Differential abundance analysis for the 2016-17 cohort

EJC

25/06/2020, based on code from May 2019

Setup

The Babraham compute cluster does not contain a global tex installation, so a local tex is added to $PATH to allow knitting to pdf.

```
Sys.setenv(PATH=paste(Sys.getenv("PATH"),
                      "/bi/home/carre/texlive/2017/bin/x86_64-linux/",sep=":"))
```

```
load(file = "../cohort_2016_17/data/SCE_QC_pass_finalised.RData")

library(edgeR)

sce$sample <- factor(paste(sce$phenotype, sce$PID, sep = "_"))

abundances <- table(sce$clusters, sce$sample)

## Make coumne metadata + DGEList
extra.info <- colData(sce)[match(colnames(abundances), sce$sample),
    ]

y.ab <- DGEList(abundances, samples = extra.info)
y.ab
```

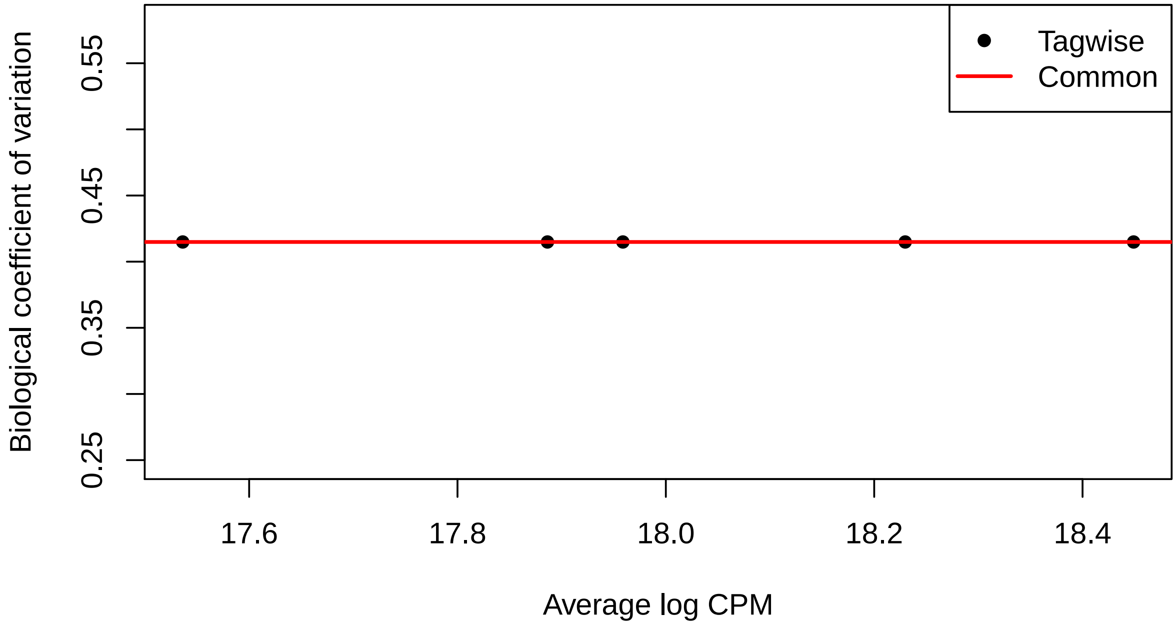
```
## An object of class "DGEList"
## $counts
##
##     old d0_520P old d0_526W old d0_536G old d0_541M old d0_543P old d0_544Q
##   1           3           0           1           5           1           0
##   2           1          10           1           1          13           0
##   3           5           1           0           0           1           5
##   4           2           1           3           0           1           5
##   5           5           1           4           5           3           0
##
##     old d0_545R old d0_652H old d0_660R old d42_520P old d42_526W old d42_536G
##   1           0           0           2            8            1            5
##   2           5           2           1            1           11            3
##   3           1           0           1            4            2            3
##   4           2           0           4            4            3            3
##   5           0           0           1           15            3           16
##
##     old d42_541M old d42_543P old d42_544Q old d42_545R old d42_643Y
##   1            7           10            8            3            0
##   2            2           18            1           10            3
##   3            3            3            8            4            2
##   4            0            5            2            3            0
##   5           18           11            9            4            2
##
##     old d42_652H old d42_660R young d0_501T young d0_559G young d0_562K
##   1            0            8             0             0             0
##   2            1            0            10             0             8
##   3            0            6             0             9             9
##   4            0            5             0             2             2
##   5            0           11             0             5             4
##
##     young d0_568R young d0_594V young d0_602D young d0_622A young d0_627F
##   1             2             8             0             0             1
##   2             3             1             0             0             8
##   3             7             4             0             3             0
##   4             1             2             0             3             0
##   5             3             2             1             3             0
##
##     young d0_637R young d42_501T young d42_559G young d42_562K young d42_568R
##   1             0              0              1              2              9
##   2             0             25              5              7              5
##   3             0              0             13              8              5
##   4             4              0              4              2              2
##   5             2              1             14              4             11
##
##     young d42_594V young d42_602D young d42_622A young d42_627F young d42_637R
##   1             20              9              9             12              4
##   2              6              1              1             10              0
##   3              2              3              2              2              5
##   4              1              1              6              2              5
##   5              7             18              4              8             18
##
##     young d42_665X
##   1             21
##   2             14
##   3              3
##   4              0
##   5              6
##
## $samples
##              group lib.size norm.factors     lane       i5       i7  lib_plate
## old d0_520P     1       16            1 lane6967 CGTACTAG AAGGAGTA cDNA190820
## old d0_526W     1       13            1 lane7055 CGAGGCTG AAGGAGTA cDNA190910
## old d0_536G     1        9            1 lane7035 CGTACTAG AAGGAGTA cDNA190919
## old d0_541M     1       11            1 lane7043 CGTACTAG AAGGAGTA cDNA190920
## old d0_543P     1       19            1 lane6966 CGTACTAG AAGGAGTA cDNA190819
##              lib_well  PID day
## old d0_520P       D2 520P  d0
## old d0_526W       D8 526W  d0
## old d0_536G       D2 536G  d0
## old d0_541M       D2 541M  d0
## old d0_543P       D2 543P  d0
##                                                                          short.name
## old d0_520P lane6967.CGTACTAG.AAGGAGTA.cDNA190820.D2.520P.d0.L001.GRCh38.hisat2.bam
## old d0_526W  lane7055.CGAGGCTG.AAGGAGTA.lib190910.D8.526W.d0.L001.GRCh38.hisat2.bam
## old d0_536G lane7035.CGTACTAG.AAGGAGTA.cDNA190919.D2.536G.d0.L001.GRCh38.hisat2.bam
## old d0_541M lane7043.CGTACTAG.AAGGAGTA.cDNA190920.D2.541M.d0.L001.GRCh38.hisat2.bam
## old d0_543P lane6966.CGTACTAG.AAGGAGTA.cDNA190819.D2.543P.d0.L001.GRCh38.hisat2.bam
##                                     fcs_name fcs.well     FSC.A    FSC.W FSC.H
## old d0_520P 520P_d0_INX_520P_d0_001_002.fcs       D2 107509.02 86902.55 81076
## old d0_526W 526W_d0_INX_526W_d0_001_021.fcs       D2  85692.48 81804.23 68651
## old d0_536G 536G_d0_INX_536G_d0_001_014.fcs       D2  57698.40 72770.91 51962
## old d0_541M 541M_d0_INX_541M_d0_001_014.fcs       D2  62003.74 73358.55 55392
## old d0_543P 543P_d0_INX_543P_d0_001_014.fcs       D2 128828.04 88785.45 95093
##                SSC.A     SSC.W SSC.H    hA.PE CD21.PE.cy7 CD38.BV421 CD20.BV605
## old d0_520P 43590.39  95069.38 30049 3.410988    3.077324  1.3262657   2.266734
## old d0_526W 30935.93  88156.23 22998 2.776826    2.751422  0.9179547   2.433115
## old d0_536G 20158.53  70365.34 18775 3.196373    2.894513  1.7420159   2.081448
## old d0_541M 23315.03  68973.67 22153 3.135865    1.321624  0.6451803   2.735065
## old d0_543P 43282.08 109675.38 25863 3.458933    3.030519  1.2805317   2.519604
##              CD27.BV711   hA.APC DUMP.APC.ef780 SA.BUV395 CD19.BUV496
## old d0_520P  1.54647852 2.704042      0.5308370 0.6831533    1.962610
## old d0_526W  1.87768920 2.092450      1.0954869 0.7579735    1.969199
## old d0_536G  1.78570126 2.478432      0.6666756 0.2750081    2.191671
## old d0_541M -0.04161669 2.594687      0.6842753 0.2911566    2.557347
## old d0_543P  2.28214291 2.805983      0.6772747 0.3217367    2.366766
##             IgD.BUV737 CD71.FITC    Time age fcs.XLoc fcs.YLoc phenotype
## old d0_520P  0.8858766  1.337242 26238.3 old        3        1    old d0
## old d0_526W  2.2615848  1.242474 21663.0 old        3        1    old d0
## old d0_536G  0.7497302  1.228306 61989.1 old        3        1    old d0
## old d0_541M  0.4045954  1.220576 50172.9 old        3        1    old d0
## old d0_543P  1.0614616  1.716234 28457.5 old        3        1    old d0
##                 sum detected percent_top_50 percent_top_100 percent_top_200
## old d0_520P 1179579     3424       35.41925        43.88142        53.85167
## old d0_526W 1300962     2279       55.42237        66.11377        78.45809
## old d0_536G 1174357     3844       31.53096        39.18289        49.30860
## old d0_541M 1012533     2793       40.58771        49.99975        61.21292
## old d0_543P 3200353     3242       46.74316        55.71729        67.01845
##             percent_top_500 subsets_Mito_sum subsets_Mito_detected
## old d0_520P        70.86452           160890                    23
## old d0_526W        94.43673           134637                    24
## old d0_536G        66.89431           120640                    26
## old d0_541M        79.16848            85760                    25
## old d0_543P        84.95813           346895                    27
##             subsets_Mito_percent   total qc_fail library clusters      sample
## old d0_520P            13.639612 1179579   FALSE       C        5 old d0_520P
## old d0_526W            10.349034 1300962   FALSE       E        2 old d0_526W
## old d0_536G            10.272856 1174357   FALSE       H        5 old d0_536G
## old d0_541M             8.469847 1012533   FALSE       I        1 old d0_541M
## old d0_543P            10.839273 3200353   FALSE       B        2 old d0_543P
## 33 more rows ...
```

```
# Filter out low abundance labels: Skipped as tends to filter
# out all labels (we know these are biologically meaningful
# clusters, so should not be filtered out on count alone).
# keep <- filterByExpr(y.ab, group=y.ab$samples$day0) y.ab <-
# y.ab[keep,] summary(keep)
```

```r
design <- model.matrix(~factor(PID) + factor(day), y.ab$samples)

y.ab <- calcNormFactors(y.ab, method = "TMMwsp")  # we need to normalise to 'library' size as day 0 has


y.ab <- estimateDisp(y.ab, design, trend = "none")
summary(y.ab$common.dispersion)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1721  0.1721  0.1721  0.1721  0.1721  0.1721
```

```r
plotBCV(y.ab, cex = 1)
```



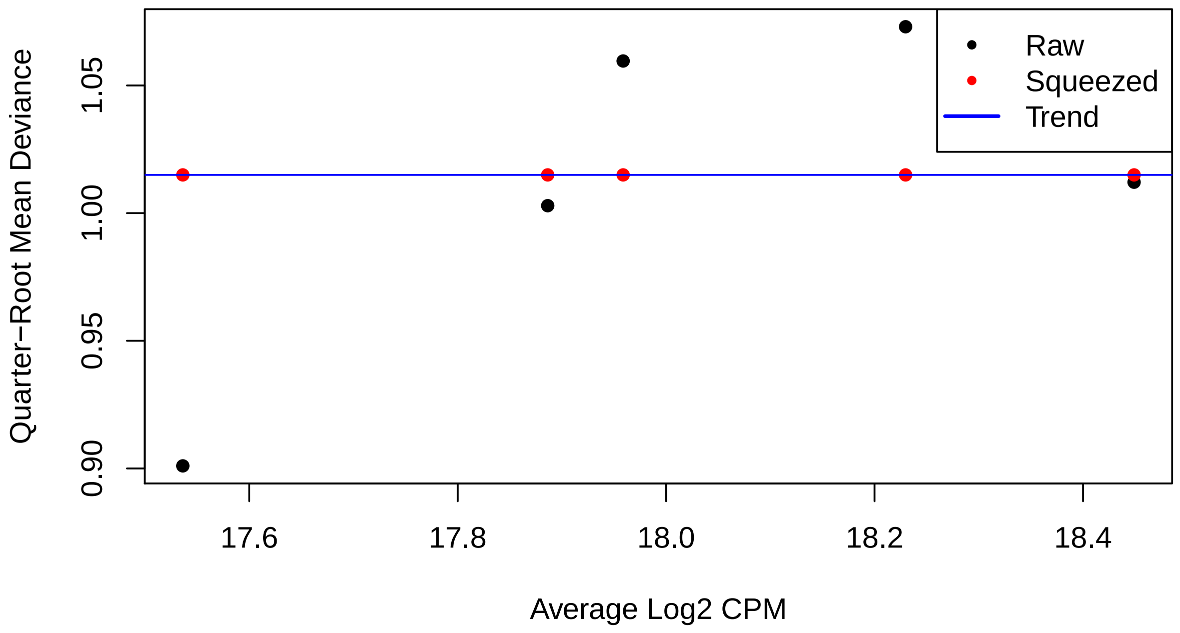
```r
fit.ab <- glmQLFit(y.ab, design, robust = TRUE, abundance.trend = FALSE)
summary(fit.ab$var.prior)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.061   1.061   1.061   1.061   1.061   1.061
```

```r
summary(fit.ab$df.prior)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     Inf     Inf     Inf     Inf     Inf     Inf
```

```r
plotQLDisp(fit.ab, cex = 1)
```



```
res <- glmQLFTest(fit.ab, coef = ncol(design))
summary(decideTests(res))
```

```
##        factor(day)d42
## Down                0
## NotSig              3
## Up                  2
```

```
topTags(res)
```

```
## Coefficient:  factor(day)d42
##        logFC   logCPM         F      PValue        FDR
## 1  1.2000102 17.95876 8.7551539 0.003998414 0.01999207
## 5  0.8484600 18.44901 6.2970082 0.013991149 0.03497787
## 4 -0.5479050 17.53626 1.8276890 0.179987206 0.29997868
## 3 -0.1631631 17.88638 0.1887136 0.665090216 0.67162254
## 2  0.1517558 18.22977 0.1809624 0.671622542 0.67162254
```

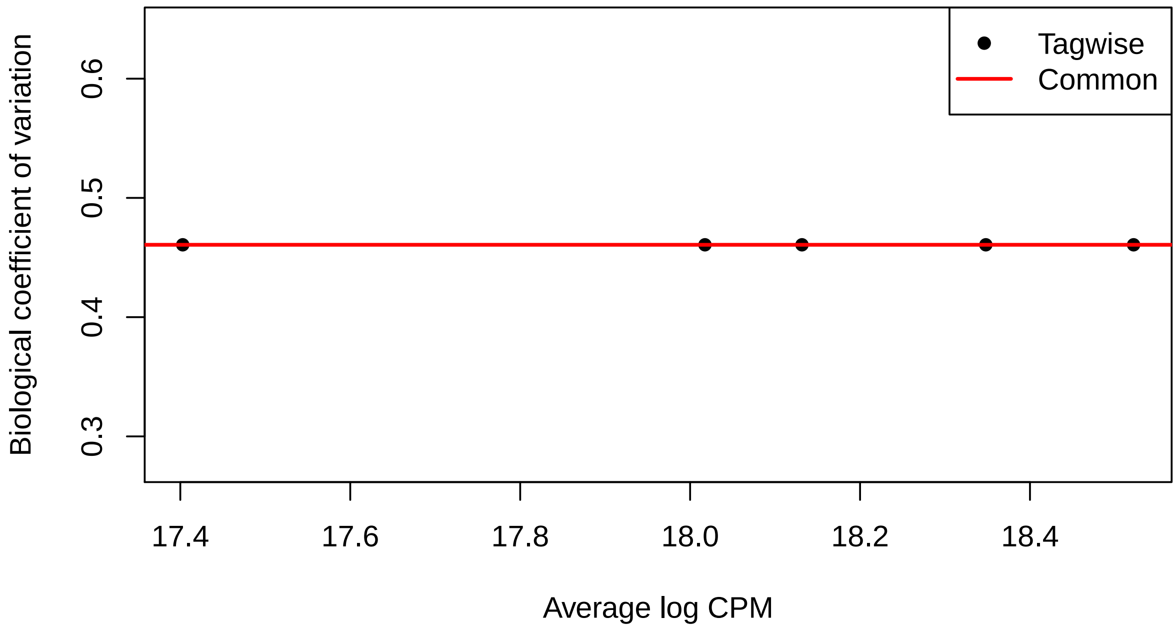
```
### For young only:

y.ab.Y <- DGEList(abundances[, grepl(colnames(abundances), pattern = "young")],
    samples = extra.info[grepl(colnames(abundances), pattern = "young"),
        ])
y.ab.Y
```

```
## An object of class "DGEList"
## $counts
##
##   young d0_501T young d0_559G young d0_562K young d0_568R young d0_594V
## 1             0             0             0             2             8
##   2            10             0             8             3             1
##   3             0             9             9             7             4
##   4             0             2             2             1             2
##   5             0             5             4             3             2
##
##     young d0_602D young d0_622A young d0_627F young d0_637R young d42_501T
##   1             0             0             1             0              0
##   2             0             0             8             0             25
##   3             0             3             0             0              0
##   4             0             3             0             4              0
##   5             1             3             0             2              1
##
##     young d42_559G young d42_562K young d42_568R young d42_594V young d42_602D
##   1              1              2              9             20              9
##   2              5              7              5              6              1
##   3             13              8              5              2              3
##   4              4              2              2              1              1
##   5             14              4             11              7             18
##
##     young d42_622A young d42_627F young d42_637R young d42_665X
##   1              9             12              4             21
##   2              1             10              0             14
##   3              2              2              5              3
##   4              6              2              5              0
##   5              4              8             18              6
##
## $samples
##               group lib.size norm.factors     lane       i5       i7  lib_plate
## young d0_501T     1       10            1 lane7055 CGTACTAG AAGGAGTA cDNA190910
## young d0_559G     1       16            1 lane6967 CGAGGCTG AAGGAGTA cDNA190820
## young d0_562K     1       23            1 lane6966 CGAGGCTG AAGGAGTA cDNA190819
## young d0_568R     1       16            1 lane7035 CGAGGCTG AAGGAGTA cDNA190919
## young d0_594V     1       17            1 lane6963 CGAGGCTG AAGGAGTA cDNA190807
##               lib_well  PID day
## young d0_501T       D2 501T  d0
## young d0_559G       D8 559G  d0
## young d0_562K       D8 562K  d0
## young d0_568R       D8 568R  d0
## young d0_594V       D8 594V  d0
##                                                                             short.name
## young d0_501T  lane7055.CGTACTAG.AAGGAGTA.lib190910.D2.501T.d0.L001.GRCh38.hisat2.bam
## young d0_559G lane6967.CGAGGCTG.AAGGAGTA.cDNA190820.D8.559G.d0.L001.GRCh38.hisat2.bam
## young d0_562K lane6966.CGAGGCTG.AAGGAGTA.cDNA190819.D8.562K.d0.L001.GRCh38.hisat2.bam
## young d0_568R lane7035.CGAGGCTG.AAGGAGTA.cDNA190919.D8.568R.d0.L001.GRCh38.hisat2.bam
## young d0_594V lane6963.CGAGGCTG.AAGGAGTA.cDNA190807.D8.594V.d0.L001.GRCh38.hisat2.bam
##                                      fcs_name fcs.well     FSC.A    FSC.W FSC.H
## young d0_501T 501T_d0_INX_637R_d0_001_014.fcs       D2  83306.88 80588.05 67747
## young d0_559G 559G_d0_INX_559G_d0_001_007.fcs       D2 133497.59 92261.68 94827
## young d0_562K 562K_d0_INX_562K_d0_001_020.fcs       D2 119693.94 86596.55 90584
## young d0_568R 568R_d0_INX_568R_d0_001_019.fcs       D2  49146.88 74580.08 43187
## young d0_594V 594V_d0_INX_594V_d0_001_019.fcs       D2 125202.00 88759.98 92443
##                  SSC.A     SSC.W SSC.H    hA.PE CD21.PE.cy7 CD38.BV421
## young d0_501T 31408.37  93537.17 22006 2.985005    3.066182  1.7504292
## young d0_559G 48985.17 100321.62 32000 4.005015    2.995239  0.7007657
## young d0_562K 53872.98  98524.33 35835 3.666481    1.249043  0.7479908
## young d0_568R 16244.47  70242.65 15156 2.946224    2.449280  0.7234898
## young d0_594V 58338.56  91846.05 41627 2.990319    1.423069  1.0976290
##               CD20.BV605 CD27.BV711   hA.APC DUMP.APC.ef780 SA.BUV395
## young d0_501T   2.365504  0.5200183 2.302632      0.4663553 0.4345870
## young d0_559G   2.345368  1.5749396 3.299656      0.8471232 0.8960040
## young d0_562K   2.607453  1.4252057 3.007837      1.3674945 1.1612936
## young d0_568R   2.272444  1.7482492 2.200919      0.9441139 1.1092720
## young d0_594V   2.817087  0.8190601 2.317705      1.0837874 0.5235748
##               CD19.BUV496 IgD.BUV737 CD71.FITC    Time   age fcs.XLoc fcs.YLoc
## young d0_501T    2.127250  2.9952726  1.253009 30610.0 young        3        1
## young d0_559G    2.052803  2.9651542  1.103903 14288.5 young        3        1
## young d0_562K    2.402106  0.4256185  1.195688 19598.7 young        3        1
## young d0_568R    2.684368  0.3721393  1.288887 24872.8 young        3        1
## young d0_594V    2.581955  0.8169065  2.006950 18128.9 young        3        1
##               phenotype     sum detected percent_top_50 percent_top_100
## young d0_501T  young d0  456978     1296       55.96112        70.27712
## young d0_559G  young d0 1453987     4460       26.61406        34.78889
## young d0_562K  young d0 2374096     3122       43.75703        52.82116
## young d0_568R  young d0 1217669     3089       38.03759        47.18721
## young d0_594V  young d0 2182491     3967       40.26385        48.64465
##               percent_top_200 percent_top_500 subsets_Mito_sum
## young d0_501T        85.45138        98.01063            78787
## young d0_559G        44.58850        60.83046           118263
## young d0_562K        62.85689        79.28066           296904
## young d0_568R        57.93717        74.36906           144367
## young d0_594V        57.79062        71.86353           265542
##               subsets_Mito_detected subsets_Mito_percent   total qc_fail
## young d0_501T                    24            17.240874  456978   FALSE
## young d0_559G                    24             8.133704 1453987   FALSE
## young d0_562K                    27            12.505981 2374096   FALSE
## young d0_568R                    25            11.856013 1217669   FALSE
## young d0_594V                    27            12.166923 2182491   FALSE
##               library clusters        sample
## young d0_501T       E        2 young d0_501T
## young d0_559G       C        3 young d0_559G
## young d0_562K       B        3 young d0_562K
## young d0_568R       H        3 young d0_568R
## young d0_594V       A        1 young d0_594V
## 14 more rows ...
```

```r
# Filter out low abundance labels: Skipped as tends to filter
# out all labels (we know these are biologically meaningful
# clusters, so should not be filtered out on count alone).
# keep <- filterByExpr(y.ab, group=y.ab$samples$day0) y.ab <-
# y.ab[keep,] summary(keep)

design.Y <- model.matrix(~factor(PID) + factor(day), y.ab.Y$samples)

y.ab.Y <- calcNormFactors(y.ab.Y, method = "TMMwsp")  # we need to normalise to 'library' size as day 0


y.ab.Y <- estimateDisp(y.ab.Y, design.Y, trend = "none")
summary(y.ab.Y$common.dispersion)
```

```
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.2122  0.2122  0.2122  0.2122  0.2122  0.2122
```

```
plotBCV(y.ab.Y, cex = 1)
```



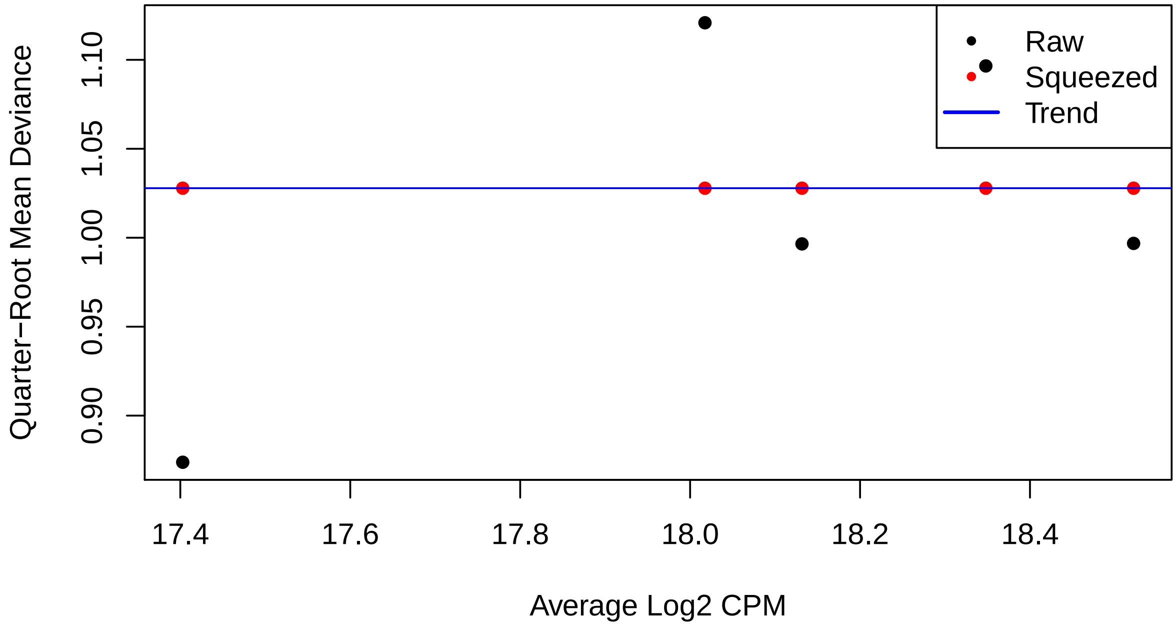
```
fit.ab.Y <- glmQLFit(y.ab.Y, design.Y, robust = TRUE, abundance.trend = FALSE)
summary(fit.ab.Y$var.prior)
```

```
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.116   1.116   1.116   1.116   1.116   1.116
```

```
summary(fit.ab.Y$df.prior)
```

```
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      Inf     Inf     Inf     Inf     Inf     Inf
```

```
plotQLDisp(fit.ab.Y, cex = 1)
```



```r
res.Y <- glmQLFTest(fit.ab.Y, coef = ncol(design.Y))
topTags(res.Y)
```

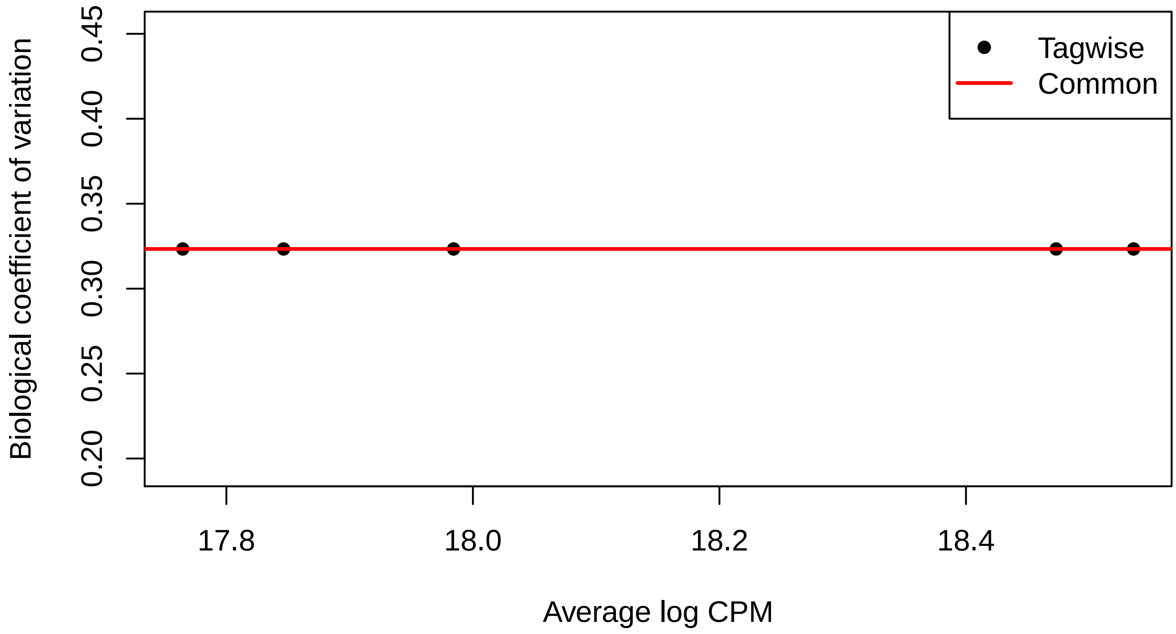
```
## Coefficient:  factor(day)d42 
##       logFC   logCPM          F       PValue         FDR
## 1 2.2701085 18.34803 14.3863850 0.0004933438 0.002466719
## 5 1.6182035 18.52196 10.5333274 0.0023726630 0.005931657
## 2 1.0755663 18.13166  3.8644556 0.0562875278 0.093812546
## 4 0.4185173 17.40284  0.4697461 0.4970570428 0.621321304
## 3 0.1627804 18.01748  0.0957215 0.7586316727 0.758631673
```

```r
### For old only:
y.ab.O <- DGEList(abundances[, grepl(colnames(abundances), pattern = "old")],
    samples = extra.info[grepl(colnames(abundances), pattern = "old"),
        ])


# Filter out low abundance labels: Skipped as tends to filter
# out all labels (we know these are biologically meaningful
# clusters, so should not be filtered out on count alone).
# keep <- filterByExpr(y.ab, group=y.ab$samples$day0) y.ab <-
# y.ab[keep,] summary(keep)

design.O <- model.matrix(~factor(PID) + factor(day), y.ab.O$samples)

y.ab.O <- calcNormFactors(y.ab.O, method = "TMMwsp")  # we need to normalise to 'library' size as day 0


y.ab.O <- estimateDisp(y.ab.O, design.O, trend = "none")
```

```
summary(y.ab.0$common.dispersion)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1046  0.1046  0.1046  0.1046  0.1046  0.1046
```

```
plotBCV(y.ab.0, cex = 1)
```



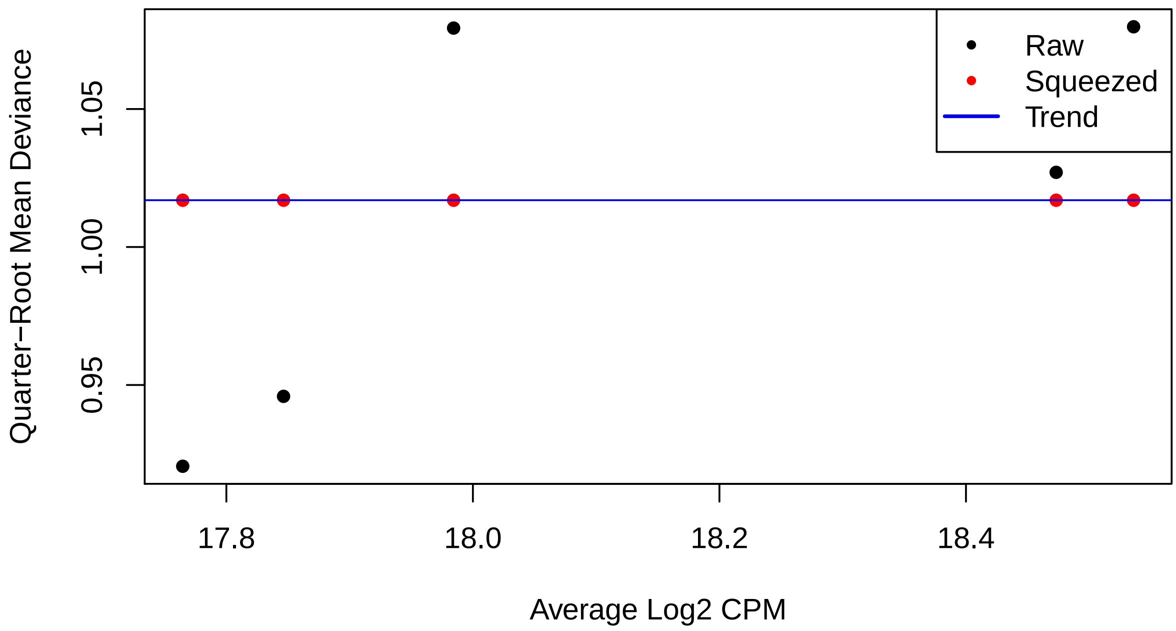
```
fit.ab.0 <- glmQLFit(y.ab.0, design.0, robust = TRUE, abundance.trend = FALSE)
summary(fit.ab.0$var.prior)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.07    1.07    1.07    1.07    1.07    1.07
```

```
summary(fit.ab.0$df.prior)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     Inf     Inf     Inf     Inf     Inf     Inf
```

```
plotQLDisp(fit.ab.0, cex = 1)
```



```
res.0 <- glmQLFTest(fit.ab.0, coef = ncol(design.0))
topTags(res.0)
```

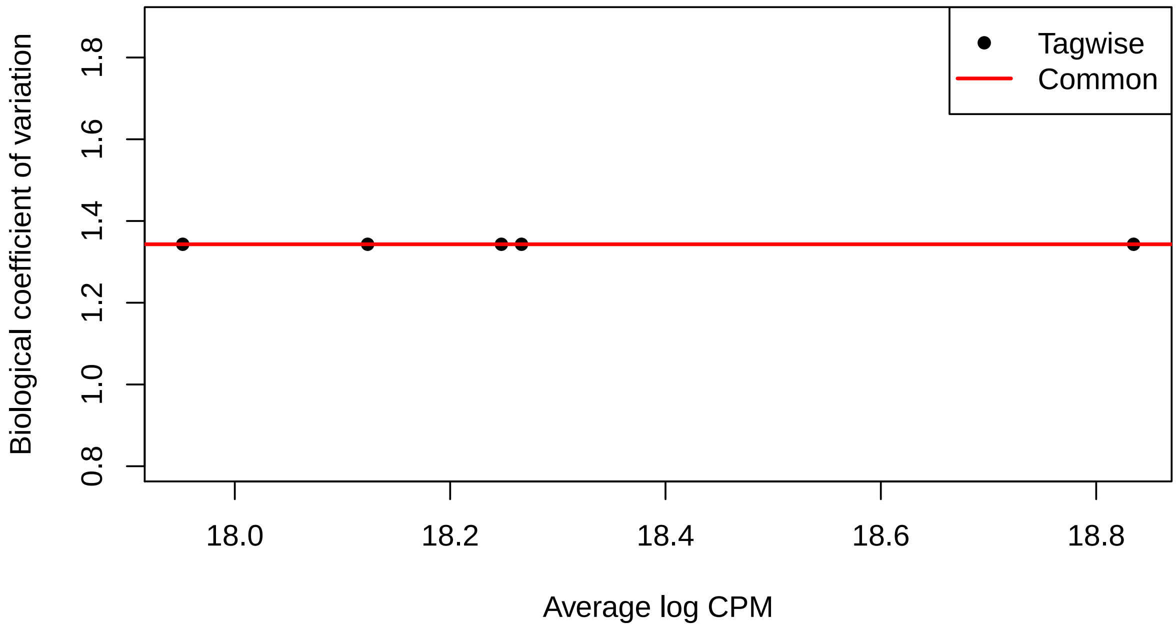
```
## Coefficient:  factor(day)d42
##        logFC   logCPM          F    PValue       FDR
## 5  0.7845489 18.53600 3.14948721 0.0835618 0.1796042
## 4 -0.8664623 17.76460 2.73742891 0.1058495 0.1796042
## 2 -0.7572025 18.47319 2.70673131 0.1077625 0.1796042
## 1  0.6231418 17.98435 1.42926387 0.2389243 0.2986553
## 3  0.1442099 17.84638 0.07208216 0.7897088 0.7897088
```

```
########################### For day0 only:
y.ab.day0 <- DGEList(abundances[, grepl(colnames(abundances),
    pattern = "d0")], samples = extra.info[grepl(colnames(abundances),
    pattern = "d0"), ])


# Filter out low abundance labels: Skipped as tends to filter
# out all labels (we know these are biologically meaningful
# clusters, so should not be filtered out on count alone).
# keep <- filterByExpr(y.ab, group=y.ab$samples$day0) y.ab <-
# y.ab[keep,] summary(keep)

design.day0 <- model.matrix(~factor(age), y.ab.day0$samples)

y.ab.day0 <- calcNormFactors(y.ab.day0, method = "TMMwsp")  # we need to normalise to 'library' size as


y.ab.day0 <- estimateDisp(y.ab.day0, design.day0, trend = "none")
```

```
summary(y.ab.day0$common.dispersion)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.804   1.804   1.804   1.804   1.804   1.804
```

```
plotBCV(y.ab.day0, cex = 1)
```



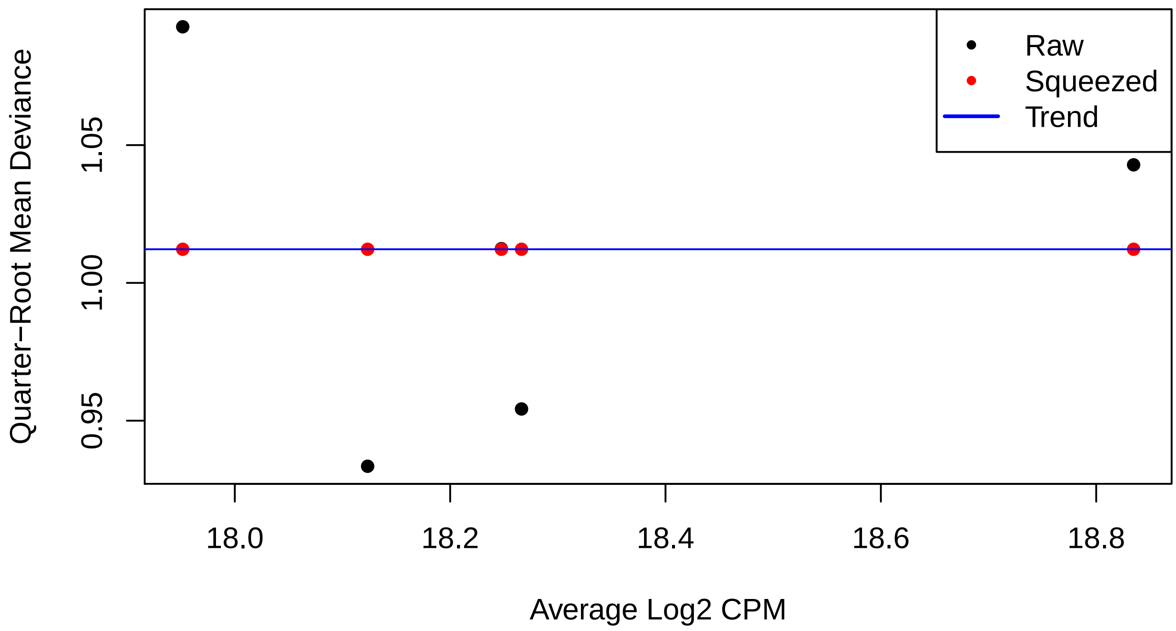
```
fit.ab.day0 <- glmQLFit(y.ab.day0, design.day0, robust = TRUE,
    abundance.trend = FALSE)
summary(fit.ab.day0$var.prior)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.05    1.05    1.05    1.05    1.05    1.05
```

```
summary(fit.ab.day0$df.prior)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     Inf     Inf     Inf     Inf     Inf     Inf
```

```
plotQLDisp(fit.ab.day0, cex = 1)
```



```
res.day0 <- glmQLFTest(fit.ab.day0, coef = ncol(design.day0))
topTags(res.day0)
```

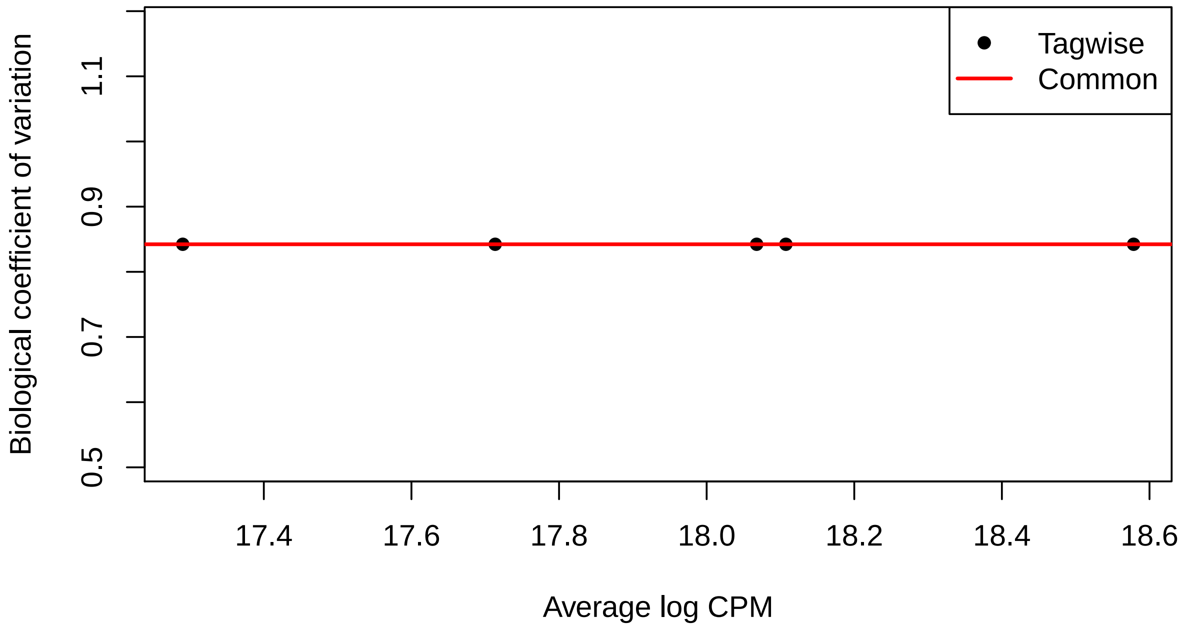
```
## Coefficient:  factor(age)young
##        logFC   logCPM         F    PValue       FDR
## 2 -1.5560264 18.83474 2.3452261 0.1296124 0.6329803
## 4 -0.9681608 18.12335 0.8526960 0.3585676 0.6329803
## 5 -0.5584090 18.26628 0.2964534 0.5876289 0.6329803
## 3  0.5480383 18.24759 0.2873902 0.5933853 0.6329803
## 1 -0.5122304 17.95164 0.2297996 0.6329803 0.6329803
```

```
######## For day42 only:
y.ab.day42 <- DGEList(abundances[, grepl(colnames(abundances),
    pattern = "d42")], samples = extra.info[grepl(colnames(abundances),
    pattern = "d42"), ])


# Filter out low abundance labels: Skipped as tends to filter
# out all labels (we know these are biologically meaningful
# clusters, so should not be filtered out on count alone).
# keep <- filterByExpr(y.ab, group=y.ab$samples$day42) y.ab
# <- y.ab[keep,] summary(keep)

design.day42 <- model.matrix(~factor(age), y.ab.day42$samples)

y.ab.day42 <- calcNormFactors(y.ab.day42, method = "TMMwsp")  # we need to normalise to 'library' size


y.ab.day42 <- estimateDisp(y.ab.day42, design.day42, trend = "none")
```

```
summary(y.ab.day42$common.dispersion)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.7094  0.7094  0.7094  0.7094  0.7094  0.7094
```

```
plotBCV(y.ab.day42, cex = 1)
```



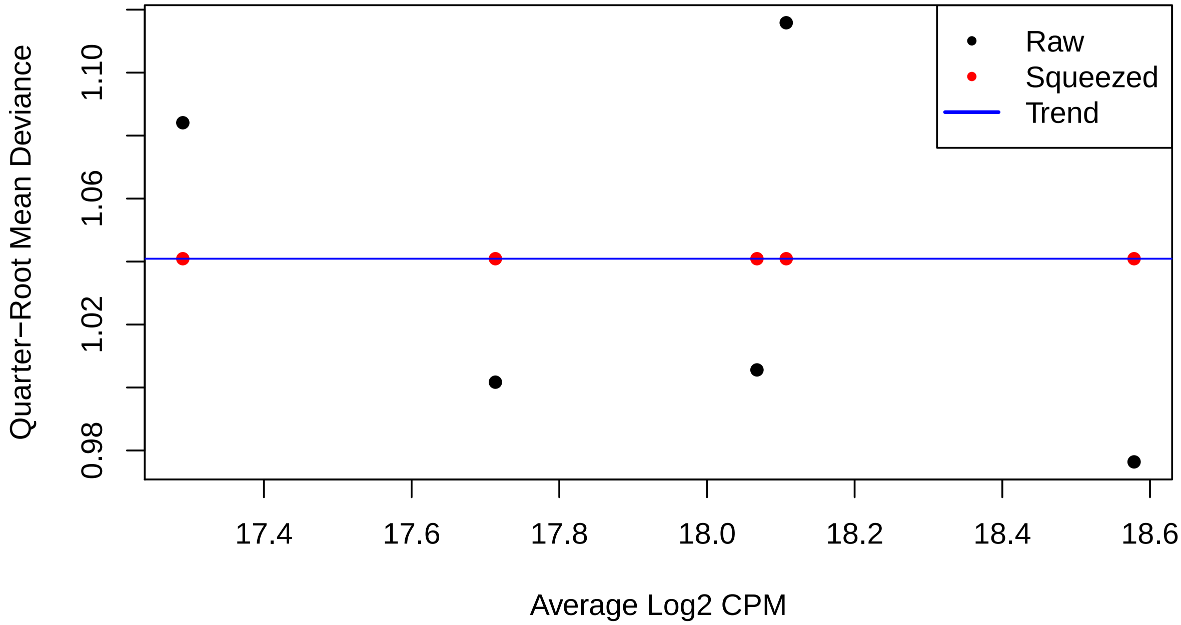
```
fit.ab.day42 <- glmQLFit(y.ab.day42, design.day42, robust = TRUE,
    abundance.trend = FALSE)
summary(fit.ab.day42$var.prior)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.174   1.174   1.174   1.174   1.174   1.174
```

```
summary(fit.ab.day42$df.prior)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     Inf     Inf     Inf     Inf     Inf     Inf
```

```
plotQLDisp(fit.ab.day42, cex = 1)
```



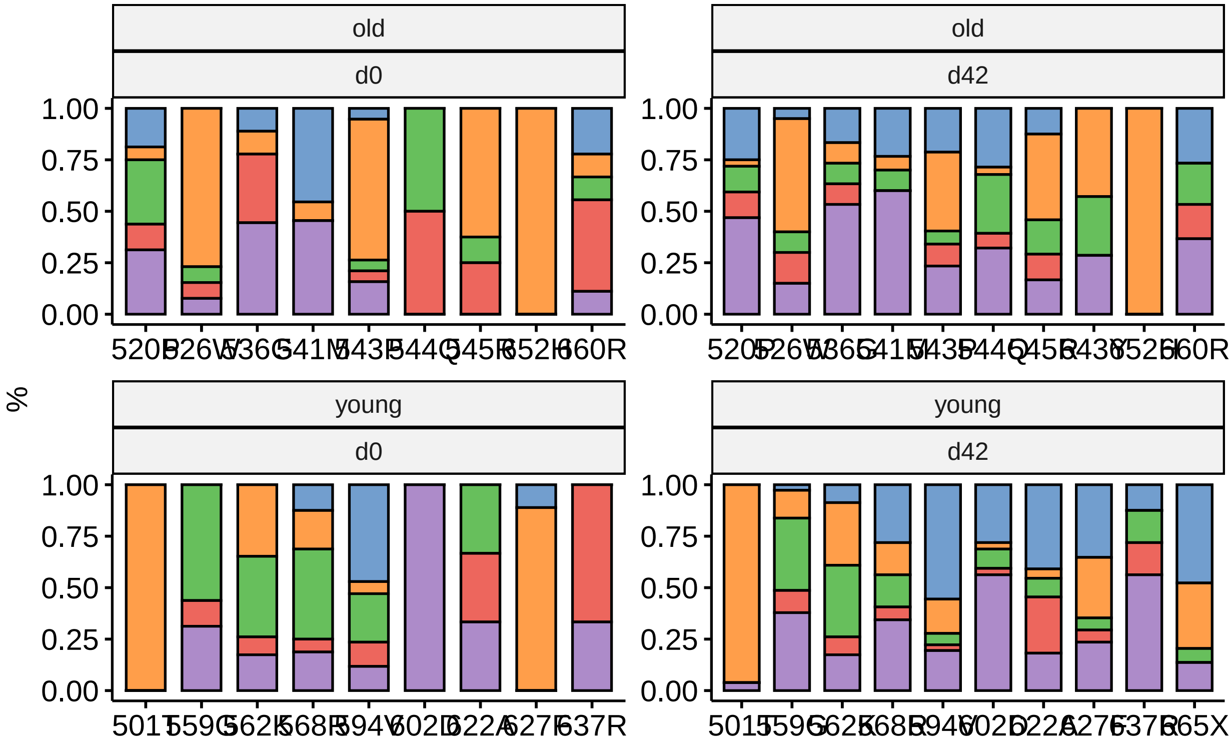
```r
res.day42 <- glmQLFTest(fit.ab.day42, coef = ncol(design.day42))
topTags(res.day42)
```

```
## Coefficient:  factor(age)young
##          logFC   logCPM          F    PValue       FDR
## 2 -0.77337580 18.10736 1.35949029 0.2467064 0.8226478
## 4 -0.53667767 17.29014 0.55182900 0.4595059 0.8226478
## 5 -0.43441434 18.57846 0.47253999 0.4935887 0.8226478
## 3 -0.29241599 17.71348 0.18472232 0.6683733 0.8354666
## 1  0.08863397 18.06773 0.01821122 0.8929536 0.8929536
```

Example plot - proportion of cells at each timepoint by individiual

See figure .Rmd for final version.

```r
library(ggpubr)

abundances.percent <- apply(abundances, MARGIN = 1, function(x) {
    x/colSums(abundances)
})

tableau10medium = c("#729ECE", "#FF9E4A", "#67BF5C", "#ED665D",
    "#AD8BC9", "#A8786E", "#ED97CA", "#A2A2A2", "#CDCC5D", "#6DCCDA")


abundances.percent <- reshape2::melt(abundances.percent)
abundances.percent$Var2 %<>% factor(.)
abundances.percent$age <- factor(ifelse(grepl(abundances.percent$Var1,
    pattern = "old"), "old", "young"))
abundances.percent$day <- factor(ifelse(grepl(abundances.percent$Var1,
    pattern = "d42"), "d42", "d0"))
abundances.percent$PID <- factor(gsub(pattern = ".*_", replacement = "",
    abundances.percent$Var1))

ggbarplot(data = abundances.percent, x = "PID", xlab = "", y = "value",
    ylab = "%", fill = "Var2", palette = tableau10medium, legend = "none") +
    facet_wrap(~age + day, scales = "free")
```



Save important objects

```
save(res, res.O, res.Y, res.day0, res.day42, abundances, abundances.percent,
    file = "data/DA_analysis_results.RData")
```