Bernd Bischl, Michel Lang, Olaf Mersmann, Joerg Rahnenfuehrer, and Claus Weihs. Computing on high performance clusters with r: Packages batchjobs and batchexperiments. Technical Report 1, TU Dortmund, 2011. URL: http://sfb876.tu-dortmund.de/PublicPublicationFiles/bischl_etal_2012a.pdf.