Description of the CAPweb Analysis Parameters

Philippe Hupé and Pierre Neuvial

March 23, 2006

Contents

1 Overview

This document describes the parameters used during the CAPweb Analysis. During the analysis two steps are peformed:

1. Data normalization is performed using the MANOR package, which includes a spatial normalization step based on a spatial trend computation by two-dimensional LOESS and unsupervised classification with spatial contraint.

2. Breakpoint detection and status assignment (gain, loss, normal, amplicon) is performed using the GLAD package.

Each parameters regarding these two steps are described in what follows.

2 MANOR parameters for spatial normalization

2.1 Short description of the algorithm

Array-CGH data need appropriate within-array normalization methods, and especially methods that handle local spatial effects. This point is illustrated figure 1): high log-ratios cluster in the upper-right corner of the array, whereas the positions of clones on the array are not correlated with their actual positions in the genome.

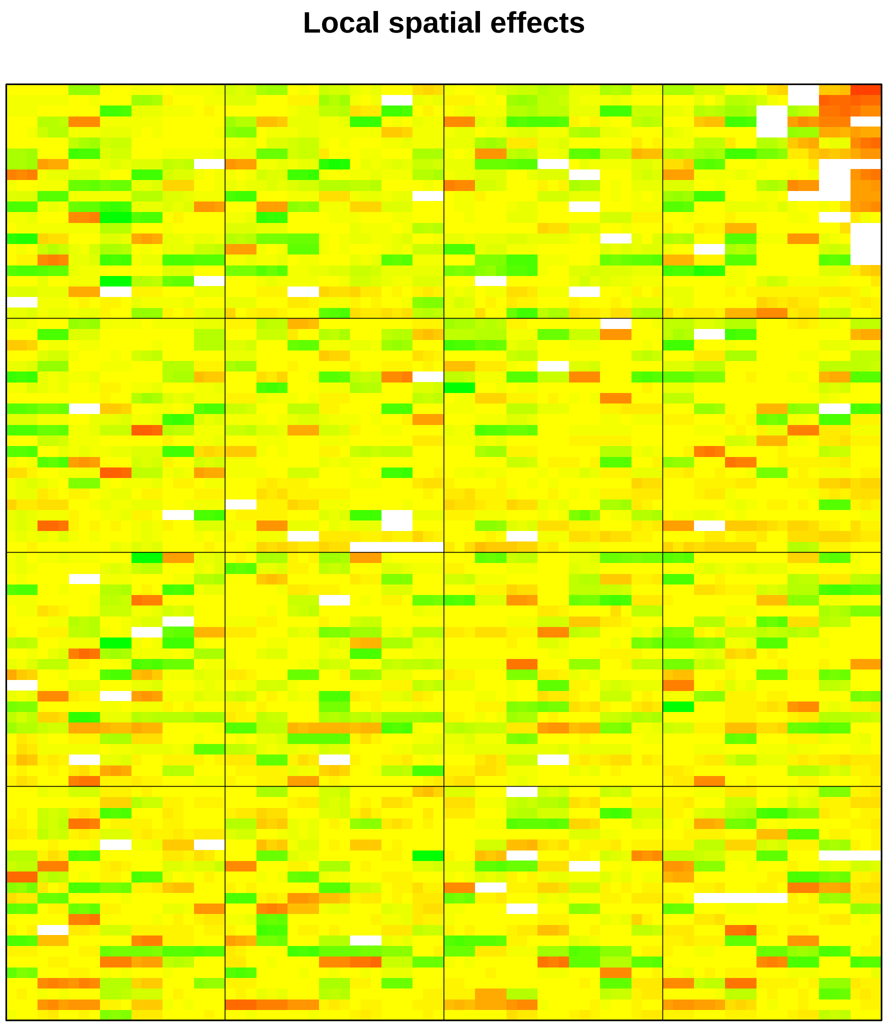

Figure 1: *Array with local spatial effects: high log-ratios cluster in the upper-right corner of the array.*

In order to eliminate such biases, we designed a three-step segmentation algorithm described in [Neuvial et al., 2006]:

`[step 1]`: Estimation of a spatial trend on the array using two-dimensional LOESS regression [Cleveland et al., 1988, Cleveland and Grosse, 1991]

`[step 2]`: Segmentation of the array into spatial areas with similar trend values, using NEM, an unsupervised classification algorithm including spatial constraints [Ambroise, 1996, Ambroise et al., 1997]

`[step 3]`: Identification of the areas affected by local spatial bias.

The last two steps of this algorithm involve parameters that may be tuned by CAPweb users at their convenience, as illustrated below.

2.2 Parameters of the segmentation algorithm

The NEM algorithm we use to segment the array into spatial areas with similar values depends on two parameters:

nk number of clusters into which the array image is segmented

β **(beta)** absolute weighting of the spatial constraint in the clustering algorithm.

Note that $\beta = 0$ corresponds to a classical clustering without spatial constraint, and $\beta = 1$ corresponds to equal weighting of classical clustering and spatial constraint. Figure 2 illustrates the influence of *nk* and β.

2.3 Parameters for the identification of the areas affected by local spatial bias

Once clustering with spatial constraint has been performed, one has to decide which clusters are affected by local spatial bias, in order to discard the corresponding spots from the array. To achieve this goal, we identify and remove those spatial clusters with signal values significantly higher (or lower) than the unbiased areas of the array.

prop maximum proportion of the array image that may correspond to local spatial bias.

thr minimum difference between cluster mean signals to call a cluster biased.

Figures 3 and 4 illustrate the influence of *prop* and *thr*.

2.4 Recommendations

Our experience after analysis of more than 3000 CGH arrays with this algorithm suggests the following recommendations:

- $\beta = 1$ is appropriate in most cases
- *nk* should be chosen such that the average cluster size is around 1500 spots, i.e. $nk = 7$ for arrays with 10000 spots, and $nk = 5$ for arrays with 7500 spots
- $prop = 0.25$ is appropriate in most cases, since local spatial bias typically applies to less than one quarter of the array
- *thr* should be chosen for each data set, depending on the amplitude of the discrepancy between biased and unbiased areas

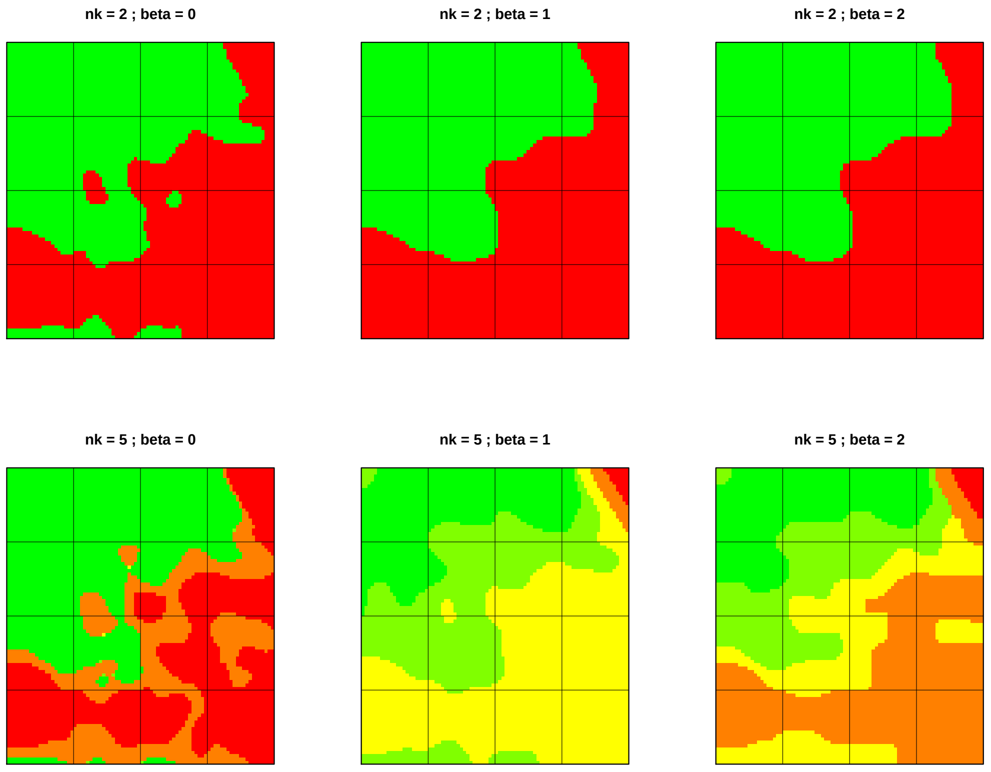

Figure 2: *Influence of nk and β on the results of NEM. Each color corresponds to a spatial cluster.*

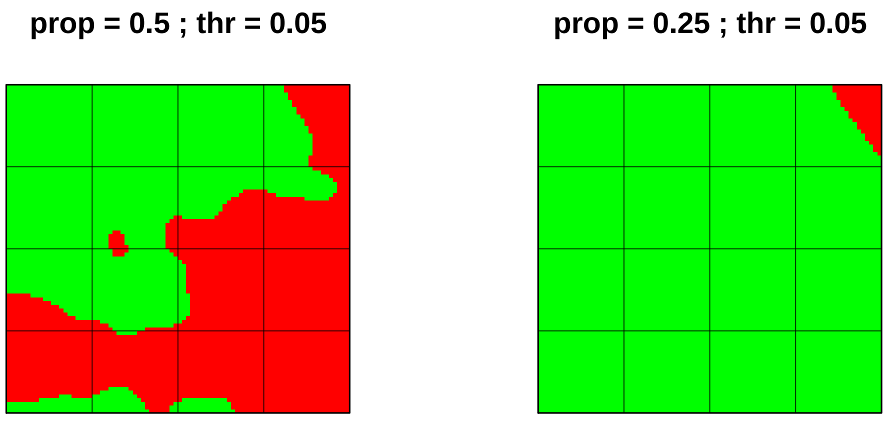

Figure 3: *Influence of prop on the results of spatial segmentation: spots considered biased are marked in red, while other spots are marked in green. Left: with prop* $= 0.5$*, three spatial clusters are considered biased, leading to numerous false positives. Right: with prop* $= 0.25$*, only two spatial clusters are considered biased, leading to an accurate delineation of the biased zone.*

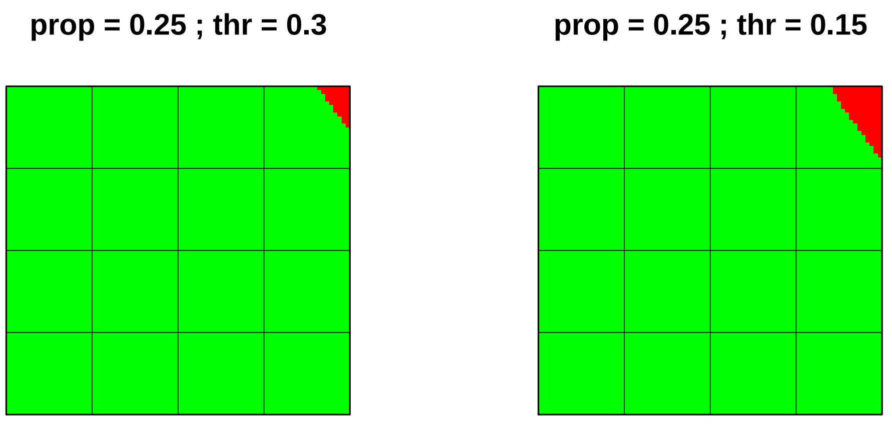

Figure 4: *Influence of thr on the results of spatial segmentation: spots considered biased are marked in red, while other spots are marked in green. Left: with thr* = 0.3*, only one spatial cluster are considered biased, leading to numerous false negatives. Right: with thr* = 0.15*, two spatial clusters are considered biased, leading to an accurate delineation of the biased zone.*

3 GLAD parameters

mediancenter If TRUE, LogRatio are center on their median.

qlambda qlambda determines the scale parameter for the stochastic penalty during the Adaptive Weight Smoothing procedure. qlambda takes values in between 0 and 1. The effect of this parameter is shown in **Figures 5 and 6**. The more you decrease the qlambda value the more you will be able to detect small genomic regions. We advise to use values like 0.99, 0.999 or 0.9999.

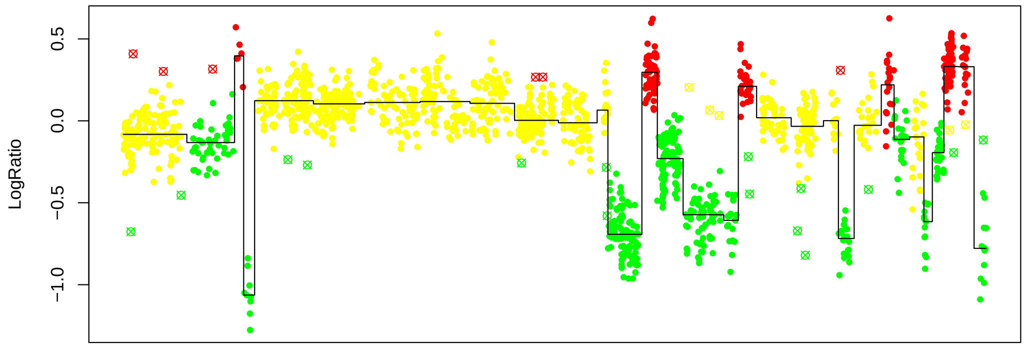

Figure 5: Results with qlambda set to 0.999.

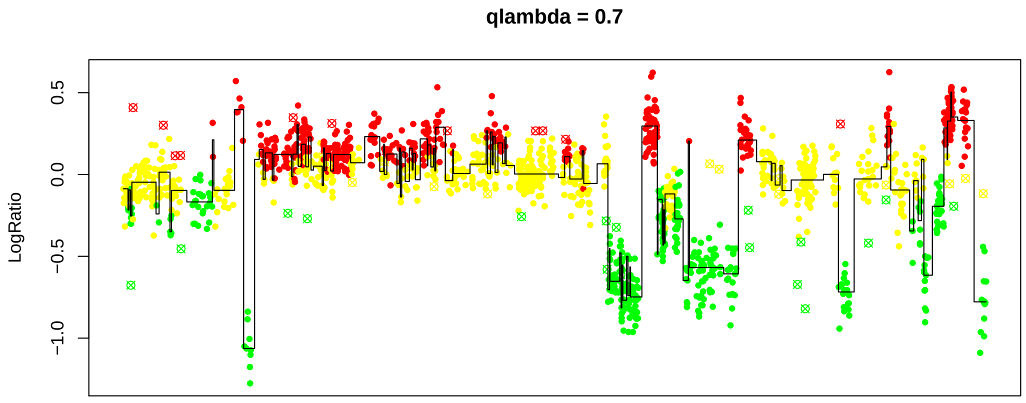

Figure 6: Results with qlambda set to 0.7.

bandwidth Set the maximal bandwidth during the Adaptive Weight Smoothing procedure. We advise to let this parameter to the default value.

lambdabreak Penalty term λ' (see **Equation 1**) used during the "Optimization of the number of breakpoints" step. The effect of this parameter is shown in **Figure 7 and 8**. The more you increase this parameter the less you will get breakpoints.

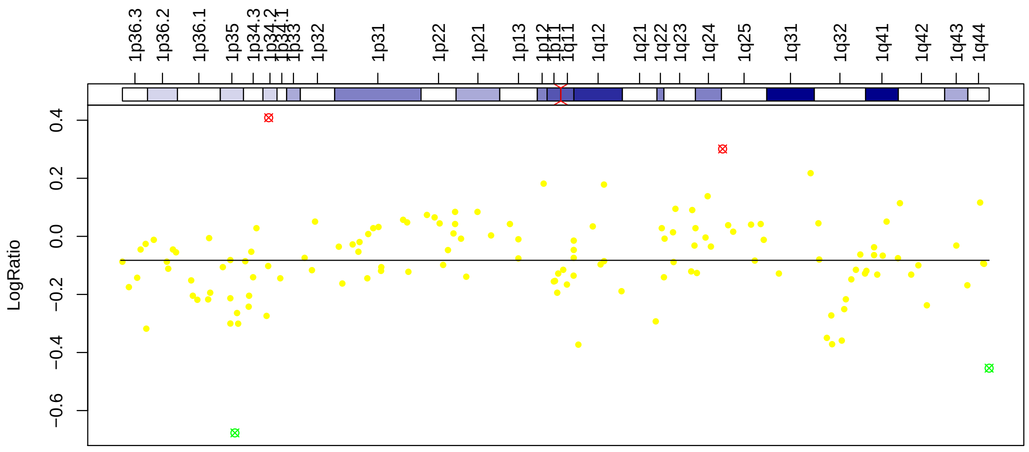

Figure 7: Results with lambdabreak set to 8.

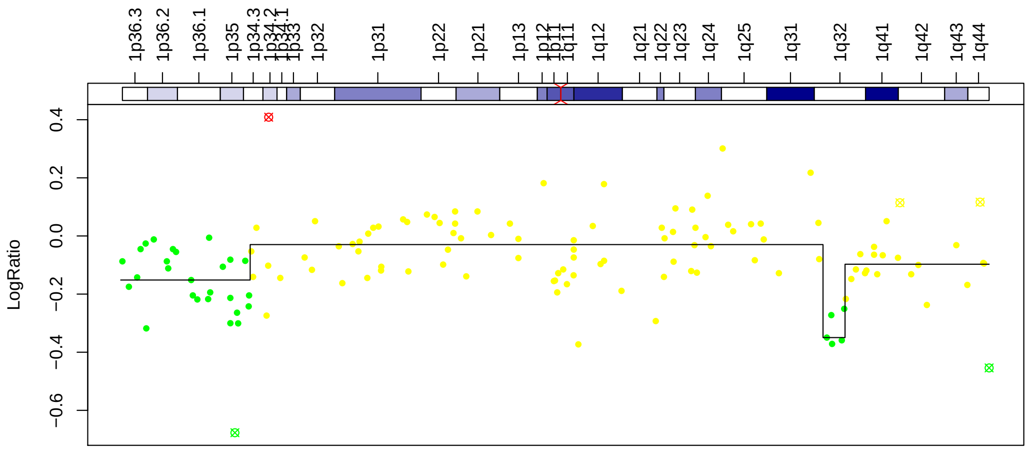

Figure 8: Results with lambdabreak set to 6.

param Parameter *d* of kernel used in the penalty term (see **Equation 1**) during the "Optimization of the number of breakpoints" step. The effect of this parameter is shown in **Figure 9 and 10**. The more you increase this parameter the less you will get breakpoints.

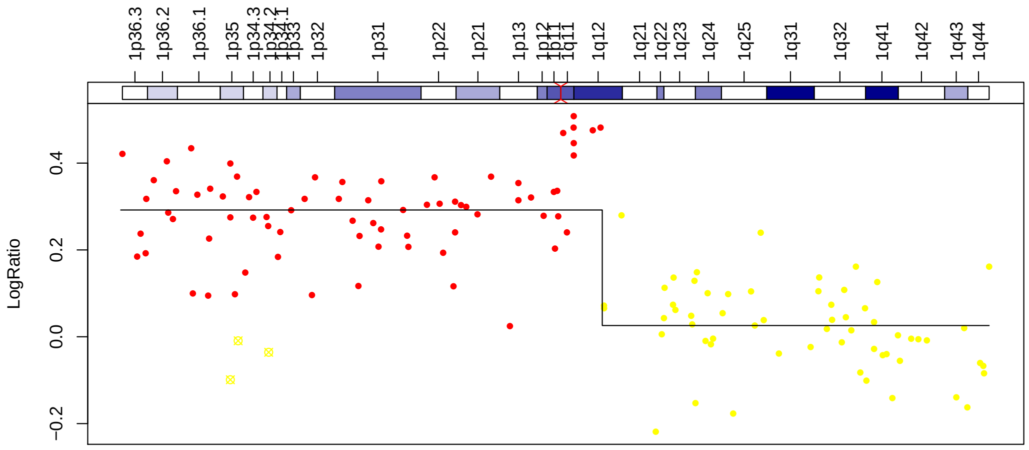

Figure 9: Results with param set to 6.

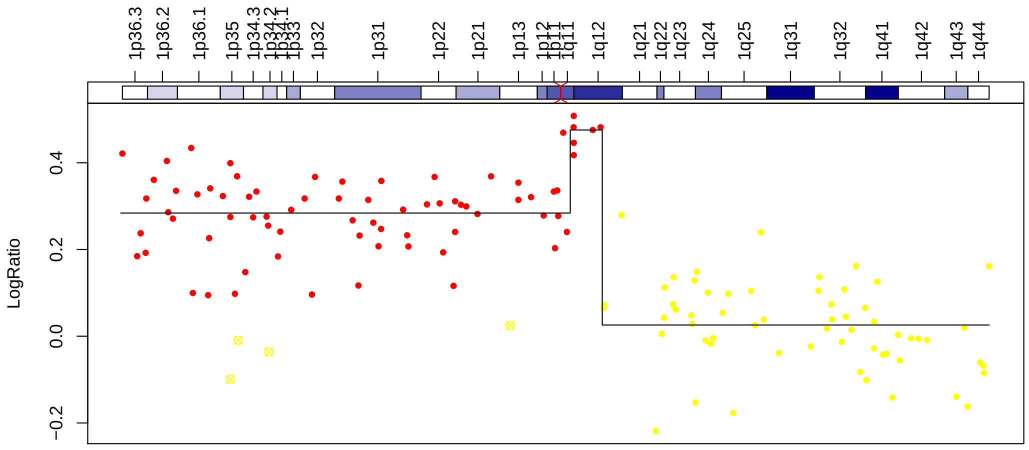

Figure 10: Results with param set to 2.

alpha Risk alpha used for the "Outlier detection" step. For each region, clone which log-ratio value in the $\alpha/2$ lower or upper tail distribution are considered as an outlier. The effect of this parameter is shown in **Figure 11 and 12**.

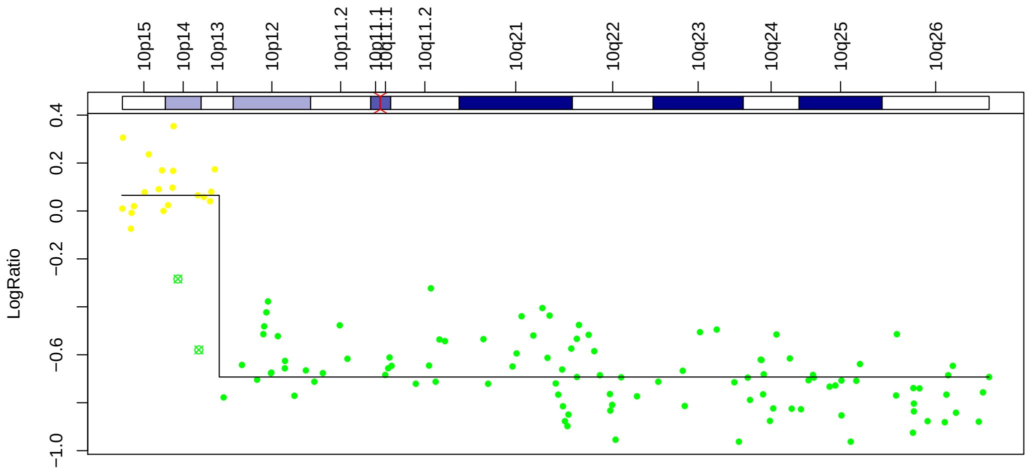

Figure 11: Results with alpha set to 0.001.

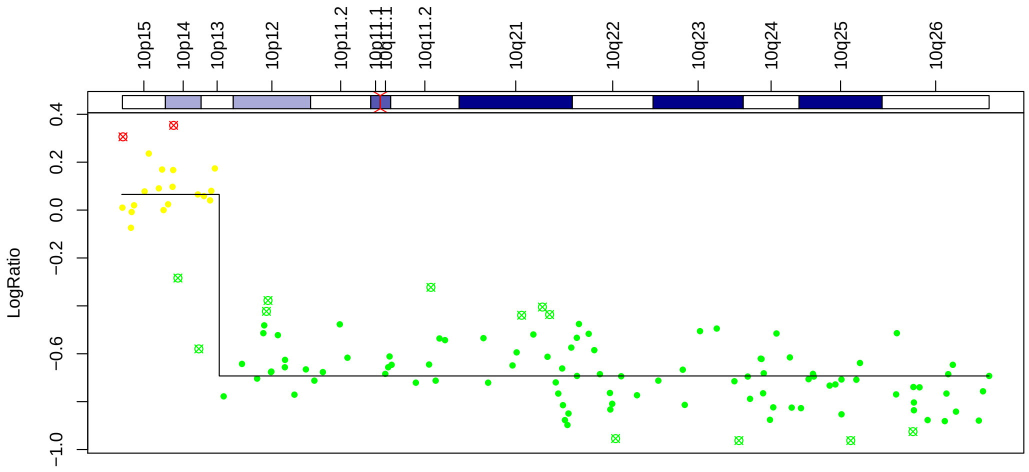

Figure 12: Results with alpha set to 0.05: more clones are detected as outlier.

msize The outliers are calculated on regions with a cardinality greater or equal to msize.

lambdaclusterGen Penalty term used during the "clustering throughout the genome" step.The effect of this parameter is shown in **Figure 13 and 14**. When you increase the lambdaclusterGen all the region will tend to be inside the same cluster and will have the same status and thus less altetartion will be indetified.

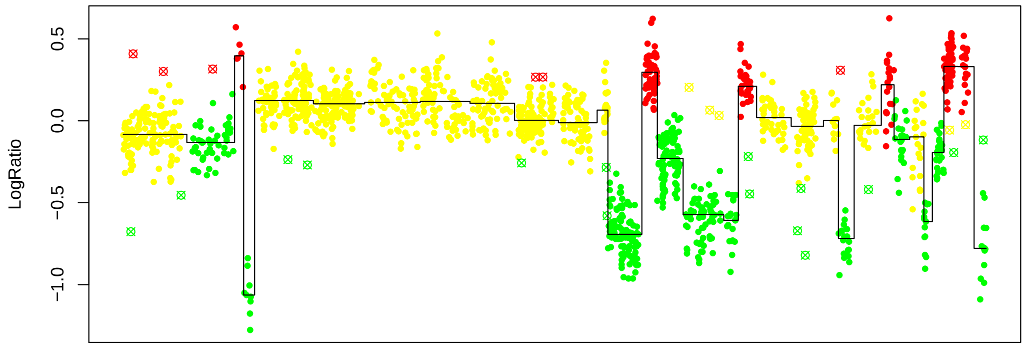

Figure 13: Results with lambdaclusterGen set to 40.

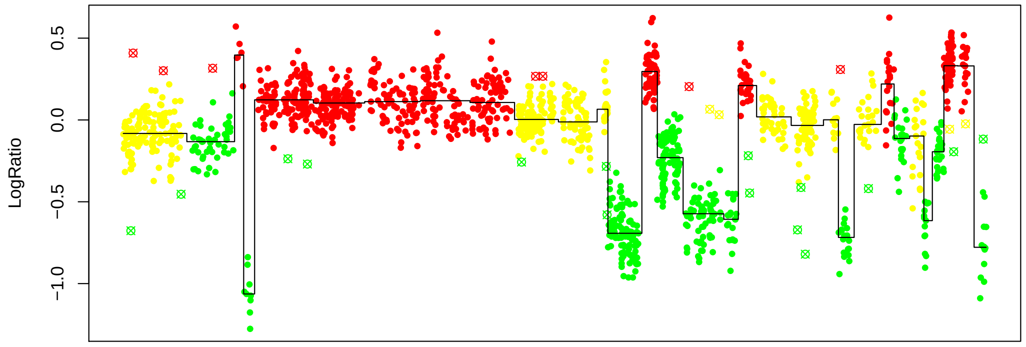

Figure 14: Results with lambdaclusterGen set to 10.

amplicon Level (and outliers) with a smoothing value (log-ratio value) greater than this threshold are consider as amplicon.

deltaN Region with smoothing values in between the interval [-deltaN,+deltaN] are supposed to be normal.

forceGL1 and forceGL2 Level with smoothing value greater (lower) than forceGL1 (forceGL2) are considered as gain (lost).

Comparing **Figure 15** and **Figure 16** shows the influence of two different sets of parameters.

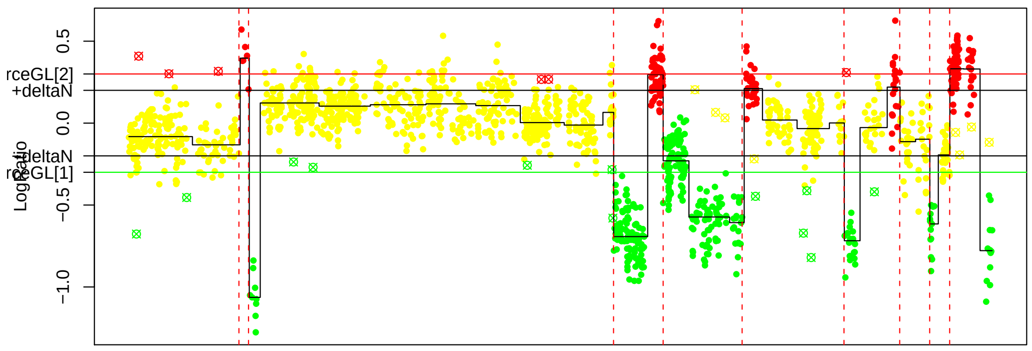

Figure 15: Results with deltaN=0.2, forceGL1=-0.3, forceGL2=0.3, amplicon=1.

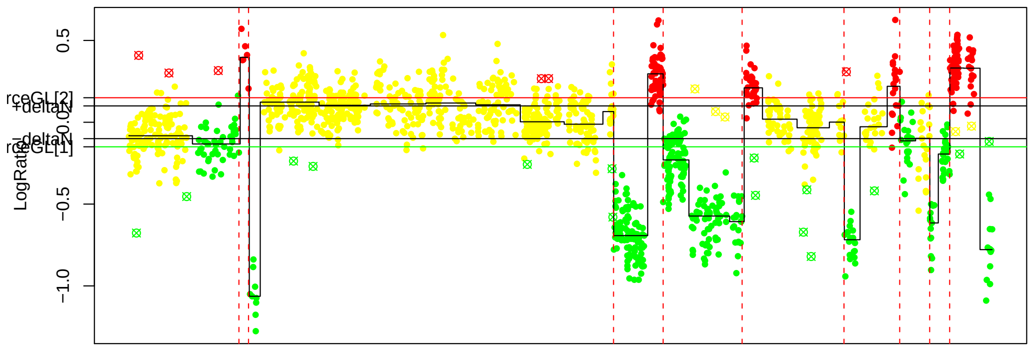

Figure 16: Results with deltaN=0.1, forceGL1=-0.15, forceGL2=0.15, amplicon=1

nbsigma For each breakpoints, a weight (see **Equation 2**) is calculated which is a function of absolute value of the gap Δ between the smoothing values of the two consecutive regions, the *nbsigma* value, and standard deviation σ of the LogRatio.

MinBkpWeight Breakpoints between two regions of same status and Weight less than MinBkpWeight are discarded.

3.1 Recommendations

The most important parameters are:

- *qlambda*
- *lambdabreak*
- *lambdacluster*
- *lambdaclusterGen*
- *param*

Decreasing those parameters will lead to a higher number of breakpoints identified. For arrays experiments with very small Signal to Noise ratio it is recommended to use a small value of *param* like $param = 3$ or less.

Moreover, it is important to tune correctly the tresholds for *amplicon*, *deltaN*, *forceGL1* and *forceGL2* which will determine the status assigned to each region.

4 appendix

4.1 Penalized log-likelihood

$$f = \sum_{k=1}^{N'+1} (B_k - B_{k-1}) \log(\hat{\sigma}_k^2) + \lambda' \sum_{k=1}^{N'} K(\hat{\sigma}^{-1} |\hat{\mu}_k - \hat{\mu}_{k+1}|) \log(N) \tag{1}$$

with $B_0 = 0$, $B_{N'+1} = N$, $\hat{\sigma}_k^2$ and $\hat{\mu}_k$ are the usual MLE of σ_k^2 and μ_k, and $\hat{\sigma}$ is the sdandard deviation of the log-ratios. The function f corresponds, up to an additive constant, to a penalized form of -log L. The function $K(x)$ is the tricubic kernel function and takes the value $(1 - (x/d)^3)^3$ for $x \in [0; d]$ and zero elsewhere.

4.2 Breakpoint weight

$$Weight = 1 - K(\Delta) \tag{2}$$

with $K(x)$ is the tricubic kernel function and takes the value $(1-(x/d)^3)^3$ for $x \in [0; d]$ and zero elsewhere. d is set to $nbsigma * \sigma$

References

[Ambroise, 1996] Ambroise, C. (1996). *Approche probabiliste en classification automatique et contraintes de voisinage*. PhD thesis, Université Technique de Compiègne, France.

[Ambroise et al., 1997] Ambroise, C., Dang, M., and Govaert, G. (1997). Clustering of spatial data by the EM algorithm. In Soares, A., Gomes-Hernandez, J., and Froidevaux, R., editors, *Geostatistics for Environmental Applications*, pages 493–504. Kluwer Academic Publisher.

[Cleveland et al., 1988] Cleveland, W., Devlin, S., E., and Grosse (1988). Regression by local fitting. *Journal of Econometrics*, 37:87 –114.

[Cleveland and Grosse, 1991] Cleveland, W. S. and Grosse, E. (1991). Computational methods for local regression. *Statistics and Computing*, 1:47–62.

[Neuvial et al., 2006] Neuvial, P., Hupé, P., Brito, I., Liva, S., Manié, E., Brennetot, C., Aurias, A., Radvanyi, F., and Barillot, E. (2006). Spatial normalization of array-CGH data. Submitted.