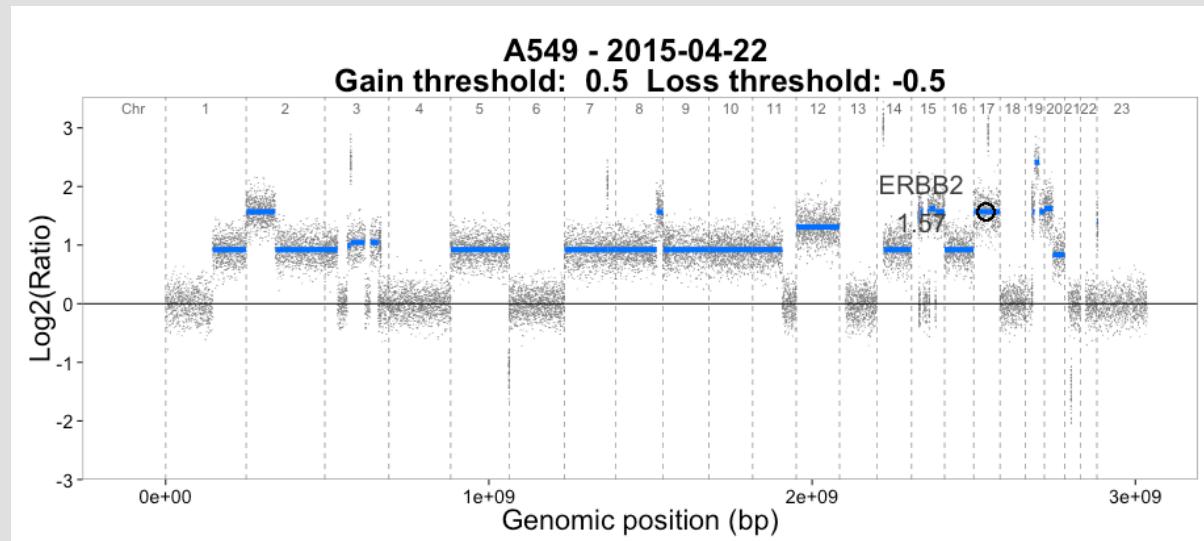


Array-based Genomic Profiles



Frederic Commo
U981 – Bioinformatics

frederic.commo@gustaveroussy.fr

In this module

- Cancer & genome, a very brief overview
- Array-based CGH Technologies
- Genomic Profile analysis
- “Do it yourself!”

- Cancer & genome, a very brief overview
- Array-based CGH Technologies
- Genomic Profile analysis
- “Do it yourself!”



« [...] malignant tumours might be the consequence of a certain abnormal chromosome constitution » Theodor Boveri (1862-1915)



« the evidence shows clearly that the characters of wild animals and plants, as well as those of domesticated races, are inherited both in the wild and in domesticated forms according to the Mendel's Law » Thomas H Morgan (1866-1945)



*The one gene-one enzyme paradigm.
Genetic Control of Biochemical Reactions in Neurospora. Beadle & Tatum, PNAS 1941*

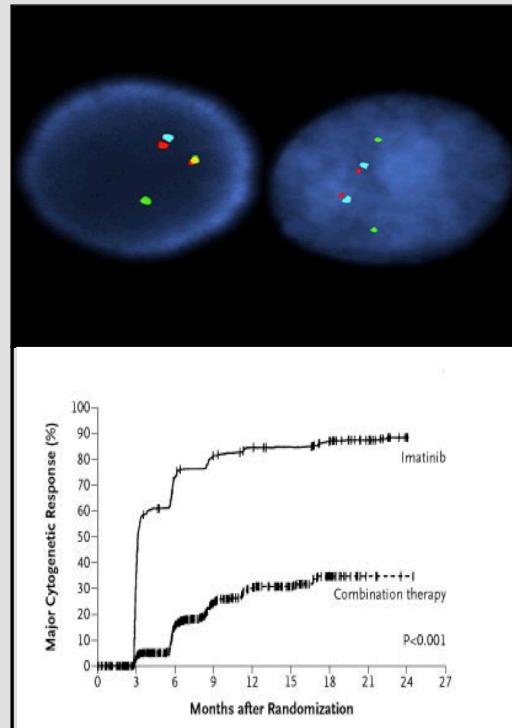


*The effect of mutated Hras on cell lines (fibroblast NCI/3T3)
Mechanism of activation of a human oncogene. Tabin C et al. Nature 1982*

Early 2000s' Relations between molecular alterations and treatment efficacy

BCR-Abl and imatinib

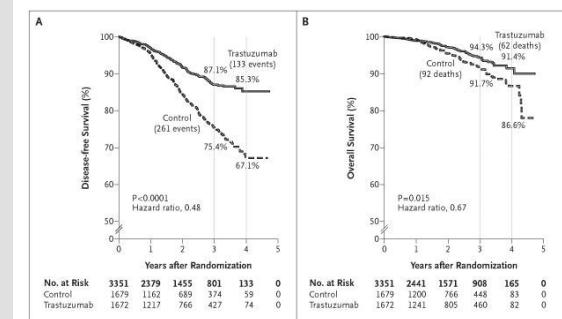
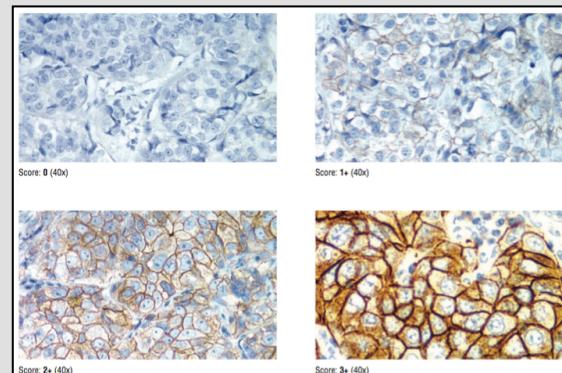
BCR-Abl FISH



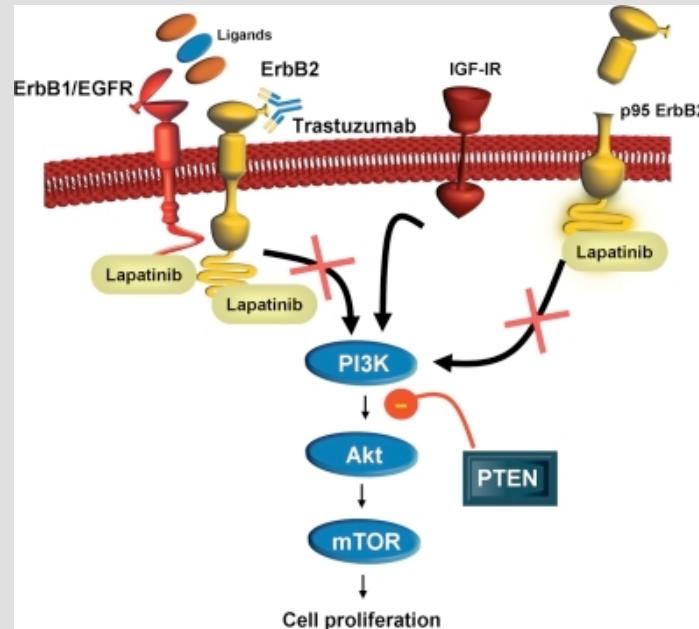
O'Brien et al. N Engl J Med. 2003

ERBB2 and trastuzumab

Her2/neu IHC

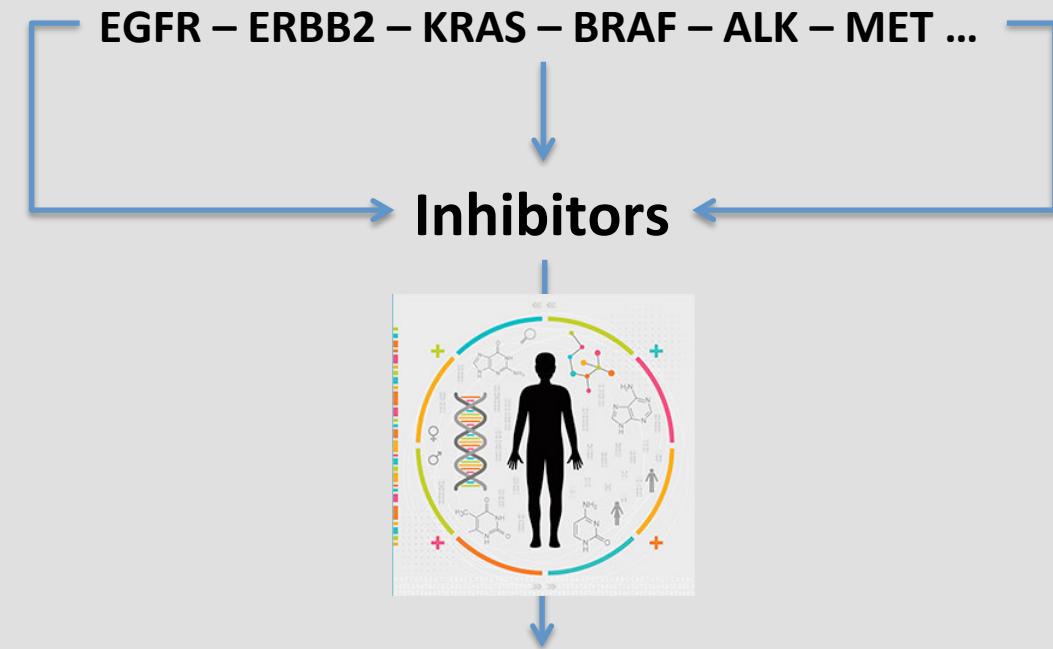


Romond et al. N Engl J Med 2005



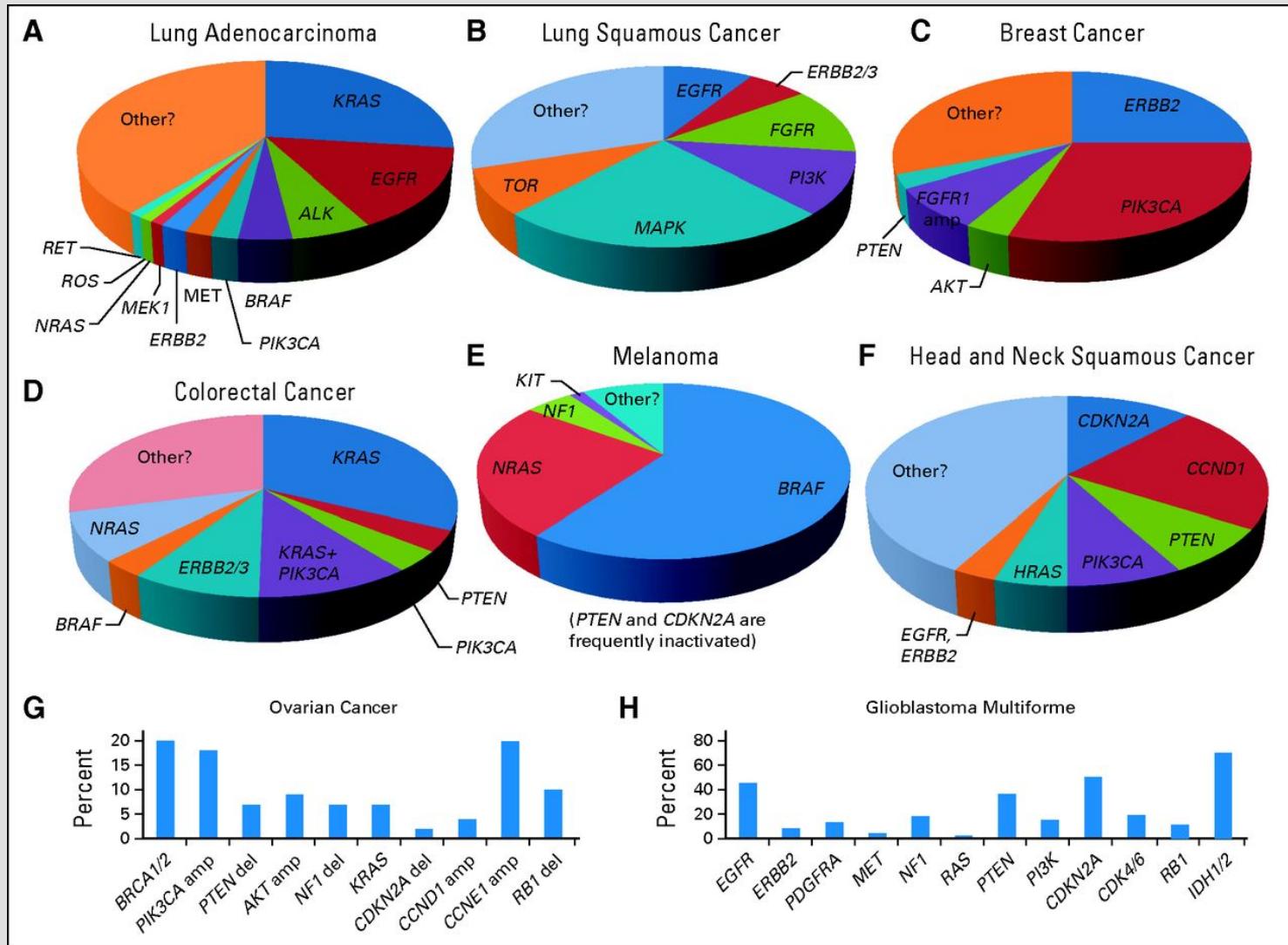
From Vogel et al. Jpn J Clin Oncol. 2010

Matching alterations to targeted therapies



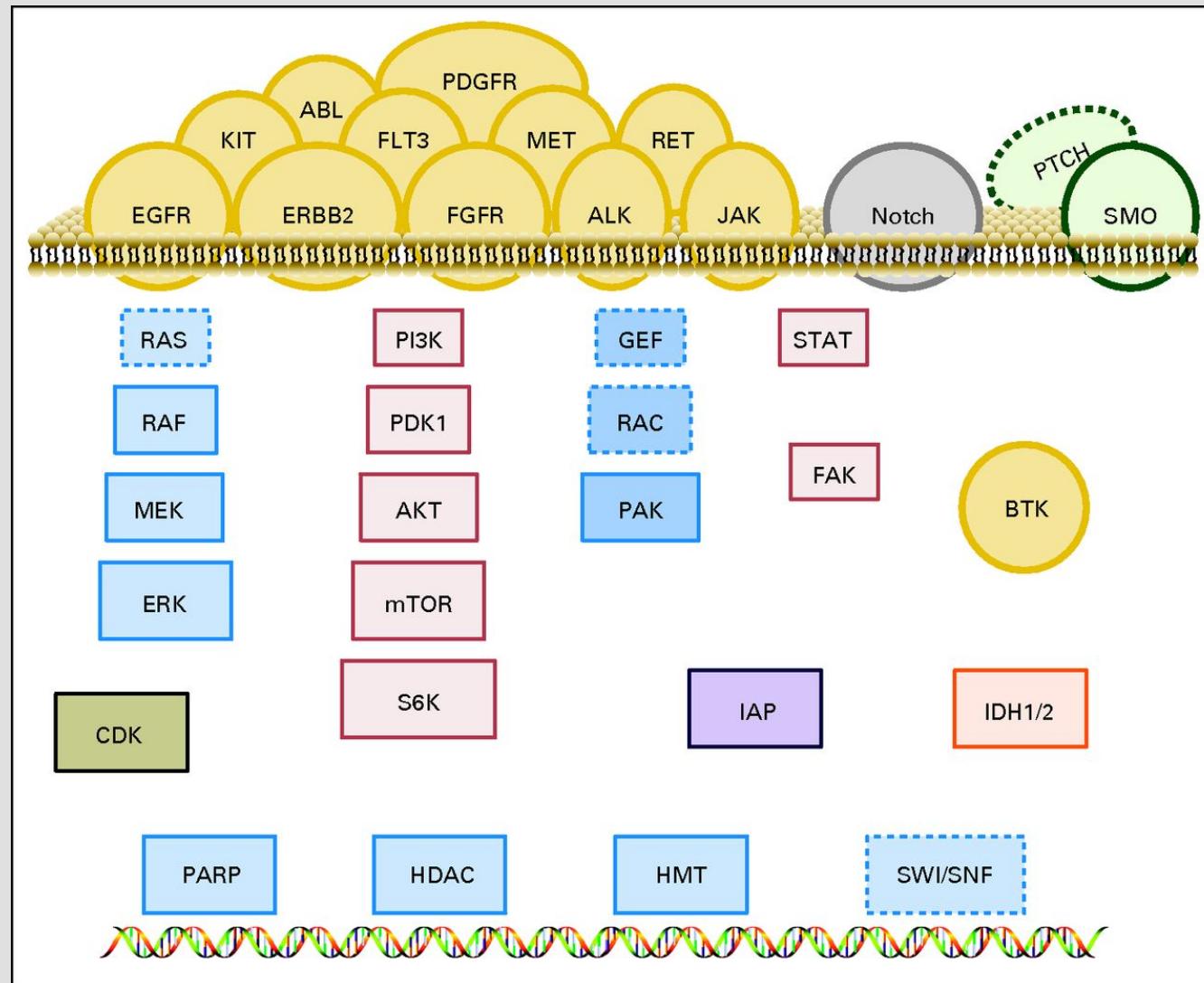
Precision Medicine

Genomic alterations affecting actionable signaling pathways in common solid tumors.



Levi A. Garraway JCO 2013;31:1806-1814

Spectrum of targeted anticancer agents in clinical development.



Levi A. Garraway JCO 2013;31:1806-1814

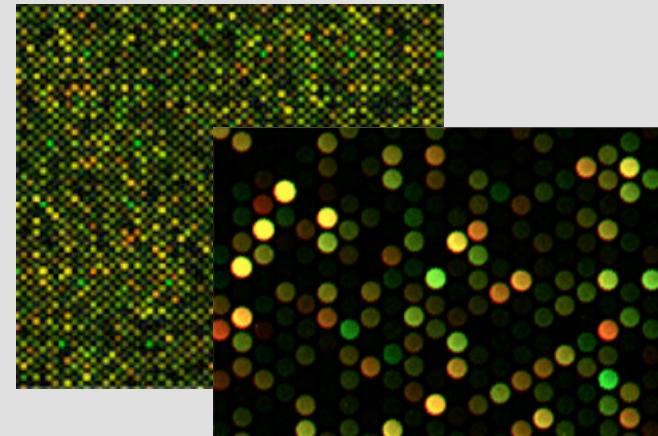
- Cancer & genome, a very brief overview
- Array-based CGH Technologies
- Genomic Profile analysis
- “Do it yourself!”



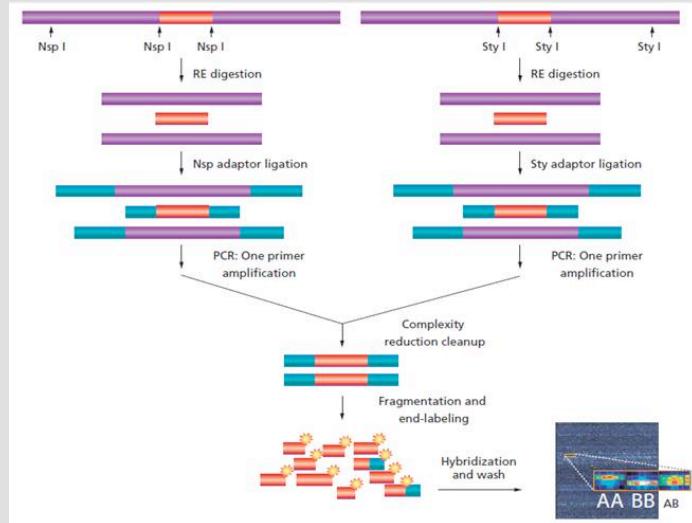
Agilent dual-color hybridization

Competitive hybridization between two DNAs

Sample Vs. Reference



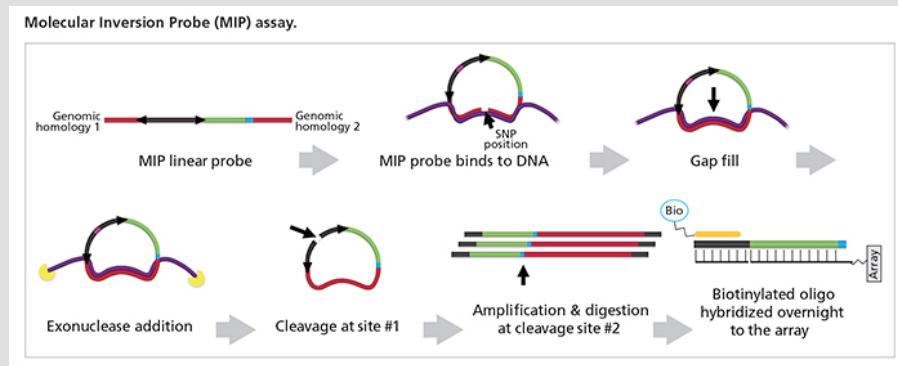
$$\log_2 \left(\frac{\text{sample}}{\text{reference}} \right) = LRR \begin{cases} > & \text{loss} \\ = & \text{neutral} \\ < & \text{gain} \end{cases}$$



Affymetrix technologies

SNP5 → SNP6 → cytoScanHD (~2.7e6 probes)

- Single-color hybridization
- CN + SNP probes
- ‘virtual’ reference (hapmap270)
- Average spacing: 880bp (range: 384 – 1737)

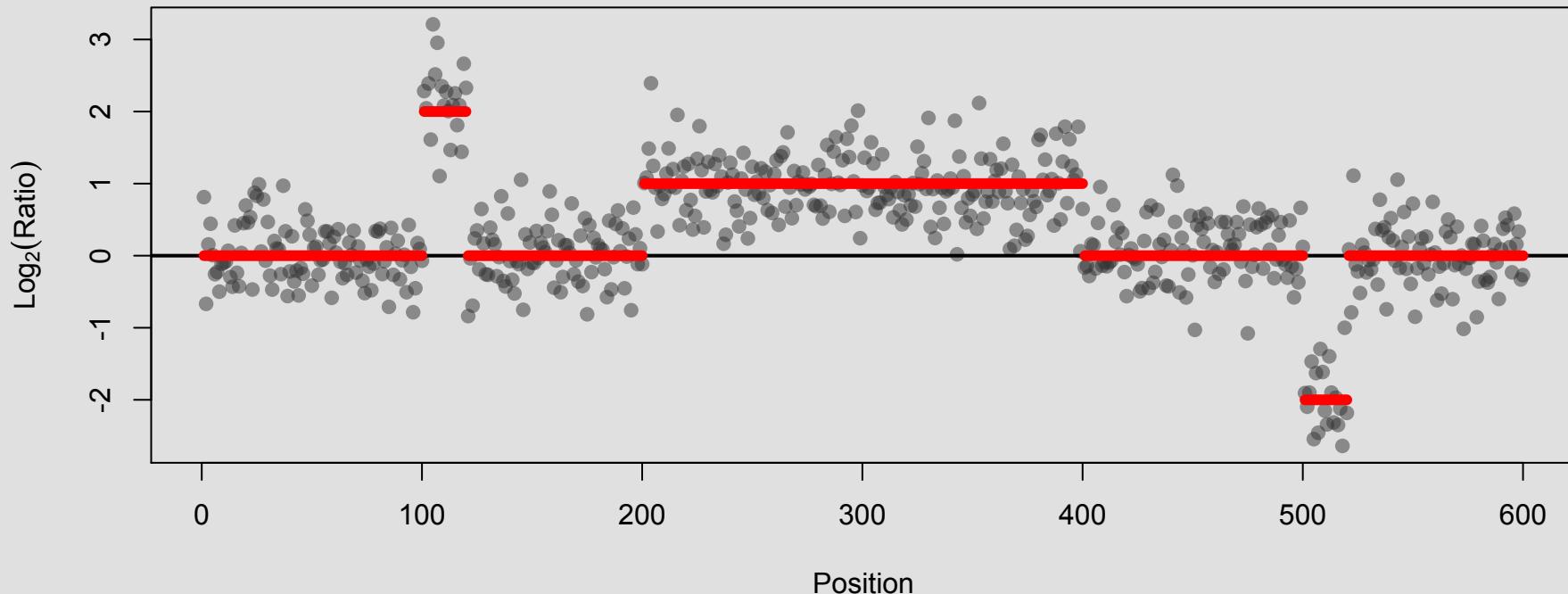


OncoScan, dedicated to FFPE samples
~230K probes exclusively SNP, new technology
- Average spacing: 2.8Kb

- Cancer & genome, a very brief overview
- Array-based CGH Technologies
- Genomic Profile analysis
- “Do it yourself!”

The main idea:

- taking into account the physical relation between markers (probes).
- Summarizing the signal over all the related points.



Issues:

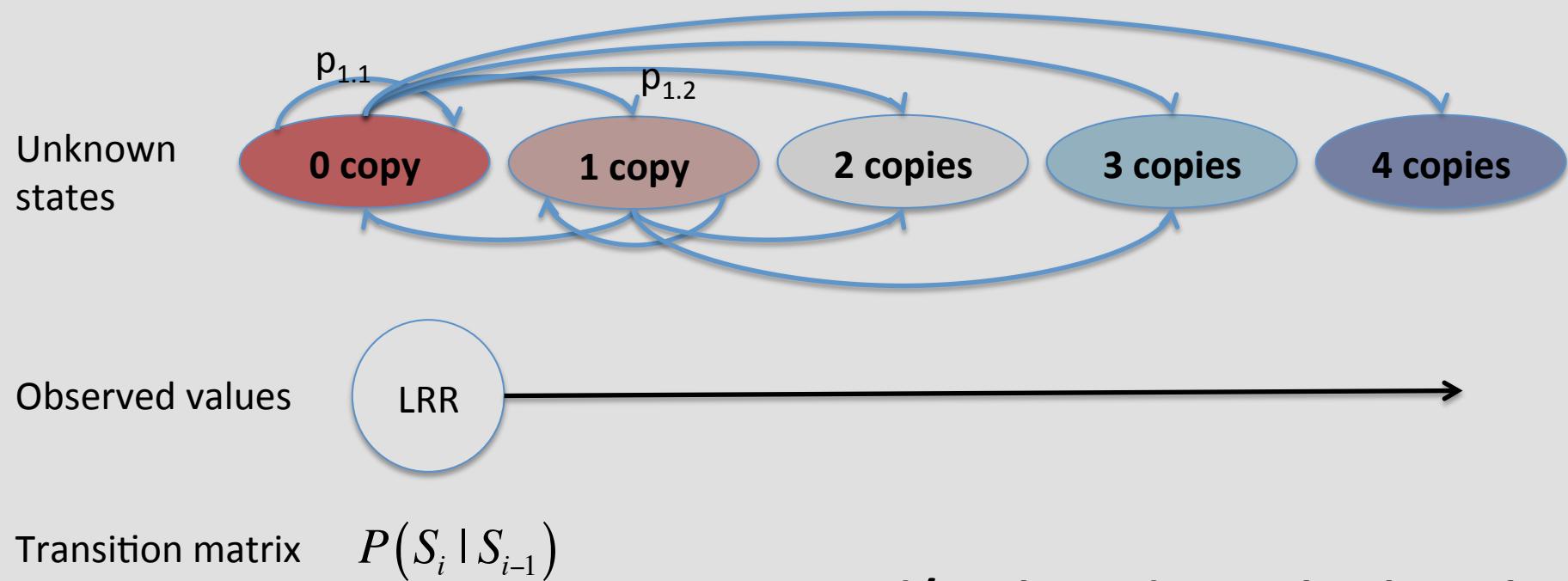
- **How to segment efficiently ?**  **A matter of algorithm**

- **How to estimate gains & losses ?**  **A matter of neutral 2-copies line ... ?**

Segmentation algorithms

The Hidden Markov Model (HMM)

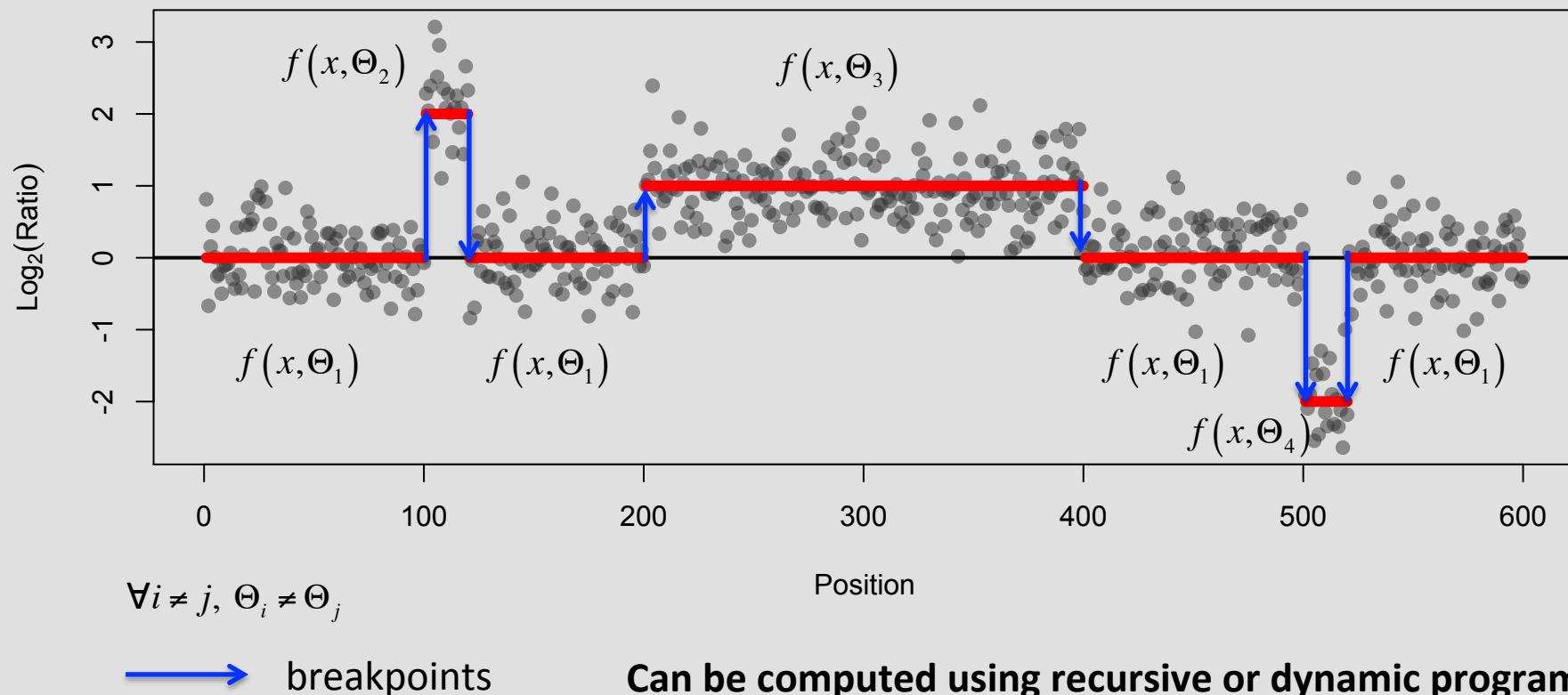
Rational: contiguous states might be somehow linked due to their contiguous positions



Segmentation algorithms

The circular binary segmentation (CBS)

Rational: there exist a sequence of k breakpoints t_1, \dots, t_k such that the LRR mean (and possibly the variance) is the same between two changes, and different from one to another.



Several solutions in R (do not reinvent the wheel...)

BioHMM: a heterogeneous hidden Markov model for segmenting array CGH data.
Marioni et al. Bioinformatics 2006

A faster circular binary segmentation algorithm for the analysis of array CGH data.
Venkatraman E S and Olshen Adam B. Bioinformatics 2007

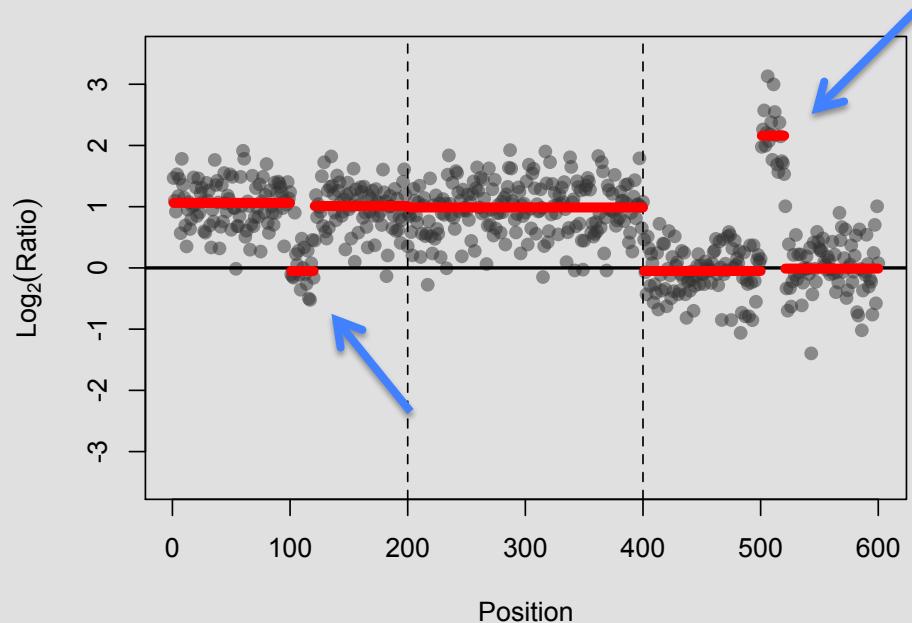
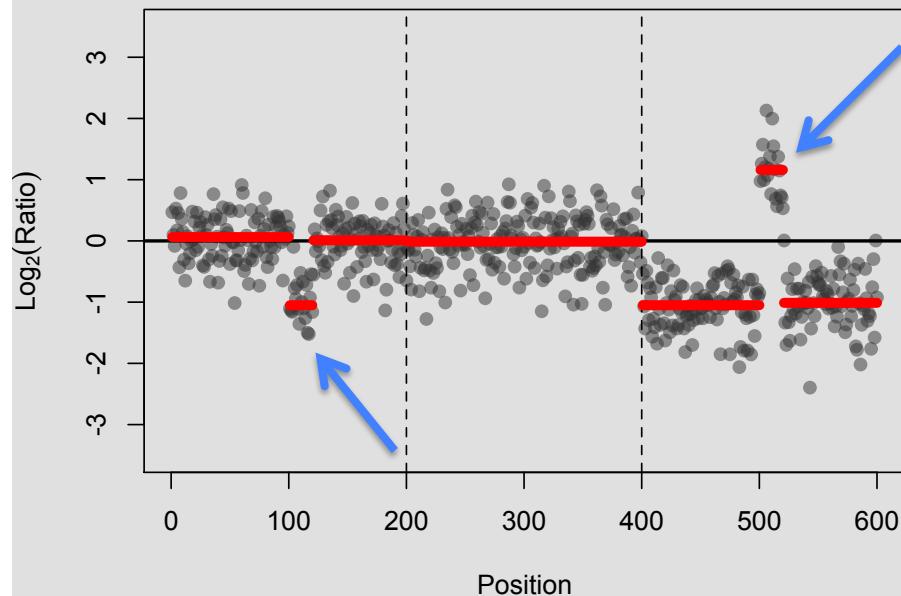
Method comparisons:

A comparison study: applying segmentation to array CGH data for downstream analyses.
Willenbrock H and Fridlyand J, Bioinformatics 2005.

Comparative analysis of algorithms for identifying amplifications and deletions in array CGH data
Lai et al. Bioinformatics 2005

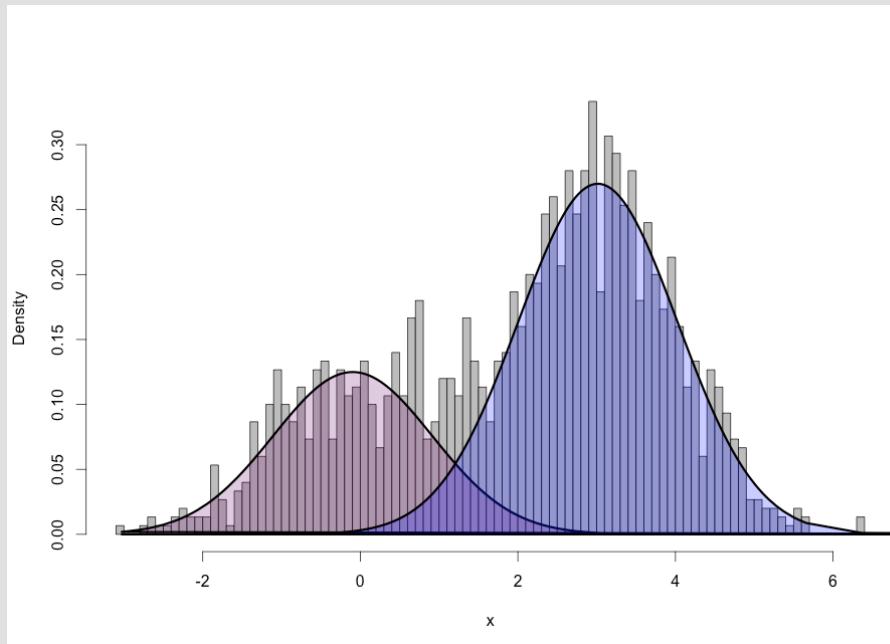
The centralization problem

Depending on the centralization choice, alterations can be different, and so the decision.



Which one is correct ?

Centralizing using the Expectation-Maximization (EM) algorithm



$$g(x, \Theta) = \sum_{k=1}^K \pi_k f_k(x, \theta_k)$$

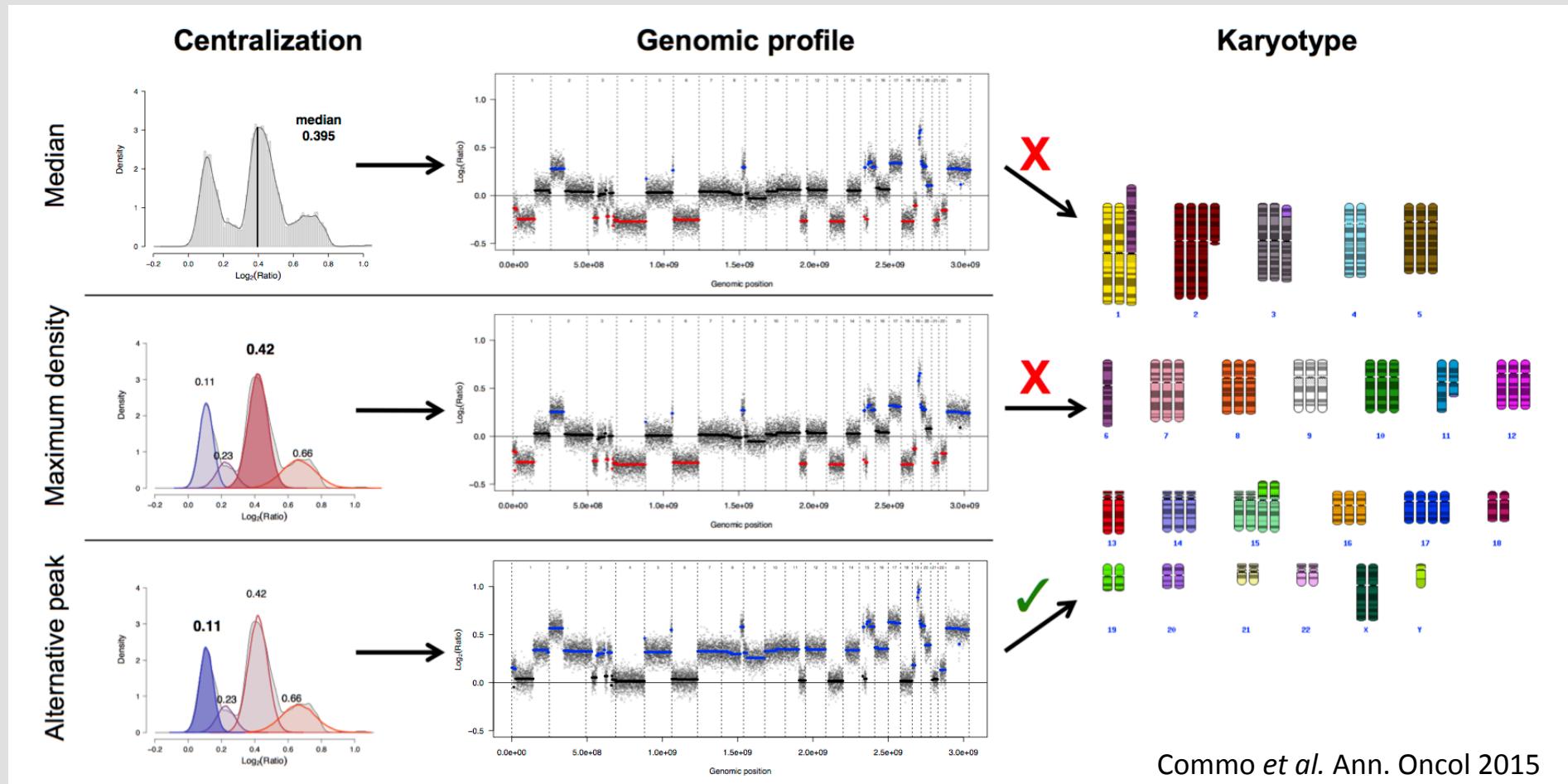
Where K is the number of classes,
 π_k the proportion of class k ,
 θ_k the set of parameters for the density f_k

Do not reinvent the wheel (again...), the R package *mclust* does the job for you.

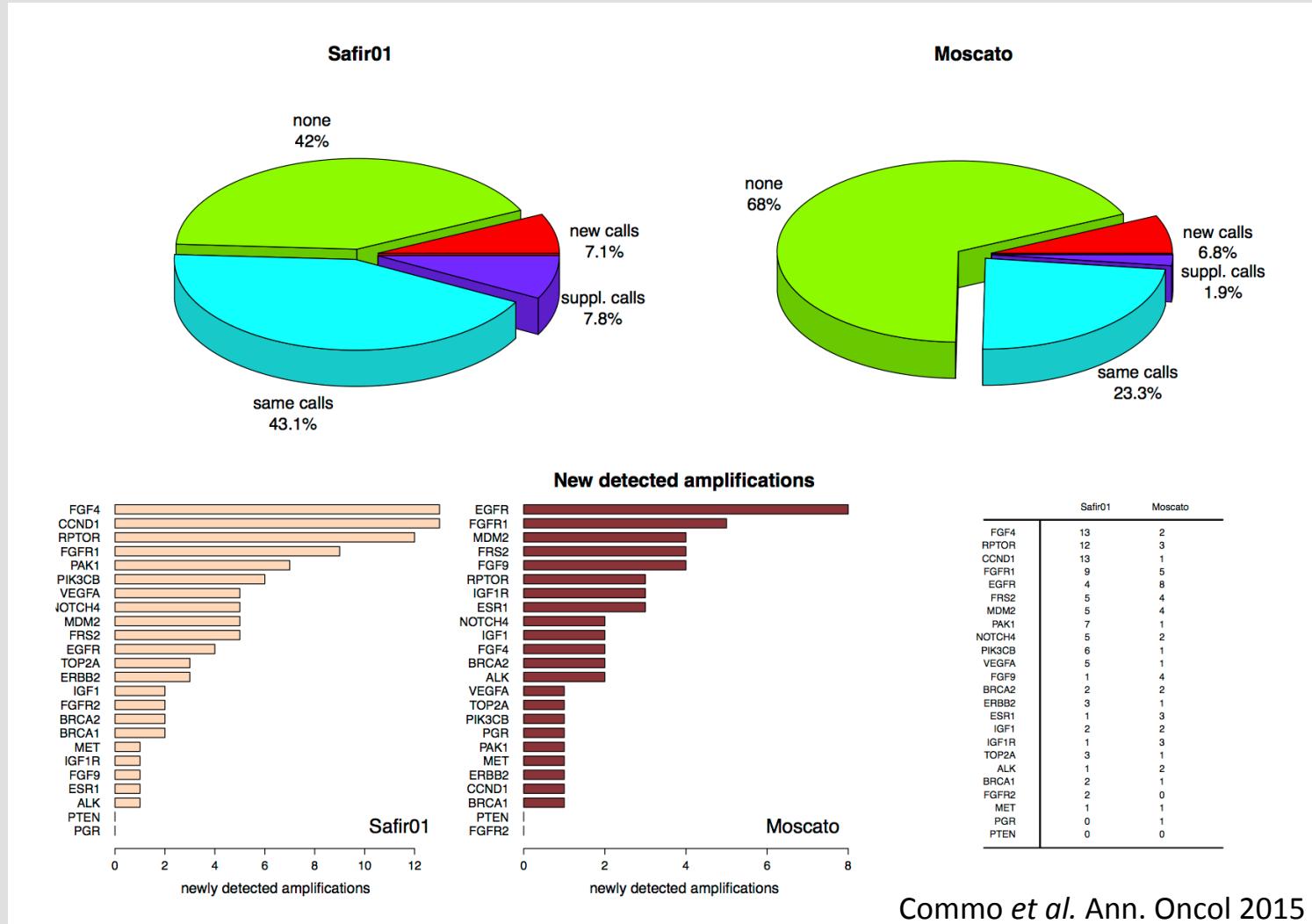
This approach was suggested in Chen *et al.* (Bioinformatics, 2008), where they choose the highest density peak, with a .95 tolerance, for centralizing the LRRs.

Real cases: the NCI-60 cell lines, aCGH profiles compared to skygrams

"NCI and NCBI's SKY/M-FISH and CGH Database (2001), <http://www.ncbi.nlm.nih.gov/sky/skyweb.cgi>"

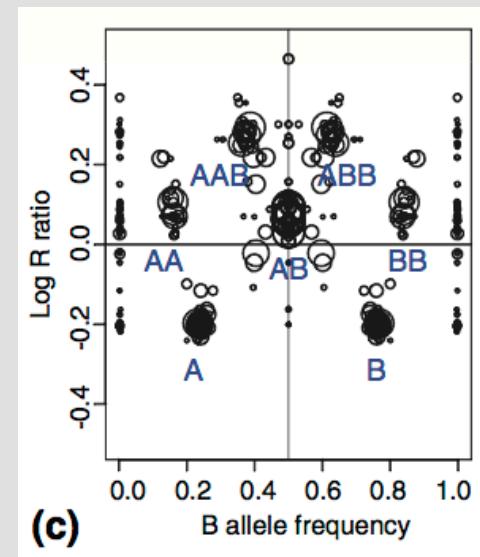
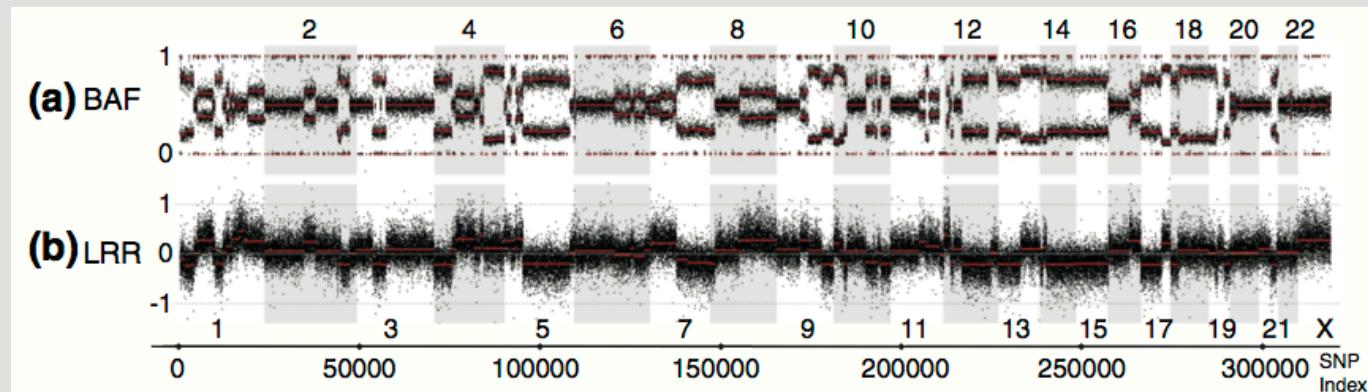


Impact on decisions for a therapeutic orientation



Alternatives

Genome Alteration Print (GAP)



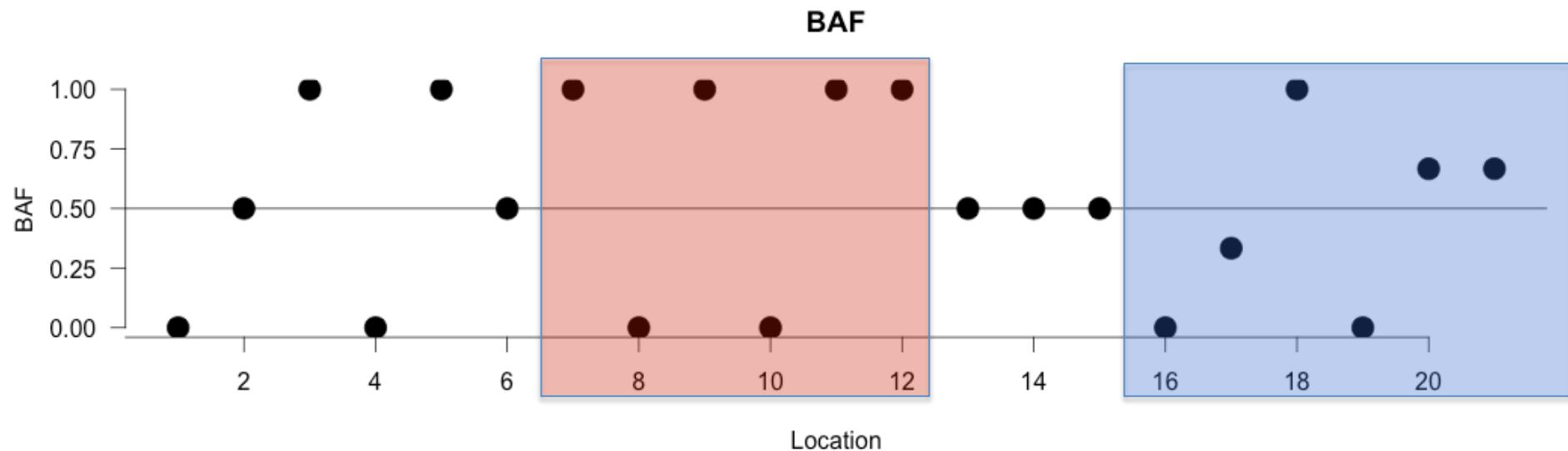
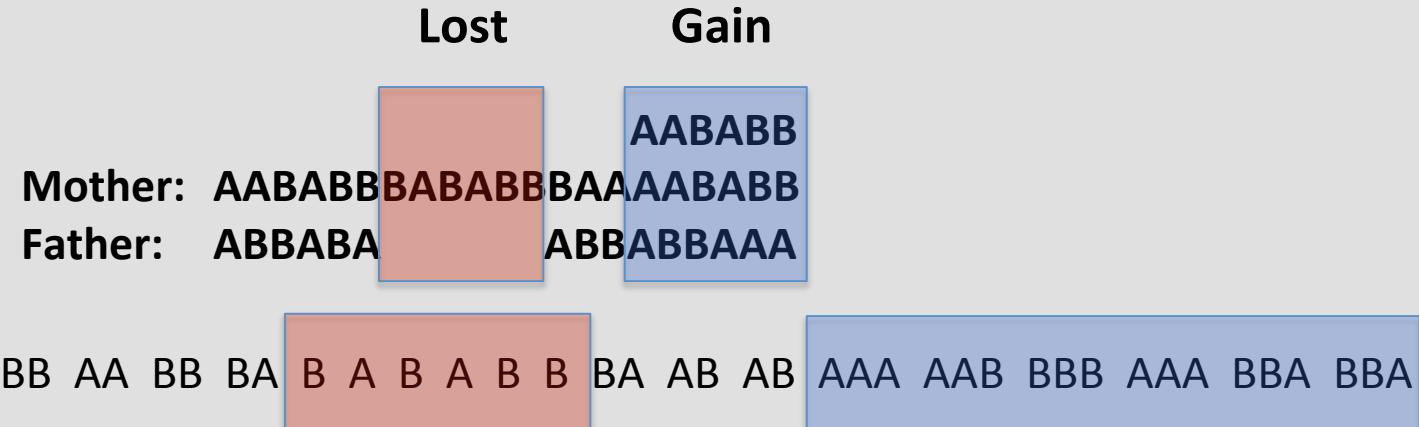
The whole-genome single-nucleotide polymorphism (SNP) array profile and genome alteration print (GAP). The whole-genome profile of genomic rearrangements in the BLC_B1_T45 sample measured by 300K Illumina SNP-array and corresponding GAP. (a) Allelic imbalances are represented by B-allele frequency (BAF). (b) Copy-number variation profile is represented by log R ratio (LRR), centered at zero. (c) The GAP of the sample is a combined sideview projection of segmented LRR and BAF. Each region of the genome is represented by two symmetric circles in the case of allelic imbalance and by one circle centered at $BAF = 0.5$ in the case of a balanced genotype. Attribution of copy numbers and genotypes corresponds to a near-diploid model of rearrangements.

Popova et al. Genome Biology 2009 10:R128 doi:10.1186/gb-2009-10-11-r128

Other methods: Absolute, ASCAT, SOMATICS, PICNIC

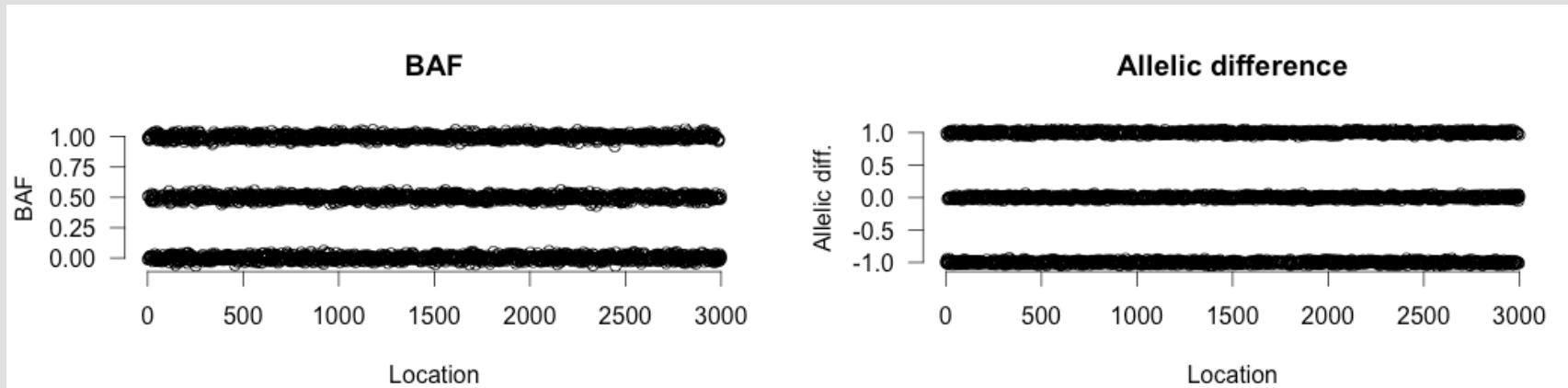
Interpreting loss of heterozygosity (LOH)

$$BAF = \frac{B}{A + B}$$

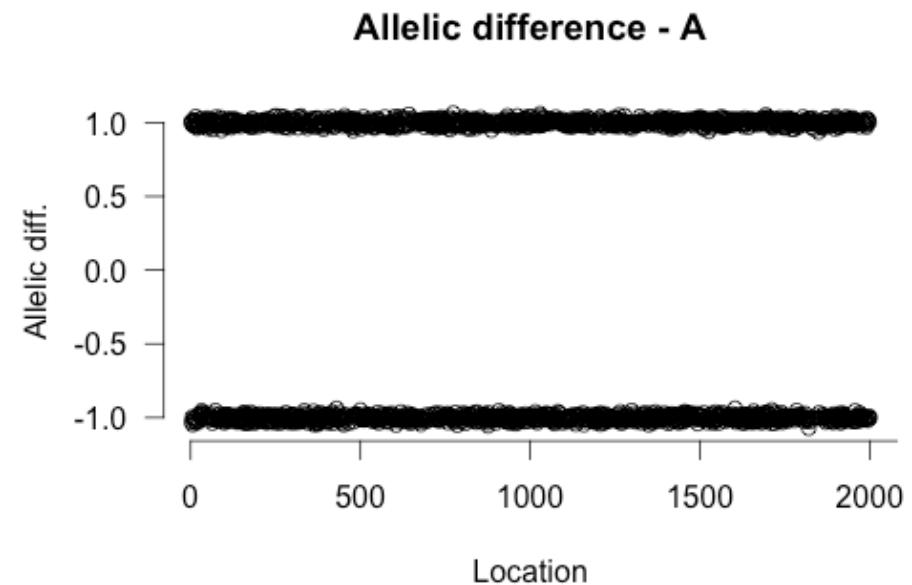
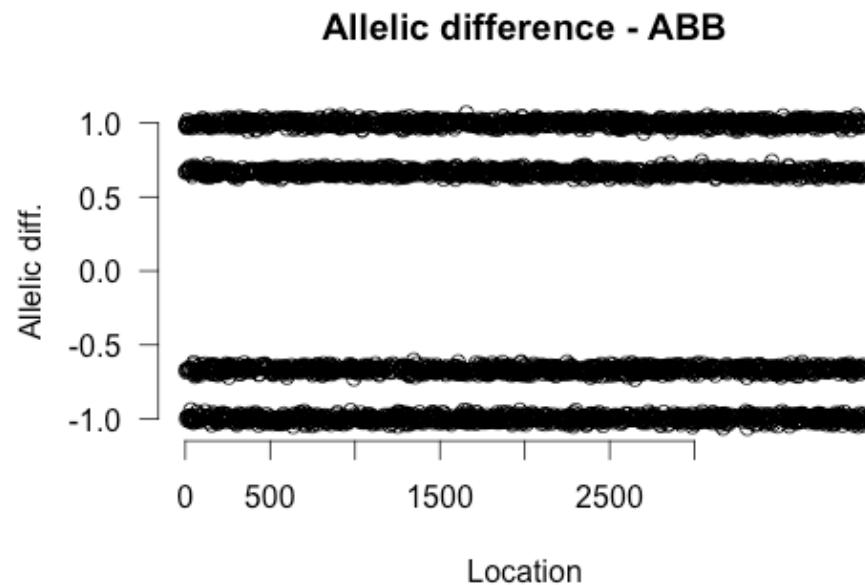


$$BAF = \frac{B}{A + B} ; \text{ Allelic.Difference} = \frac{A - B}{A + B}$$

| snp | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----------------------------|---|----|-----|----|---|---|---|---|----|-----|
| Genotype A | a | b | a | b | b | a | a | b | b | a |
| Genotype B | a | b | b | b | a | a | a | a | b | b |
| BAF = B/(A+B) | 0 | 1 | 0.5 | 1 | 0 | 0 | 0 | 0 | 1 | 0.5 |
| Allelic diff = (A-B)/(A+B) | 1 | -1 | 0 | -1 | 0 | 1 | 1 | 0 | -1 | 0 |

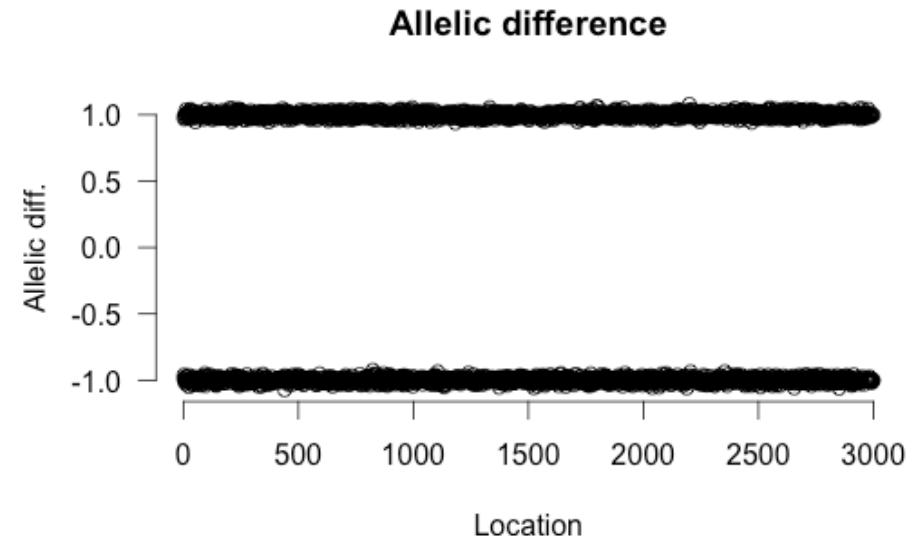
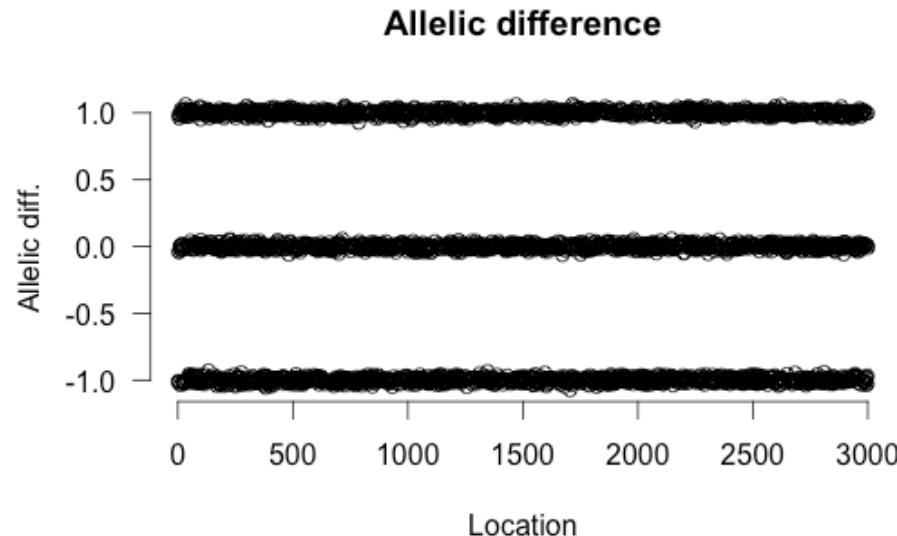


| Probe location | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----------------------------|---|----|------|----|-----|---|---|-----|----|------|
| Genotype A | a | b | a | b | b | a | a | b | b | a |
| Genotype B | a | b | b | b | a | a | a | a | b | b |
| Genotype B | a | b | b | b | a | a | a | a | b | b |
| Allelic diff = (A-B)/(A+B) | 1 | -1 | -2/3 | -1 | 2/3 | 1 | 1 | 2/3 | -1 | -2/3 |



The problem is...

Since signals are adjusted such that $AA = 1$ and $BB = -1$, and so 0 is AB ...

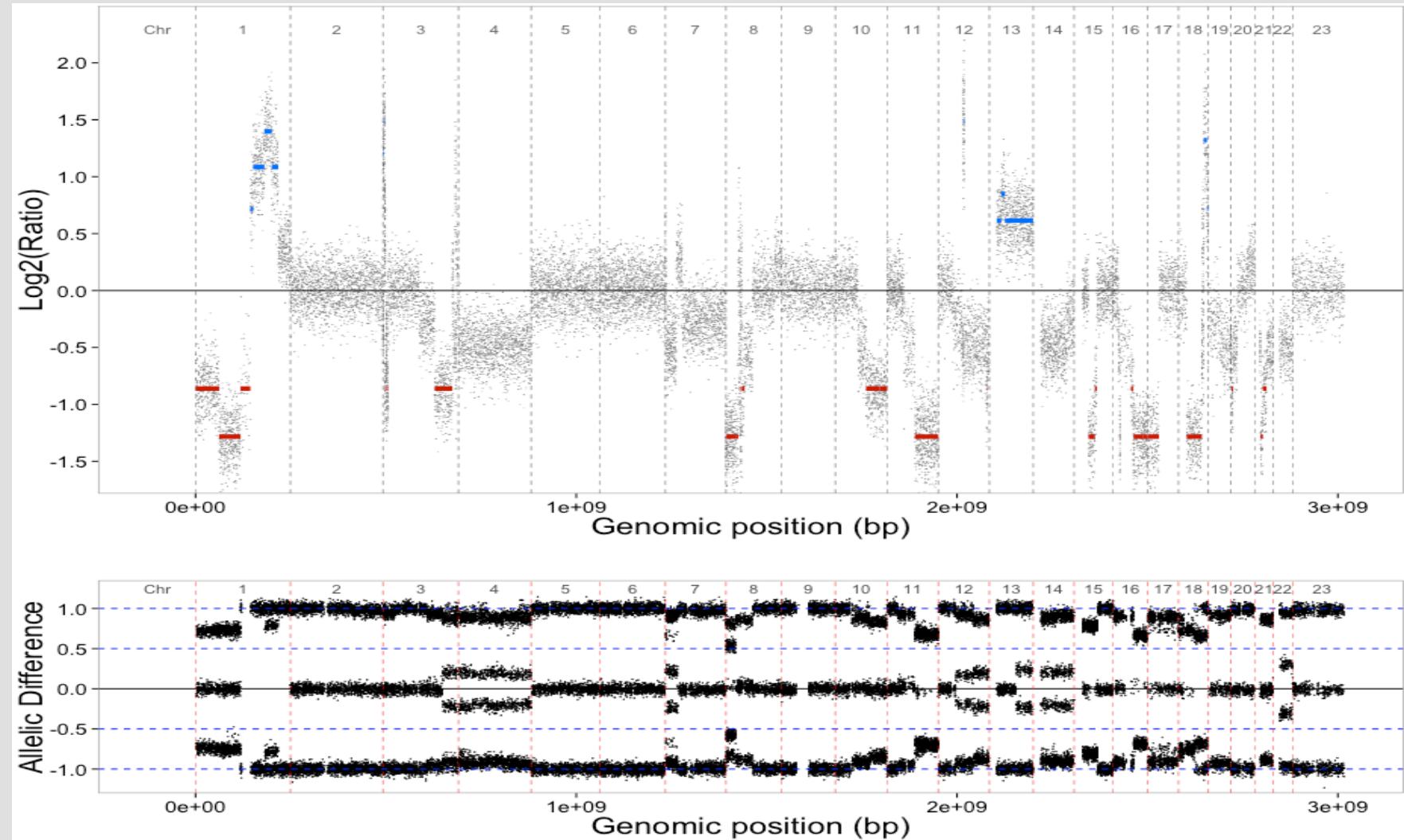


Can be any balanced genotype: AB or AABB

Can be any genotype A or AA or AAA

In case of mosaicism, things can be even more complicated...

A real case...



- Cancer & genome, a very brief overview
- Array-based CGH Technologies
- Genomic Profile analysis
- “Do it yourself!”

```
#####
# Using DNAcopy
#####

> require(DNAcopy)
> op <- par(no.readonly = TRUE)

# Loading supp. data
> path <- "/Users/fredcommo/Documents/myProjects/IFSBM/data"
> load(file.path(path, "hg19.rda"))
> load(file.path(path, "geneDB.rda"))

# Reading file
> filePath <- file.path(path, "Affy_cytoScan.cyhd.CN5.CNCHP_short.txt.gz")
> foo <- readLines(filePath, n=750)
> idx <- grep("ProbeSet", foo)
> cnSet <- read.delim(filePath, skip=idx-1, stringsAsFactors=FALSE)
> dim(cnSet)
> head(cnSet)
```

| | ProbeSetName | Chromosome | Position | CNState | Log2Ratio | SmoothSignal | LOH | Allele.Difference |
|---|--------------|------------|----------|---------|-----------|--------------|-----|-------------------|
| 1 | C-7SARK | 1 | 849467 | 2 | 0.059410 | 1.859233 | 0 | NA |
| 2 | C-6HYCN | 1 | 874571 | 2 | -0.209187 | 1.855083 | 0 | NA |
| 3 | C-7SEBI | 1 | 874841 | 2 | -0.391675 | 1.852993 | 0 | NA |
| 4 | S-3WRNV | 1 | 882803 | -1 | 0.000000 | 2.000000 | 0 | -0.47663 |
| 5 | C-7SBEJ | 1 | 884724 | 2 | -0.058026 | 1.854296 | 0 | NA |
| 6 | C-5YZUW | 1 | 884761 | 2 | 0.128548 | 1.850769 | 0 | NA |

```
# Filtering SNP probes
> cnSet <- cnSet[grep("^S", cnSet$ProbeSet),]
> dim(cnSet)
> head(cnSet)

  ProbeSetName Chromosome Position CNState Log2Ratio SmoothSignal LOH Allele.Difference
4      S-3WRNV          1    882803      -1  0.000000  2.000000  0       -0.476630
9      S-4GXBG          1    888659       2 -0.296734  1.856312  0        0.604524
23     S-4LYTY          1    918573       2  0.440532  1.888353  0        0.258405
24     S-4HQZX          1    920733       2  0.092134  1.891228  0        1.078725
46     S-4KCPC          1   1039857       2 -0.184646  1.958233  0       -0.706869
47     S-3UOZS          1   1041366       2 -0.071947  1.960523  0       -0.014035
```

```
# Checking Chr names
> table(cnSet$Chromosome)

 1   10   11   12   13   14   15   16   17   18   19   2   20   21   22   3
14913  9329 10249  8950  7328  7017  6624  5488  4664  5467  2904 16500  4255  2667  2280 13735
      4      5      6      7      8      9      X
12947 12383 13733 12208 10245  8060  5748

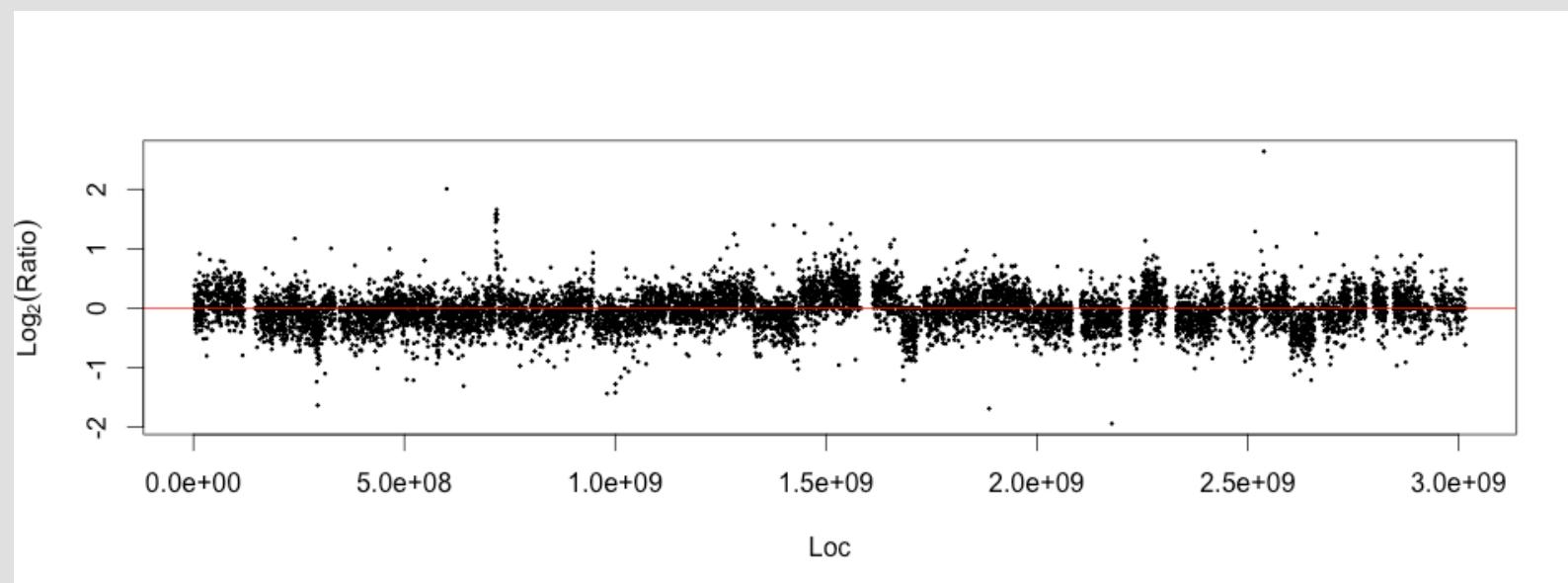
> cnSet$Chromosome[cnSet$Chromosome=="X"] <- 23
> cnSet$Chromosome <- as.numeric(cnSet$Chromosome)
> table(cnSet$Chromosome)

 1   2   3   4   5   6   7   8   9   10   11   12   13   14   15   16
14913 16500 13735 12947 12383 13733 12208 10245  8060  9329 10249  8950  7328  7017  6624  5488
      17    18    19    20    21    22    23
      4664   5467   2904   4255   2667   2280   5748

> cnSet <- cnSet[order(cnSet$Chromosome, cnSet$Position),]
```

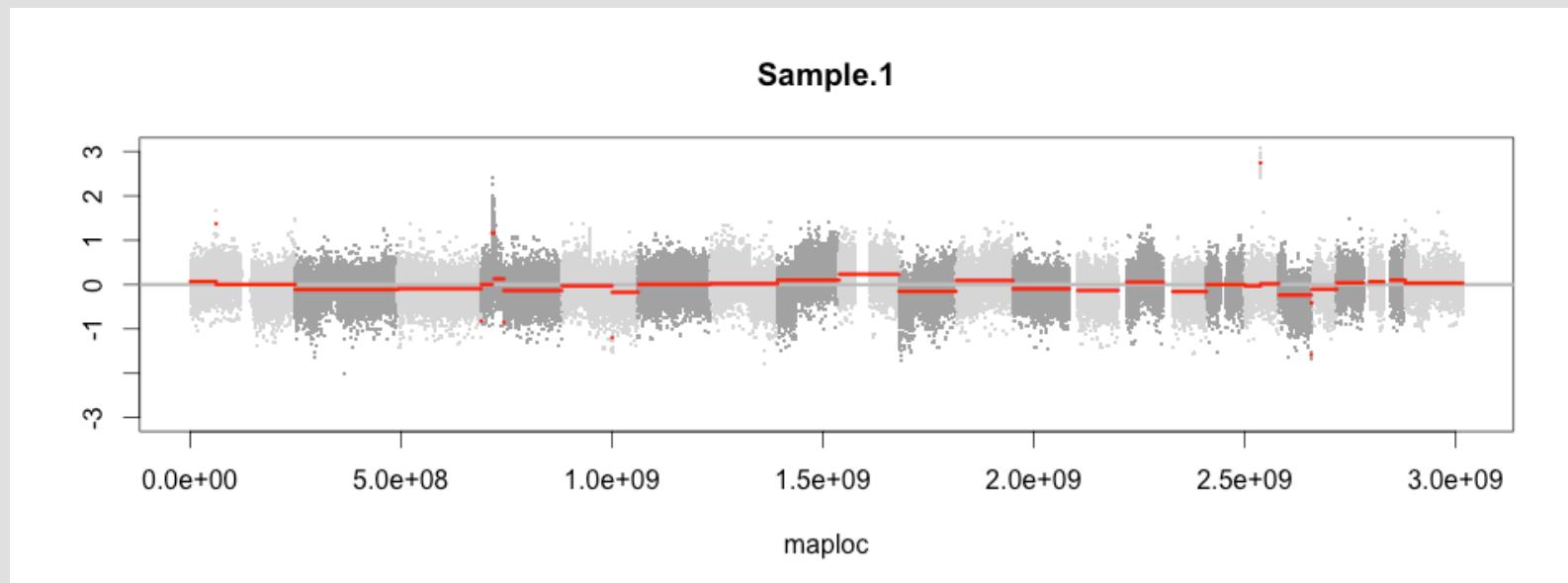
```
# Calculating genomic positions
> locs <- lapply(1:24, function(chr) {
  l <- cnSet$Position[cnSet$Chromosome==chr]
  return(l + hg19$cumlen[chr])
})
> locs <- do.call(c, locs)
> s <- sample(1:nrow(cnSet), 1e4)

# A quick preview
> plot(locs[s], cnSet$Log2Ratio[s], cex=.2, xlab="Loc", ylab=expression(Log[2]
(Ratio)))
> abline(h = 0, col="red")
```



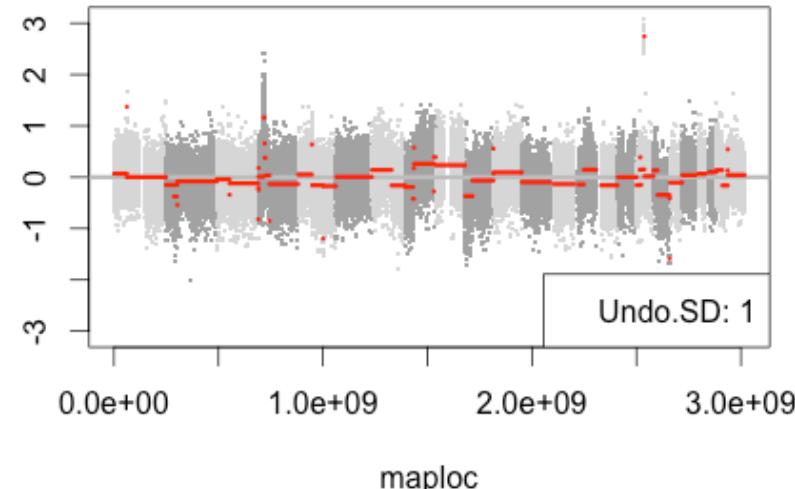
```
# Segmenting using DNAcopy and the default params
  # Constructing a DNAcopy object
> LR <- cnSet$Log2Ratio
> Chr <- cnSet$Chromosome
> ptcols <- c("grey65", "grey85")
> cnaObj <- CNA(LR, Chr, locs, presorted = TRUE)
> cnaObj <- smooth.CNA(cnaObj)
> segObj <- segment(cnaObj, undo.splits = "sdundo")

  # Plotting
> ptcols <- c("grey65", "grey85")
> plot(segObj, xmaploc=TRUE, pt.cols=ptcols)
```

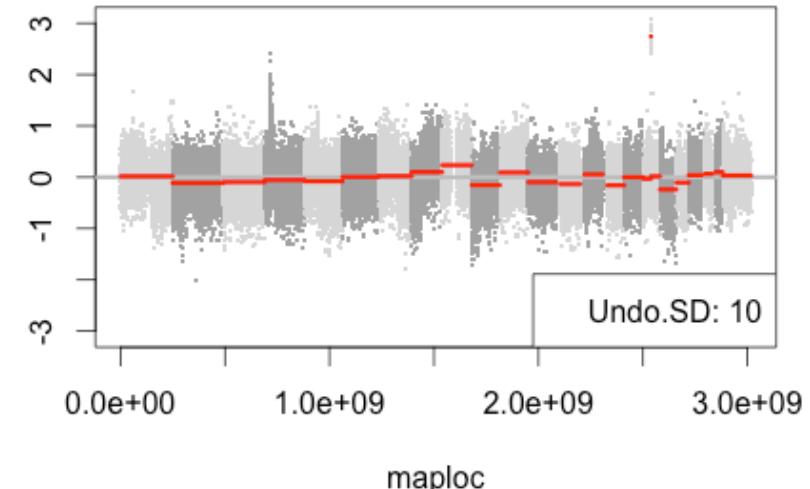


```
# The Undo.SD effect
> par(mfrow=c(1, 2))
> for(s in c(1, 10)){
  segObj <- segment(cnaObj, undo.splits = "sdundo", undo.SD = s)
  plot(segObj, xmaploc=TRUE, pt.cols=ptcols)
  legend("bottomright", legend=sprintf("Undo.SD: %s", s))
}
> par(op)
```

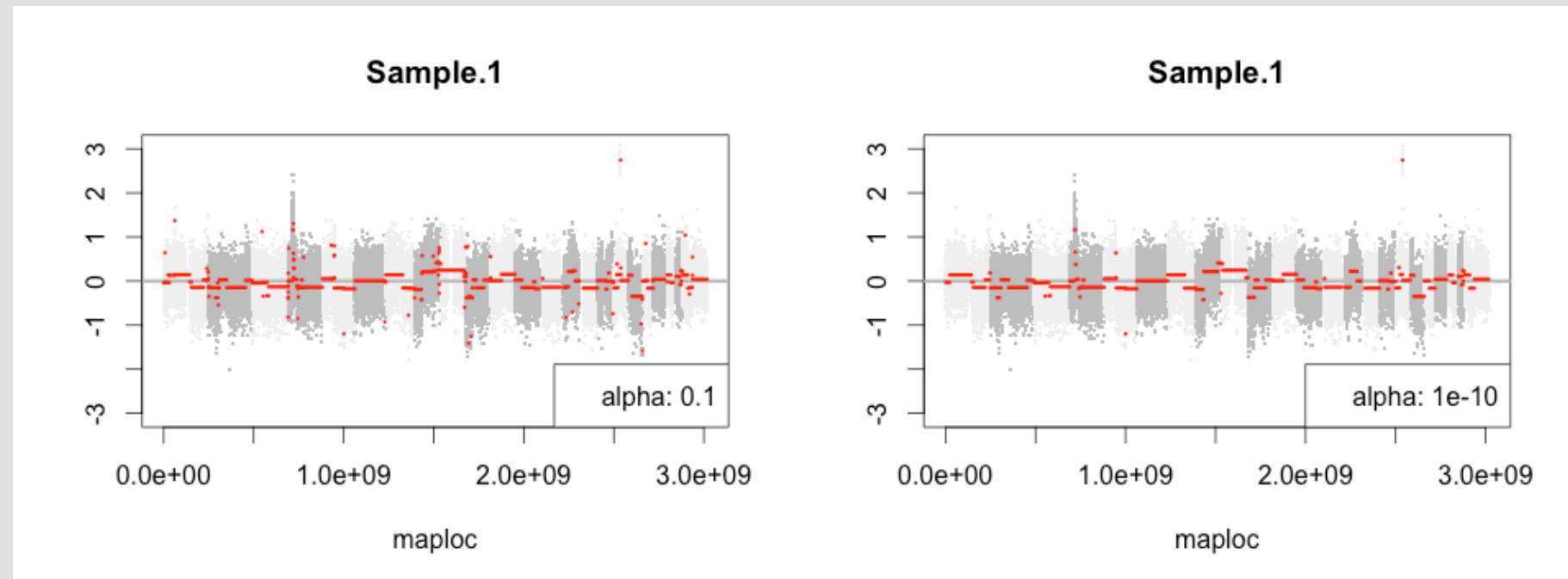
Sample.1



Sample.1



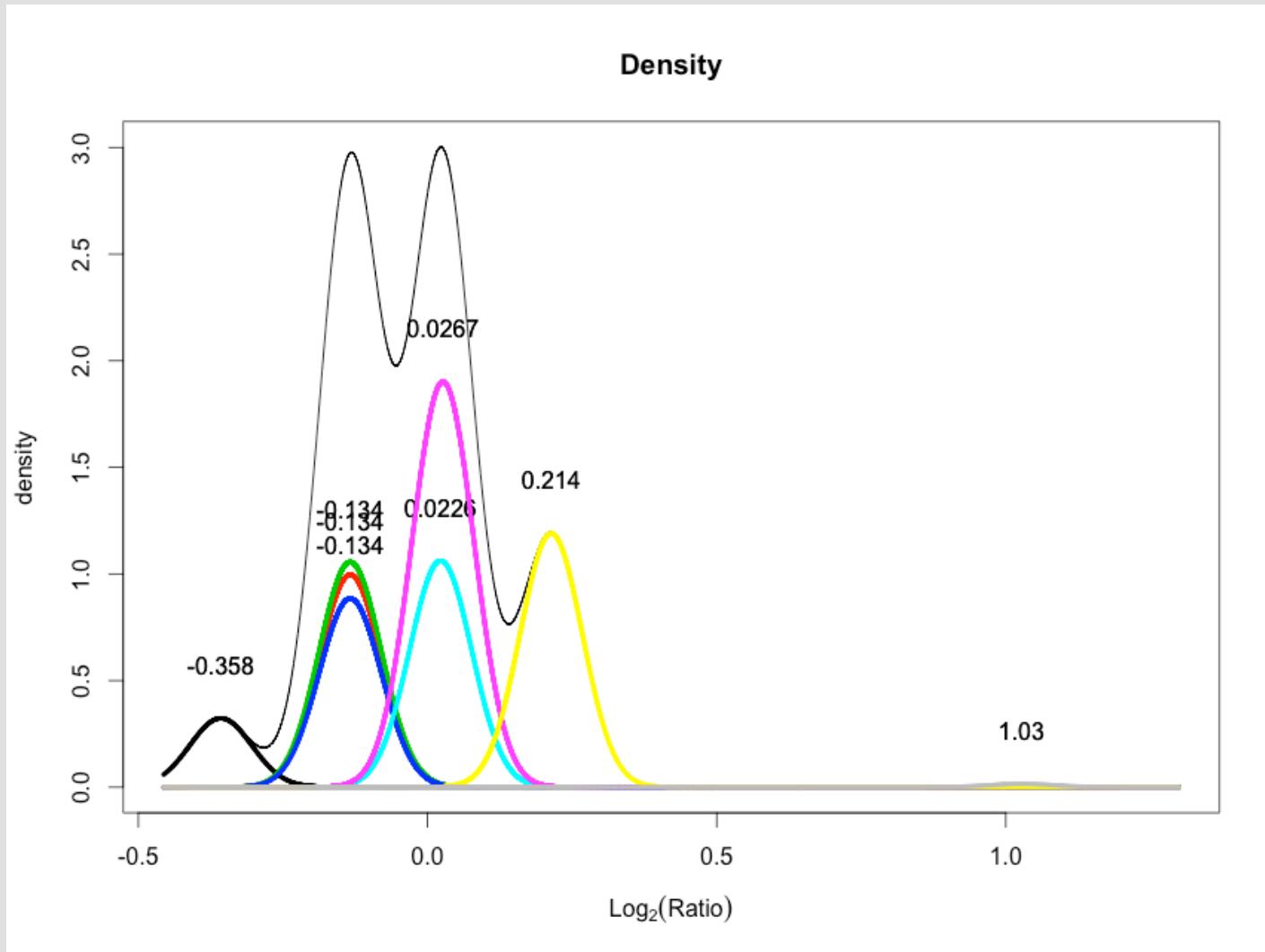
```
# The alpha tolerance effect
> par(mfrow=c(1, 2))
> for(a in c(1e-1, 1e-10)){
  segObj <- segment(cnaObj, undo.splits = "sdundo", undo.SD = .5, alpha = a)
  plot(segObj, xmaploc=TRUE, pt.cols=c("grey75", "grey95"))
  legend("bottomright", legend=sprintf("alpha: %s", a))
}
> par(op)
```



```
# Centering the profile
> require(mclust)
> rLR <- runmed(LR, k=101)
> rLR <- sort(rLR)
> idx <- seq(1, length(rLR), len=25e3)
> model <- Mclust(rLR[idx])
> means <- model$parameters$mean
> props <- model$parameters$pro
> s2 <- model$parameters$variance$sigmasq
> if(length(s2)==1) s2 <- rep(s2, length(means))

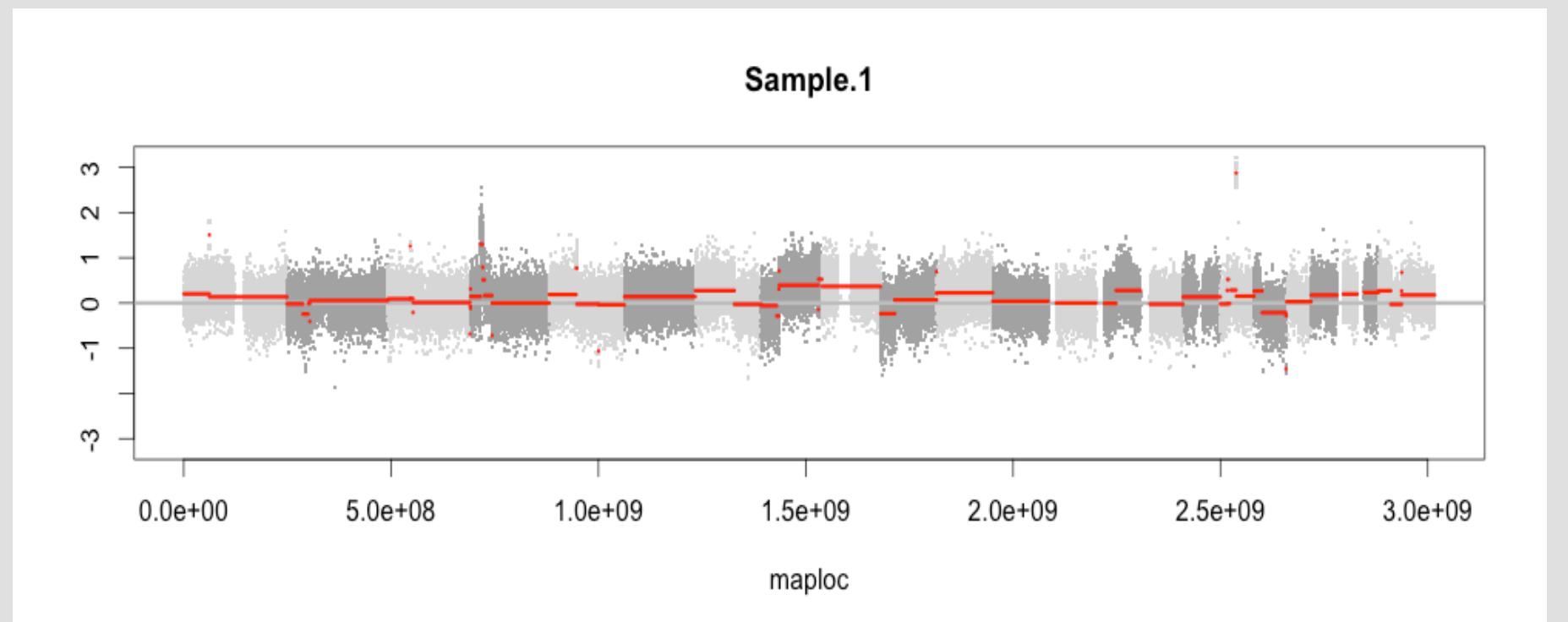
> plot(model, what="density", xlab=expression(Log[2](Ratio)))
> for(ii in 1:length(means)){
  m <- means[ii]; s <- sqrt(s2[ii]); p <- props[ii]
  d <- dnorm(rLR, m, s)*p
  lines(rLR, d, col=ii, lwd=4)
  text(m, max(d)+.25, labels=format(m, digits=3))
}
```

Expectation-Maximization modeling



```
# Final profile
> choice <- -0.134
> LR <- LR - choice

> cnaObj <- CNA(LR, Chr, locs, presorted = TRUE)
> cnaObj <- smooth.CNA(cnaObj)
> segObj <- segment(cnaObj, undo.splits = "sdundo", undo.SD = 1, alpha = 1e-2)
> plot(segObj, xmaploc=TRUE, pt.cols=ptcols)
```



```
# Getting the gene list within a specific segment
> st <- segObj$output
> chr <- 17
> sgt <- st[st$chrom==chr,]
> subdb <- geneDB[geneDB$chr==chr,]
> s <- sgt$loc.start[6]
> e <- sgt$loc.end[6]

> idx <- which(s <= subdb$genomStart & subdb$genomEnd<=e)
> subdb[idx, c("symbol", "fullName")]

  symbol                               fullName
4560   CDK12          cyclin-dependent kinase 12
8489  ERBB2 v-erb-b2 avian erythroblastic leukemia viral oncogene homolog 2
9428   FBXL20         F-box and leucine-rich repeat protein 20
11279  GRB7           growth factor receptor-bound protein 7
13682  IKZF3          IKAROS family zinc finger 3 (Aiolos)
15386  TCAP           titin-cap
16967  MED1           mediator complex subunit 1
17252  MIEN1          migration and invasion enhancer 1
18670  MIR4728        microRNA 4728
20625  NEUROD2        neuronal differentiation 2
23462  PNMT           phenylethanolamine N-methyltransferase
23572  PGAP3          post-GPI attachment to proteins 3
24598  PPP1R1B        protein phosphatase 1, regulatory (inhibitor) subunit 1B
35517  STARD3         STAR-related lipid transfer (START) domain containing 3
```

Any question ?