Genomic imprinting analysis (Pignatta Ler-Col samples)

- this is an example for ColxLer and LerxCol samples from Pignatta et al. (2014) with diagnostic plots
- runs GLM analysis based on edgeR to identify statistically significantly imprinted genes for RNA-seq samples of a reciprocal F1 cross
- assumes that allelic count tables are located in the same directory with file names "Counts_Alleles_SRRxxxxxxx" where xxxxxxx is the SRR sample ID
- *NOTE* this script is not fully generalized - several things in and are hard-coded and have to be adapted for different datasets

usage: Rscript run_edgeR_LerCol.R

2016-08-14 Stefan Wyder

```
library("edgeR")
```

```
## Loading required package: limma
```

```
library("doBy")
library("ggplot2")


cross <- "Pignata_Ler_Col"  # used for file names of diagnostic plots
fdr_cutoff <- 0.05          # FDR cut-off 5%
# Col x Ler samples, all other samples are assumed to be from reciprocal cross Ler x Col
ColxLer_Samples <- c("SRR1039929","SRR1508239","SRR1508241")
```

Functions

```
#' Merges allelic count tables in the current folder (output of Classify_Alleles.py)
#'
#' @param FileNameStart The name start of allelic count files to be merged
#' @param ClipString Part of file names to be clipped from column/sample names
#'
#' @return A data frame with allelic counts for all samples, gene names as row names
#' @author Stefan Wyder
#'
multiMerge <- function(FileNameStart, ClipString){
  filenames <- list.files(path=".", pattern=paste0(FileNameStart, "*"))
  shortnames <- gsub(ClipString, "", filenames)
  datalist <- lapply(filenames, function(x) {read.csv(file=x, header=T, sep="\t")})
  for (i in 1:length(filenames)) {
    names(datalist[[i]]) <- c("Gene", paste0("ref_", shortnames[i]), paste0("nef_", shortnames[i]))
  }
  merged <- Reduce(function(x,y) {merge(x,y, by="Gene", all.x=TRUE)}, datalist)
  row.names(merged) <- merged$Gene
  merged$Gene <- NULL
  return(merged)
}
```

load and prepare data

```
# merges all count files in current directory named 'Counts_Alleles_SRRxxxxxxx'
#  clips "Counts_Alleles_" from samples names in merged count table
counts_summed <- multiMerge("Counts_Alleles_SRR", "Counts_Alleles_")
# missing genes have 0 counts
counts_summed[is.na(counts_summed)] <- 0

# Assign maternal and paternal depending on cross direction
# ref: reference allele Col, nef: non-reference allele Ler
Assign_mat_pat_CxL <- function(x) {
    ifelse(grepl("ref_", x), sub("ref_", "mat_", x), sub("nef_", "pat_", x))
}
Assign_mat_pat_LxC <- function(x) {
    ifelse(grepl("nef_", x), sub("nef_", "mat_", x), sub("ref_", "pat_", x))
}
is_ColxLer_Sample <- sapply(colnames(counts_summed),
    FUN=function(x) {strsplit(x, fixed=T, split="_")[[1]][2]}) %in% ColxLer_Samples
colnames(counts_summed)[is_ColxLer_Sample] <- sapply(
    colnames(counts_summed)[is_ColxLer_Sample], Assign_mat_pat_CxL)
colnames(counts_summed)[!is_ColxLer_Sample] <- sapply(
    colnames(counts_summed)[!is_ColxLer_Sample], Assign_mat_pat_LxC)

print(dim(counts_summed))
```

```
## [1] 15264    14
```

```
# filter out genes with less than 10 counts overall
counts_summed_over10 <- counts_summed[rowSums(counts_summed) >= 10, ]
print(dim(counts_summed_over10))
```

```
## [1] 14367    14
```

```
######################
# create design matrix
######################

design <- data.frame(row.names=colnames(counts_summed_over10),
                     mother=ifelse(is_ColxLer_Sample, "Col", "Ler"),
                     type=ifelse(grepl("mat_", colnames(counts_summed_over10)),"mother","father"),
                     cross=ifelse(is_ColxLer_Sample, "1","2"))
edgeR.design <- model.matrix(~design$cross + design$type)
print(colnames(edgeR.design))
```

```
## [1] "(Intercept)"       "design$cross2"     "design$typemother"
```

```
print(edgeR.design)
```

```
##    (Intercept) design$cross2 design$typemother
## 1            1             1                 0
```

```
## 2            1             1                 1
## 3            1             0                 1
## 4            1             0                 0
## 5            1             1                 0
## 6            1             1                 1
## 7            1             0                 1
## 8            1             0                 0
## 9            1             1                 0
## 10           1             1                 1
## 11           1             0                 1
## 12           1             0                 0
## 13           1             1                 0
## 14           1             1                 1
## attr(,"assign")
## [1] 0 1 2
## attr(,"contrasts")
## attr(,"contrasts")$`design$cross`
## [1] "contr.treatment"
##
## attr(,"contrasts")$`design$type`
## [1] "contr.treatment"
```

run edgeR

```
edgeR <- DGEList(counts=counts_summed_over10, genes=row.names(counts_summed_over10))
print(head(counts_summed_over10, 5))
```

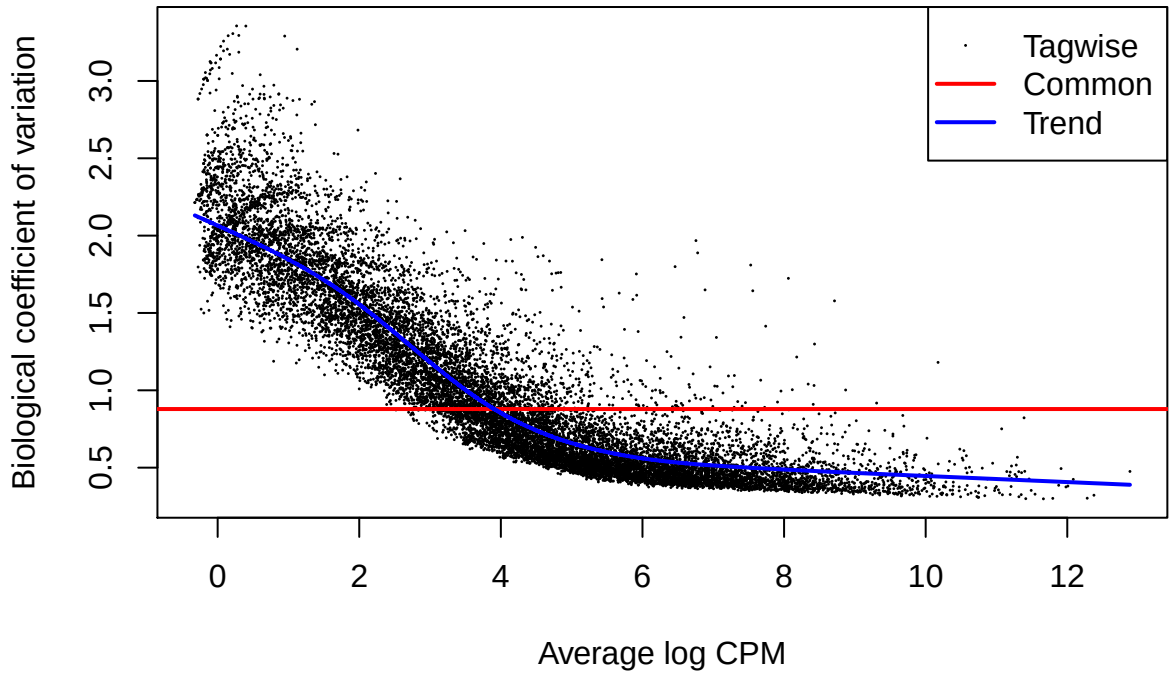
```
##           pat_SRR1039928 mat_SRR1039928 mat_SRR1039929 pat_SRR1039929
## AT1G01150              0              2              9              4
## AT1G01225             33             68            134             64
## AT1G01230              7              9             26             13
## AT1G01240              4             16             42              9
## AT1G01250              2             10              0              0
##           pat_SRR1508238 mat_SRR1508238 mat_SRR1508239 pat_SRR1508239
## AT1G01150              0              0              0              0
## AT1G01225             82            155            185             82
## AT1G01230              0              0              0             23
## AT1G01240             34             99             41             31
## AT1G01250              1              1              0              0
##           pat_SRR1508240 mat_SRR1508240 mat_SRR1508241 pat_SRR1508241
## AT1G01150              0              0              0              0
## AT1G01225            199            318            301             66
## AT1G01230              0              0             23              0
## AT1G01240             37            228             27             79
## AT1G01250             38            270             25              0
##           pat_TEST_SRR1039928 mat_TEST_SRR1039928
## AT1G01150                   0                   2
## AT1G01225                  33                  68
## AT1G01230                   7                   9
## AT1G01240                   4                  16
## AT1G01250                   2                  10
```

```
edgeR <- calcNormFactors(edgeR)
print(edgeR$samples$norm.factors)
```

```
##  [1] 0.9874812 0.9528513 0.9283708 0.9717485 1.1685469 1.0663293 0.9872314
##  [8] 1.0959780 1.0442605 0.9671975 0.9177490 1.0018351 0.9875151 0.9528498
```

```
edgeR <- estimateGLMCommonDisp(edgeR, edgeR.design)
edgeR <- estimateGLMTrendedDisp(edgeR, edgeR.design)
edgeR <- estimateGLMTagwiseDisp(edgeR, edgeR.design)

plotBCV(edgeR)
```

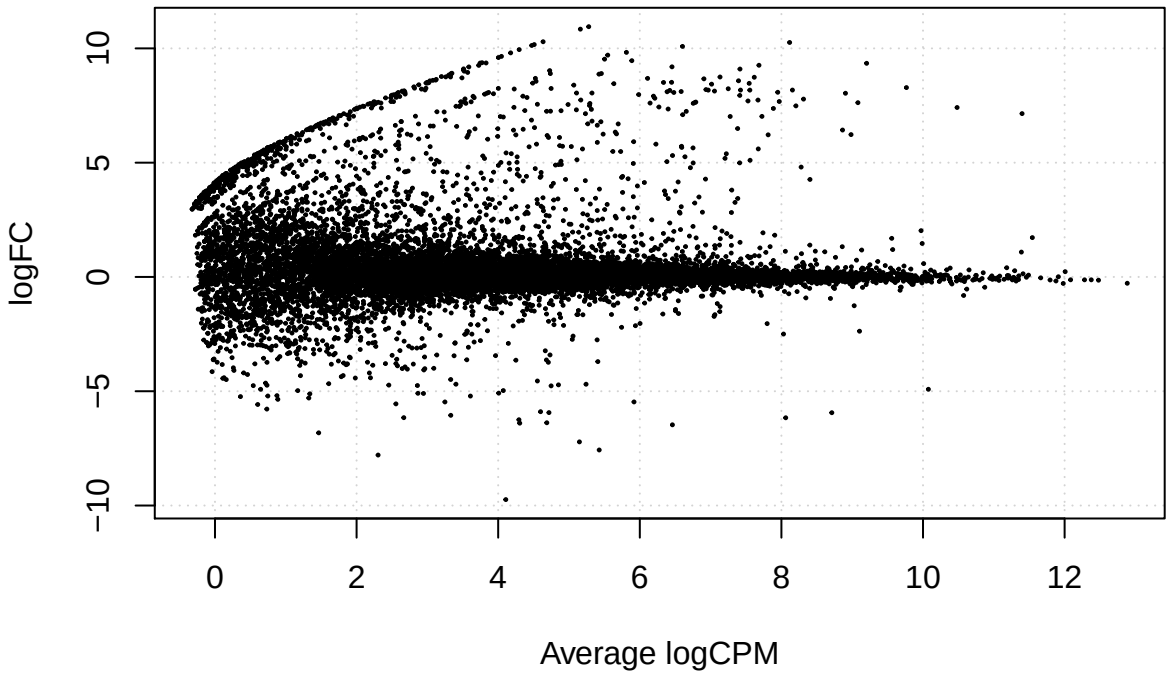



```
print(paste0("edgeR common dispersion: ", edgeR$common.dispersion))
```

```
## [1] "edgeR common dispersion: 0.772531447579586"
```

```
edgeR.fit <- glmFit(edgeR, edgeR.design)
edgeR.lrt <- glmLRT(edgeR.fit, coef="design$typemother")

plotSmear(edgeR.lrt)
```




```
# print number of MEGs and PEGs
#-1: PEGs, 0:non-significant, +1: MEGs
print(summary(de <- decideTestsDGE(edgeR.lrt, p=fdr_cutoff, adjust="BH")))
```

```
##    design$typemother
## -1                54
## 0              13740
## 1                573
```

```
print(topTags(edgeR.lrt))
```

```
## Coefficient:  design$typemother
##           genes    logFC    logCPM       LR       PValue          FDR
## 2490  AT1G61720 8.284372  9.766254 309.0855 3.454995e-69 4.963792e-65
## 2044  AT1G51500 9.261511  7.683510 270.4327 9.132763e-61 6.560520e-57
## 3018  AT1G70830 8.035965  8.905242 254.3157 2.975774e-57 1.425098e-53
## 12288 AT5G26120 8.174521  7.554425 250.3008 2.232845e-56 8.019820e-53
## 10089 AT4G25960 7.649995  6.791271 246.1637 1.781637e-55 5.119355e-52
## 4558  AT2G25450 8.176909  8.155996 244.8296 3.480819e-55 8.334822e-52
## 11327 AT5G08260 9.192497  6.455011 229.7812 6.653600e-52 1.365604e-48
## 1387  AT1G28590 8.667474  6.906001 222.0471 3.234922e-50 5.809516e-47
## 9246  AT4G12960 7.145847 11.399389 219.3365 1.262162e-49 2.014831e-46
## 2574  AT1G62990 8.509056  6.335765 212.0351 4.941776e-48 7.099850e-45
```

```
print("Raw Counts of top hits")
```

```
## [1] "Raw Counts of top hits"
```

```
print(edgeR$counts[as.integer(rownames(topTags(edgeR.lrt))),])
```

```
##           pat_SRR1039928 mat_SRR1039928 mat_SRR1039929 pat_SRR1039929
## AT1G61720              3           3250          20287              21
## AT1G51500              1           1000           4627               3
## AT1G70830              6           1960           6747              11
## AT5G26120              3            626           3789               9
## AT4G25960              1            707            996               6
## AT2G25450              2           1028           7745              15
## AT5G08260              0            591           1275               2
## AT1G28590              0            309           1546               5
## AT4G12960             79          21334          38721             180
## AT1G62990              0            412           1833               3
##           pat_SRR1508238 mat_SRR1508238 mat_SRR1508239 pat_SRR1508239
## AT1G61720             13           7409          11261              31
## AT1G51500              1           2068           2724               1
## AT1G70830              1           3674           5873               7
## AT5G26120              0           1313           2283               2
## AT4G25960              3           1092            852               1
## AT2G25450              1           1166           3981              13
## AT5G08260              0            353            923               1
## AT1G28590              1            646           1769               1
## AT4G12960             57          24319          22998              14
## AT1G62990              2            434            895               1
##           pat_SRR1508240 mat_SRR1508240 mat_SRR1508241 pat_SRR1508241
## AT1G61720              2           1996           9233              16
## AT1G51500              1            844            973               1
## AT1G70830              0           3139           6279              23
## AT5G26120              0            760           2623               1
## AT4G25960              3           1128            942               0
## AT2G25450              0           1203           2741               0
## AT5G08260              0            523            944               1
## AT1G28590              0           1172           1764               2
## AT4G12960             52          14237          15716              59
## AT1G62990              0            282            821               0
##           pat_TEST_SRR1039928 mat_TEST_SRR1039928
## AT1G61720                   3                3250
## AT1G51500                   1                1000
## AT1G70830                   6                1960
## AT5G26120                   3                 626
## AT4G25960                   1                 707
## AT2G25450                   2                1028
## AT5G08260                   0                 591
## AT1G28590                   0                 309
## AT4G12960                  79               21334
## AT1G62990                   0                 412
```

```
print("=====================================================================")
```

```
## [1] "====================================================================="
```

```r
print("Normalized Counts of top hits")
```

```
## [1] "Normalized Counts of top hits"
```

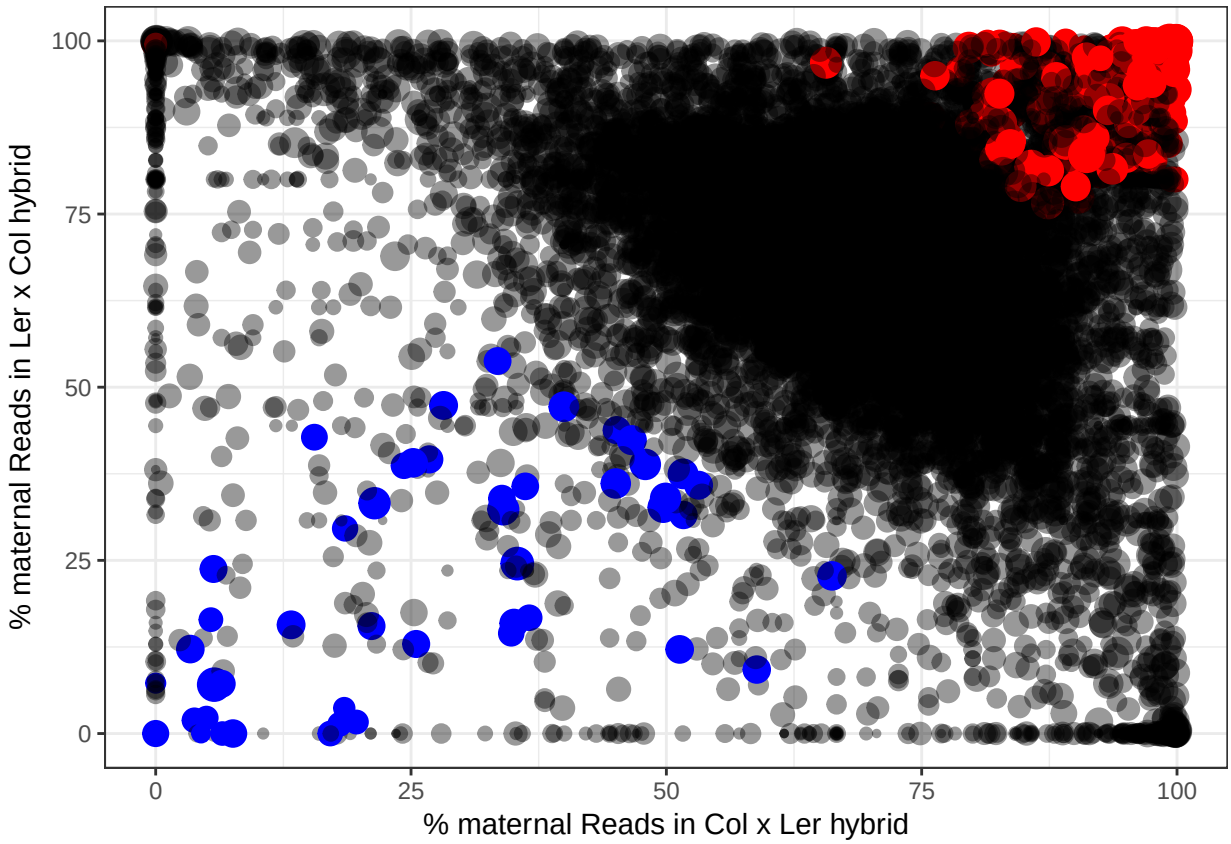
```r
print(cpm(edgeR$counts[as.integer(rownames(topTags(edgeR.lrt))),]))
```

```
##           pat_SRR1039928 mat_SRR1039928 mat_SRR1039929 pat_SRR1039929
## AT1G61720       31578.95     104109.940      231676.68      82352.941
## AT1G51500       10526.32      32033.828       52840.14      11764.706
## AT1G70830       63157.89      62786.302       77050.45      43137.255
## AT5G26120       31578.95      20053.176       43270.22      35294.118
## AT4G25960       10526.32      22647.916       11374.28      23529.412
## AT2G25450       21052.63      32930.775       88447.57      58823.529
## AT5G08260           0.00      18931.992       14560.45       7843.137
## AT1G28590           0.00       9898.453       17655.25      19607.843
## AT4G12960      831578.95     683409.681      442192.18     705882.353
## AT1G62990           0.00      13197.937       20932.78      11764.706
##           pat_SRR1508238 mat_SRR1508238 mat_SRR1508239 pat_SRR1508239
## AT1G61720      164556.96     174436.126      210254.11      430555.56
## AT1G51500       12658.23      48688.610       50859.80       13888.89
## AT1G70830       12658.23      86499.976      109654.77       97222.22
## AT5G26120           0.00      30913.029       42625.89       27777.78
## AT4G25960       37974.68      25709.846       15907.69       13888.89
## AT2G25450       12658.23      27452.088       74329.24      180555.56
## AT5G08260           0.00       8310.967       17233.33       13888.89
## AT1G28590       12658.23      15209.305       33029.00       13888.89
## AT4G12960      721518.99     572562.038      429395.62      194444.44
## AT1G62990       25316.46      10218.016       16710.54       13888.89
##           pat_SRR1508240 mat_SRR1508240 mat_SRR1508241 pat_SRR1508241
## AT1G61720       34482.76       78943.21      219645.07     155339.806
## AT1G51500       17241.38       33380.79       23146.83       9708.738
## AT1G70830           0.00      124149.66      149371.97     223300.971
## AT5G26120           0.00       30058.54       62398.90       9708.738
## AT4G25960       51724.14       44613.19       22409.36          0.000
## AT2G25450           0.00       47579.50       65206.01          0.000
## AT5G08260           0.00       20685.02       22456.94       9708.738
## AT1G28590           0.00       46353.43       41964.03      19417.476
## AT4G12960      896551.72      563083.37      373870.02     572815.534
## AT1G62990           0.00       11153.30       19530.88          0.000
##           pat_TEST_SRR1039928 mat_TEST_SRR1039928
## AT1G61720            31578.95           104109.940
## AT1G51500            10526.32            32033.828
## AT1G70830            63157.89            62786.302
## AT5G26120            31578.95            20053.176
## AT4G25960            10526.32            22647.916
## AT2G25450            21052.63            32930.775
## AT5G08260                0.00            18931.992
## AT1G28590                0.00             9898.453
## AT4G12960           831578.95           683409.681
## AT1G62990                0.00            13197.937
```

Allelic Imbalance Plot

```
# Sum up counts of biological replicates
counts_summarized <- data.frame(Gene=row.names(counts_summed_over10))
counts_summarized$mat_CxL <- rowSums(
    counts_summed_over10[, is_ColxLer_Sample & grepl("mat_", colnames(counts_summed_over10))])
counts_summarized$pat_CxL <- rowSums(
    counts_summed_over10[, is_ColxLer_Sample & grepl("pat_", colnames(counts_summed_over10))])
counts_summarized$pat_LxC <- rowSums(
    counts_summed_over10[, !is_ColxLer_Sample & grepl("pat_", colnames(counts_summed_over10))])
counts_summarized$mat_LxC <- rowSums(
    counts_summed_over10[, !is_ColxLer_Sample & grepl("mat_", colnames(counts_summed_over10))])
counts_summarized$CxL_total <- with(counts_summarized, mat_CxL + pat_CxL)
counts_summarized$LxC_total <- with(counts_summarized, pat_LxC + mat_LxC)

# allelic imbalance plot, significantly imprinted genes (PEGs in blue, MEGS in red)
# we add 0.5 counts to divisor to prevent division by zero
ggplot(data=counts_summarized, aes(
            x=mat_CxL/(CxL_total+.5)*100,
            y=mat_LxC/(LxC_total+.5)*100,
            size=log10(CxL_total+LxC_total),
            color=as.factor(de))) +
  geom_point(alpha=c(1,0.4,1)[as.factor(de)]) +
  theme_bw() +
  scale_color_manual(values=c("blue","black","red")) +
  theme(legend.position = "none") +
  xlab("% maternal Reads in Col x Ler hybrid") +
  ylab("% maternal Reads in Ler x Col hybrid")
```


% maternal Reads in Ler x Col hybrid
100
75
50
25
0
0
25
50
75
100
% maternal Reads in Col x Ler hybrid