clrDV vignette

Hongxiang Li and Tsung Fei Khang

12 April 2023

Introduction

This guide provides on an overview of the R package *clrDV*, a statistical methodology for identifying genes that show differential variability (DV) between two conditions. *clrDV* is based on a compositional data analysis (CoDA) framework. The skew-normal distribution with centered parameters is used to model gene-wise null distribution of centered log-ratio (CLR) transformed RNA-Seq data. The main function for running DV test is `clrDV()`.

Installation

Install *clrDV* from GitHub:

```
library(devtools)
install_github("Divo-Lee/clrDV")
```

Getting Started

Load the *clrDV* package:

```
library(clrDV)
```

Differential Variability Analysis

We first provide an example of performing DV test on a simulated dataset of CLR-transformed RNA-Seq counts, `clrCounts2`. This dataset contains 1000 genes, with the first 100 genes exhibiting differential variability. Each group has a sample size of 200 (control vs. case). First, we load `clrCounts2`:

```
data("clrCounts2")
# 1000 genes, 200 samples per group, differential variability for the first 100 genes,
# CLR-transformed counts table
dim(clrCounts2)
#> [1] 1000  400
clrCounts2[1:5, c(1:3, 201:203)]
#>         control1   control2   control3       case1     case2     case3
#> gene1 0.03244982  1.2652100  1.3987549 -6.67691562 -3.257939  2.182924
#> gene2 0.17267916 -1.1125588  1.1066562 -1.72108856 -2.564792 -2.657446
#> gene3 3.12573804 -0.6980028 -1.4365760 -0.06219002  2.285168  2.176272
#> gene4 1.48277854 -0.3693177  0.4163379  3.88936299  3.480840  0.846851
#> gene5 0.13068826  2.4335490  0.9431369 -6.67691562  2.768495  1.558259
```

Each row represents a gene, and each column represents a sample.

Now we can apply `clrDV()` to perform a DV test. Note that `clrDV()` does not perform the CLR-transformation itself; the CLR-transformed counts must be provided as input. Thus:

```
group2 = c(rep(0,200), rep(1,200))
clrDV_result <- clrDV(data = clrCounts2, group = group2)
head(clrDV_result, 5)
#>               DV        se         z         pval     adj_pval   sigma1
#> gene1  0.8184170 0.1585811  5.160873 2.458013e-07 1.978429e-05 1.546710
#> gene2  1.2275625 0.1931241  6.356340 2.066173e-10 2.008607e-08 1.561095
#> gene3 -0.8568991 0.0901118 -9.509288 1.919711e-21 6.531793e-19 1.485030
#> gene4  1.6866450 0.1923920  8.766711 1.839605e-18 3.721705e-16 1.523563
#> gene5  1.9487495 0.1957370  9.955957 2.375273e-23 1.616367e-20 1.350196
#>       se.sigma1 z.sigma1     p.sigma1    sigma2  se.sigma2 z.sigma2
#> gene1 0.08484662 18.22948 3.011780e-74 2.3651267 0.13397397 17.65363
#> gene2 0.08742686 17.85601 2.595393e-71 2.7886574 0.17220182 16.19412
#> gene3 0.08339285 17.80765 6.164764e-71 0.6281314 0.03414336 18.39688
#> gene4 0.08491930 17.94130 5.612629e-72 3.2102076 0.17263657 18.59518
#> gene5 0.07653587 17.64135 1.185868e-69 3.2989458 0.18015341 18.31187
#>           p.sigma2
#> gene1 9.542430e-70
#> gene2 5.548320e-59
#> gene3 1.391392e-75
#> gene4 3.515526e-77
#> gene5 6.654501e-75
tail(clrDV_result, 5)
#>                   DV        se          z      pval adj_pval   sigma1
#> gene996  -0.06696298 0.1107142 -0.6048272 0.5452939        1 1.444097
#> gene997  -0.04657379 0.1137911 -0.4092920 0.6823254        1 1.461016
#> gene998  -0.18443986 0.1178862 -1.5645592 0.1176863        1 1.592520
#> gene999   0.09552836 0.1294366  0.7380323 0.4604948        1 1.584475
#> gene1000  0.08027653 0.1351108  0.5941534 0.5524095        1 1.651214
#>           se.sigma1 z.sigma1     p.sigma1   sigma2  se.sigma2 z.sigma2
#> gene996  0.08084520 17.86250 2.310686e-71 1.377134 0.07564191 18.20597
#> gene997  0.08313581 17.57385 3.907620e-69 1.414442 0.07769719 18.20455
#> gene998  0.08838318 18.01836 1.398249e-72 1.408080 0.07800999 18.05000
#> gene999  0.08680371 18.25354 1.939401e-74 1.680003 0.09601531 17.49724
#> gene1000 0.09326542 17.70447 3.873381e-70 1.731491 0.09775729 17.71214
#>               p.sigma2
#> gene996  4.628012e-74
#> gene997  4.749704e-74
#> gene998  7.889388e-73
#> gene999  1.503835e-68
#> gene1000 3.379677e-70
```

Genes with `adj_pval < 0.05` are flagged as showing statistically significant differential variability.

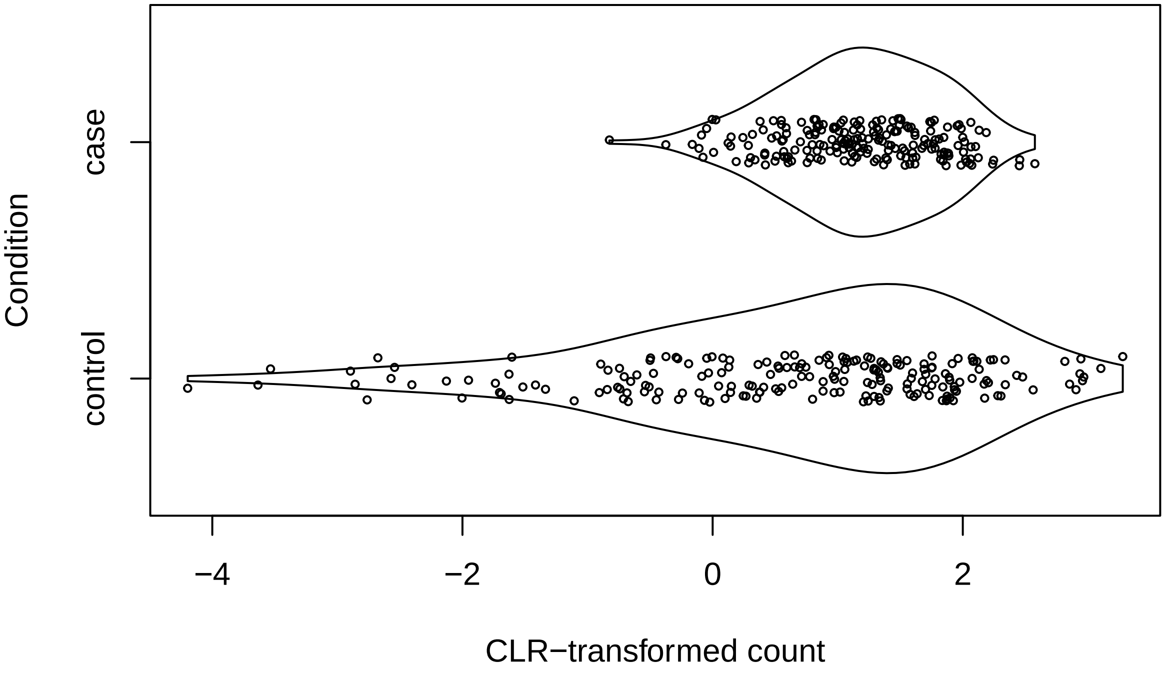
```
sum(clrDV_result$adj_pval < 0.05) # DV genes called
#> [1] 101
sum(clrDV_result$adj_pval[1:100] < 0.05) # true DV genes called
#> [1] 98
# observed FDR = (101-98)/101 = 0.0297; probability of Type II Error = (100-98)/100 = 0.02
```

The function `top.DV.genes()` extracts top-ranked DV genes, ranked using the SD ratio (case vs. control) of the CLR-transformed counts. Here, the `top="abs"` argument ranks the genes by $|\log_2(SD\_ratio)|$, that is, $|LFC|$.

```
top.DV.genes(clrDV_result, top = "abs", n = 10)
#>                 DV         se          z         pval     adj_pval  SD_ratio
#> gene31  2.0560046 0.19884127  10.339929 4.648830e-25 8.699671e-22 2.6755625
#> gene94 -1.0018582 0.09449064 -10.602724 2.894256e-26 2.135929e-22 0.3751806
#> gene14 -0.8974986 0.08515929 -10.539056 5.706867e-26 2.135929e-22 0.3790775
#> gene41  2.1142863 0.20659007  10.234211 1.393261e-24 1.738203e-21 2.6111755
#> gene27 -0.9947079 0.09755600 -10.196276 2.060201e-24 1.927697e-21 0.3857555
#> gene8  -0.9541758 0.09306702 -10.252566 1.152432e-24 1.725299e-21 0.3986518
#> gene20 -0.9722673 0.09532761 -10.199220 1.998709e-24 1.927697e-21 0.4001926
#> gene29  1.8643515 0.19181759   9.719398 2.492505e-22 1.373064e-19 2.4867692
#> gene67  2.0280983 0.21198653   9.567109 1.099368e-21 4.114642e-19 2.4819493
#> gene28 -0.9659782 0.09621564 -10.039721 1.019618e-23 8.480352e-21 0.4054308
#>              LFC
#> gene31  1.419842
#> gene94 -1.414343
#> gene14 -1.399435
#> gene41  1.384699
#> gene27 -1.374241
#> gene8  -1.326799
#> gene20 -1.321233
#> gene29  1.314273
#> gene67  1.311474
#> gene28 -1.302472
```

We can use `violin.plot.clrDV()` to produce violin plots for graphically inspecting the variance of the distribution of CLR-transformed count between two groups. These plots are useful for checking that the computational results are reasonable. The violin plots in the figure below show an example of a gene that has significant DV. For gene 10, the control group has significantly larger variance ($sigma1 = 1.415998$) than the case group ($sigma2 = 0.6185685$), as shown by the skew to negative values.

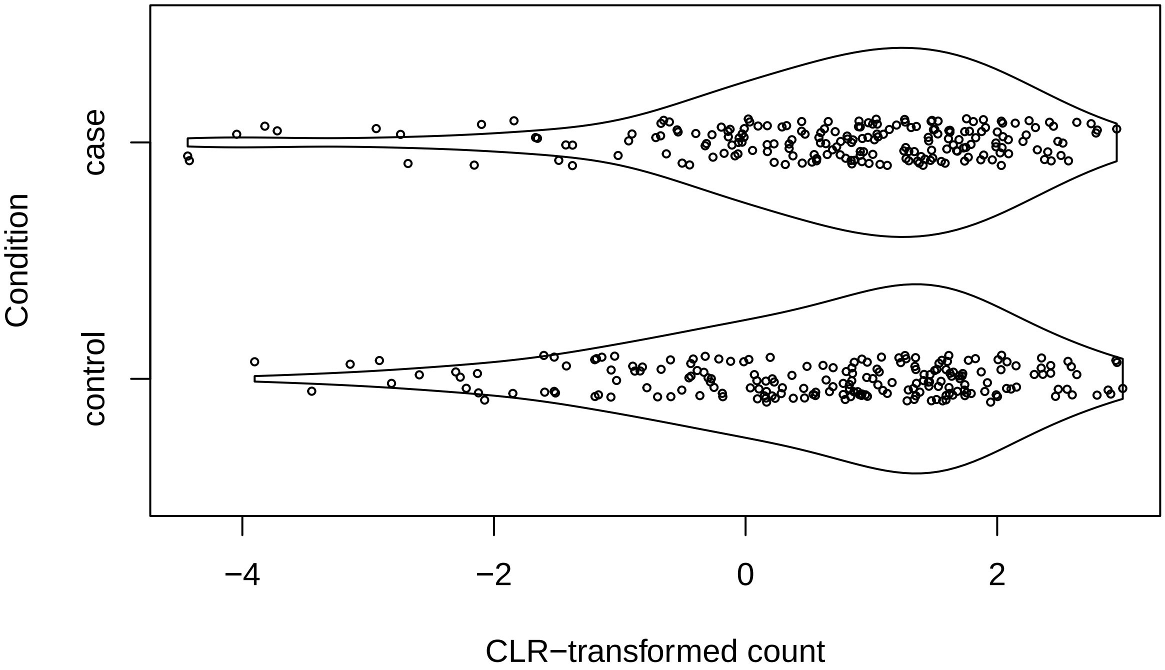
```
violin.plot.clrDV(data = clrCounts2, "gene10",
                  group = group2,
                  group.names = c("control", "case"))
```



```
clrDV_result[10, ]
#>                  DV         se         z         pval     adj_pval   sigma1
#> gene10 -0.7974292 0.08493435 -9.388772 6.070407e-21 1.747687e-18 1.415998
#>          se.sigma1 z.sigma1     p.sigma1    sigma2  se.sigma2 z.sigma2
#> gene10 0.07819351 18.10889 2.711615e-73 0.6185685 0.03316049 18.65378
#>             p.sigma2
#> gene10 1.176523e-77
```

The figure below shows an example of a gene that is not significant for DV. For gene 150, violin plots show that the spread of the values for both groups is about the same (*sigma1* = 1.37357, *sigma2* = 1.323281).

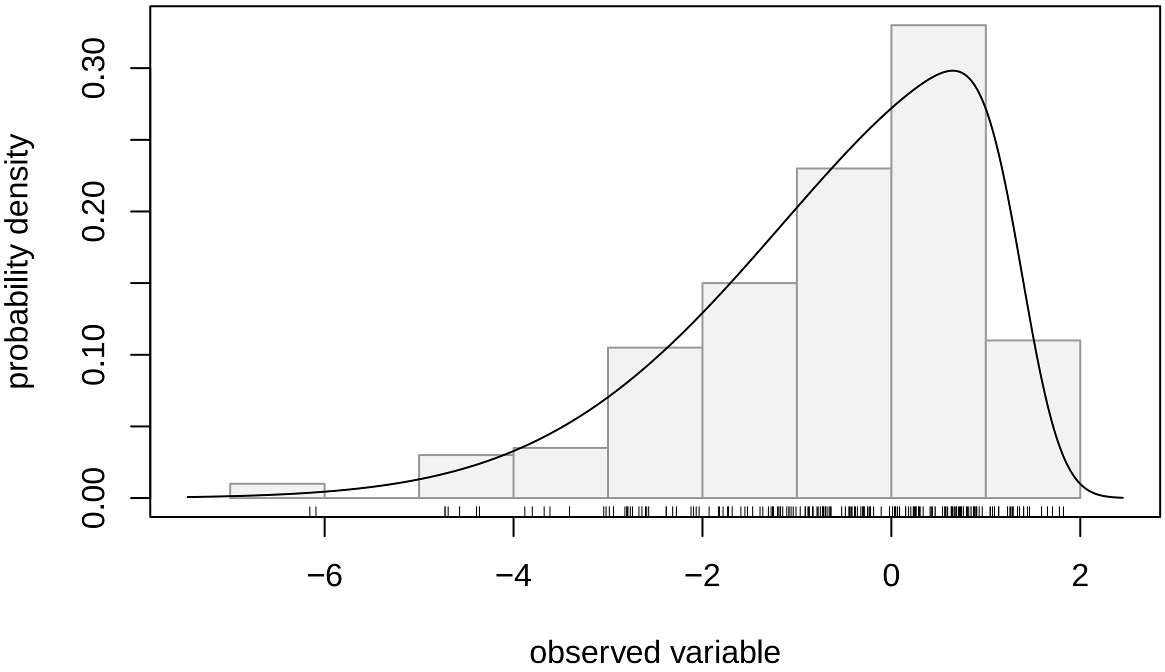
```
violin.plot.clrDV(data = clrCounts2, "gene150",
                  group = group2,
                  group.names = c("control", "case"))
```



```
clrDV_result[150, ]
#>                   DV        se          z      pval adj_pval  sigma1  se.sigma1
#> gene150 -0.0502891 0.1059458 -0.4746683 0.6350234        1 1.37357 0.07630776
#>          z.sigma1     p.sigma1   sigma2  se.sigma2 z.sigma2     p.sigma2
#> gene150  18.0004 1.934142e-72 1.323281 0.07349581 18.00485 1.784798e-72
```

The function `SN.plot()` produces a histogram of observed CLR-transformed counts, along with the fitted skew-normal probability density function for a particular gene/transcript. It can be used to graphically check how well the skew-normal distribution fits the data.

```
SN.plot(clrCounts2[1, 1:200])
```



```
clr.SN.fit(clrCounts2[1, 1:200])
#>              mu          se.mu           z.mu           p.mu          sigma
#>  -6.207045e-01   1.064579e-01  -5.830513e+00   5.525710e-09   1.546710e+00
#>        se.sigma        z.sigma        p.sigma          gamma      se.gamma.
#>   8.484662e-02   1.822948e+01   3.011780e-74  -9.268736e-01   3.420058e-02
#>         z.gamma        p.gamma
#>  -2.710111e+01  9.555226e-162
```

The Kolmogorov-Smirnov (KS) test can be used to compare the distribution of CLR-transformed counts of a particular gene with a skew-normal distribution. For illustration, we apply the KS test to the control group of `clrCounts2` to evaluate whether the distribution of CLR-transformed counts in this group and the skew-normal distribution are statistically similar. The distribution of p-values obtained form KS-tests conducted on all 1000 genes in the control group indicates that for 995 out of 1000 genes (99.5%), the skew-normal model fits the CLR-transformed count data well.

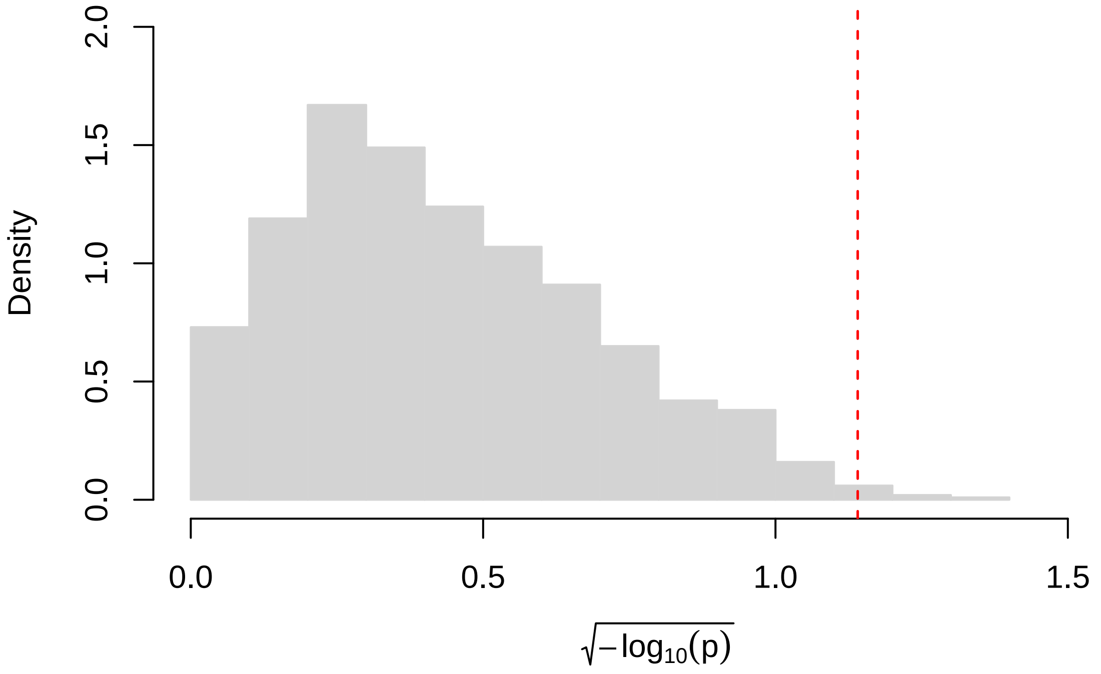
```
library(sn) # R package for skew-normal distribution and related distributions
cp_to_dp <- function(mean=NULL, sd=NULL, skewness=NULL){
  b <- sqrt(2/pi)
  if(skewness >= 0){
    r <- (2*skewness/(4-pi))^(1/3)
  } else {
    r <- -(2*(- skewness)/(4-pi))^(1/3)
  }
  alpha <- r/sqrt(2/pi - (1-2/pi)*r^2)
  delta <- alpha/sqrt(1 + alpha^2)
  omega <- sd/sqrt(1 - (b^2)*delta^2)
  xi <- mean - b*omega*delta
  return(c(xi, omega, alpha))
} # map centered parameters to direct parameters

control_clr_SN_fit <- clr.SN.fit(clrCounts2[, 1:200]) # MLE, control group
control_sn_CP <- control_clr_SN_fit[, c("mu", "sigma", "gamma")]
```

```
control_sn_DP <- matrix(NA, nrow = dim(control_sn_CP)[1], ncol = 3)
for (i in 1:dim(control_sn_CP)[1]) {
  control_sn_DP[i,] <- c(cp_to_dp(control_sn_CP[i,1],
                                  control_sn_CP[i,2],
                                  control_sn_CP[i,3]))
}
colnames(control_sn_DP) <- c("xi", "omega", "alpha") # direct parameters
control_sn_DP <- as.data.frame(control_sn_DP)

KS_test_pvalue <- vector()
for (i in 1:dim(control_sn_CP)[1]) {
  ks <- ks.test(clrCounts2[i, 1:200],
                "psn",
                xi = control_sn_DP$xi[i],
                omega = control_sn_DP$omega[i],
                alpha = control_sn_DP$alpha[i])
  KS_test_pvalue[i] <- ks$p.value
}
sum(KS_test_pvalue < 0.05)
#> [1] 5
# the number of genes where the skew-normal distribution fit is poor
# (1000-5)/1000 = 99.5%
# 99.5% of the genes in the control group are well fitted by skew-normal distribution

hist(sqrt(-log10(KS_test_pvalue)), freq = F, breaks = 15,
     main = NULL, border = "grey83",
     xlim = c(0, 1.6), ylim = c(0, 2),
     xlab = expression(sqrt(-log[10](p))))
abline(v = sqrt(-log10(0.05)), lty = 2, lwd = 1.25, col = "red")
```



```
# the dashed red line represents the p-value of 0.05
```