Analyzing paired count data using edgeR

Lizhong Chen

Smyth Lab, Bioinformatics Division

WEHI

September 30, 2025

A paired count object: PCList

```
?PCList

Create a PCList object

Description:

     Assembles a PCList object from its components, especially the paired counts as a couple matrix.

Usage:

     PCList(counts, counts2, samples = NULL, group = NULL, genes = NULL, ...)

Arguments:

  counts: numeric matrix containing sequence read counts, with rows corresponding to genes (genomic features) and columns
          to samples. Negative values or NAs are not allowed.

 counts2: numeric matrix containing sequence read counts, with rows corresponding to genes (genomic features) and columns
          to samples. Negative values or NAs are not allowed.

 samples: data.frame containing sample information, with a row for each sample. This data.frame will be appended to the
          samples component of the DGEList object.

   group: vector or factor giving the experimental group or treatment condition for each sample. Defaults to a single group.

   genes: data.frame containing gene annotation.

     ...: other arguments are not currently used.
```

Methylation data

- Each row represents the CpG loci site
- For each loci, we have a paired count, methylated Y_{gi} and unmethylated Z_{gi}
- The proportion of methylated signals at each loci is

$$X_{gi} = \frac{Y_{gi}}{Y_{gi} + Z_{gi}}$$

- The M-value is the base2 logit transformation of X_{gi}
- Test the difference of the M-values under various conditions

Methylation data

```
> y
An object of class "PCList"
$counts
              P6_1 P6_4 P7_2 P7_5 P8_3 P8_6
chr1-3020689   16   20    4   13    6    2
chr1-3020690   33   46   18   46   15   22
chr1-3020724   15   21    4   14    6    2
chr1-3020725   37   45   18   45   15   21
chr1-3020815    3   10    5    7    4    2
1745912 more rows ...

$samples
     group
P6_1     1
P6_4     1
P7_2     1
P7_5     1
P8_3     1
P8_6     1

> dim(y)
[1] 1745917       6
```

```
$counts2
              P6_1 P6_4 P7_2 P7_5 P8_3 P8_6
chr1-3020689    0    1    0    0    0    0
chr1-3020690    4    1    0    0    0    0
chr1-3020724    1    0    0    0    0    0
chr1-3020725    0    2    0    1    0    1
chr1-3020815    3    0    0    0    0    0
1745912 more rows ...

$genes
               Chr   Locus EntrezID Symbol Strand Distance  Width
chr1-3020689 chr1 3020689   497097   Xkr4      -  -721032 457017
chr1-3020690 chr1 3020690   497097   Xkr4      -  -721031 457017
chr1-3020724 chr1 3020724   497097   Xkr4      -  -720997 457017
chr1-3020725 chr1 3020725   497097   Xkr4      -  -720996 457017
chr1-3020815 chr1 3020815   497097   Xkr4      -  -720906 457017
1745912 more rows ...
```

F1-hybrids related data (F1 allelic analysis)

- Sequencing data from F1-hybrids (Usually from B6 and CAST mice)
- Each row represents the genomic feature (RNA-seq, ATAC-seq, HiC ...)
- For each feature, the paired count consists of Y_{gi} from the B6 and Z_{gi} from CAST
- The allelic ratio for each genomic feature is

$$X_{gi} = \frac{Y_{gi}}{Z_{gi}}$$

- Test the allelic ratio under various conditions or for each group

Binomial model

- To model the paired count data, we assume the conditional distribution

$$Y_{gi}|X_{gi} \sim \text{Binomial}(X_{gi}, \pi_{gi})$$

  where $X_{gi} = Y_{gi} + Z_{gi}$ is the total count

- We fit a logit regression

$$\log \frac{\pi_{gi}}{1-\pi_{gi}} = \mathbf{X}\boldsymbol{\beta}$$

  where $\mathbf{X}$ is the design matrix and $\boldsymbol{\beta}$ is the coefficient vector

- The quasi-dispersion is usually estimated by Pearson statistic (in `glm` function)

- The paired count Y_{gi} and Z_{gi} might be over-dispersed

Beta-binomial model

- We assume the prior probability

$$\pi_{gi} \sim \text{Beta}(\alpha_{gi}, \beta_{gi})$$

where α_{gi} and β_{gi} are the shape parameters

- The posterior probability is

$$P(Y_{gi}|X_{gi}, \alpha_{gi}, \beta_{gi}) = \binom{X_{gi}}{Y_{gi}} \frac{B(\alpha_{gi} + Y_{gi}, \beta_{gi} + Z_{gi})}{B(\alpha_{gi}, \beta_{gi})}$$

- The mean and dispersion are

$$\pi_{gi} = \frac{\alpha_{gi}}{\alpha_{gi} + \beta_{gi}} \quad \text{and} \quad \theta_{gi} = \frac{1}{\alpha_{gi} + \beta_{gi}}$$

Challenge of beta-binomial model

- The estimation (MLE) of the prior parameters α_{gi} and β_{gi} usually requires full likelihood and one might be biased
- Maximizing the full likelihood is performed with `optim` function
- The accuracy of the MLE by small samples is bad
- Shrinkage of dispersion estimation by EB method is difficult, maybe a similar way to `DESeq2` is possible
- Likelihood ratio test is widely used assuming the estimated dispersion is a constant

edgeR v3 approach to paired count

- It is based on the paired design approach
- The design matrix is created by `modelMatrixMeth`
- The library sizes are set to be the same for the paired counts
- No normalization is performed
- Following the same edgeR QL pipeline

Limitations of edgeR v3 approach

- It does not suit the edgeR QL pipeline perfectly
- It requires setting the library size manually
- It also requires filtering the data manually
- It is always a complex design and becomes slow with increase of sample size
- It maybe not be very powerful

edgeR v4 quasi-binomial model

- We fit the quasi-binomial model
- We estimate the quasi-dispersion by adjusted deviance statistics
- We apply empirical Bayes method to stabilize the quasi-dispersion estimation
- We perform the quasi-F test
- This approach is almost identical to current NB model

edgeR v4 QL pipeline

```
# edgeR v3 pipeline from user's guide
# read data
y <- readBismark2DGE(files, sample.names=targets$Sample)
# filtering
Methylation <- gl(2,1,ncol(y), labels=c("Me","Un"))
Me <- y$counts[, Methylation=="Me"]
Un <- y$counts[, Methylation=="Un"]
Coverage <- Me + Un
HasCoverage <- rowSums(Coverage >= 8) == 6
HasBoth     <- rowSums(Me) > 0 & rowSums(Un) > 0
y <- y[HasCoverage & HasBoth,, keep.lib.sizes=FALSE]
# correct library sizes
TotalLibSize <- 0.5 * y$samples$lib.size[Methylation=="Me"] +
              + 0.5 * y$samples$lib.size[Methylation=="Un"]
y$samples$lib.size <- rep(TotalLibSize, each=2)
# MDS plot
M <- log2(Me + 2) - log2(Un + 2)
plotMDS(M, label=Group, col=rep(2:4,each=2),
        main="MDS(Population")
# design matrix
designSL <- model.matrix(~0+Group, data=targets)
design   <- modelMatrixMeth(designSL)
# model fitting
fit   <- glmQLFit(y, design)
contr <- makeContrasts(Group60vs40 = Group60um - Group40um,
                       levels=design)
qlf   <- glmQLFTest(fit, contrast=contr)
```

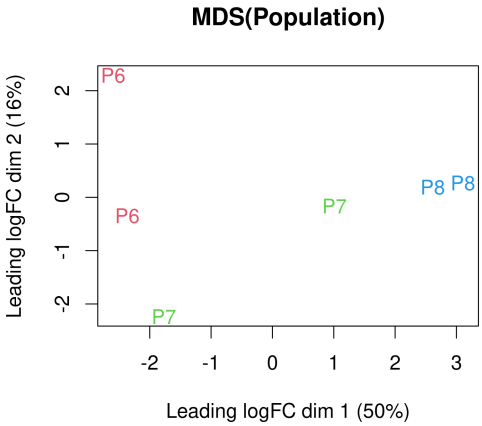
```
# edgeR v4 QL pipeline
# read data
y <- readBismark2PC(files, sample.names=targets$Sample)
# filtering
Coverage <- y$counts + y$counts2
keep <- rowSums(Coverage >= 3) == 6
y    <- y[keep,]
```

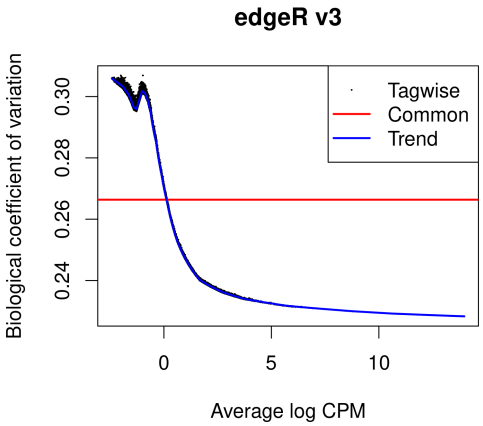
```
# MDS plot
plotMDS(y, label=Group, col=rep(2:4,each=2),
        main="MDS(Population")

# design matrix
design <- model.matrix(~0+Group, data=targets)

# model fitting
fit   <- glmQLFit(y, design)
contr <- makeContrasts(Group60vs40 = Group60um - Group40um,
                       levels=design)
qlf   <- glmQLFTest(fit, contrast=contr)
```

Case study: Methylation






```
> dim(y)
[1] 1396779       6
```

Model fitting

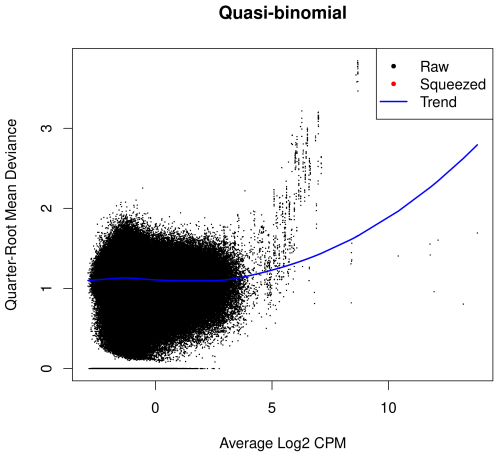
```
> des
  (Intercept) P7 P8
1           1  0  0
2           1  0  0
3           1  1  0
4           1  1  0
5           1  0  1
6           1  0  1
```

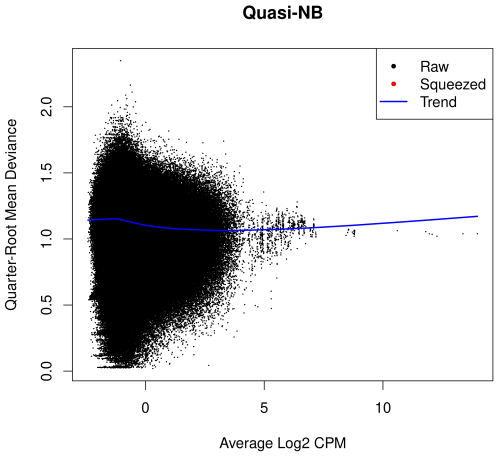
```
> system.time(fit <- glmQLFit(y, des, robust=FALSE))
   user  system elapsed
  9.199   0.140  10.129
> fit$df.prior
[1] 8134.8
> summary(fit$s2.prior)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.44    1.55    1.61    1.58    1.63   60.89
> summary(fit$df.residual.adj)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    2.18    2.99    3.11    4.28    8.33
```

```
> desM
   Sample1 Sample2 Sample3 Sample4 Sample5 Sample6 (Intercept) P7 P8
1        1       0       0       0       0       0           1  0  0
2        1       0       0       0       0       0           0  0  0
3        0       1       0       0       0       0           1  0  0
4        0       1       0       0       0       0           0  0  0
5        0       0       1       0       0       0           1  1  0
6        0       0       1       0       0       0           0  0  0
7        0       0       0       1       0       0           1  1  0
8        0       0       0       1       0       0           0  0  0
9        0       0       0       0       1       0           1  0  1
10       0       0       0       0       1       0           0  0  0
11       0       0       0       0       0       1           1  0  1
12       0       0       0       0       0       1           0  0  0
```

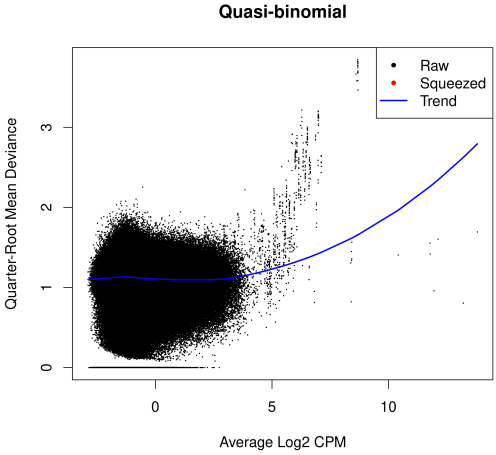
```
> system.time(fit3 <- glmQLFit(z, desM, robust=FALSE))
   user  system elapsed
105.734   0.319 114.845
> fit3$df.prior
[1] 8134.8
> summary(fit3$s2.prior)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.28    1.56    1.70    1.64    1.75    1.88
> summary(fit3$df.residual.adj)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    2.21    3.24    3.20    4.47    7.48
> fit3$dispersion
[1] 0.051639
```

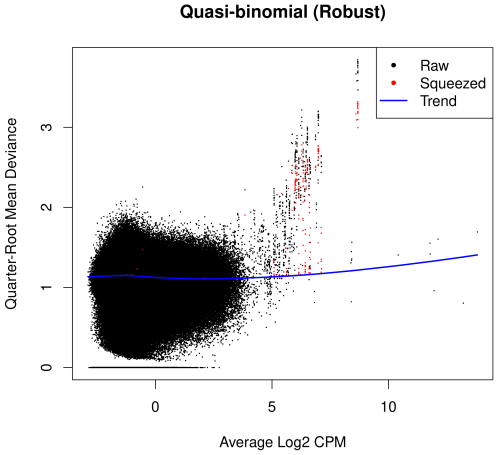
Quasi-dispersion plots





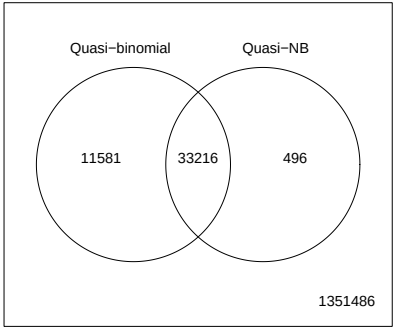
Robust empirical Bayes



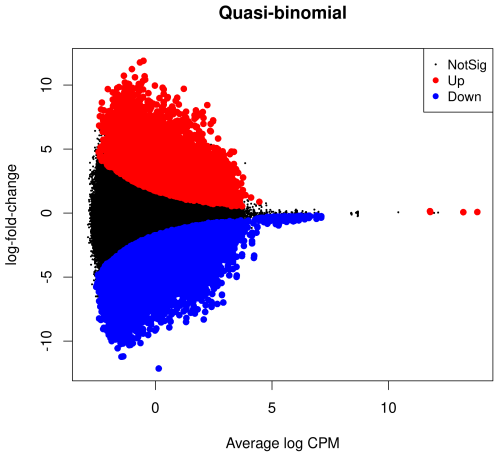


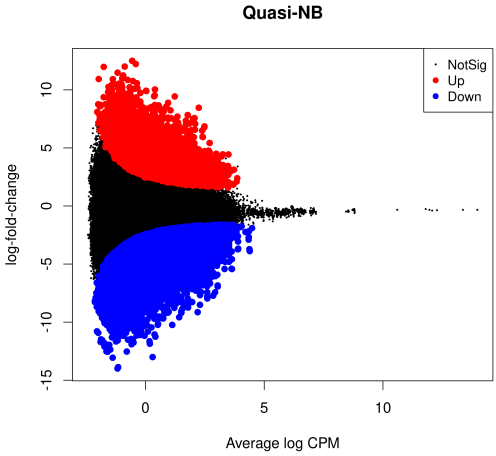
Hypothesis test: P6 vs P7

```
# Quasi-binomial
> summary(decideTests(qlf2))
       PopulationP7
Down          30877
NotSig      1351982
Up            13920
> topTags(qlf2)[,-(1:7)]
Coefficient:  P7
                logFC logCPM   F    PValue       FDR
chr6-23162321   -3.32   4.24 522 4.88e-112 6.82e-106
chr13-45709467  -7.23   2.41 489 3.50e-105  2.44e-99
chr15-58379194  -6.37   2.85 481 1.40e-103  6.53e-98
chr6-23162270   -3.50   4.22 453  6.86e-98  2.40e-92
chr13-45709480  -7.32   2.41 444  5.17e-96  1.44e-90
>
# Quasi-NB
> summary(decideTests(qlf3))
       PopulationP7
Down          27567
NotSig      1363067
Up             6145
> topTags(qlf3)[,-(1:7)]
Coefficient:  P7
                logFC logCPM   F   PValue      FDR
chr13-45709467  -7.79   2.60 167 7.29e-38 1.02e-31
chr17-46572098  -9.14   2.19 163 6.00e-37 4.19e-31
chr13-45709480  -7.96   2.60 161 1.51e-36 7.05e-31
chr10-100094048 -7.73   2.85 154 5.30e-35 1.52e-29
chr8-120068504  -7.67   2.30 154 5.43e-35 1.52e-29
```

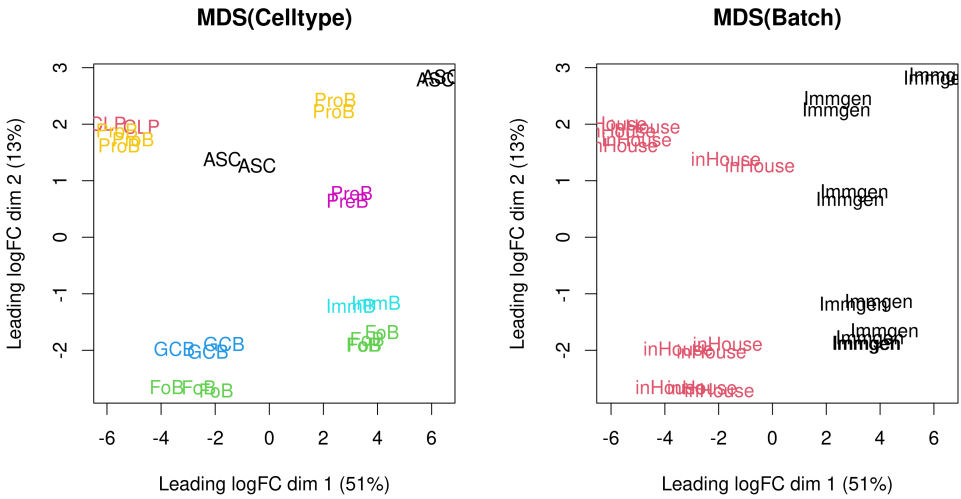


Hypothesis test: P6 vs P7

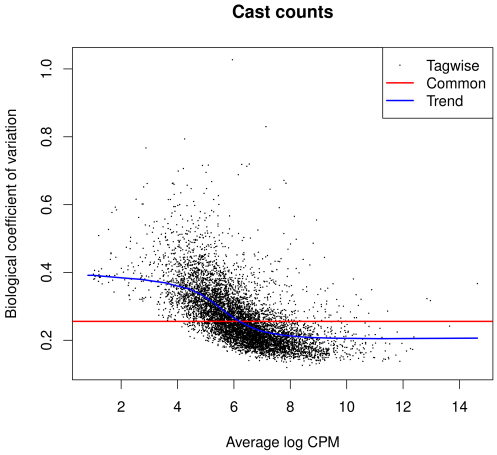


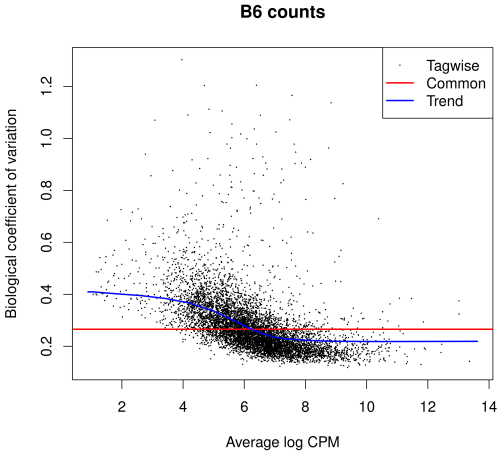


Case study: F1-hybrids data (RNA-seq)



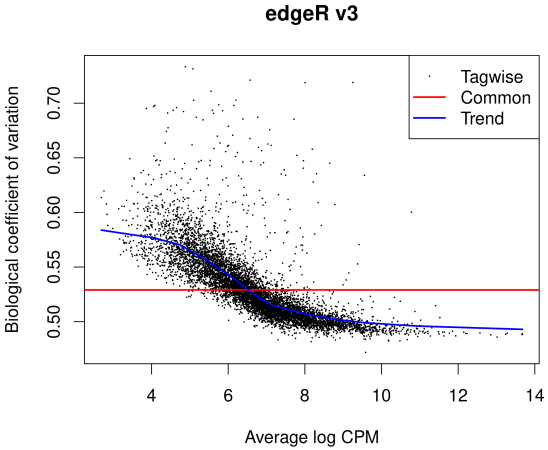
BCV plots for single count data



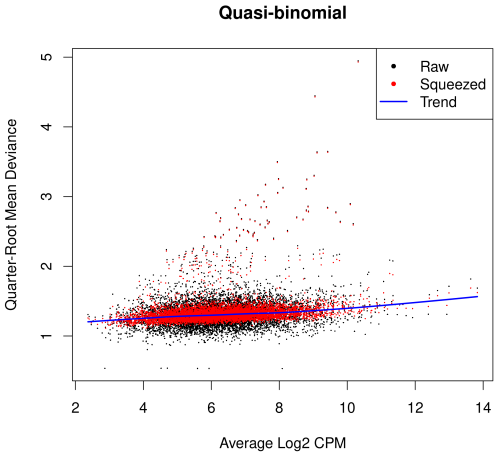


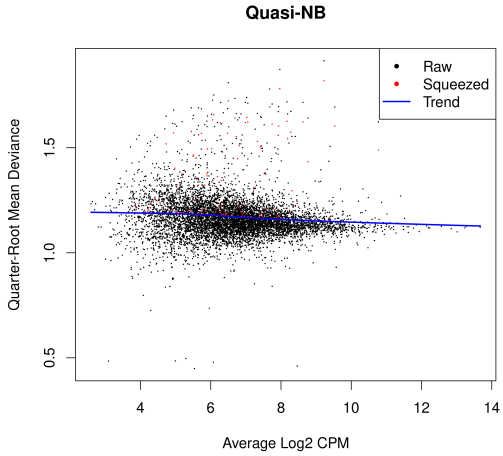
Model fitting

```
> dim(y)
[1] 7232   25
# Quasi-binomial
> system.time(fit <- glmQLFit(y, des, robust=TRUE))
   user  system elapsed
  0.458   0.000   0.471
> summary(fit$df.prior)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.24   21.04   21.04   20.60   21.04   21.04
> summary(fit$s2.prior)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   2.10    2.76    2.88    2.91    3.04    6.00
>
# Quasi-NB
> system.time(fitM <- glmQLFit(z, desM, robust=TRUE))
   user  system elapsed
 22.022   0.000  21.993
> summary(fitM$df.prior)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   4.59 8134.85 8134.85 7991.61 8134.85 8134.85
> summary(fitM$s2.prior)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.62    1.83    1.89    1.89    1.96    2.02
> fitM$dispersion
[1] 0.24133
```



Quasi-dispersion plots





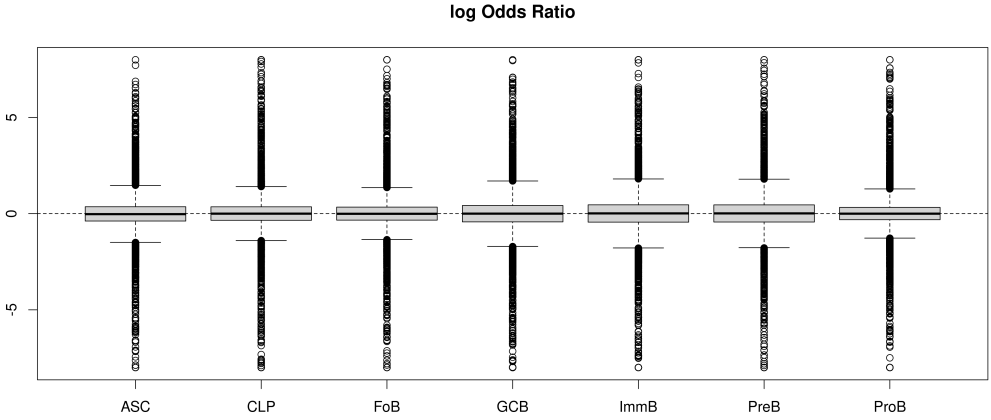
Hypothesis test: single group

```
# Quasi-binomial
> delist <- list()
> for(i in 1:ncol(des))){
+     delist[[i]] <- glmQLFTest(fit, coef = i)
+ }
> deindex  <- lapply(delist, decideTests)
> detable  <- lapply(deindex, summary)
> do.call(cbind, detable)[,-8]
          ASC  CLP  FoB  GCB ImmB PreB ProB
Down      461  439  853  516  512  552  680
NotSig 6209 6275 5315 6127 6119 5994 5710
Up        562  518 1064  589  601  686  842
>
#  We check how many shared genes among groups
> as.vector(colSums(y$counts)-colSums(y$counts2))
 [1]  77357 120599  86711  87616  61081  42342  53105  28500  45147  28693  49260  77421  26956
[14]  66933  26066  52475  47103  56851  40621  61472  31891  28579  57405  30488  28791
> ind <- rowSums(do.call(cbind, deindex))
> table(ind)

  -7   -6   -5   -4   -3   -2   -1    0    1    2    3    4    5    6    7
 155   71  122  123  130  216  494 4371  578  246  162  121  138   87  218
> head(rownames(y)[ind==7])
[1] "2610035D17Rik" "4833420G17Rik" "5830428M24Rik" "AI506816"      "Acp1"          "Actr3"
> head(rownames(y)[ind==-7])
[1] "5430405H02Rik" "Acad12"        "Actg1"         "Actr2"         "Akap13"        "Amd2"
```

```
# Quasi-NB
> delistM <- list()
> for(i in 1:ncol(des)){
+     delistM[[i]] <- glmQLFTest(fitM, coef = 25+i)
+ }
> deindexM  <- lapply(delistM, decideTests)
> detableM  <- lapply(deindexM, summary)
> do.call(cbind, detableM)[,-8]
          ASC  CLP  FoB  GCB ImmB PreB ProB
Down       75   18  194   53   95   91  120
NotSig 6867 7095 6474 7103 7000 6998 6853
Up        290  119  564   76  137  143  259
```

Log Odds for single group

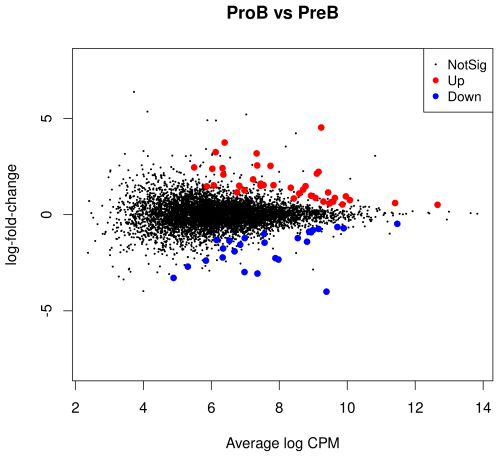


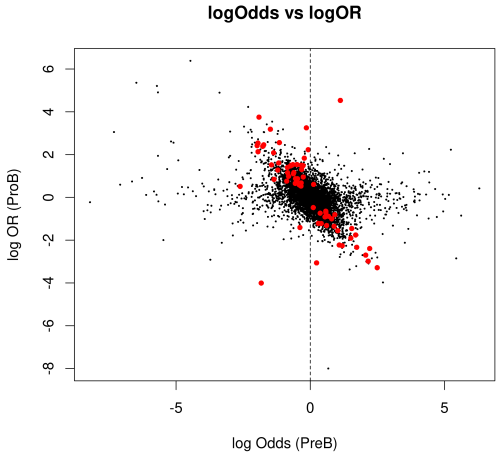
Hypothesis test: PreB vs ProB

```
# Quasi-binomial
> contr <- c(0,0,0,0,0,-1,1,0)
> system.time(qlf2 <- glmQLFTest(fit, contrast=contr))
   user  system elapsed
  0.117   0.000   0.117
> summary(decideTests(qlf2))
       -1*PreB 1*ProB
Down                26
NotSig            7166
Up                  40
> topTags(qlf2)
Coefficient:  -1*PreB 1*ProB
           logFC  logCPM      F      PValue        FDR
Lgals9   0.86990  9.6362 91.328 1.1762e-11 8.1798e-08
Cd38    -1.40789  8.8169 87.114 2.2621e-11 8.1798e-08
Xrcc6    1.53084  7.8294 70.538 3.4215e-10 7.3512e-07
Rogdi   -3.06258  7.3575 69.563 4.0659e-10 7.3512e-07
Rac2     0.60628 11.4088 56.557 4.8803e-09 7.0589e-06
Fermt3  -2.33403  7.9760 48.725 2.5999e-08 3.1337e-05
Ucp2    -0.47581 11.4728 42.968 9.8476e-08 1.0174e-04
AU020206  1.15514  9.4350 40.288 1.8941e-07 1.7123e-04
Desi1    1.53440  7.5367 36.849 4.5603e-07 3.6644e-04
Cyp4f18  2.22818  9.1561 36.421 5.6190e-07 4.0637e-04
```

```
# Quasi-NB
> contrM <- c(rep(0,25),contr)
> system.time(qlf2M <- glmQLFTest(fitM, contrast=contrM))
   user  system elapsed
  3.392   0.000   3.392
> summary(decideTests(qlf2M))
       -1*PreB 1*ProB
Down                0
NotSig           7232
Up                  0
> topTags(qlf2M)[,-(1:9)]
Coefficient:  -1*PreB 1*ProB
           logFC logCPM      F      PValue FDR
Piezo1   4.1726 6.8930 11.4784 0.00070744   1
Gimap7   5.2842 6.1346  9.4332 0.00213797   1
Coro2a   3.7682 6.5075  9.0516 0.00263272   1
Mettl27  3.5931 5.2851  8.9384 0.00280067   1
Camkmt   7.5549 3.7984  8.3451 0.00387742   1
Rogdi   -2.8982 7.8929  8.3439 0.00388005   1
Gimap4   4.0168 8.4603  8.3326 0.00390426   1
Gvin3    4.1602 9.2681  8.1283 0.00436899   1
Tor3a   -3.4336 7.2809  7.8453 0.00510727   1
Hip1     3.5998 6.0577  7.2070 0.00727685   1
```

Hypothesis test: PreB vs ProB





Comparison between methylation and F1-hybrids allelic data

- The generating process of counts are different
- The requirement of filtering should be less strict for F1-hybrids data
- It makes a lot of technical problems for binomial model
- The biological questions are more complicated for F1-hybrids data
- Maybe F1-hybrids specific results should be considered

Future work

- All new or updated functions should be well tested and documented
- `PCList` object and related helper functions
- `binFit`, `mbinOneWay`, `mbinIWLS` fitting functions
- `glmQLFit`, `glmQLFTest` QL pipeline functions
- `plotQLDisp`, `plotMD` visualization functions

Future work

- Testing log fold change relative to a threshold
- Introducing sample weights to QL pipeline similar to `limma`
- Explore the properties or structure of single-cell RNA-seq data
- Explore the properties or structure of transcripts data

Cell composition analysis

- Let Y_{ji} be the number of the type j cells and the sample i and $Z_i = \sum_j Y_{ji}$ be total number of cells of sample i. The cell proportion of type j is

$$\pi_{ji} = \frac{Y_{ji}}{Z_i}$$

- We test the change of cell proportion between groups
- It can come from the single-cell RNA-seq data after clustering
- It can come from the spatial transcript data after clustering

Multinomial model

- Let $(Y_{1i}, \cdots, Y_{Ji})$ be the cell number of sample i, we assume

$$(Y_{1i}, \cdots, Y_{Ji}) | Z_i \sim \text{Multinom}(Z_i, \pi_{1i}, \cdots, \pi_{Ji})$$

  where $\sum_j \pi_{ji} = 1$

- We assume the prior Dirichlet distribution

$$(\pi_{1i}, \cdots, \pi_{Ji}) \sim \text{Dir}(\alpha_{1i}, \cdots, \alpha_{Ji})$$

- Let $\alpha_{0i} = \sum_j \alpha_{ji}$ and $\pi_{ji} = \alpha_{ji}/\alpha_{0i}$, the mean and variance are

$$\mathbf{E}[Y_{ji}] = Z_i \pi_{ji} \quad \text{and} \quad \mathbf{Var}[Y_{ji}] = \left(1 + \frac{Z_i - 1}{1 + \alpha_{0i}}\right) \times Z_i \pi_{ji}(1 - \pi_{ji})$$

Multinomial model

- Let X be the design matrix and choose the last cell type J as the reference cell type. For $j = 1, \cdots, J-1$, we have

$$\log \boldsymbol{\pi}_j = X\boldsymbol{\beta}_j$$

where $\boldsymbol{\pi}_j = (\pi_{j1}, \cdots, \beta_{jI})^T$, $\boldsymbol{\beta}_j = (\beta_{j1}, \cdots, \beta_{jp})^T$

- Write the coefficient matrix as $\boldsymbol{\beta} = [\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_{J-1}]$ of dimension $p \times (J-1)$. Then

$$\log [\boldsymbol{\pi}_1, \cdots, \boldsymbol{\pi}_{J-1}] = X\boldsymbol{\beta}$$

where $\boldsymbol{\pi}_J = 1 - \sum_{j=1}^{J-1} \boldsymbol{\pi}_j$

- The multinomial model is fitted by `multinom` from the package `nnet`

Application to cell composition data

- We fit the multinomial model for both full and null model using `multinom`
- We break it into single quasi-binomial model and calculate the deviance and `logOR`
- We estimate the quasi-dispersion using adjusted deviance statistics
- We apply empirical Bayes method to stabilized the quasi-dispersion estimate
- We perform a quasi-F test for each cell type

Case study

```
> head(t(counts)[,1:10])
       [,1]  [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
Clst_0 2562  2391 1765 2614 6103 2846 3245 2940 2217  5474
Clst_1 1093 16628  629  933 2036 2796 3929 1540 1089   947
Clst_2  316   659  118  286  793  515  684  574  468   472
Clst_3 1471  1708  889 1246 2609 1381 1680 1257  932  1705
Clst_4  663   724    0    1    9    4    2    2    1     2
Clst_5 1286  1299  974 1866 4477 2021 2630 1730 1446  1976
>
> design <- model.matrix(~ 0 + status)
> contr  <- makeContrasts(MARG_vs_NONI = MARG - NONI,
                          levels = design)
>
> clf    <- diffComposition(t(counts), design, contrast=contr)
> names(clf)
 [1] "counts"          "fitted.values"   "deviance"        "deviance.adj"
 [5] "df.residual"     "df.residual.adj" "df.prior"        "s2.post"
 [9] "s2.prior"        "table"
> clf$df.prior
[1] 2.042112
> clf$s2.prior
[1] 160.9391
> clf$s2.post
 [1]  678.83425  758.33969  261.17489  275.04895  415.84236  424.38202
 [7] 1687.78308  521.61091   49.63230  700.80769  205.91202  329.32410
[13]   38.24316  203.91890  119.55747  173.70278  789.30518   54.42218
```

```
> clf$table
              logOR          F      PValue       FDR
Clst_0 -0.160670319 0.611824340 0.43778734 0.7163793
Clst_1  0.019207460 0.008514539 0.92684855 0.9505895
Clst_2 -0.401099694 2.270247242 0.13816603 0.4415724
Clst_3 -0.040627414 0.066068086 0.79820404 0.9505895
Clst_4  0.985599749 2.580566669 0.11447641 0.4415724
Clst_5 -0.235073315 1.715501256 0.19625441 0.4415724
Clst_6  0.705699240 3.164247709 0.08134164 0.4415724
Clst_7 -0.032593023 0.032204383 0.85830449 0.9505895
Clst_8 -0.144595362 0.432205296 0.51392154 0.7708823
Clst_9 -0.253110135 1.743839551 0.19266091 0.4415724
Clst_10 -0.025843126 0.056750189 0.81268094 0.9505895
Clst_11 -0.136601071 0.710976742 0.40313218 0.7163793
Clst_12 -0.415625359 1.740560070 0.19307347 0.4415724
Clst_13 -0.163174301 1.936118424 0.17024563 0.4415724
Clst_14 -0.005049756 0.003878560 0.95058946 0.9505895
Clst_15  0.162279181 1.526146737 0.22246017 0.4449203
Clst_16  0.353535785 3.118761764 0.08349558 0.4415724
Clst_17  0.047388368 0.142779520 0.70713134 0.9505895
```

Potential to differential transcript usage (DTU) analysis

- For those transcripts from the same gene, the usage analysis is the same
- We estimate the overall quasi-dispersion for genes, not for transcripts
- We apply EB method to stabilized the gene-wise quasi-dispersion estimate
- We perform a quasi-F test for each transcript
- The essential idea behind is different from `diffSplice`
- It should be faster than `diffSplice` because of the complex design of null hypothesis for `diffSplice`, especially when the number of transcripts increases

Acknowledgement

Smyth Lab

Gordon Smyth

Mengbo Li

Hannah Coughlan

Waruni Abeysekera

Shama Deb

Chen Lab

Andy Chen

Nutt Lab

Junli Nie

WEHI
brighter together

Australian Government
National Health and
Medical Research Council

NHMRC

Thank you

WEHI_research WEHIresearch WEHImovies WEHI_research Walter and Eliza Hall Institute