The qlasso user's guide

Kwame Okrah kwame.okrah@gmail.com   Hector Corrado Bravo hcorrada@gmail.com
Stephanie C. Hicks shicks@jimmy.harvard.edu
Rafael A. Irizarry rafa@jimmy.harvard.edu

Modified: March 5, 2015. Compiled: March 5, 2015

Contents

1 Introduction

Add introduction here.

2 Getting Started

Load the qlasso package in R.

```
library(qlasso)
```

3 Data

3.1 flowSorted Data Example

Load an example data set. Here we use the flowSorted data set in quantro. (but we can change this to whatever).

```
library(quantro)
library(minfi)
```

```
data(flowSorted)
p <- getBeta(flowSorted, offset = 100)
pd <- pData(flowSorted)
```

4 Using `qlasso` for normalization

4.1 Computing quantiles

The sample quantiles of the raw data can be computed using the `scals()` function. The `scals()` function also computes the distribution of all the samples averaged across the quantiles. We call this the quantile reference.

$$\bar{Q}_{..}(u) = \frac{1}{n_T} \sum_{n_T}^{i=1} Q_{ik}(u)$$

```
alpha = p - rowMeans(apply(p, 2, sort))
# what about median normalization?
# what about geometric mean?
# alpha = scals(p)£alpha
```

4.2 Using the `qlasso()` function

Compute F-statistics for each quantile.

```
# I think we should create a qlasso() function that wraps up fstat + fitCoeffs
fVals = fStat(alpha = alpha, groupFactor = pd$CellType)
```

Use qlasso to fit a linear model at each quantile.

```
qlassoFit = fitCoeffs(alpha = alpha, groupFactor = pd$CellType,
                      lambda = 1 / fVals$fstat)
head(qlassoFit$betas)

##                 NeuN_neg    NeuN_pos
## cg11805814 0.745649323 0.790179226
## cg27272293 0.745646130 0.724215860
## cg21785710 0.753945438 0.772728239
## cg27189973 0.150035005 0.279568646
## cg06532611 0.621730859 0.852632472
## cg18838207 0.006808749 0.006578851
```

4.3 Obtaining qlasso normalized values

The normalized values are computed using

5 SessionInfo

```
sessionInfo()

## R version 3.1.2 (2014-10-31)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats4    parallel  stats     graphics  grDevices datasets  utils     methods
## [9] base
##
## other attached packages:
##  [1] minfi_1.12.0         bumphunter_1.6.0     locfit_1.5-9.1       iterators_1.0.7
##  [5] foreach_1.4.2        Biostrings_2.34.1    XVector_0.6.0        GenomicRanges_1.18.4
##  [9] GenomeInfoDb_1.2.4   IRanges_2.0.1        S4Vectors_0.4.0      lattice_0.20-30
## [13] Biobase_2.26.0       BiocGenerics_0.12.1  quantro_1.0.0        qlasso_0.0.0.9000
## [17] knitr_1.9
##
## loaded via a namespace (and not attached):
##  [1] annotate_1.44.0       AnnotationDbi_1.28.1  base64_1.1
##  [4] beanplot_1.2          BiocStyle_1.4.1       codetools_0.2-10
##  [7] colorspace_1.2-4      DBI_0.3.1             digest_0.6.8
## [10] doParallel_1.0.8      doRNG_1.6             evaluate_0.5.5
## [13] formatR_1.0           genefilter_1.48.1     ggplot2_1.0.0
## [16] grid_3.1.2            gtable_0.1.2          highr_0.4
## [19] illuminaio_0.8.0      limma_3.22.6          MASS_7.3-39
## [22] matrixStats_0.14.0    mclust_4.4            multtest_2.22.0
## [25] munsell_0.4.2         nlme_3.1-120          nor1mix_1.2-0
## [28] pkgmaker_0.22         plyr_1.8.1            preprocessCore_1.28.0
## [31] proto_0.3-10          quadprog_1.5-5        RColorBrewer_1.1-2
## [34] Rcpp_0.11.4           registry_0.2          reshape_0.8.5
## [37] reshape2_1.4.1        rngtools_1.2.4        RSQLite_1.0.0
## [40] scales_0.2.4          siggenes_1.40.0       splines_3.1.2
## [43] stringr_0.6.2         survival_2.38-1       tools_3.1.2
## [46] XML_3.98-1.1          xtable_1.7-4          zlibbioc_1.12.0
```