# Identifying copy number alteration using CBS

**Background: Why are we looking at copy number**

DNA copy number refers to the abundancy of a small piece of chomosome, and is of both biological and medical interests. Cancer initiation is often associated with accumulated DNA mutation followed by DNA alteration at a larger scale, because of the inefficiency of DNA repairing mechanism. Copy numeval.after = c('fig.cap','code'),ber alteration can also be observed in germline cells, often associated with genetic diseases. Hence, reliable detection of copy number alteration would enable non-invasive cancer diagnosis and screens for genetic diseases. It also serves as an intermediate between chromosome morphology analysis and sequencing analysis, often conducted through microarray.

**Data simulation**

The data is simulated by concatenating a sample of segments. The copy-number of the segments follows a multinomial distribution, whereas their length follows Poisson distribution ([**?**]nstruction ). At the end an artificial chromosome of 200Mbp is generated, as represented by 30K equally-spaced probe, giving an inter-probe distance of appx. 6kbp. We experimented with different parameters of $L_{loss}$, $L_{dup}$,$L_{norm}$, $p_{loss}$, $p_{dup}$, $p_{norm}$, and selected a parameter set that generated aberrations that span around $10^2$ probes so that there is enough signal to perform CBS segmentation, while keeping the "normal" to be the longest and most likely segment type. The parameters read:

$$\begin{aligned} L_{loss} &= 20 \cdot 10^4 bp \\ L_{norm} &= 60 \cdot 10^4 bp \\ L_{dup} &= 20 \cdot 10^4 bp \\ p_{loss} &= 0.25 \\ p_{norm} &= 0.5 \\ p_{dup} &= 0.25 \end{aligned}$$

Each probe is then assigned a copy-number response of 1/2/3 according to its segment. The response is log-transformed into logR and masked with a gaussian noise

$$\begin{aligned} logR &= \log_2(R/2) + \epsilon \\ \epsilon &\sim N(0, \sigma^2) \end{aligned}$$

**Segmentation under noise**

Segmentation is performed on the observed logR with DNACopy (1) , for different $\sigma$. For each dataset, we calculated mean_absolute_error between the fitted response and the theoretical model according to eq(1) to evaluate the quality of the fit. As noise becomes more prevalent, the prediction becomes less reliable (see figure  and 1). Note the mean_absolute_error needs to be compared against a completely null prediction, that is, the expected error if no segment is predictied. (mean_absolute_error=0.179). For example, at sigma=1.691, the MAE appears worse than not predicting any copy number changes.

(1)
$$MAE = \frac{\sum_{probe} |logR(probe) - logR_M(probe)|}{N(probe)}$$

where $logR_M$ denotes the logR from theoretical model, $logR$ denotes that from the observed probe response.

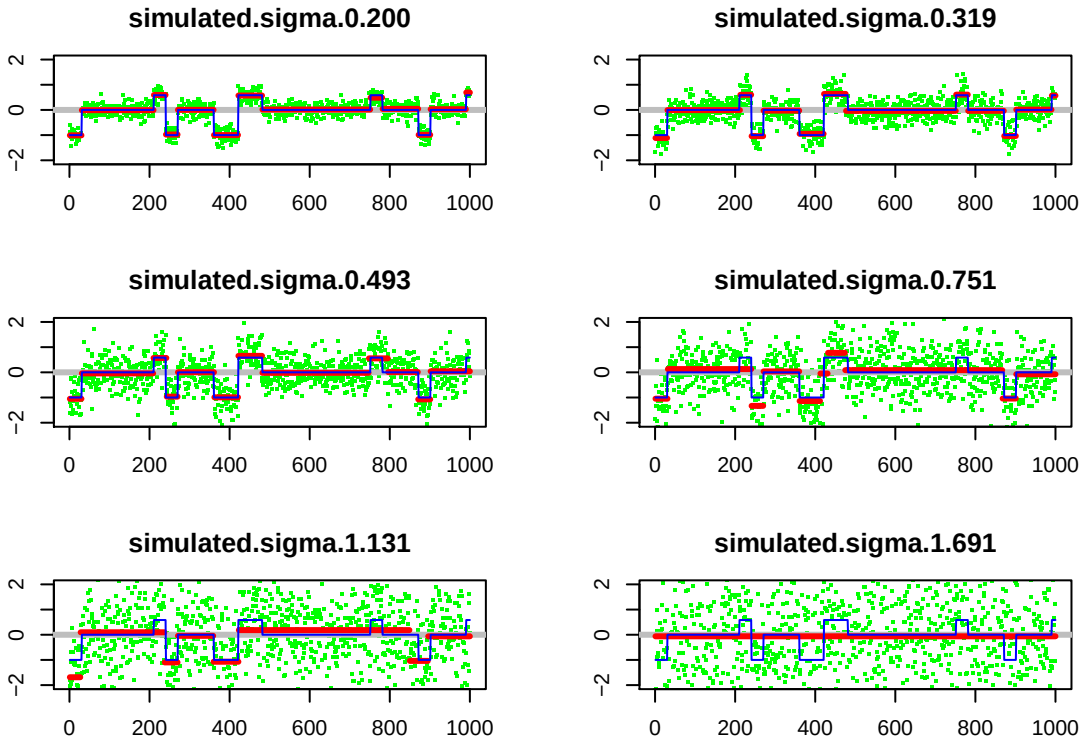


Figure 1: CBS result for different noise level. Green: raw probe response; Blue: theoretical model; Red: CBS segmentation

### Calling copy numbers

In order to define regions with DNA losses, we first attempted a simple thresholding algorithm, where segment means are compared against a fixed threshold, below which the segment is declared to have underwent a "loss" event. To compare the merit of different thresholds, we plotted Precision-Recall curve for different thresholds under all noise levels.

To simplify the evaluation, we ignored the segments for now and compute on a per-probe basis. A model is binariesd with "$logR_M(probe) == -1.0$", whereas a prediction is binarised with "$logR(probe) < threshold$", which are then combined to calculate TP,FP,TN,FN, and converted into precision and recall (eq(2)). The use of PR curve is justified by the imbalanced nature of the sample.

(2)

$$Precision : P = \frac{TP}{TP + FP}$$
$$Recall : R = \frac{TP}{TP + FN}$$
$$TN = N(model = 1)$$
$$\leftrightarrow TN = N(\log_2(model/2) = -1)$$
$$TN + FP = N(fitted < threshold)$$

As can be seen from the PR-curve ( figure 3), it is possible to achieve P=1 and R=1 for less noisy data, whereas some trade-off is necessary in treating noisy data. The question then becomes how to find the sweet spot without referencing the actual model. This must involves some calculation on the observation side, e.g by looking at the distribution of fitted segment means.

The distribution of segment means follow a nice multimodal trend (see figure). Intuitively, optimal threshold should be placed at where the value jumps discontinously. This jump is easy to identify manually, and here

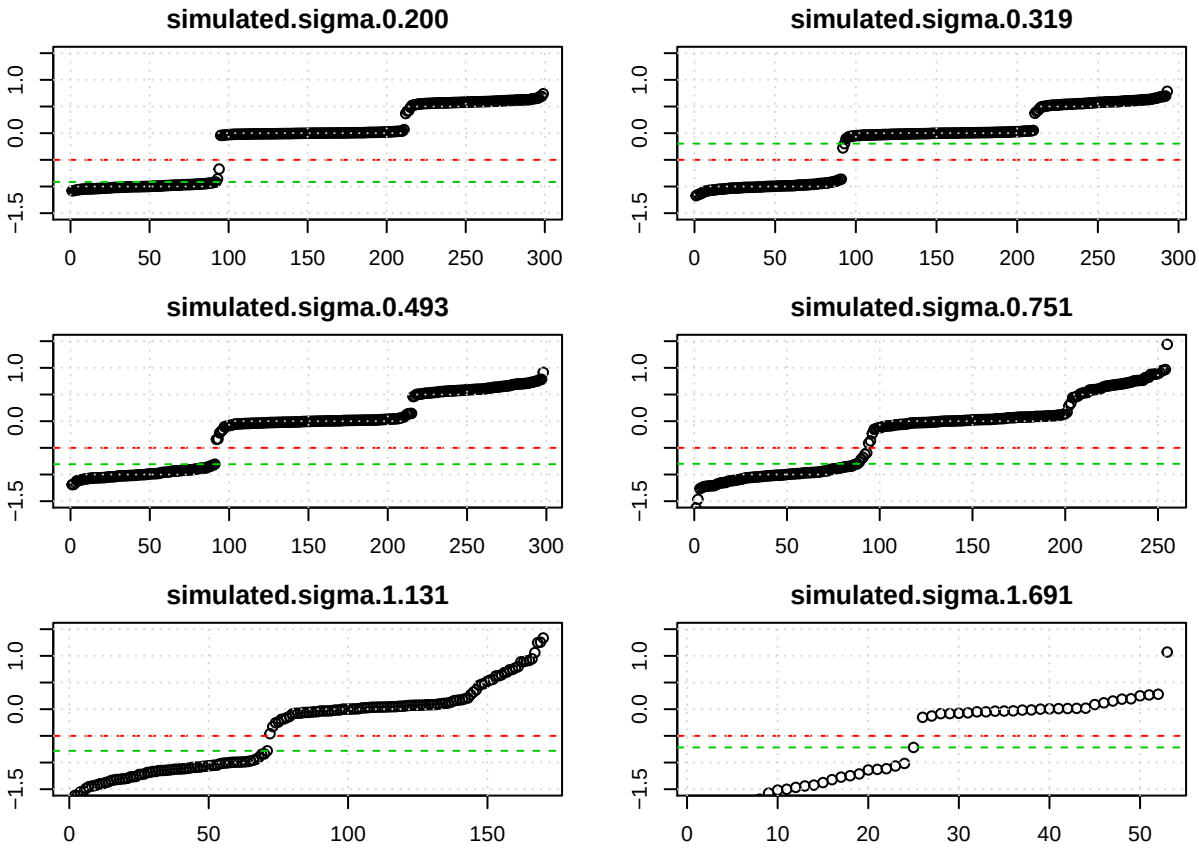


Figure 2: Sorted segmented means. Red: threshold = -0.5; Green: threshold by cluster

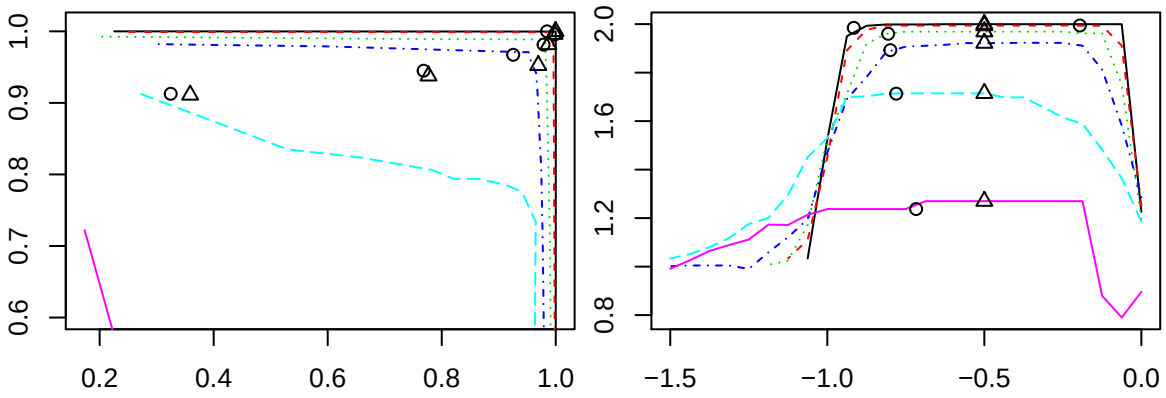


Figure 3: Evaluation of different threshold. Circle: Thresholded by cluster. Triangle: Naive thresholding (This plot is somehow broken after switching dataset)

we apply a simple gaussian/t-test based clustering to demarcate such discontinous jumps. The clustering algorithm works as follows: we start with an empty vector, and add the segment means one by one from the smallest. Once the vector size reach 5, we start to estimate mean and standard deviation (using MAD as estimator) from the sample defined by the vector, so as to estimate the likelihood of the incoming element to belong to this sample. If the incoming value is significantly deviating from the sample (using a Z-score cutoff), then this sample is dumped as a cluster and we start with an empty vector again. As a result, the whole distribution is segmented into multiple homogeneous normal distributions.

The resultant clusters are then subject to a one-tail t-test with the alternative hypothesis that $\mu < -0.5$, and the cluster is retained only if there is significant evidence to reject the null (P<0.001). In other words, we threshold segment means on a pre-cluster basis to ensure the consistency of thresholding. As shown in the PR-curve (figure **??**), using the selected threshold we achieved precision $> 90\%$ at all tested level of noises. Moreover, in cases where a smooth elbow is present, it is always identified by the cluster-based threshold, achieving the best possible compromise between precision and recall.

Unlike result from simple thresholding at -0.5, the cluster-based result always sits on the start of the elbow, does not suffer from the precision reduction due to noise.

For furture reference, this segmentation algorithm requires several parameters: (1) MIN_LEN: is the smallest sample size for a standard deviation to be estimated (2) Z_MIN: is the z-score cutoff to declare the ending of a cluster.(3) SUPER_THRES: is the prior belief of where the cutoff lies.

### 3. Impact of mixing a normal cell population

We consider a normal cell contamination that attenuates our signal, so that (@eq01)

$$R_{cM}(probe) = c \cdot 2 + (1 - c) \cdot R_M(probe)$$
$$\log_2 \left(R_{cM}(probe)/2\right) = \log_2 \left(c + (1 - c) \cdot R_M(probe)/2\right)$$

We plotted $R_{cM}(probe)$ w.r.t. $R_M(probe)$ on a log-log scale, showing a approximately linear trend near R(probe)=2, indicating that the mapped signals should still be separable, albeit less tolerant to noise. In other words, during outlier-based clustering, SUPER_THRES needs to be set dynamically considering the possibility of contamination, rather than fixed at -0.5.

Firstly, To understand the effect of a normal cell contamination, we plotted noise-MAE curve for 6 contamination level. Indeed, samples with higher contamination are more prone to noise (see figure ).

Secondly, in order to infer this ratio of contamination, we compute the mean absolute error (MAE, also known as the L1-norm) for a range of candidate contamination ratio and select the one with the least MAE.

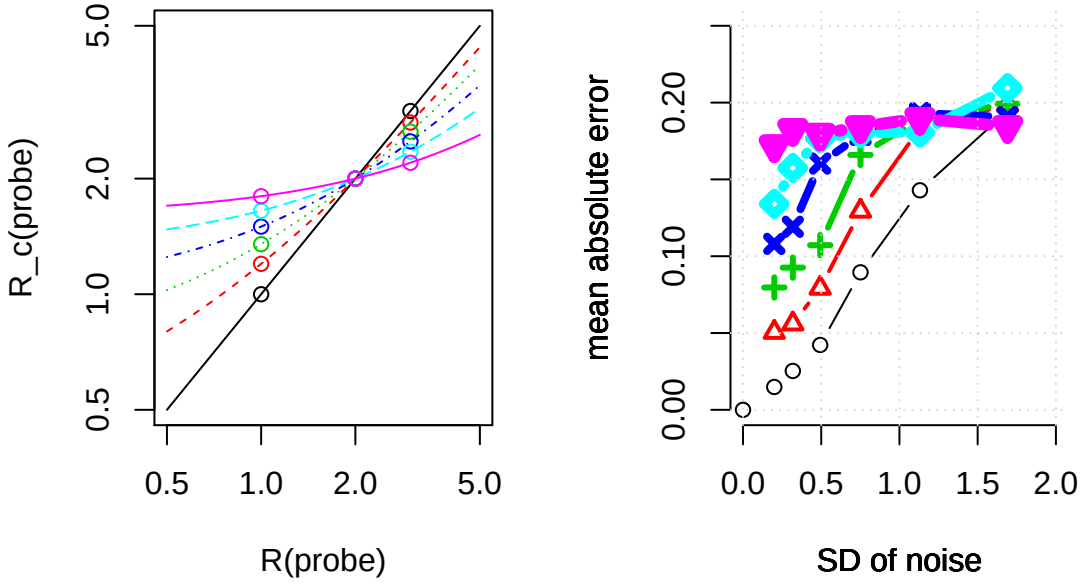


Figure 4: Higher contamination are more prone to noise

This is done by calculating the average for distances from predicted segment mean to their most probable copy-number given a candidiate contamination ratio ( eq(@eq01) for $R(probe) \in \{1, 2, 3\}$ ) . It appears that as contamination increases, the algorithm tends to under-estimate the ratio (figure 5), conceivably due to less segments being available to support the inference.

For summary, we compared 5 threshold estimating scheme on the PR-curve: (1) Naive thresholding at -0.5 (2) Naive thresholding using theoretical contamination (3) Cluster-based thresholding at -0.5 (4) Cluster-based thresholding using theoretical contamintaion (5) Cluster-based thresholding using guessed contamination

From the summarizing plot, we observe that thresholding using theoretical contamination reasonably improved the result (type2, type4, as compared to type1, type3). However, due to the difficulty in inferring the contamination, it is not always possible to perfrom type5 reliably.

In other words, the putative "loss" clusters are shifted towards the neutral signal, with their shape invaried.

**Conclusion**

We have tested the efficacy of CBS algorithm using a simulated dataset and assessed its performance under different noise level. Furthermore, we suggested to use cluster-based thresholding to improve the copy-number calling. We then go on to adapt this methodology under a normal cell contamination by incorporating an estimator for contamination, but uncovering the unfortunate fact that the weakness in contamination estimation is destroying the whole algorithm. However, we highlight the usefulness of both naive thresholding and cluster-based thresholding, given that the contamination ratio. Thus the whole system, will benefit from a robust estimator for contamintation ratio.

```
tpbs = pbs[1:10]
i = 1
(probes[i] <= tpbs[-1]) & (probes[i] > tpbs[-length(tpbs)])
```

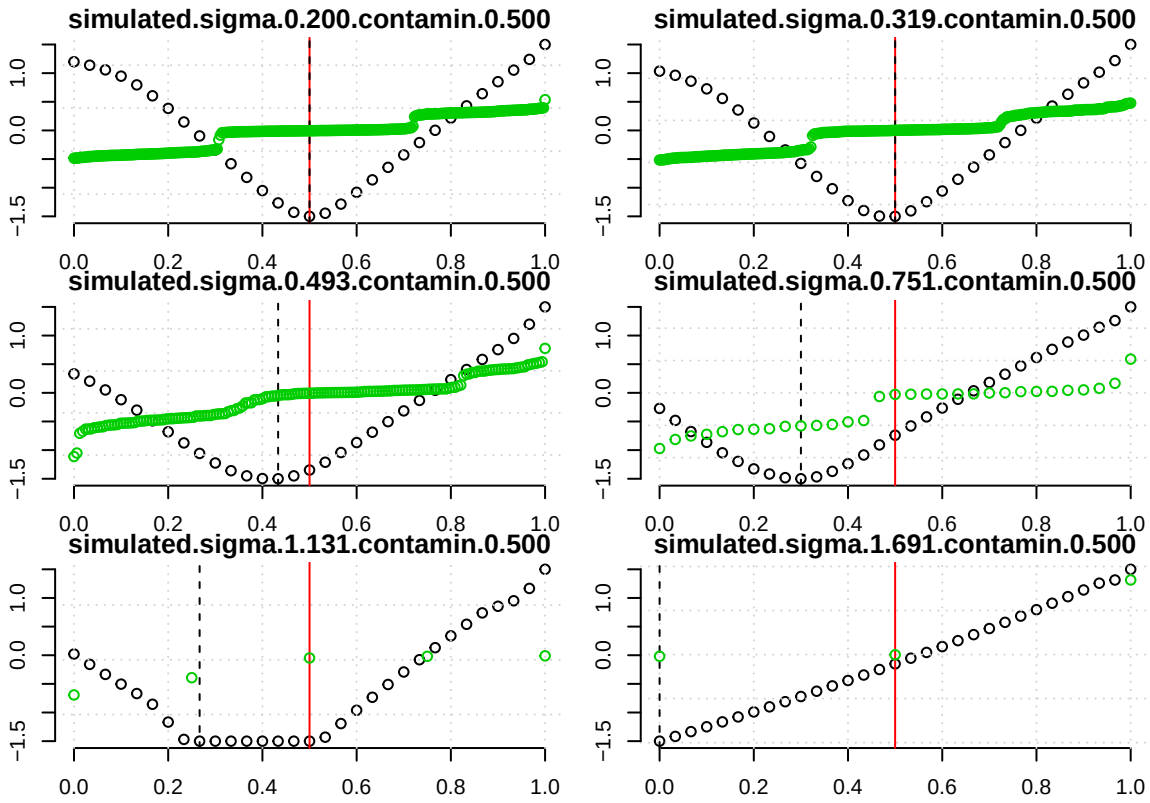


Figure 5: Estimating conatmination level by minimising MAE. Red: Actual contamination ratio (c-ratio) Black: estimated c-ratio

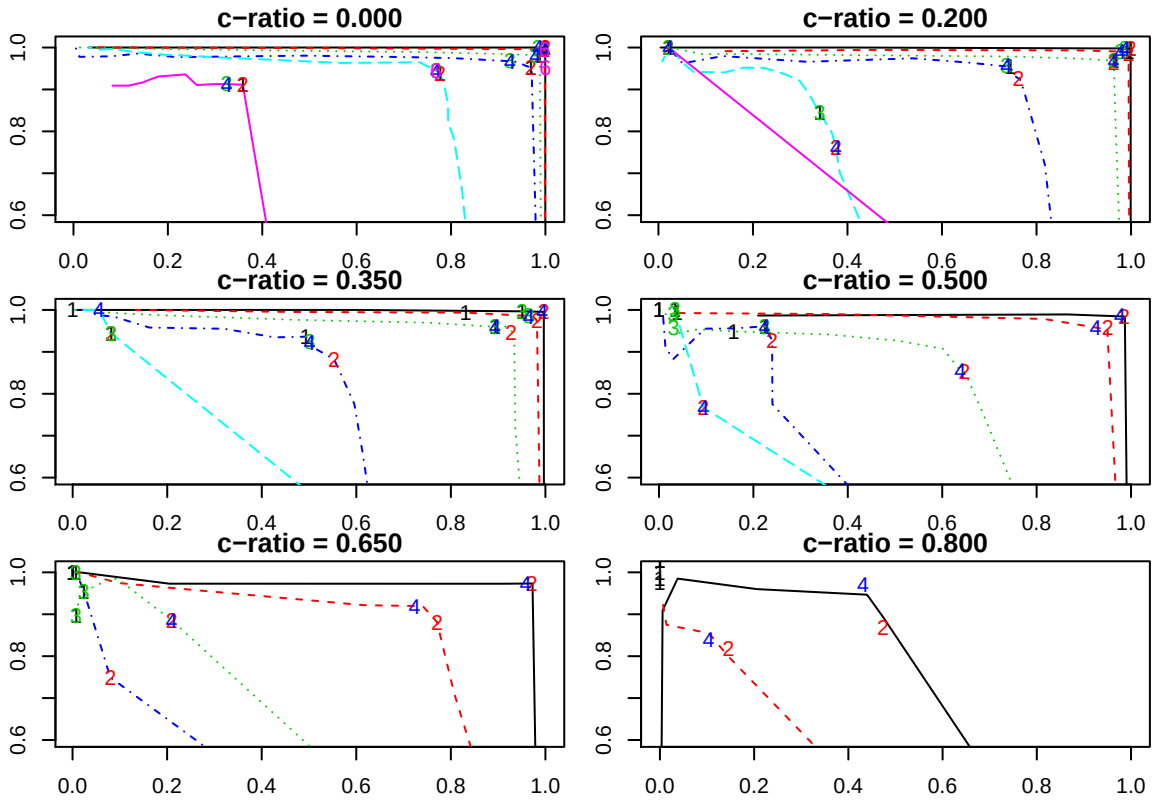


Figure 6: Evaluation of different threshold. Coloured by increasing noise level)

```
# probes_mat = matrix(probes,ncol = 1)
probe = probes[1]

# copy_number = (probes_mat <= segEND[-1]) & (probes_mat >= segEND[-length(segEND)])
# copy_number%f%dim
# copynumber
```

References: 1. Adam B. Olshen, E. S. Venkatraman, Robert Lucito, Michael Wigler; Circular binary segmentation for the analysis of array based DNA copy number data, Biostatistics, Volume 5, Issue 4, 1 October 2004, Pages 557–572,