Clonality: A Package for Clonality testing

Irina Ostrovnaya

November 1, 2022

Department of Epidemiology and Biostatistics
Memorial Sloan-Kettering Cancer Center
ostrovni@mskcc.org

Contents

1 Overview

This document presents an overview of the Clonality package. This package can be used to test whether two tumors from the same patient are clonal (metastases) or independent (double primaries) using their somatic mutations, copy number or loss of heterozygosity (LOH) profiles. For LOH data it implements Concordant Mutations (CM) test (Begg et al., 2007) and Likelihood Ratio (LR) test (Ostrovnaya et al., 2008). For copy number profiles the package implements the methodology based on the likelihood ratio described in (Ostrovnaya et al., 2010). For somatic mutations we included the methods described in (Ostrovnaya et al., 2015) and (Mauguen et al., 2018).

2 Inference using profiles of somatic mutations

In (Ostrovnaya et al., 2015) we presented statistical test for evaluating evidence for clonality against null hypothesis that the two tumors are independent using their mutational profiles obtained by next generation sequencing, such as targeted panel sequencing or whole exome sequencing. It utilizes conditional likelihood model where for each patient only loci where at least one tumor has a mutation are contributing to the test statistic.

2.1 Estimating frequency of observed mutations using TCGA or other datasets

Marginal frequencies of mutations are assumed to be known and used as fixed quantities in the likelihood model. They can be estimated using function *get.mutation.frequencies* using either the built-in mutational data from TCGA pancancer study, or reference mutational data, or combination of the two. Currently, the built-in mutational frequencies datset 'freqdata' contains 3 cancer types: COAD, LUAD and BRCA - see note about other types below.

Note that these frequencies used in the test can never be zero even if the mutation was never seen before, so the observed mutations seen in the particular patient are essentially added to the external data. For example, if in TCGA/reference dataset mutation was observed in X/n samples, then the frequency of this mutation is (X+1)/(n+1). The whole dataset for which clonality is tested can also be added to the frequency calculation. The mutations are matched with TCGA/reference dataset using mutation IDs that have the following format: Chromosome Location RefAllele AltAllele, each entry separated by space, where chromosome is a number 1-22 or X or Y; location is genomic location in GRCh37 build; RefAllele is a reference allele and AltAllele is Alternative allele. The ref and alt alleles follow standard TCGA maf file notations.

```
>   library(Clonality)
> data(lcis)
> data(freqdata)
> mut.matrix<-create.mutation.matrix(lcis )
> freq<-get.mutation.frequencies(rownames(mut.matrix),"BRCA")
```

If other cancer types are needed then internal object freqdata can be replaced by the full set of 33 cancer types using

```
> load(url("https://github.com/IOstrovnaya/MutFreq/blob/master/freqdata.RData?raw=true"))
```

after which the same analysis pipeline applies.

This full object freqdata was obtained using the following code:

```
> #  The file mc3.v0.2.8.PUBLIC.maf can be downloaded from https://gdc.cancer.gov/about-data/p
>  pancan<-read.table(  "mc3.v0.2.8.PUBLIC.maf",sep="\t",header=T,quote="")
> #  The file clinical.tsv.csv can be downloaded from https://portal.gdc.cancer.gov/projects -
> # click on 11,315 Cases across 33 Projects on the upper right corner -> click on Clinical->T
>  pancaninfo<-read.table("clinical.tsv",sep="\t",header=T)
>  pancaninfo$project_id<-substr(pancaninfo$project_id,6,11)
>  pancan$Tumor_Sample_Barcode<-substr(pancan$Tumor_Sample_Barcode,1,12)
>  pancan$mut<-paste(pancan$Chromosome,pancan$Start_Position,pancan$Reference_Allele,pancan$Tu
>  pancan$type<-pancaninfo$project_id[match(pancan$Tumor_Sample_Barcode,pancaninfo$submitter_i
>  pancan<-pancan[!duplicated(paste(pancan$Tumor_Sample_Barcode,pancan$mut)),]
>  pancan<-pancan[!is.na(pancan$type),]
>  pancan$mut<-as.factor(pancan$mut)
>  types<-unique(pancan$type)
>  ntypes<-NULL
> freqdata<-NULL
> for (i in types)
+ {w<-pancan$type==i
+ s<-length(unique(pancan$Tumor_Sample_Barcode[w]))
+  ntypes<-c(ntypes,s)
+  freqdata<-cbind(freqdata,table(pancan$mut[w]))
+
+  }
>  rownames(freqdata)<-names(table(pancan$mut[w]))
>  colnames(freqdata)<-types
>  freqdata<-rbind(ntypes,freqdata)
>  freqdata<-freqdata[,c(c("COAD","LUAD","BRCA"))]
>  freqdata<-freqdata[apply(freqdata,1,sum)>0,]
```

2.2 Likelihood model

Here we download the exome sequencing data from study of Lobular Carcinoma in Situ (LCIS) and Invasive lobular carcinomas (ILC) and Invasive Ductal Carcinomas (IDC) in the same patients ((Begg et al., 2016)). Patient 53 has one match between IDC and LCIS and based on BRCA data in TCGA we estimated its probability as 0.000979.

```
>   table(mut.matrix$TK53IDC2,mut.matrix$TK53LCIS2 )

       0    1
  0 1011   16
  1   22    1
```

```
> freq[mut.matrix$TK53IDC2+mut.matrix$TK53LCIS2==2]

22 28195110 G A
   0.0009794319
```

Below is the test of clonality of these two tumors. Note that the p-value is calculated using the simulated null distribution, thus setting the random seed is recommended for reproducibility.

```
>   set.seed(1)

NULL
```

```
> SNVtest(mut.matrix$TK53IDC2,mut.matrix$TK53LCIS2  ,freq)

         n1          n2     n_match      LRstat      maxKsi    LRpvalue
23.00000000 17.00000000  1.00000000  2.98918791  0.04907477  0.01400000
```

The p-value of 0.014 confirms that these two tumors are likely clonal, i.e. originate from the same cell harboring the matching mutation.

2.3 Two-site test

It is also possible to test clonality of two tumors that come from two different sites or organs with different mutational profiles using function "SNVtest2". The null hypothesis is that two tumors come from 2 different sites with different marginal proabilities, and there are two alternative hypotheses: that two tumors are clonal and come from site 1, and that they are clonal and come from site 2. Here we simulate two tumors from lung and colon tumors on the set of mutations common in either of these two types.

```
> #restricting set of loci to 10,000 so that the data are more similar to targeted sequencing
> mut.list<-row.names(freqdata[freqdata[,"COAD"]>=1 | freqdata[,"LUAD"]>=1,][-1,])[1:10000]
> freqCOAD<-get.mutation.frequencies(mut.list,"COAD")/freqdata[1,"COAD"]
> #arbitrarily specified background mutation rate
> freqCOAD[freqCOAD==min(freqCOAD)]<-1/(freqdata[1,"COAD"] +freqdata[1,"LUAD"]  )
> freqLUAD<-get.mutation.frequencies(mut.list,"LUAD")/freqdata[1,"LUAD"]
> #arbitrarily specified background mutation rate
> freqLUAD[freqLUAD==min(freqLUAD)]<-1/(freqdata[1,"COAD"] +freqdata[1,"LUAD"]  )
>
```

Below is the test of clonality of the two randomly generated tumors. Note that the p-value is calculated using the simulated null distribution, thus setting the random seed is recommended for reproducibility.

```
>   set.seed(1)
> x1<-as.numeric(runif(length(mut.list))<=freqCOAD)
> x2<-as.numeric(runif(length(mut.list))<=freqLUAD)

> SNVtest2(x1,x2  ,cbind(freqCOAD,freqLUAD))
```

```
   n.match n.site1only n.site2only    xi.site1    xi.site2     p.value 
     0e+00       5e+00       7e+00       1e-05       1e-05       1e+00 
```

The p-value confirms that these two tumors are independent.

2.4 Random effects model

Here we show how to test the independence of the somatic mutation profile, following the random effects model proposed by Mauguen et al (http://biostats.bepress.com/mskccbiostat/paper33, (Mauguen et al., 2018)). The example uses the data from 17 cases with both lobular or ductal carcinoma in situ (LCIS or DCIS) and an invasive lobular or ductal breast cancer (ILC or IDC) (Begg et al., 2016). Data from whole-exome sequencing were available and used to compare the mutation profile of the two tumors. The random-effect model is estimated on the data using the following code:

```
>   data(lcis)
> mut.matrix<-create.mutation.matrix(lcis ,rem=TRUE)
> freq<-get.mutation.frequencies(rownames(mut.matrix),"BRCA")
> mod <- mutation.rem(cbind(freq, mut.matrix))

>   print(mod)
```

```
Estimation done on  17 pairs

___ Parameter estimates

Random-effect distribution
 mean mu = -0.87
 standard-deviation sigma = 0.59

Proportion of clonal pairs
 pi = 0.628

___ Model likelihood and convergence

likelihood  -308.1262
convergence status  1
number of iterations used 13
NULL
```

In this example, the estimation converged (convergence status=1). The proportion of clonal cases in the LCIS dataset is estimated to be 57%. The individual probabilities of clonality for those pairs of tumors are obtained using:

```
>   mod <- mutation.rem(cbind(freq,mut.matrix), proba=TRUE)
>

>   print(mod)
```

```
Estimation done on  17 pairs

___ Parameter estimates

Random-effect distribution
 mean mu = -0.87
 standard-deviation sigma = 0.59

Proportion of clonal pairs
 pi = 0.628

___ Model likelihood and convergence

likelihood  -308.1262
convergence status  1
number of iterations used 13
NULL
```

The individual probability of clonality varies from <1% for case 16 with no shared mutations to >99% for several cases having 2 or more mutations shared between the two tumors.

Finally, once the model is estimated on a given population, it is possible to estimate the probability of clonality of a new case using:

```
> pi <- runif(30,0.001,0.13)
> newpair <- cbind(pi,rbinom(30,1,1-pi*pi)+1)
> new.likmat <- grid.lik(xigrid=c(0, seq(0.0005, 0.9995, by=0.001)),
+                         as.matrix(newpair[,c(-1)]), newpair[,1])
> proba <- mutation.proba(c(mod$mu, mod$sigma, mod$pi), t(as.matrix(new.likmat)) )

>   print(proba)

[1] 0.023
[1] 0.023
```

For this hypothetical case with 30 private mutations, the probability of being clonal is 24%.

3 Copy number profiles

We will show how to test independence of the copy number profiles from the same patient using simulated data. First we simulate the dataset with 10 pairs of tumors with 22 chromosomes, 100 markers each. Simulated log-ratios are equal to signal + noise. The signal is defined in the following way: each chromosome has 50% chance to be normal, 30% to be whole-arm loss/gain, and 20% to be partial arm loss/gain, where endpoints are drawn at random, and loss/gain means are drawn from standard normal distribution. There are no chromosomes with recurrent losses/gains. Noise is drawn from normal distribution with mean 0, standard deviation 0.4 and added to the signal. First 9 patients have independent tumors, while last patient has two tumors with identical signal, but independent noise.

```
>  library(Clonality)
> set.seed(100)
> chrom<-paste("chr",rep(c(1:22),each=100),"p",sep="")
> chrom[nchar(chrom)==5]<-paste("chr0",substr(chrom[nchar(chrom)==5] ,4,5),sep="")
> maploc<- rep(c(1:100),22)
> data<-NULL
> for (pt in 1:9)
+ {
+ tumor1<-tumor2<- NULL
+ mean1<- rnorm(22)
+ mean2<- rnorm(22)
+ for (chr in 1:22)
+ {
+   r<-runif(2)
+ if (r[1]<=0.5) tumor1<-c(tumor1,rep(0,100))
+   else if   (r[1]>0.7)  tumor1<-c(tumor1,rep(mean1[chr],100))
+   else  { i<-sort(sample(1:100,2))
+         tumor1<-c(tumor1,mean1[chr]*c(rep(0,  i[1]),rep(1, i[2]-i[1]), rep(0,  100-i[2])))
+         }
+ if (r[2]<=0.5) tumor2<-c(tumor2,rep(0,100))
+   else if   (r[2]>0.7)  tumor2<-c(tumor2,rep(mean2[chr],100))
+   else   {i<-sort(sample(1:100,2))
+        tumor2<-c(tumor2,mean2[chr]*c(rep(0,  i[1]),rep(1, i[2]-i[1]), rep(0,  100-i[2])))
+          }
+ }
+ data<-cbind(data,tumor1,tumor2)
+ }
> tumor1<- NULL
> mean1<- rnorm(22)
> for (chr in 1:22)
+ {
+   r<-runif(1)
+ if (r<=0.4) tumor1<-c(tumor1,rep(0,100))
+   else if   (r>0.6)  tumor1<-c(tumor1,rep(mean1[chr],100))
+   else  { i<-sort(sample(1:100,2))
+         tumor1<-c(tumor1,mean1[chr]*c(rep(0,  i[1]),rep(1, i[2]-i[1]), rep(0,  100-i[2])))
+         }
+
+ }
> data<-cbind(data,tumor1,tumor1)
> data<-data+matrix(rnorm( 44000,mean=0,sd=0.4) ,nrow=2200,ncol=20)
> samnms<-paste("pt",rep(1:10,each=2),rep(1:2,10),sep=".")
>
```

Rows of data correspond to probes (genomic markers). The first column is the chromosome and the second column is probe's genomic position. All subsequent columns correspond to the samples and contain log-ratios. Here the genomic is an index, but normally it would be actual probe's location along the genome, and then 'splitChromosomes' function should be used to divide the chromosome into p and q arms, thus increasing the number of independent units for the analysis.

```
> dim(data)

[1] 2200   20
```

As the next step of data preparation, we have to create a CNA (copy number array) object as described DNAcopy.

```
> dataCNA<-CNA(data,chrom=chrom,maploc=maploc,sampleid=samnms)
> as.matrix(dataCNA)[1:5,1:10]

  chrom    maploc pt.1.1           pt.1.2          pt.2.1
1 "chr01p" "   1"  " 1.787454e-01" "-0.0747496473" " 3.863461e-01"
2 "chr01p" "   2"  "-3.404918e-01" " 0.2797033500" " 1.739630e-01"
3 "chr01p" "   3"  "-4.191789e-01" " 0.3877484789" " 2.237324e-01"
4 "chr01p" "   4"  " 1.597503e-03" " 0.6996900997" "-1.257982e-01"
```

```
5 "chr01p" "  5"  "-5.678337e-01" "-0.1219955963" "-3.305056e-02"
  pt.2.2          pt.3.1          pt.3.2          pt.4.1
1 "-1.121270e+00" " 7.806864e-01" " 0.6435773066" "-7.334312e-01"
2 "-7.469143e-01" " 9.833035e-01" "-0.2639643722" "-7.559545e-01"
3 "-1.155145e+00" " 9.314601e-01" "-0.9361022506" "-1.273873e+00"
4 "-7.048441e-01" " 1.111832e+00" "-0.1264779640" "-9.182190e-01"
5 "-1.788286e+00" " 3.425941e-01" "-0.7664538697" "-4.629290e-02"
  pt.4.2
1 "-1.7398678901"
2 "-1.4075129961"
3 "-1.4278995167"
4 "-1.5514615241"
5 "-1.4944360492"

>
```

Our methodology allows at most one genomic change per chromosome arm, estimated by the one-step Circular Binary Segmentation (CBS) algorithm ((Venkatraman and Olshen, 2007)).

If the data had many more than 15,000 markers, most outstanding, and likely a short change would be picked up, which would not be representative of the chromosome pattern. To avoid this, one can use the following function:

```
> dataAve<- ave.adj.probes(dataCNA,2)

Total number of markers after averaging is 1100
```

Here we have averaged every two consecutive markers. For this dataset, though, averaging is not necessary.

Next we have to create a vector of patient labels that matches the samples.

```
> ptlist<- paste("pt",rep(1:10,each=2),sep=".")
```

Finally, we can run the clonality analysis:

```
> results<-clonality.analysis(dataCNA, ptlist,  pfreq = NULL, refdata = NULL, nmad = 1,  refer

Calculating LR..........
Calculating reference LR: %completed 10, 20, 30, 40, 50, 60, 70, 80, 90, 100,
```

The main information is in the output LR:

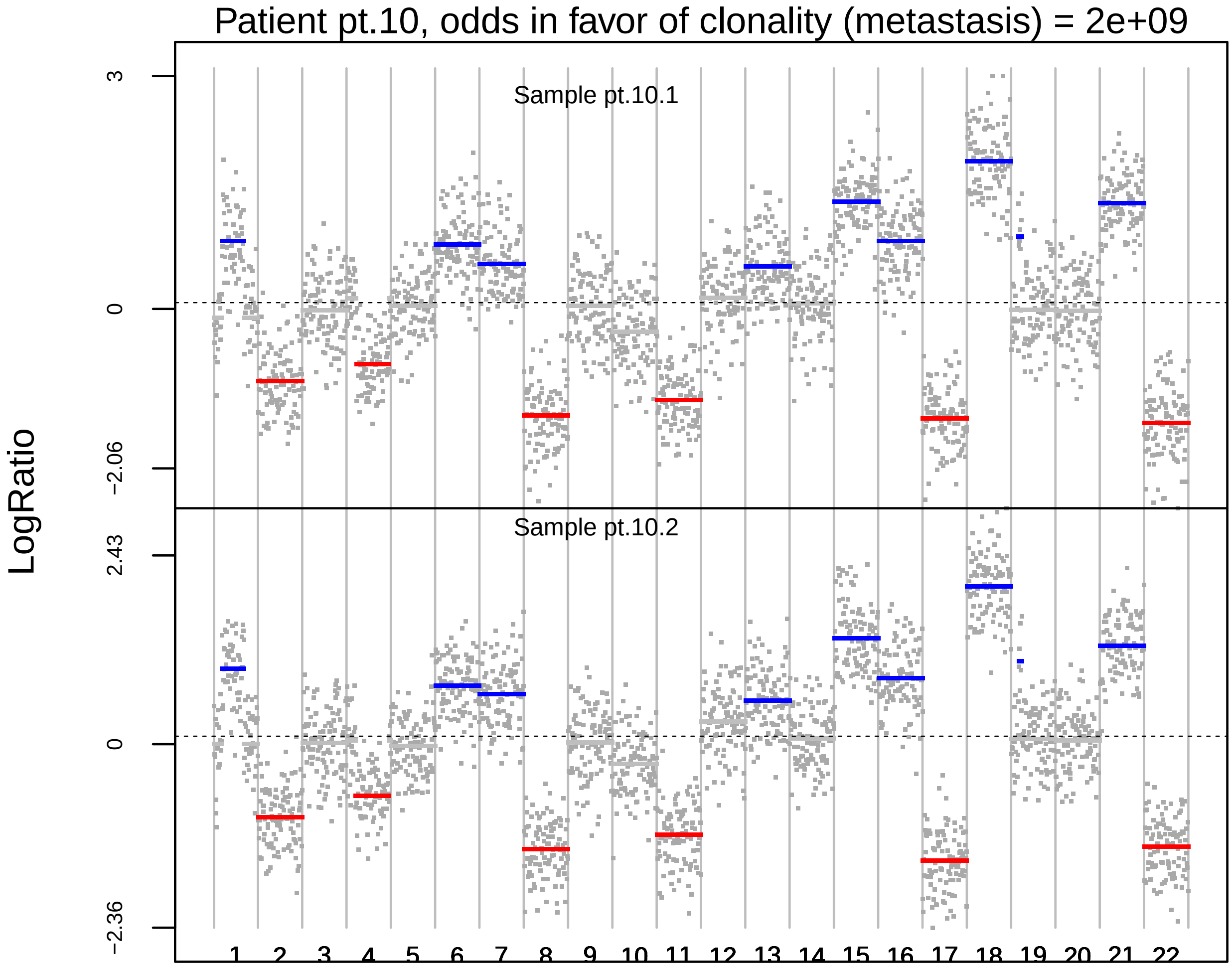
```
> results$LR

   Sample1 Sample2          LR1          LR2 GGorLL NN GL GNorLN
1   pt.1.1  pt.1.2 7.545701e-02 7.545701e-02      0 14  0      8
2   pt.2.1  pt.2.2 2.295601e-03 2.295601e-03      0  6  0     16
3   pt.3.1  pt.3.2 4.848796e-02 4.848796e-02      0 13  0      9
4   pt.4.1  pt.4.2 2.133004e-02 2.133004e-02      1  8  1     12
```

```
5   pt.5.1  pt.5.2 2.381485e-01 2.394420e-02      2 10 1      9
6   pt.6.1  pt.6.2 2.193293e-02 2.193293e-02      1  9 2     10
7   pt.7.1  pt.7.2 4.598862e-02 4.598862e-02      2 10 3      7
8   pt.8.1  pt.8.2 1.986364e-02 1.986364e-02      0 13 2      7
9   pt.9.1  pt.9.2 8.997422e-03 8.997422e-03      1  8 2     11
10 pt.10.1 pt.10.2 2.546268e+04 1.984131e+09     15  7 0      0
                      IndividualComparisons LR2pvalue
1                                           0.2833333
2                                           0.9222222
3                                           0.3833333
4                                           0.5388889
5                                chr15p 0.1 0.5222222
6                                           0.5388889
7                                           0.4055556
8                                           0.5888889
9                                           0.7444444
10 chr01p 69.73; chr04p 26.03; chr19p 42.93 0.0000000
```

The likelihood ratios LR2 for patients 1:9 are much smaller than 1, therefore these tumors are independent. Patient 10 has LR2 much higher than one, and we can conclude that this patient's tumors are clonal. The reference distribution for LR2 under the hypothesis of independence is constructed by pairing tumors from different patients that are independent by default. The p-value column reflects the percentiles of a particular patient's LR2 in the reference distribution: clonal tumors would have small p-values.

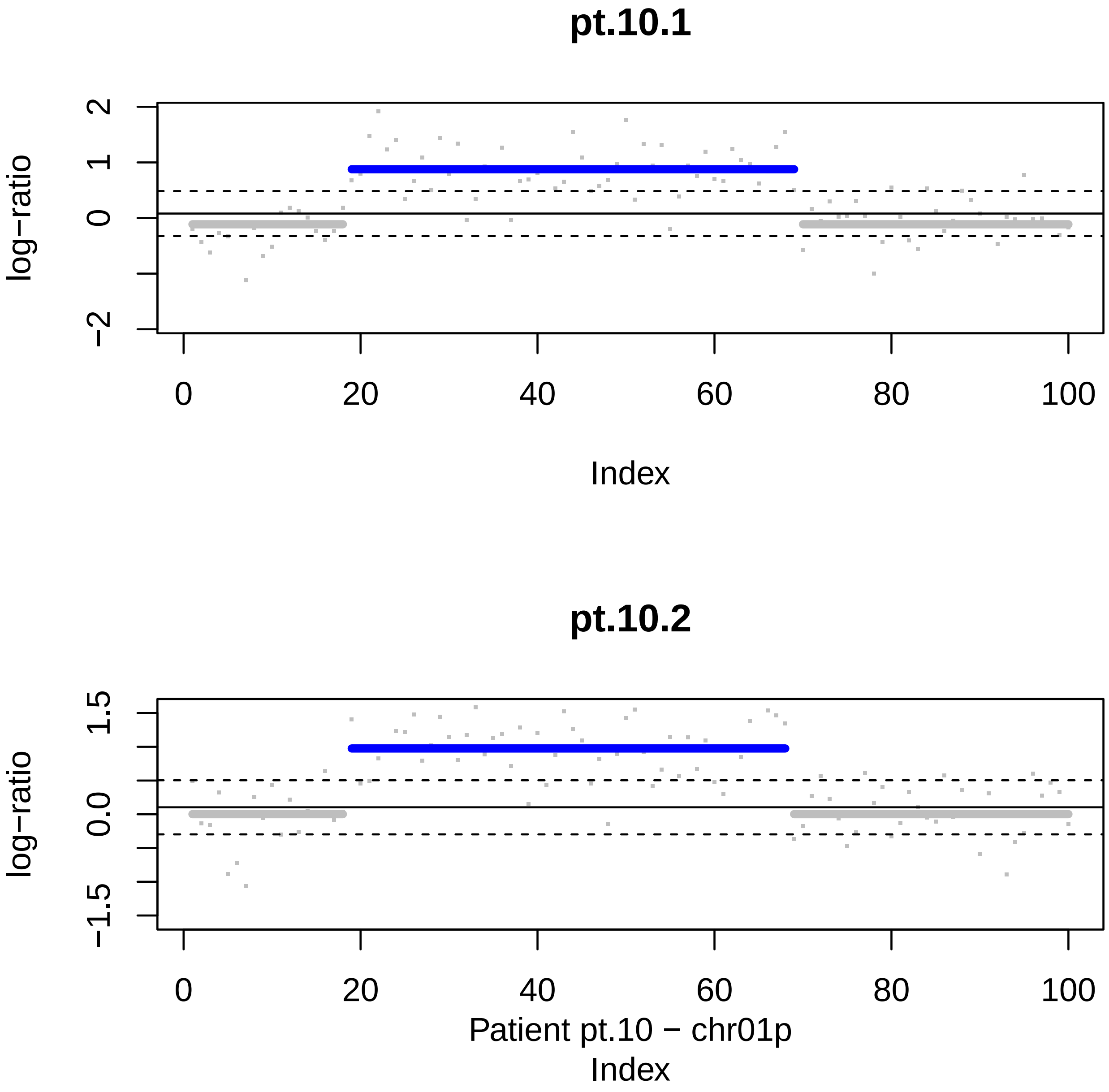
We can view the genomewide plots of patient 10 using:

```
> genomewidePlots(results$OneStepSeg, results$ChromClass,ptlist , c("pt.10.1", "pt.10.2"),resu
```



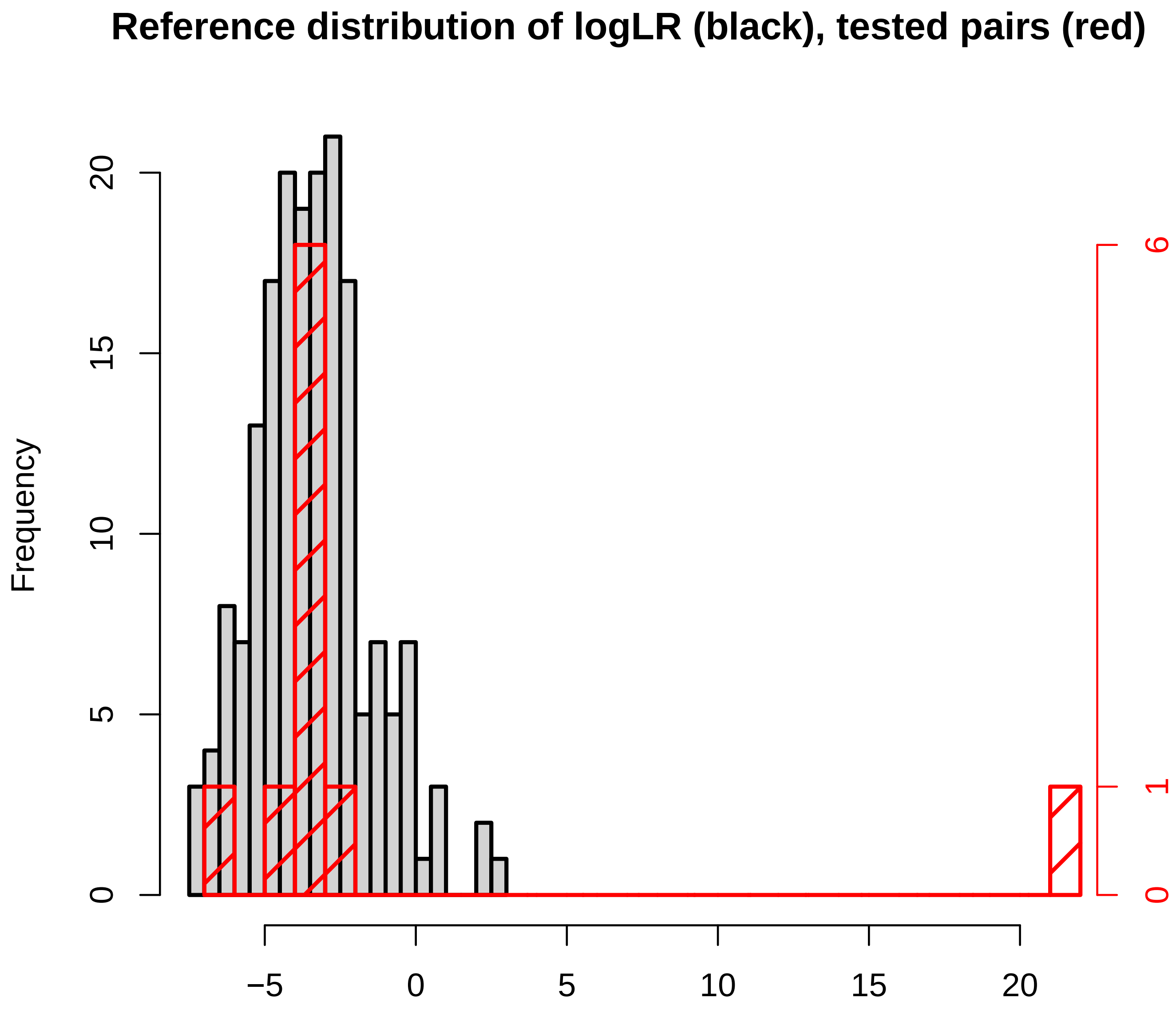
Patterns for each chromosome would be plotted by:

```
> chromosomePlots(results$OneStepSeg, ptlist,ptname="pt.10",nmad=1)
```



The overlap between the histograms of LR2 from original pairs of tumors and the reference distribution are produced by:

```
> histogramPlot(results$LR[,4], results$refLR[,4])
```



3.1 Choice of segmentation algorithm

Note that the user can potentially specify the segmentation method to be used. Currently the default behavior of the clonality.analysis function is to use the CBS algorithm to identify the most significant change in each chromosome arm. The internal function for this purpose is "oneseg" called as oneseg(x, alpha, nperm, sbdry)

There are 4 arguments to oneseg:

x: is the finite logratio data ordered by genomic position.
alpha: the significance level used by CBS.
nperm: the number of permutations for the reference distribution.
sbdry: early stopping boundary for declaring no change (calculated from alpha and nperm).

The output of this function is a vector of 3 numbers where the first is the number of change-points detected (must be 0, 1 or 2), and the second and the third numbers are the start and end of the left segment if there is only one change-point, and of the middle segment when there are 2 change-points.

The function allows the user to specify alternative alpha and nperm for 'oneseg' as a list using the segpar argument e.g. segpar=list(alpha=0.05, nperm=1000). Since sbdry is always calculated in clonality.analysis function from alpha and nperm it is not specified.

Alternate segmentation algorithm can be used. It requires the user to create a function that takes the ordered logratio from one chromosome arm as argument "x" as in oneseg. The name of this function should not be 'oneseg' and is passed through the 'segmethod' argument and all other necessary arguments that are needed passed as a list through 'segpar' argument.

4 LOH data

The LOH data has to be combined in a matrix where first column has marker names and the following columns have LOH calls for each sample. Here we simulate a dataset with 10 pairs of tumors and 20 markers. First pair of tumor is clonal, and the rest of them are independent. If the marker is heterozygous and there is no LOH, then it is denoted by 0. LOH at maternal or paternal alleles is marked by 1 or 2.

```
> set.seed(25)
> LOHtable<-cbind(1:20,matrix(sample(c(0,1,2),20*20,replace=TRUE),20))
> LOHtable[,3]<-LOHtable[,2]
> LOHtable[1,3]<-0
```

```
> LOHtable[,1:5]
```

```
      [,1] [,2] [,3] [,4] [,5]
 [1,]    1    2    0    1    0
 [2,]    2    0    0    0    2
 [3,]    3    0    0    1    2
 [4,]    4    0    0    2    1
 [5,]    5    0    0    2    0
 [6,]    6    1    1    2    2
 [7,]    7    2    2    0    0
 [8,]    8    1    1    1    1
 [9,]    9    0    0    2    1
[10,]   10    2    2    2    2
[11,]   11    0    0    1    2
[12,]   12    1    1    2    0
[13,]   13    2    2    1    2
```

```
[14,]   14    2    2    1    0
[15,]   15    0    0    2    1
[16,]   16    0    0    1    0
[17,]   17    1    1    0    1
[18,]   18    0    0    2    2
[19,]   19    2    2    0    1
[20,]   20    0    0    1    2
```

```
> LOHclonality(LOHtable,rep(1:10,each=2),pfreq=NULL,noloh=0,loh1=1,loh2=2)

Testing clonality for patient 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,  Done
   Sample1 Sample2 a  e f g  h Ntot            CMpvalue LRpvalue
1        1       1 9  9 0 1 10   20 1.16285127411288e-07        0
2        2       2 4 11 3 5  1   20    0.890104091081334    0.632
3        3       3 3  9 5 5  1   20    0.932925591404839    0.517
4        4       4 4 11 5 3  1   20    0.890104091081334    0.595
5        5       5 3  9 3 4  4   20    0.825019072698981    0.519
6        6       6 5 11 3 2  4   20    0.506005585378169    0.238
7        7       7 3  8 2 6  4   20     0.75441116002322    0.659
8        8       8 5 10 3 6  1   20    0.658509623524574    0.937
9        9       9 4  7 5 5  3   20    0.517258428113465    0.302
10      10      10 3  9 3 2  6   20    0.708622670111187    0.519
```

First p-value is small, indicating clonality, for both CM and LR tests. The rest of the p-values are not significant.

Markers that are not informative (e.g. homozygous) in a particular tumor should be given NA instead of a call. Such markers will be dropped from the analysis of this specific patient.

5 LOH data for 3 and more tumors

It is possible to test clonality of 3 or more tumors using Extended Concordant Mutations test, implemented in function 'ECMtesting'. The input LOH matrix can be in the same format as for 'LOHclonality' function: first column of a matrix contains marker names, subsequent columns are samples. For each patient all possible subsets of tumors are tested for clonality, with adjustment for multiple comparison performed using permutation MinP procedure.

Likelihood model can be extended for 3 or 4 tumors with function 'LRtesting3or4tumors'. The likelihood function depends on 2 parameters for 3 tumors, and 3 parameters for 4 tumors, allowing for non-symmetric relationship among tumors. Likelihood ratio test is computed and p-value is calculated using permutations.

Below are the details of the session information:

```
R version 4.2.1 (2022-06-23)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 20.04.5 LTS

Matrix products: default
BLAS:   /home/biocbuild/bbs-3.16-bioc/R/lib/libRblas.so
```

```
LAPACK: /home/biocbuild/bbs-3.16-bioc/R/lib/libRlapack.so

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_GB              LC_COLLATE=C
 [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] Clonality_1.46.0 DNAcopy_1.72.0

loaded via a namespace (and not attached):
[1] compiler_4.2.1 tools_4.2.1
```

References

Begg, C., Eng, K., and Hummer, A. (2007). Statistical tests for clonality. *Biometrics*, 63:522–530.

Begg, C., Ostrovnaya, I., Carniello, J., Sakr, R., Giri, D., Towers, R., Schizas, M., DeBrot, M., Andrade, V., Mauguen, A., Seshan, V., and King, T. (2016). Clonal relationships between lobular carcinoma in situ and other breast malignancies. *Breast Cancer Res*, 18(1):66.

Mauguen, A., Seshan, V., Ostrovnaya, I., and Begg, C. (2018). Estimating the probability of clonal relatedness of pairs of tumors in cancer patients. *Biometrics*, 74(1):321–330.

Ostrovnaya, I., Olshen, A., Seshan, V., Orlow, I., Albertson, D., and Begg, C. (2010). A metastasis or a second independent cancer? evaluating the clonal origin of tumors using array copy number data. *Statistics in Medicine*, 29:1608–1621.

Ostrovnaya, I., Seshan, V., and Begg, C. (2008). Comparison of properties of tests for assessing tumor clonality. *Biometrics*, 68:1018–1022.

Ostrovnaya, I., VE, S., and CB, B. (2015). Using somatic mutation data to test tumors for clonal relatedness. *Annals of Applied Statistics*, 9(3):1533–1548.

Venkatraman, E. and Olshen, A. (2007). A faster circular binary segmentation algorithm for the analysis of array cgh data. *Bioinformatics*, 23:657–663.