

Exact and fast segmentation of large SNP/CGH profiles

Guillem Rigail

January 2010



Outline

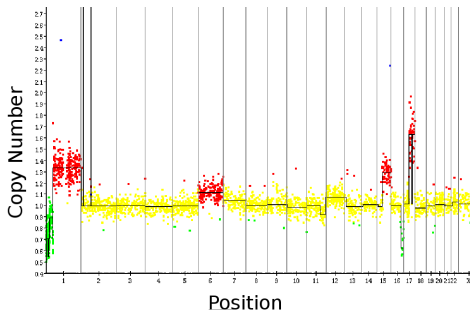
- 1 DNA copy number analysis, statistical model
- 2 Fast segmentation algorithm
- 3 Conclusion

Outline

- 1 DNA copy number analysis, statistical model
- 2 Fast segmentation algorithm
- 3 Conclusion

DNA copy number and Cancer

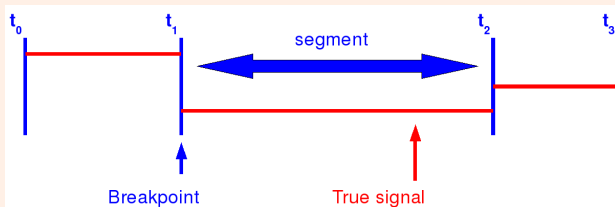
- In normal cells: copy number = 2 (pairs of chromosome)
- In tumor cells: copy number $\neq 2$ on many points of the genome
- Gain and loss of DNA:
 - ▶ chromosomes
 - ▶ smaller regions up to 10Kb



Multiple change-point detection

The data

- A succession of segments that share the same copy number
- The signal is affected by abrupt changes



Segments and segmentations

\mathcal{M}_K the set of all possible segmentations with K segments

$m \in \mathcal{M}_K$ a specific segmentation

$r \in m$ a segment of m with n_r observations

Statistical model

Normal homosedastic segmentation

- A succession of segments that share the same copy number
- We observe a noisy signal Y_t :

$$\forall t \in r \quad Y_t \sim \mathcal{N}(\mu_r, \sigma^2) \quad \{Y_t\}_t \text{ are independent}$$

Maximum likelihood

- Equivalent to a minimal residual sum of squares

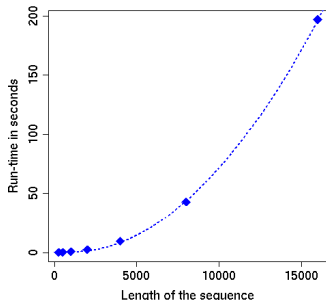
$$\min_{m \in \mathcal{M}_K} \left\{ \sum_{r \in m} \min_{\mu} \left\{ \sum_{t \in r} (Y_t - \mu)^2 \right\} \right\}$$

How to find breakpoint positions ?

- There is C_{n-1}^{K-1} possible segmentations
- Use of dynamic programming (Bellman and Roth 1969)
- Application to CGH data (Picard et al. 2005)

Run-time complexity in $\mathcal{O}(n^2)$

- $n = 10^5 \rightarrow 2.4 \text{ hours}$
- $n = 10^6 \rightarrow 10 \text{ days}$
- SNP profiles : $10^5 \leq n \leq 10^6$



How to find breakpoint positions when n is large ?

Solutions

1 Heuristics to minimize the least square criterion

- ▶ CART + dynamic programming

2 Different optimization problem

- ▶ Fused lasso (Tibshirani and Wang 2007)

$$\min \left\{ \sum_i (y_i - \beta_i)^2 \right\}$$

$$, \text{ subject to } \sum_i |\beta_i| < \mathbf{s}_1 \text{ and } \sum_i |\beta_i - \beta_{i+1}| < \mathbf{s}_2$$

Purpose :

3 **Fast algorithm to retrieve the exact solution for the least square criterion**

Outline

- 1 DNA copy number analysis, statistical model
- 2 Fast segmentation algorithm**
- 3 Conclusion

Dynamic programming

Optimization problem

- $\mathcal{M}_{K,t}$: all possible segmentations in K segments up to point t
- $C_{K,t}$: optimal cost in K segments up to point t

$$C_{K,t} = \min_{\{m \in \mathcal{M}_{K,t}\}} \left\{ \sum_{r \in m} \min_{\mu} \left\{ \sum_{t \in r} (Y_t - \mu)^2 \right\} \right\}.$$

Segment additivity : $t - K$ comparisons at each step $\Rightarrow \mathcal{O}(n^2)$

$$C_{K,t} = \min_{K-1 \leq t_0 < t} \left\{ C_{K-1,t_0} + \min_{\mu} \left\{ \sum_{i=t_0+1}^t (Y_i - \mu)^2 \right\} \right\}$$

If we know:

- the best solution in $K - 1$ segments up to any $t_0 < t$
- \Rightarrow We get the best solution in K segments up to point t

Known optimal value of the current segment μ^*

Optimization problem

$$P_{K,t}(\mu^*) = \min_{K-1 \leq t_0 < t} \left\{ C_{K-1,t_0} + \sum_{i=t_0}^t (Y_i - \mu^*)^2 \right\}$$

Point additivity : 1 comparison at each step $\Rightarrow \mathcal{O}(n)$

$$P_{K,t+1}(\mu^*) = \min \{ P_{K,t}(\mu^*) , C_{K-1,t} \} + (Y_{t+1} - \mu^*)^2$$

So if we know:

- 1 the best solution in K segments up to point t
 - 2 the best solution in $K - 1$ segments up to point t
- \Rightarrow We get the best solution in K segments up to point $t + 1$

Unknown optimal value of the current segment μ^*

Test p possible values of μ^*

- For example a grid of p regularly spaced values
- Run-time in $\mathcal{O}(p.n)$
- However it does not retrieve the best solution

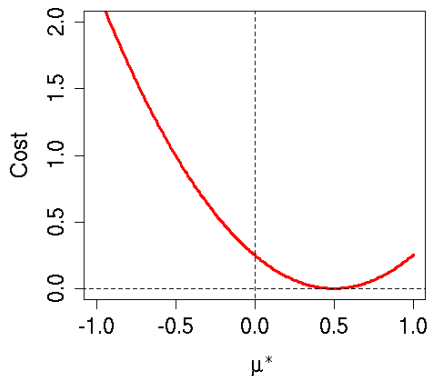
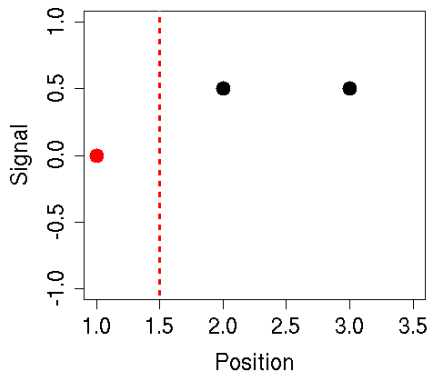
Test all possible values of μ^*

- Close values of μ^* correspond to the same last optimal breakpoint
- We need to store critical values of μ^* corresponding to a change in the last optimal breakpoint

An example, initialization

- Best segmentation in 2 segments up to point $t = 2$
- $P_{2,2}(\mu^*)$: cost for a break at $t = 1$:

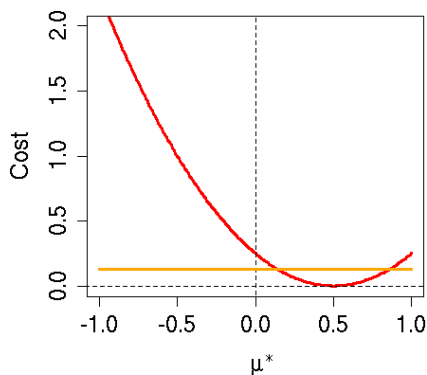
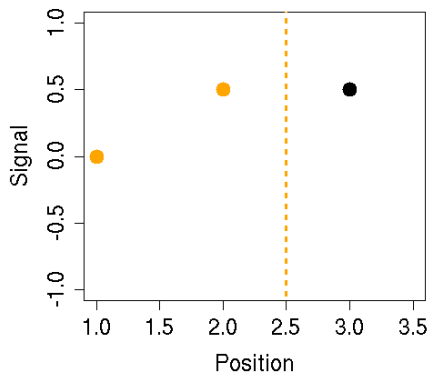
$$P_{2,2}(\mu^*) = C_{1,1} + (y_2 - \mu^*)^2 = 0 + (y_2 - \mu^*)^2$$



An example, step 1.1

- Best segmentation in 2 segments up to point $t = 3$
- Compare $P_{2,2}(\mu^*)$ with the cost of a break at $t = 2$:

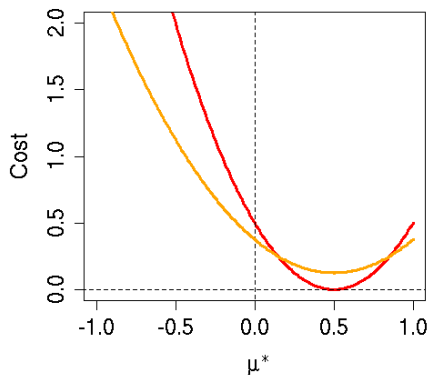
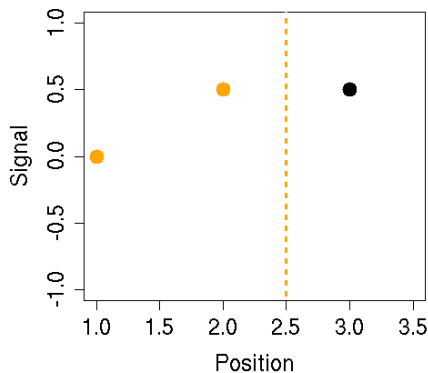
$$\min\{P_{2,2}(\mu^*), C_{1,2}\}$$



An example, step 1.2

- Best segmentation in 2 segments up to point $t = 3$
- Add the cost of the third observation y_3

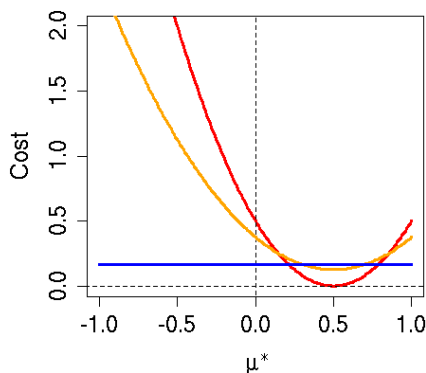
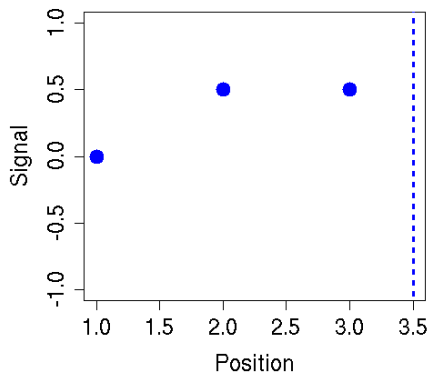
$$P_{2,3}(\mu^*) = \min\{ P_{2,2}(\mu^*) , C_{1,2} \} + (y_3 - \mu^*)^2$$



An example, step 2.1

- Best segmentation in 2 segments up to point $t = 4$
- Compare $P_{2,3}(\mu^*)$ with the cost of a break at $t = 3$:

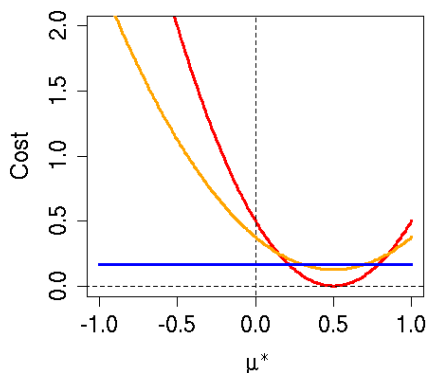
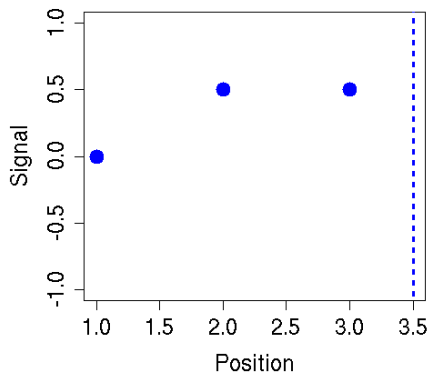
$$\min\{ P_{2,3}(\mu^*) , C_{1,3} \}$$



An example, filtering

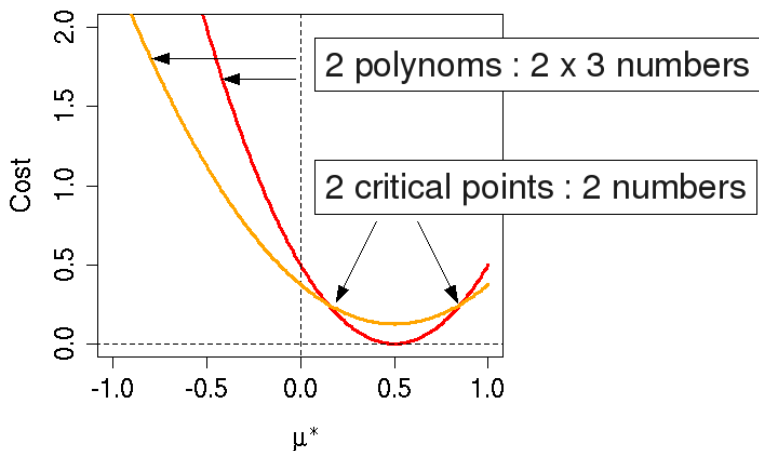
- Best segmentation in 2 segments up to point $t = 4$
- We no longer need the orange curve : last break at $t = 2$

$$\min\{ P_{2,3}(\mu^*), C_{1,3} \}$$



Required information

- Second degree polynomial functions
 - ▶ Corresponding to $3n$ real numbers
- Critical points



Worst case and mean time complexity

Worst case

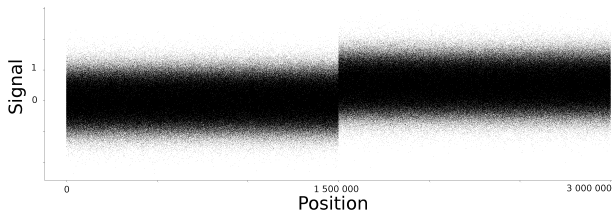
- Correspond to a maximum number of critical points
- It can be demonstrate that at most there is $2n - 1$ critical points
- Worst complexity in $\mathcal{O}(n^2)$
- Equivalent to the classic dynamic programming algorithm

Mean case

- In practise very few critical points \rightarrow run-time $\ll \mathcal{O}(n^2)$

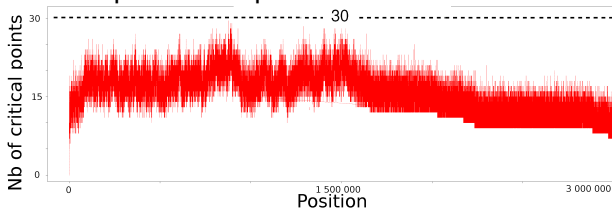
Number of critical points stored at each step

- A simulated sequence of $3 \cdot 10^6$ observations:



- Number of critical points at each step:

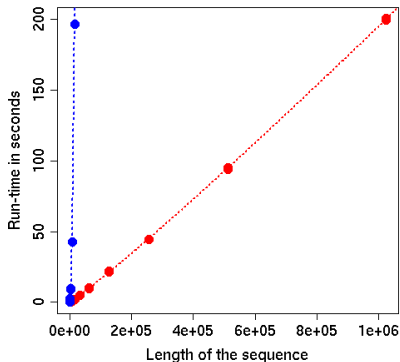
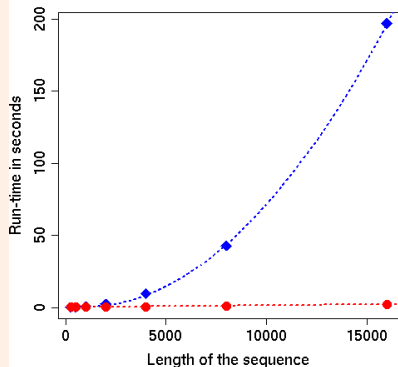
Less than 30 points compare to a worst case of $6 \cdot 10^6 - 1$



Mean time complexity

Mean time to analyze sequences of increasing size

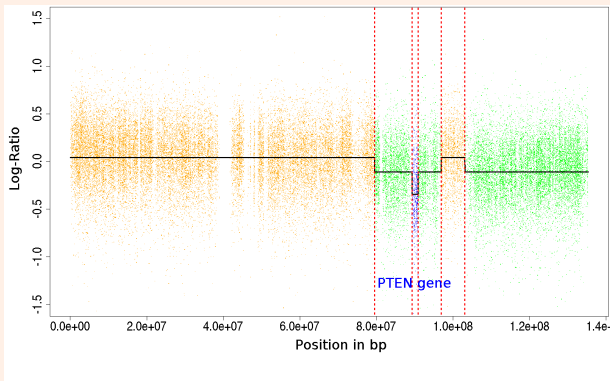
- Computer of 1.8 GHz and 1 Go RAM
- For $n = 10^6$ and $K = 50$: **3 minutes** instead of **10 days**



Application

Breast cancer sample, chromosome 10

- $n = 50\,000$, $K = 100$
- Retrieve the best segmentation in 12s



Outline

- 1 DNA copy number analysis, statistical model
- 2 Fast segmentation algorithm
- 3 Conclusion**

Conclusion

- Fast segmentation algorithm
- For $n = 10^6$ and $K = 100$ the run-time is a few minutes
- Can be generalized to other losses
 - ▶ For example : Poisson model

Thank you

Aknowledgements

- Stéphane Robin, Emilie Lebarbier, Michel Koskas
- Emmanuel Barillot
- Thierry Dubois