Clonality: A Package for Clonality testing

Irina Ostrovnaya

May 3, 2016

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 are clonal (metastases) or independent (double primaries) using their copy number or loss of heterozygosity (LOH) profiles. For LOH data it implements Concordant Mutations (CM) test (**?**) and Likelihood Ratio (LR) test (**?**). For copy number profiles the package implements the methodology based on the likelihood ratio described in (**?**).

2 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)  #first 9 patients have independent tumors
```

```
+ {
+ 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)
+ }
> #last patient has identical profiles
> 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"  " 5.229029e-01" " 0.2959505888" "-3.479070e-01"
2 "chr01p" "  2"  " 1.787454e-01" "-0.0747496473" " 3.863461e-01"
3 "chr01p" "  3"  "-3.404918e-01" " 0.2797033500" " 1.739630e-01"
4 "chr01p" "  4"  "-4.191789e-01" " 0.3877484789" " 2.237324e-01"
5 "chr01p" "  5"  " 1.597503e-03" " 0.6996900997" "-1.257982e-01"
  pt.2.2          pt.3.1          pt.3.2         pt.4.1
1 " 3.365784e-01" " 0.5740303360" "-0.138725302" "-5.097874e-01"
2 "-2.887743e-01" " 0.1649959341" " 0.643577307" "-1.060686e-01"
3 " 8.558146e-02" " 0.3676130117" "-0.263964372" "-1.285919e-01"
4 "-3.226487e-01" " 0.3157696773" "-0.936102251" "-6.465105e-01"
5 " 1.276517e-01" " 0.4961418049" "-0.126477964" "-2.908564e-01"
  pt.4.2
1 " 1.196744e-01"
2 "-4.829711e-01"
3 "-1.506162e-01"
4 "-1.710028e-01"
5 "-2.945648e-01"

>
```

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

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,  re
```

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

The main information is in the output LR:

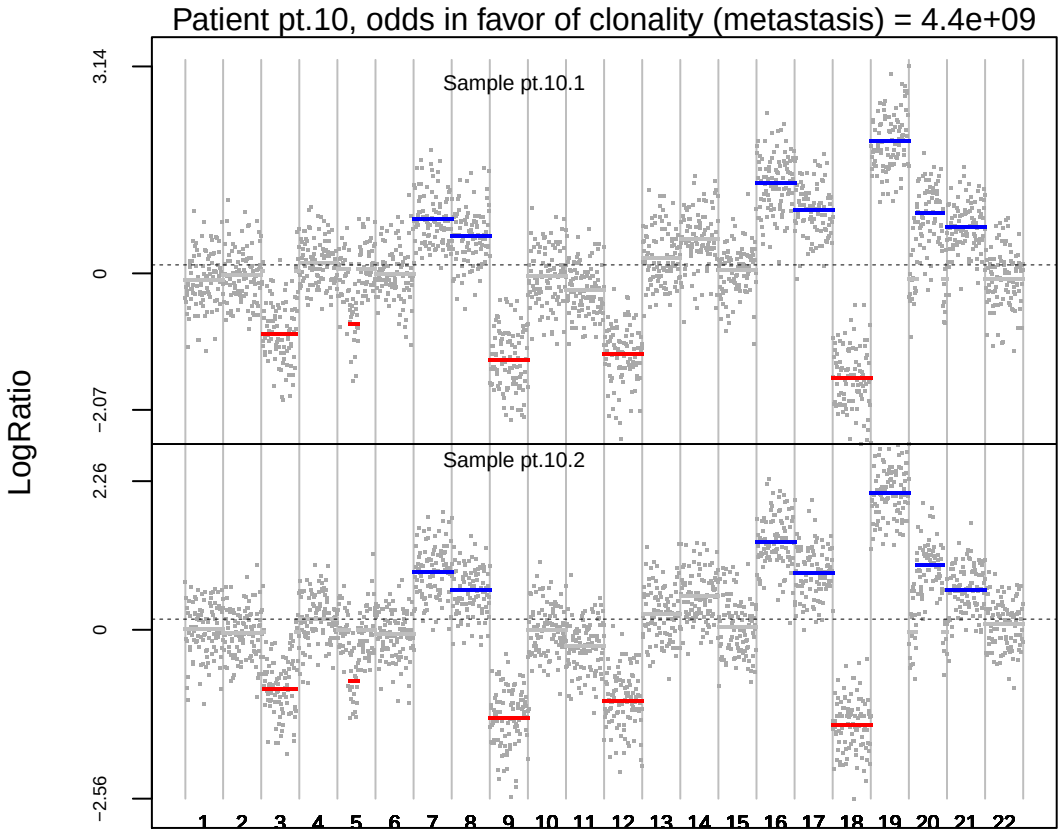
```
> results$LR
```

```
   Sample1 Sample2          LR1          LR2 GGorLL NN GL GNorLN
1   pt.1.1  pt.1.2 2.682477e-02 2.682477e-02      0 12  0     10
2   pt.2.1  pt.2.2 2.830516e-02 4.816474e-03      1  9  0     12
3   pt.3.1  pt.3.2 7.263437e-03 7.263437e-03      0  9  0     13
4   pt.4.1  pt.4.2 1.793088e-01 1.793088e-01      2 10  0     10
5   pt.5.1  pt.5.2 1.897357e-02 2.700118e-03      2  8  3      9
6   pt.6.1  pt.6.2 7.441437e-03 7.441437e-03      0 11  2      9
7   pt.7.1  pt.7.2 1.084280e+00 1.784246e-01      4  8  1      9
8   pt.8.1  pt.8.2 1.350562e-01 1.350562e-01      1 15  3      3
9   pt.9.1  pt.9.2 9.918617e-03 9.918617e-03      1  8  1     12
10 pt.10.1 pt.10.2 5.790525e+04 4.402231e+09     12 10  0      0
                    IndividualComparisons LR2pvalue
1                                          0.3944444
2                              chr20p 0.17 0.7944444
3                                          0.7388889
4                                          0.1555556
5                              chr15p 0.14 0.8777778
6                                          0.6833333
7                              chr18p 0.16 0.1555556
8                                          0.2000000
9                                          0.6388889
10 chr03p 27.5; chr05p 52.81; chr20p 52.34 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.

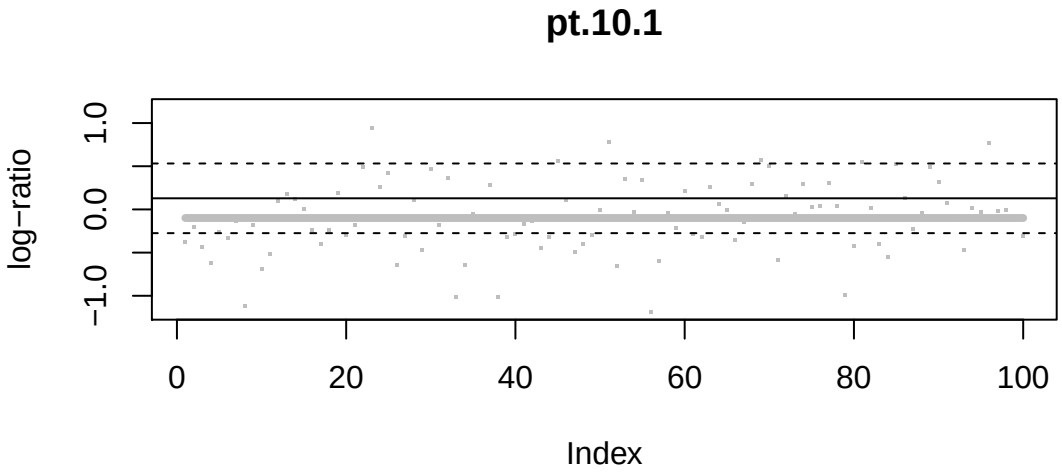
We can view the genomewide plots of patient 10 using:

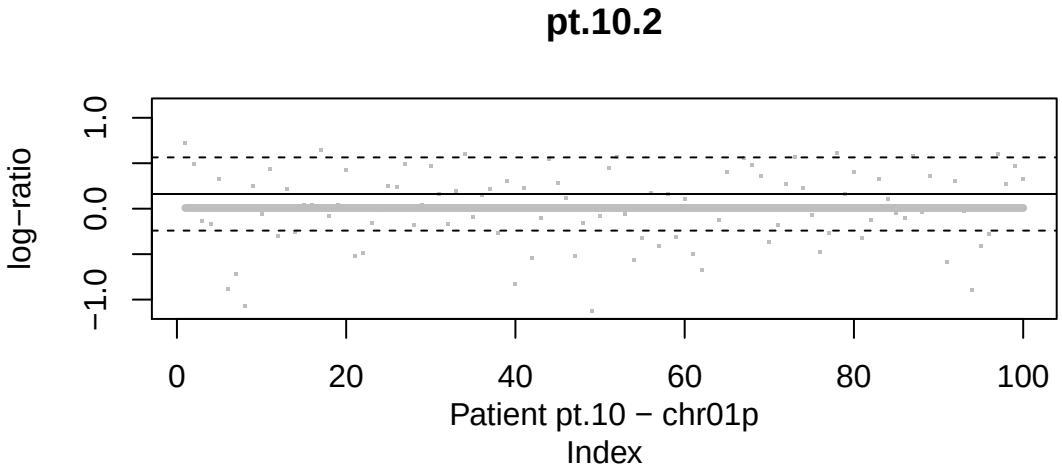
```
> genomewidePlots(results$OneStepSeg, results$ChromClass,ptlist , c("pt.10.1", "pt.10.2"),r
```



Patterns for each chromosome would be plotted by:

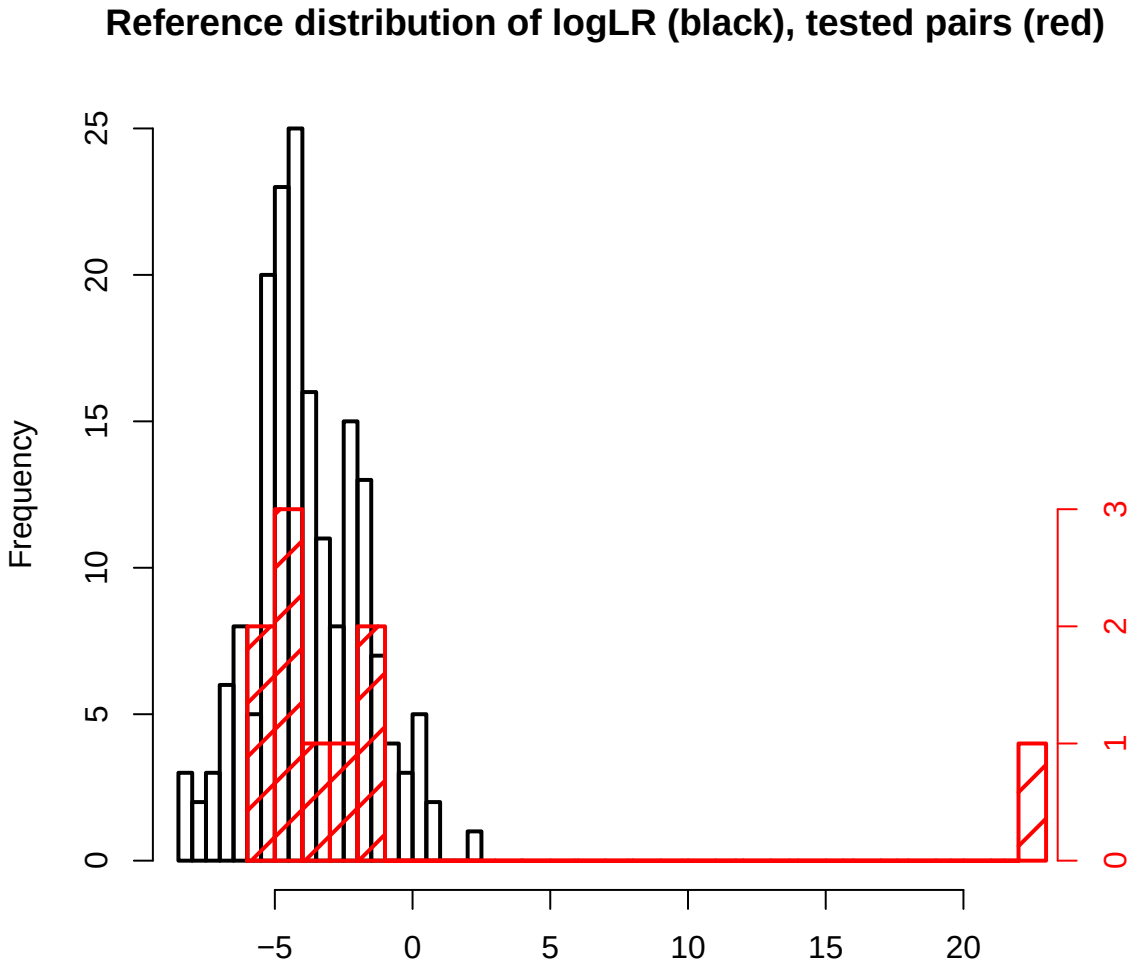
```
> chromosomePlots(results$OneStepSeg, ptlist,ptname="pt.10",nmad=1)
```





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])
```



2.1 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.

3 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    1    0    2    0
 [2,]    2    2    2    0    0
 [3,]    3    0    0    1    0
 [4,]    4    2    2    2    2
 [5,]    5    0    0    1    1
 [6,]    6    2    2    0    2
 [7,]    7    1    1    2    1
 [8,]    8    1    1    2    2
 [9,]    9    0    0    0    1
[10,]   10    0    0    2    0
[11,]   11    0    0    0    2
[12,]   12    1    1    2    0
[13,]   13    2    2    2    0
[14,]   14    1    1    1    2
[15,]   15    2    2    1    0
```

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

```
> 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 13 13 0 1 6   20 2.20457220717234e-08        0
2        2       2  3  6 4 6 4   20    0.633257174327221        1
3        3       3  6  9 2 5 4   20     0.13247418005031    0.458
4        4       4  6  9 4 5 2   20    0.271731009940983    0.807
5        5       5  3  6 7 5 2   20    0.768026723950271        1
6        6       6  1  5 8 3 4   20    0.964059678147575    0.442
7        7       7  6 12 3 4 1   20    0.607636663320756        1
8        8       8  5 11 4 2 3   20    0.585520481546597    0.719
9        9       9  4  7 6 5 2   20    0.597049677704141    0.911
10      10      10  6 10 3 6 1   20    0.424944369195046    0.794
```

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.

4 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 3.3.0 RC (2016-04-26 r70550)
Platform: x86_64-apple-darwin13.4.0 (64-bit)
Running under: OS X 10.9.5 (Mavericks)

locale:
```

```
[1] C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

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

other attached packages:
[1] Clonality_1.20.0 DNAcopy_1.46.0

loaded via a namespace (and not attached):
[1] tools_3.3.0
```