CNVtools

Vincent Plagnol and Chris Barnes

October 17, 2016

Contents

1 What CNVtools is meant to do

1.1 Input data

CNVtools is a R package meant to perform robust case-control and quantitative trait association testing of copy number variants (CNVs). The association testing procedure for CNVtools requires a one-dimensional normalized data summary per sample. Another vector is required to provide the case-control status or, alternatively, a quantitative trait. Another optional input vector is used to separate the samples in batches in situations where one expects the signal distribution to differentially affect each batch because of potential technical artifacts. This issue of differential bias is discussed in details in the companion publication published in Nature Genetics in 2008 (see Reference at the end of this vignette).

In addition, CNVtools fits a mixture model to the one-dimensional CNV data summary. For this approach to make sense, it is essential that the distinct components can at least be distinguished in the histogram of the normalized signal. If this is not the case mixture models are not an modeling appropriate tool and CNVtools cannot be used to analyze such data.

Note that no code is provided in this package for CNV signal normalization. However, is it frequent for multiple CNV probes located in the same chromosome region to provide information about the same CNV. It is therefore useful to combine the information across a small number of CNV probes to obtain a one-dimensional signal for each sample. In fact, this summary step is required to use the CNVtools routines which need a one-dimensional data summary as input. This data summary step is the purpose of the functions apply.pca and apply.ldf which suggest two methods to summarize CNV data across multiple probes: principal component and canonical correlation analysis. Unlike the actual association testing routines (like CNVtest.binary) and the underlying C code, these two functions are not essential to CNVtools but we thought it would be convenient to include them.

1.2 Genome-wide or single CNV data

Unlike a R package like "snpMatrix" specifically designed for genome-wide data, the CNVtools functions are implemented to analyze each CNV separately. To be more explicit, we did not design data structure to store and analyze data from multiple CNVs (for example genome-wide). However, this package was initially designed to analyze CNV data generated by the Wellcome Trust Case Control Consortium, which performed a genome-wide scan for association for eight common disorders (approximately 19,000 samples). Therefore CNVtools is well suited for large scale CNV association studies but it is the userś task to write a wrapper in order to apply the CNVtools code separately for each CNV. To give some explicit computation time: for the WTCCC CNV association study, and using recently purchased computers (as of 2009, a mixture of Dual-Core AMD Opteron Processor 2220 and Quad-Core AMD Opteron Processor 2384), CNVtools could analyze approximately 3,000 CNVs typed in 19,000 individuals for an approximate total time of 20 days of computing (6h using a 80 nodes computing cluster).

2 First look at the data

We first load an example CNV data set, called A112, in the two WTCCC control groups (1958 British Birth cohort and National Blood Services). The data required for CNVtools is a matrix of normalized signal intensities. In this example each row represents an individual and each column represents a locus within the CNV. To get a feel for the data, we plot the histograms for the mean intensity as well as the first principal component analysis (Figure **??**).

```
> #source("../CNVtools.r"); dyn.load("../../src/CNVtools.so"); load("../../CNVtools/data/A112.RData")
> library(CNVtools)
> data(A112)
> head(A112)

          subject cohort        SNP0       SNP1       SNP2       SNP3
1 WTCCC01-11474A1    58C -0.12647400 -0.1214220 -0.1423570  0.0449446
2 WTCCC01-11474A2    58C -0.21574200  0.0265778 -0.0964269  0.0617480
3 WTCCC01-11474A3    58C -0.00150499  0.0820076 -0.2853430  0.1589580
4 WTCCC01-11474A4    58C -0.05538290 -0.1691450 -0.0592800  0.0264289
5 WTCCC01-11474A5    58C -0.12926900  0.2014540 -0.8474870 -0.2647420
6 WTCCC01-11474A6    58C -0.06209860  0.1826130  0.1245160 -0.1731720
        SNP4       SNP5       SNP6        SNP7        SNP8       SNP9
1  0.0259435  0.1351870  0.0746991  0.40581000 -0.18601600  0.0990579
2  0.1521360 -0.0445652 -0.3751110 -0.39122600  0.10114500  0.1816270
3  0.0320422  0.1823220  0.0699921  0.29014900  0.00885492 -0.0387201
4 -0.0208353 -0.2740840  0.0310302  0.20566300  0.12842100 -0.2219500
5 -0.0502723 -0.2150250 -0.2254730  0.00162372  0.08069250  0.0562238
6 -0.0870918 -0.0902743 -0.0634414 -0.80391700  0.37845800 -0.1880560
       SNP10      SNP11      SNP12       SNP13     SNP14      SNP15      SNP16
1 -0.1969750  0.0448241 -0.0193997  0.13117800 -0.163383  0.1545760  0.0253607
2  0.0688791 -0.1166620  0.0217019 -0.05719720 -0.138044 -0.0554405 -0.0536655
3  0.1131100  0.0609800  0.2402140  0.23635400 -0.111235  0.5082330  0.0272966
4 -0.2299260  0.0198905 -0.3210060  0.14955900 -0.534339 -0.7596830 -0.1940050
5 -0.0636589 -0.0433160 -0.5579070  0.13913000 -0.778225 -0.9224910  0.0343805
6 -0.1368910 -0.0779523  0.1212290  0.00857489  0.179257 -0.0675581 -0.1812210
       SNP17      SNP18     SNP19      SNP20      SNP21      SNP22      SNP23
1 -0.0560689 -0.0751385 -0.485160 -0.0288187 -0.1945410 -0.0456346  0.0929479
2  0.1212250  0.1018410 -0.200404 -0.1797650 -0.0456029  0.2835270 -0.0813351
3 -0.1532160 -0.1135340  0.183407 -0.0960403  0.1230410 -0.1076840 -0.0180287
4 -0.1035420  0.1661650 -0.318173 -0.7149550 -0.7436040 -0.2483910 -0.2552810
5 -0.0130905  0.1538550 -0.589194 -0.4773230 -0.6345150  0.1788480 -0.4428020
6 -0.1676740  0.3261350 -0.199970  0.0908316  0.1268390  0.1787620  0.1138070
      SNP24     SNP25      SNP26       SNP27      SNP28        SNP29      SNP30
1 -0.222375 -0.368043 -0.1448800 -0.00706918  0.0356588 -0.346104000 -0.1318280
2 -0.143908 -0.105819 -0.2330800 -0.07807670  0.0980952 -0.152811000  0.0728393
3  0.401502  0.240364 -0.1334340 -0.00942116 -0.0514102 -0.254315000  0.0708932
4 -0.106774  0.203908  0.3008000 -0.24017000  0.1681400  0.298436000  0.1303020
5 -0.124996 -0.191220 -0.1863940 -0.08408520 -0.2589270 -0.000203031 -0.0516899
6 -0.228779 -0.409863  0.0208064  0.01472170  0.2187790 -0.384239000 -0.0265277
       SNP31      SNP32
1  0.0140277  0.0583939
2 -0.2176910  0.0172098
3 -0.1251580 -0.1050300
4 -0.0139433  0.0432413
5  0.0381275 -0.0992932
6 -0.2410430 -0.0577618

> raw.signal <- as.matrix(A112[, -c(1,2)])
> dimnames(raw.signal)[[1]] <- A112$subject
> mean.signal <- apply(raw.signal, MAR=1, FUN=mean)
> pca.signal <- apply.pca(raw.signal)
> pdf("fig/mean_pca_signal.pdf", width=10, height=5)
> par(mfrow=c(1,2))
> hist(mean.signal, breaks=50, main='Mean signal', cex.lab=1.3)
> hist(pca.signal, breaks=50, main='First PCA signal', cex.lab=1.3)
> dev.off()
```

```
null device
          1
```

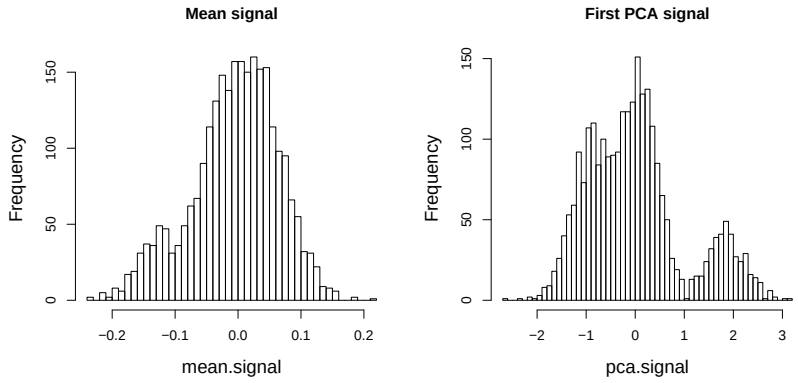

Figure 1: Histograms for the mean intensity and the first principal component of the CNV A112

3 Model selection using the Bayesian Information Criterion (BIC)

To determine the number of components of the CNV for the downstream analysis we can run the model selection algorithm. The models scanned by this function can be specified by the user, however the default settings allow for determining the number of components using some general models for the mean and variance of the components. We specify 3 iterations under H0 to increase the chances of locating a global maximum for each model. The output of the model selection gives the BIC (and AIC) for the different models. The model that minimizes the chosen statistic is the most likely model. This is demonstrated in Figure **??**.

```
> batches <- factor(A112$cohort)
> sample <- factor(A112$subject)
> set.seed(0)
> results <- CNVtest.select.model(signal=pca.signal, batch = batches, sample = sample, n.H0 = 3, method="B
> ncomp <- results$selected
> pdf("fig/modelselect.pdf",width=5,height=5)
> plot(-results$BIC, xlab="n comp", ylab="-BIC", type="b", lty=2, col="red", pch = '+')
> dev.off()
```

```
null device
          1
```

4 Clustering the PCA transformed data

We can then cluster the result of the pca analysis under the null hypothesis of no association between the number of copies and the case-control status. The data will be clustered only once, assuming the null hypothesis $\mathbb{H}_0$. Because of this we do not have to specify any case-control status. We assume free model for the means and the variances for each number of copies using '$\sim$ strata(cn)'. We could have chosen free variances for each combination of batch and copy number using '$\sim$ strata(cn, batch)'. Alternatively a variance model proportional to the number of copies is possible using '$\sim$ cn'. Note, however, that the formulation using strata is much quicker and numerically robust, and should be used when possible. We can also provide an optional vector of starting values for the mean locations of the three clusters.

Note that we must check the status of the fit. Only 'C' should be accepted for further analysis. The possibilities include

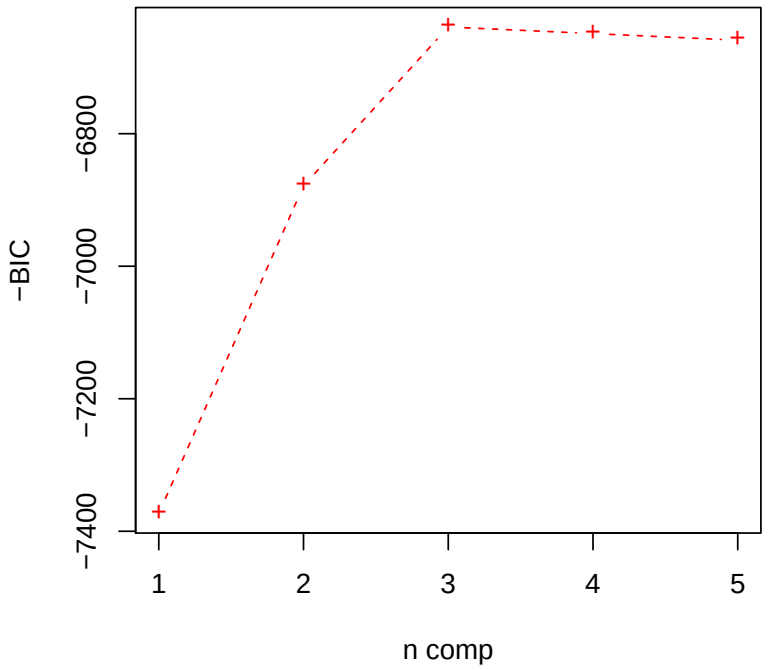

Figure 2: The BIC as a function of the number of components fit to the data output from the model selection stage. The most appropriate model is the one that minimises the BIC.

'C' Converged. This is the only acceptable status.

'M' Maximum iterations reached. The EM algorithm did not converge.

'P' Posterior density problem. The posterior probabilities are not monotonic.

'F' Fit failed. Most likely due to singularity at $\sigma = 0$.

The output contains a list, and the first element of this list is the data frame of interest. In Figure **??** we plot the result of the clustering.

```
> ncomp <- 3
> batches <- factor(A112$cohort)
> sample <- factor(A112$subject)
> fit.pca <- CNVtest.binary ( signal = pca.signal, sample = sample, batch = batches, ncomp = ncomp, n.H0=3
> print(fit.pca$status.H0)

[1] "C"

> pdf("fig/pca-fit.pdf", width=10, height=5)
> par(mfrow=c(1,2))
> cnv.plot(fit.pca$posterior.H0, batch = '58C', main = 'Cohort 58C', breaks = 50, col = 'red')
> cnv.plot(fit.pca$posterior.H0, batch = 'NBS', main = 'Cohort NBS', breaks = 50, col = 'red')
> dev.off()

null device
          1
```

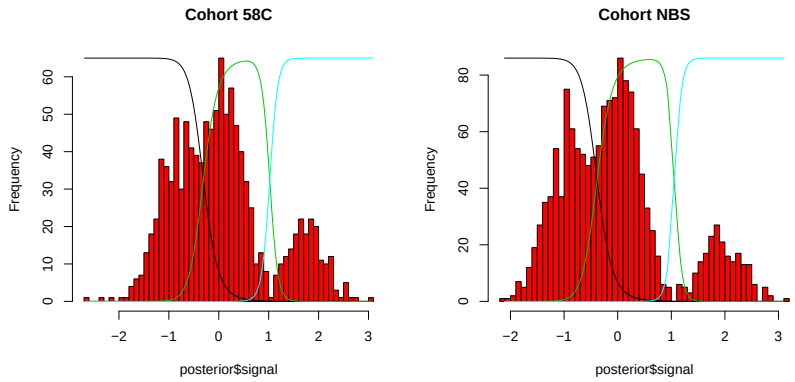

Figure 3: Output of the clustering procedure using the pca (under the null hypothesis $\mathbb{H}_0$ of no allele frequency difference between both cohorts). The colored lines show the posterior probability for each of the three copy number classes (copy number $= 1, 2$ or 3). For clarity the scale for the posterior probabilities is not shown but the maximum is 1 and the three posterior probabilities always add up to 1.

5 Assigning individuals to a copy number genotype

The output from the clustering under $\mathbb{H}_0$ can be used to obtain the posterior probabilities and MAP estimates of an individuals cluster membership. This can also be applied after the LDF improvement (see below). The columns P1, P2, P3 represent the posterior probability of belonging to component 1, 2, 3. The column labeled cn is the MAP assignment.

```
> head(fit.pca$posterior.H0)
```

```
          subject batch     signal trait         P1         P2        P3 cn
1 WTCCC01-11474A1   58C  0.1918171     0 0.04565297 0.95428972 0.0000000  2
2 WTCCC01-11474A2   58C  0.1187269     0 0.06635041 0.93362452 0.0000000  2
3 WTCCC01-11474A3   58C -0.6572859     0 0.92980569 0.07019431 0.0000000  1
4 WTCCC01-11474A4   58C  1.4943201     0 0.00000000 0.00240271 0.9975957  3
5 WTCCC01-11474A5   58C  2.1352589     0 0.00000000 0.00000000 0.9999996  3
6 WTCCC01-11474A6   58C -0.5394552     0 0.84146496 0.15853503 0.0000000  1
```

6 Improving using the LDF procedure

It is now possible to use the posterior probabilities from the pca procedure to improve the fit. This is done by using a linear discriminant analysis.

```
> ncomp <- 3
> pca.posterior <- as.matrix((fit.pca$posterior.H0)[, paste('P',seq(1:ncomp),sep='')])
> dimnames(pca.posterior)[[1]] <- (fit.pca$posterior.H0)$subject
> ldf.signal <- apply.ldf(raw.signal, pca.posterior)
> pdf("fig/ldf_pca_signal.pdf", width=10, height=5)
> par(mfrow=c(1,2))
> hist(pca.signal, breaks=50, main='First PCA signal', cex.lab=1.3)
> hist(ldf.signal, breaks=50, main='LDF signal', cex.lab=1.3)
> dev.off()

null device
          1
```

The results of the LDF analysis can now be see in Figure **??** and we can observe a clear improvement. The data will then be much easier to cluster.

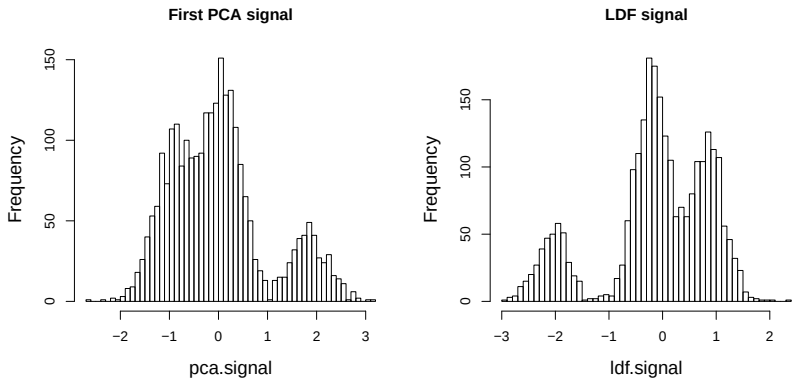

Figure 4: Comparing the LDF and PCA analysis, both clustered under the null hypothesis of no association.

7 Testing for genetic association with a dichotomous disease trait

7.1 Some mathematical details

The association testing approach has been described previously (see reference below) but for completeness we sketch the principle. We use a likelihood ratio approach to test for association between the genotype calls and the case-control status. Genotypes are called using a finite mixture model. Formally, this association test can be as summarized as jointly fitting two linear models:

$$X = \gamma + \theta^t Z + \epsilon \quad (1)$$

$$logit(Y) = \alpha + \beta X \quad (2)$$

The first model is the Gaussian or T mixture model, and the second model is a traditional generalized logit linear model. Notations are:

- X is a N-dimensional vector of signal intensities, where N is the number of samples in the study.
- Z is the (N, G) matrix of genotype assignment, where G designates the number of copy number classes: $Z_{i,j} = 1$ if and only if the sample i has genotype j. Each row z_i of Z is sampled from a multinomial distribution with probabilities $(\Phi_i)_{i=1}^{G}$ representing the genotype frequencies in the sampled population.
- The error term ϵ is normally distributed with mean 0.
- θ is a G dimensional vector, linking the genotype status with the mean value of the signal intensity.
- α and β are scalar and $\beta \neq 0$ under the alternative $\mathbb{H}_1$. Our default assumption is that the log-odds ratio is proportional to the genotype X.
- Y is the N dimensional binary vector describing the case-control status.

7.2 Example

Now that we have summarised the intensity data in an efficient manner we can use it to test for genetic association between both cohorts. Here we have defined an artificial trait, 0 for NBS and 1 for 58C. We can specify the number of iterations under $\mathbb{H}_0$ and $\mathbb{H}_1$, and in that case we will use one iteration for each scenario because the data quality is sufficient and does not require multiple iterations to be fitted properly.

```
> ncomp <- 3
> trait <- ifelse( A112$cohort == '58C', 0, 1)
> fit.ldf <- CNVtest.binary ( signal = ldf.signal, sample = sample, batch = batches, disease.status = trait
> print(fit.ldf$status.H0)

[1] "C"
```

```
> print(fit.ldf$status.H1)

[1] "C"

> pdf("fig/ldf-fit.pdf", width=10, height=5)
> par(mfrow=c(1,2))
> cnv.plot(fit.ldf$posterior.H0, batch = '58C', main = 'Cohort 58C', breaks = 50, col = 'red')
> cnv.plot(fit.ldf$posterior.H0, batch = 'NBS', main = 'Cohort NBS', breaks = 50, col = 'red')
> dev.off()

null device
          1

> LR.statistic <- -2*(fit.ldf$model.H0$lnL - fit.ldf$model.H1$lnL)
> print(LR.statistic)

[1] 1.546307
```

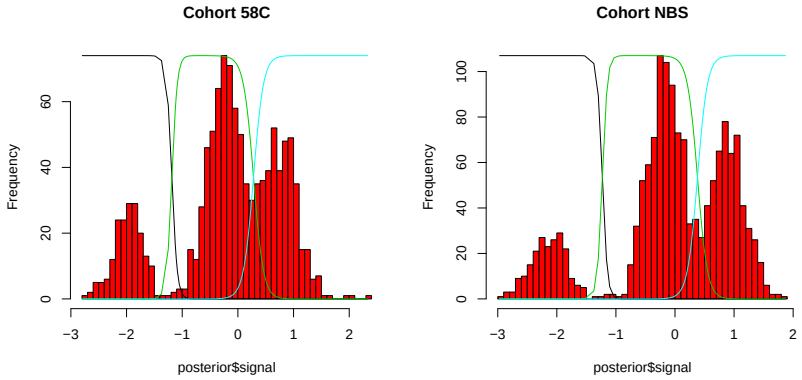

Figure 5: Output of the clustering procedure using the LDF, under the null hypothesis $\mathbb{H}_0$ of no association. The colored lines show the posterior probability for each of the three copy number classes. For clarity the scale for the posterior probabilities is not shown but the maximum is 1 and the three posterior probabilities always add up to 1.

If the fit is correct and there is indeed no association, this statistic should be distributed as χ^2 with one degree of freedom. The fit can be checked in Figure **??**.

Note that the assumed disease model, under the alternate hypothesis $\mathbb{H}_1$, is a linear odds model, which means that the effect on the log-odds is proportional to the number of alleles. We might be interested in testing an allelic model, where the odds are not constrained by a linear trend. This is done by specifying the `model.disease` formula when fitting the data, as follows:

```
> fit.ldf <- CNVtest.binary ( signal = ldf.signal, sample = sample, batch = batches, disease.status = trai
> print(fit.ldf$status.H0)

[1] "C"

> print(fit.ldf$status.H1)

[1] "C"

> LR.statistic <- -2*(fit.ldf$model.H0$lnL - fit.ldf$model.H1$lnL)
> print(LR.statistic)

[1] 3.112371
```

The default for `model.disease` is $\sim$ `cn`. Introducing the `factor` adds one degree of freedom, canceling the default linear constraint. The resulting statistic is now distributed, under the null, as χ^2 with 2 degrees of freedom.

8 Testing for genetic association with a quantitative trait

Now consider the testing of association with a quantitative trait. The model now consists of a standard regression instead of a logistic regression. For this example we will generate a gaussian hypothetical trait and test for association with the combined NBS and 58C individuals. The association test is done in a completely analogous way, namely the LR under $\mathbb{H}_0$ should be distributed as χ^2 with one degree of freedom assuming a linear trend model.

```
> batches <- rep("ALL",length(sample))
> qt <- rnorm(length(sample), mean=9.0, sd=1.0)
> fit.ldf <- CNVtest.qt(signal = ldf.signal, sample = sample, batch = batches, qt = qt, ncomp = ncomp, n.H
> print(fit.ldf$status.H0)

[1] "C"

> print(fit.ldf$status.H1)

[1] "C"

> LR.statistic <- -2*(fit.ldf$model.H0$lnL - fit.ldf$model.H1$lnL)
> print(LR.statistic)

[1] 0.7683424

> pdf("fig/qt-fit.pdf", width=15, height=5)
> qt.plot(fit.ldf)
> dev.off()

null device
          1
```

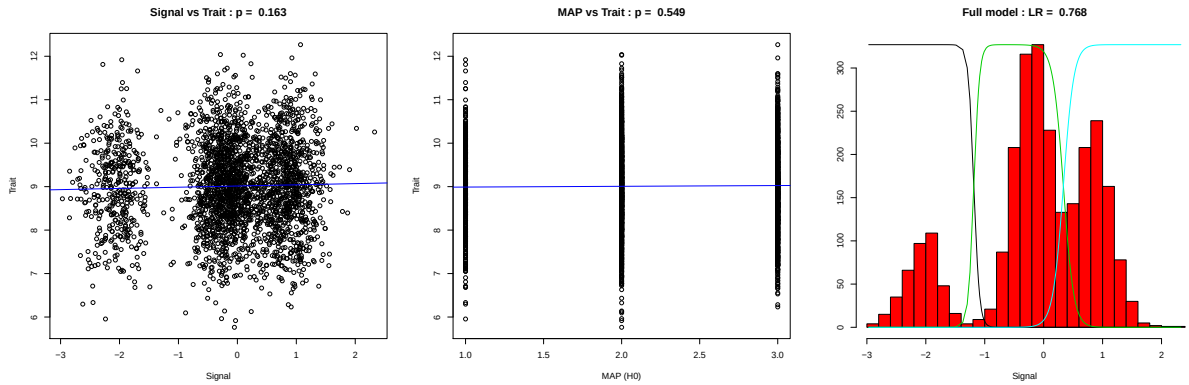

Figure 6: Output of the quantitative trait association procedure on the LDF transformed data. For the rightmost graph the colored lines show the posterior probability for each of the three copy number classes.

9 Reference

The statistical ideas underlying this package have been published in:
A robust statistical method for case-control association testing with copy number variation, Chris Barnes, Vincent Plagnol, Tomas Fitzgerald, Richard Redon, Jonathan Marchini, David G. Clayton, Matthew E. Hurles, Nature Genetics 2008