# *Pruned dynamic programming for optimal multiple change-point detection*

Guillem Rigaill

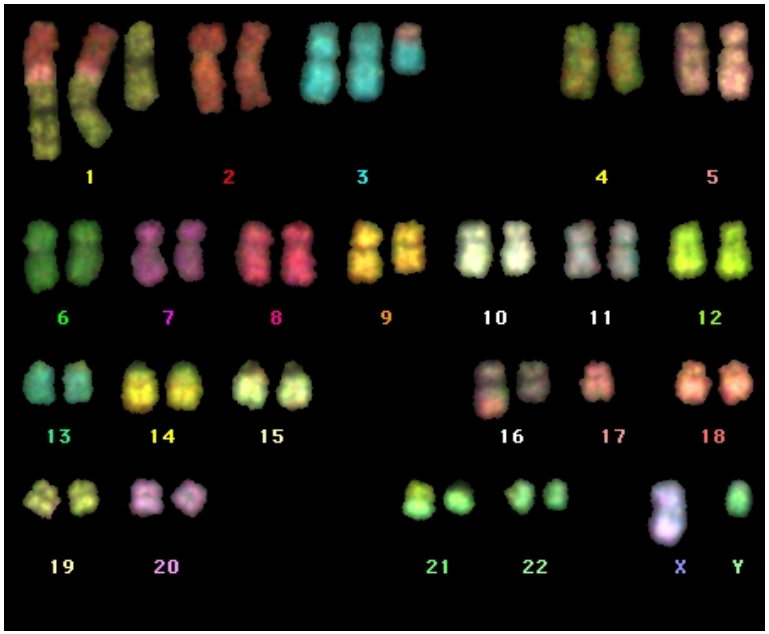
May 2010

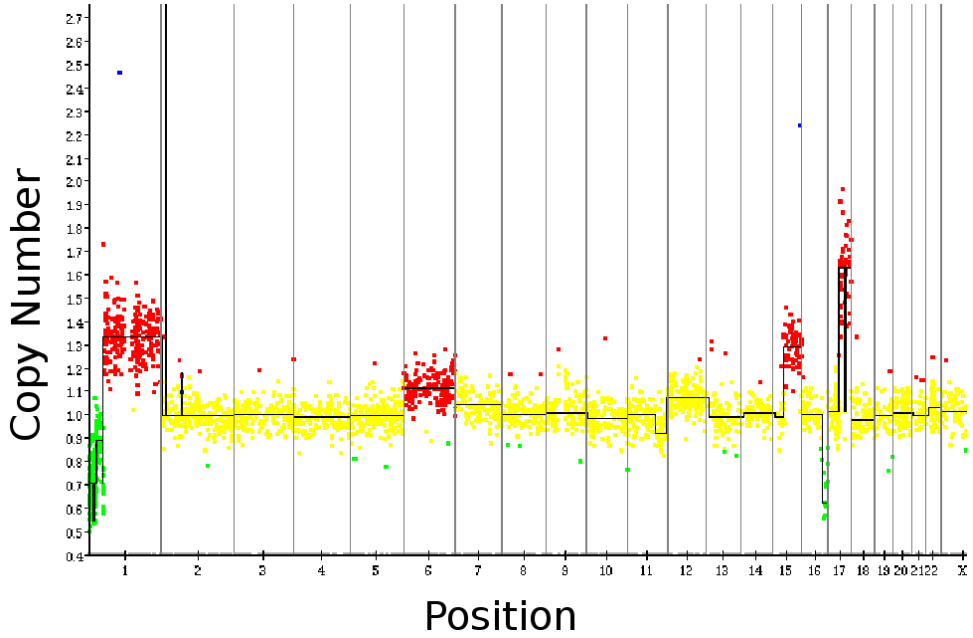
# Outline

# Outline

# DNA copy number data

- Gain and loss of DNA:
  - In normal cells: copy number = 2 (pairs of chromosomes)
  - In tumor cells: copy number $\neq 2$ on many points of the genome

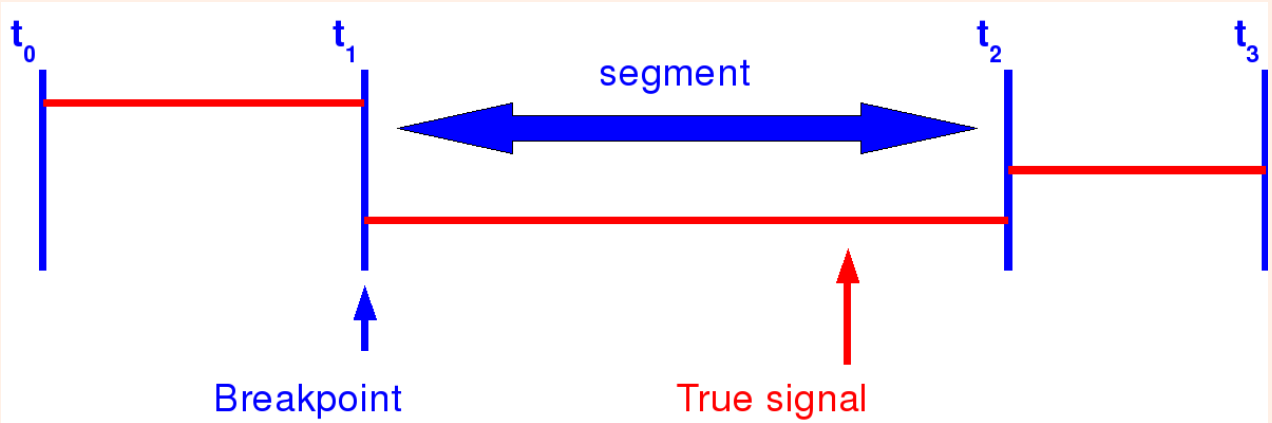


# Multiple change-point detection

## 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 homoscedastic segmentation

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

## Means

- For a given $m$ the estimation is straightforward:

$$\hat{\mu}_r = \frac{1}{n_r} \sum_{t \in r} Y_t$$

## Change-points

- Maximum likelihood and quadratic loss:

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

# Finding optimal change-point positions?

## The problem

- A lot of possible segmentations $\binom{n-1}{K-1}$:
  - $n = 10^5,\ K = 100 \rightarrow 10^{342}$
- Dynamic programming for segmentation (Bellman 1961)

## Dynamic programming (DP)

- Time complexity $\Theta(Kn^2)$
- Space complexity $\Theta(n^2)$

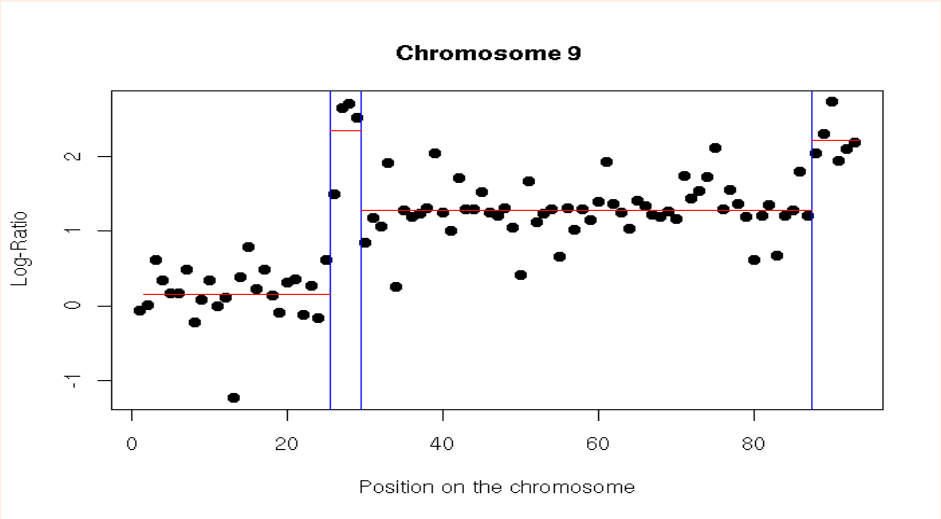
## Application to Copy Number Data

- Application to CGH data (Picard *et al.* 2005)
- One of the best methods for CGH data (Lai *et al.* 2005)

# One example

## CGH array

1. Use the DP algorithm
   - to recover the best segmentation in 1, 2, ... K segments
2. Select the number of change-points



# Outline

# Finding optimal change-point positions?

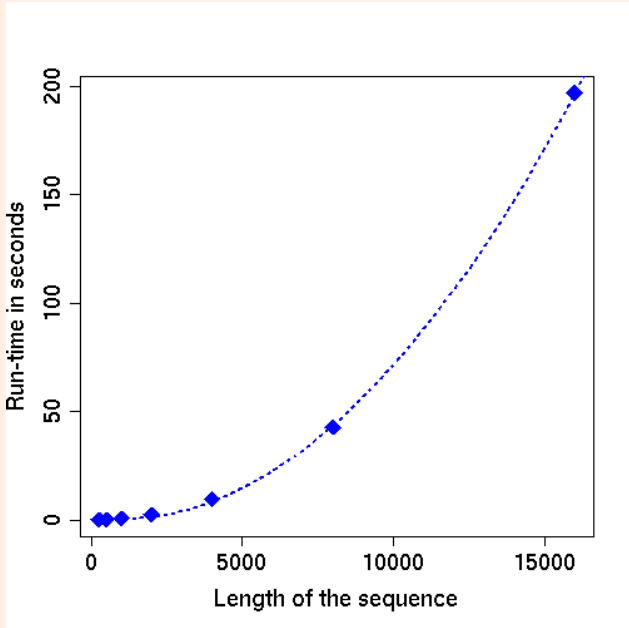
## Space Complexity - Cost matrix $\Theta(n^2)$

- Guédon (2008): $\Theta(Kn^2)$ time and $\Theta(Kn)$ space

## Time Complexity $\Theta(Kn^2)$

With a computer of $1.8 GHz$

- $n = 10^5 \rightarrow 2 - 3$ *hours*
- $n = 10^6 \rightarrow 9 - 10$ *days*
- CGH / SNP profiles:
  - $10^4 \leq n \leq 10^6$



# How to find change-point positions when $n$ is large?

## Many different ways

1. Heuristics to minimize the least square criterion
   - CART + dynamic programing (Gey and Lebarbier 2008)
2. Different optimization problem
   - Lasso (Harchaoui and Lévy-Leduc 2007):

$$min\left\{\sum_i (y_i - \beta_i)^2\right\}, \quad \text{subject to } \sum_i |\beta_i - \beta_{i+1}| < s_2$$

3. ...

- But does not retrieve the optimal solution w.r.t. the quadratic loss

# Optimal change-points w.r.t. the quadratic loss ?

## Pruned DP algorithm

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

- Can be used for large SNP profiles $\sim 10^6$

# Classical DP

## 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 $t$

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

## Segment additivity: $\Theta(t)$ comparisons at each step $\Rightarrow$ $\Theta(Kn^2)$

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

- If we know the best solutions in $K-1$ segments up to any $t_0 < t$
- We get the best solution in $K$ segments up to point $t$

# Known optimal value of the current segment $\mu^*$

## Optimization problem

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

## Point additivity: 1 comparison at each step $\Rightarrow$ $\Theta(n)$

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

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 $t$

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

# Unknown optimal value of the current segment $\mu$

## Test $P$ possible values of $\mu$

- For example a grid of $P$ regularly spaced values
- Run-time in $\Theta(Pn)$
- But does not retrieve the best solution

## Test all possible values of $\mu$ ?

- Close values of $\mu$ correspond to the same last optimal change-point
- We need to store critical values of $\mu$ corresponding to a change in the last optimal breakpoint

# Candidate last change point: cost functions

## Cost function $Cost_{k,t'}(\mu)$

- Best candidate in $k$ segments with a last change-point at $t'$:

$$\forall\ t' < t \qquad h_{k,t,t'}(\mu) = C_{k-1,t'} + \sum_{i=t'+1}^{t} \gamma(Y_i, \mu),$$

## Update

$$\forall\ t > t' \qquad h_{k,t+1,t'}(\mu) = h_{k,t,t'}(\mu) + \gamma(Y_{t+1}, \mu)$$

## Optimal solution

$$H_{k,t}(\mu) = min_{\{t' \in [\![k-1,t-1]\!]\}}\{\ h_{k,t,t'}(\mu)\ \}.$$

$$C_{k,t} = min_{\mu}\{H_{k,t}(\mu)\}$$

# Candidate last change point: winning intervals

## Set of winning intervals $Set_{k,t'}$

- Set of values such that a last change-point at $t'$ is optimal:

$$S_{k,t,t'} = \{\mu \mid h_{k,t,t'}(\mu) = H_{k,t}(\mu)\}.$$

- Set of values such that a change at $t'$ is better than a change at $t$:

$$I_{k,t,t'} = \{\ \mu \mid h_{k,t,t'}(\mu)\ \leq\ C_{k-1,t}\ \}.$$

## Update and Pruning

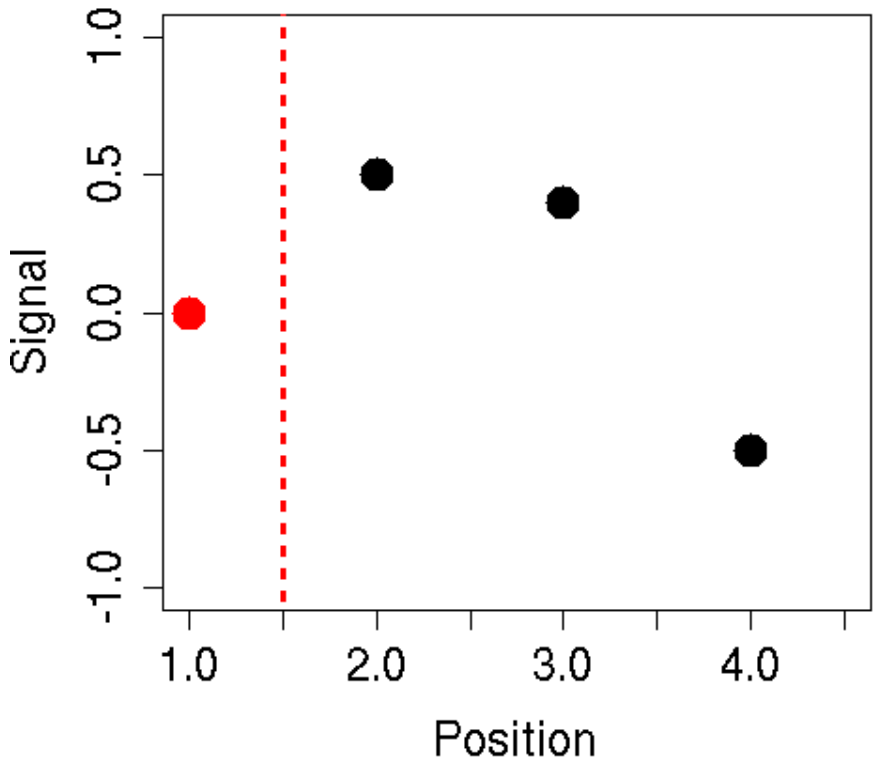
- Update:

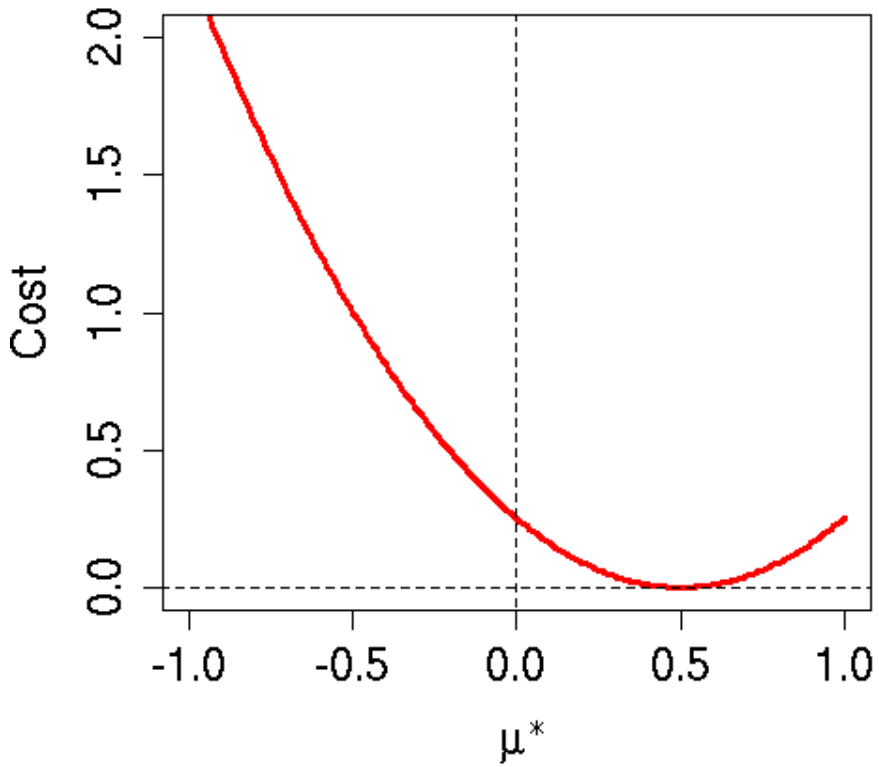
$$\begin{array}{lll} \forall\ t > t' \geq k, & S_{k,t+1,t'} & = \quad S_{k,t,t'} \cap I_{k,t,t'} \\ \forall\ t' \geq k, & S_{k,t',t'}, & = \quad \complement_{\mathbb{R}}(\cup_{t \in [\![k-1,t'-1]\!]} I_{k,t,t'}) \end{array}$$

- Pruning: $S_{k,t,t'} = \emptyset \quad \Rightarrow \quad \forall\ t^* \geq t \quad S_{k,t^*,t'} = \emptyset$

# An example

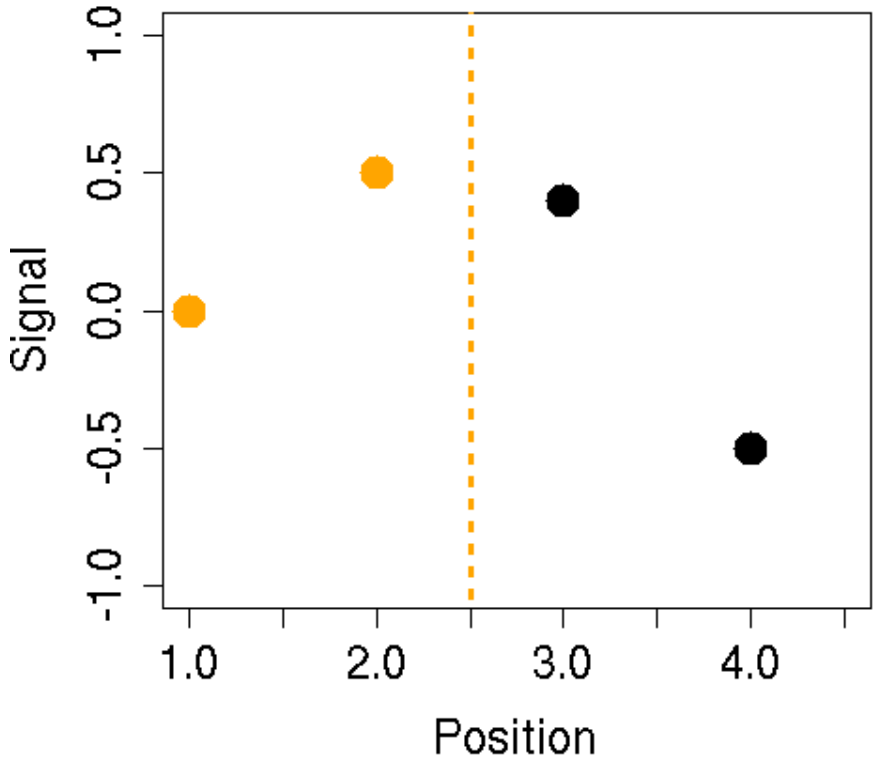
| Candidate | Cost function | Set of Intervals |
|---|---|---|
| $t' = 1$ | $Cost_{2,1} = 0 + (0.5 - \mu)^2$ | $Set_{2,1} = [-0.5,\ 0.5\ ]$ |

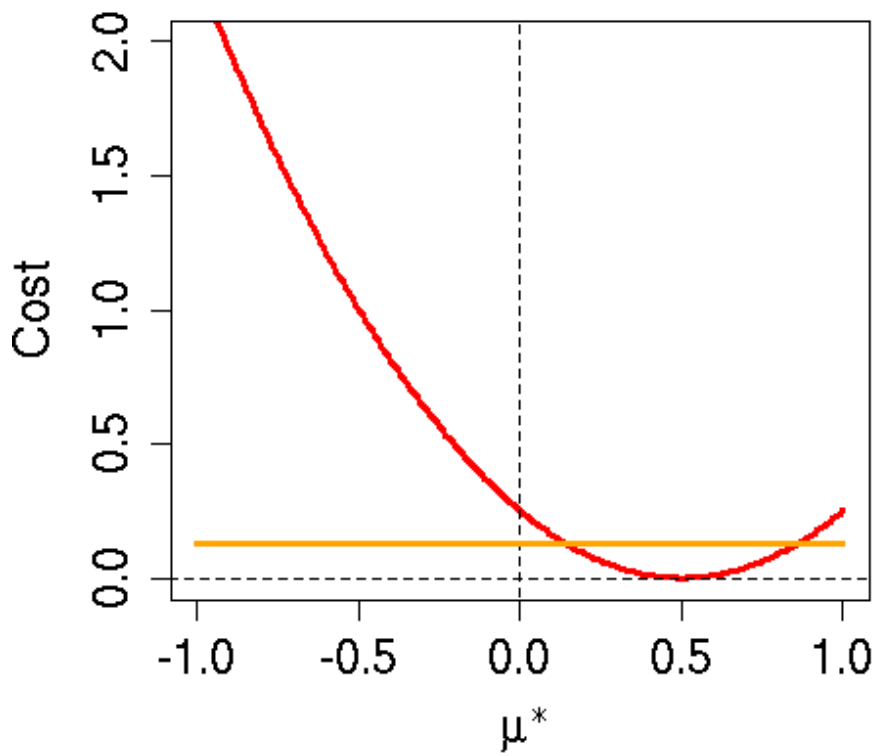


# An example

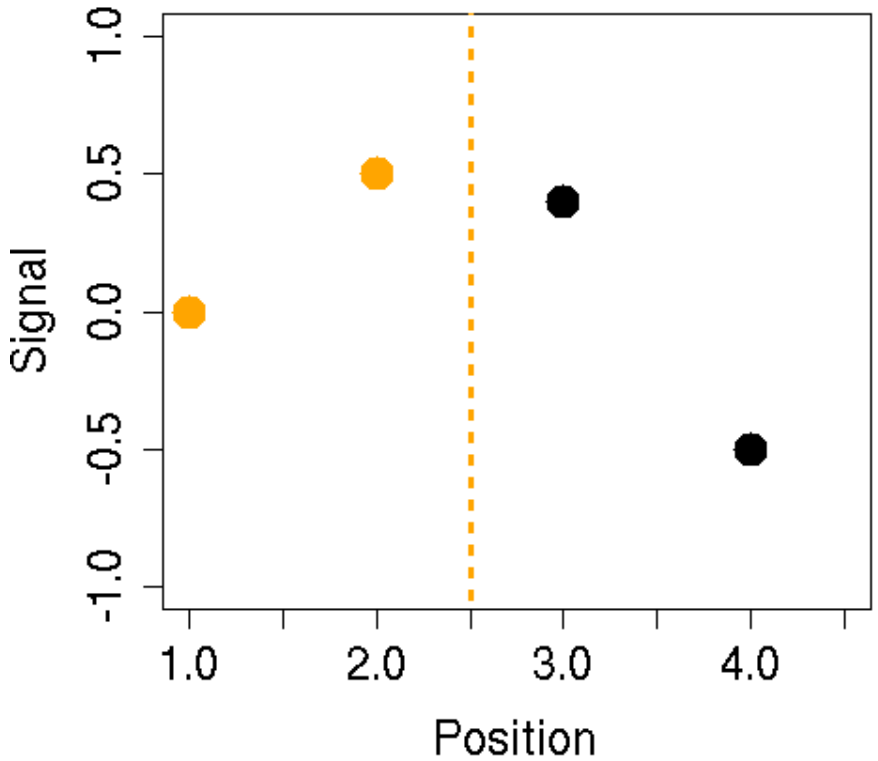
| Candidate | Cost function | Set of Intervals |
|---|---|---|
| $t' = 1$ | $Cost_{2,1} = 0.25 - \mu + \mu^2$ | $Set_{2,1} = [\ 0.146\ ,\ 0.5\ ]$ |
| $t' = 2$ | $Cost_{2,2} = C_{1,2} = 0.125$ | $Set_{2,2} = [-0.5, 0.146\ ]$ |

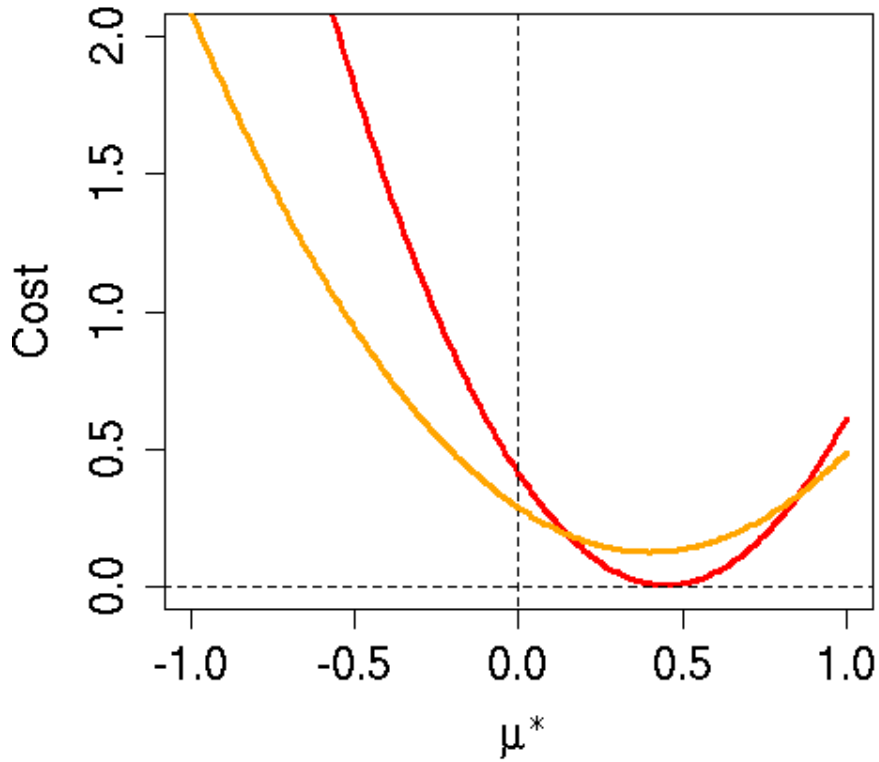


# An example

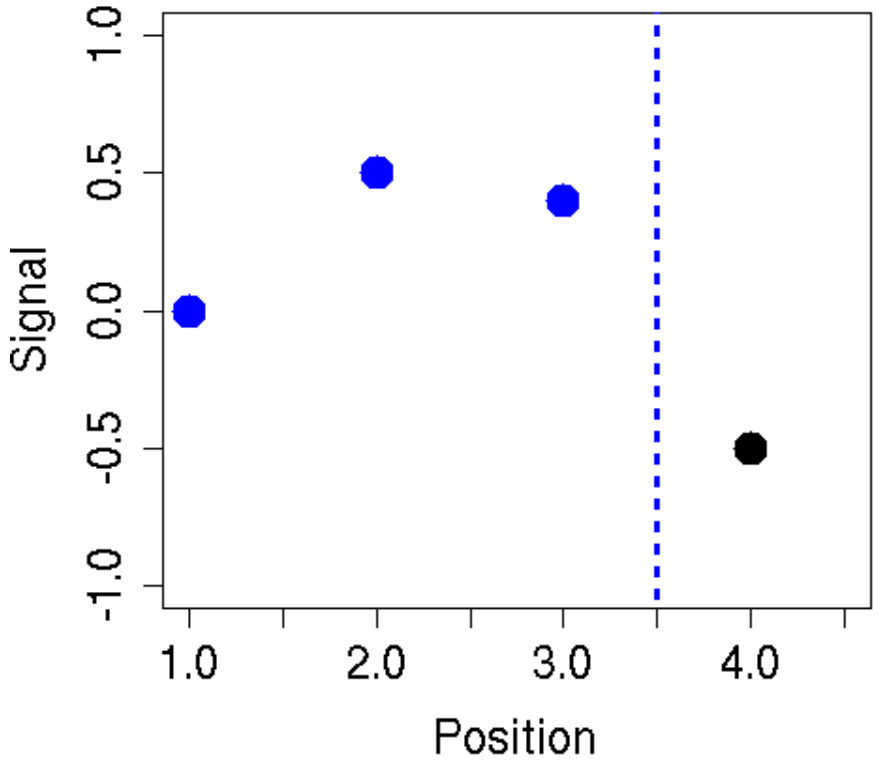
| Candidate | Cost function | Set of Intervals |
|---|---|---|
| $t' = 1$ | $Cost_{2,1} = 0.41 - 1.8\mu + 2\mu^2$ | $Set_{2,1} = [\ 0.146\ ,\ 0.5\ ]$ |
| $t' = 2$ | $Cost_{2,2} = 0.285 - 0.8\mu + \mu^2$ | $Set_{2,2} = [-0.5, 0.146\ ]$ |

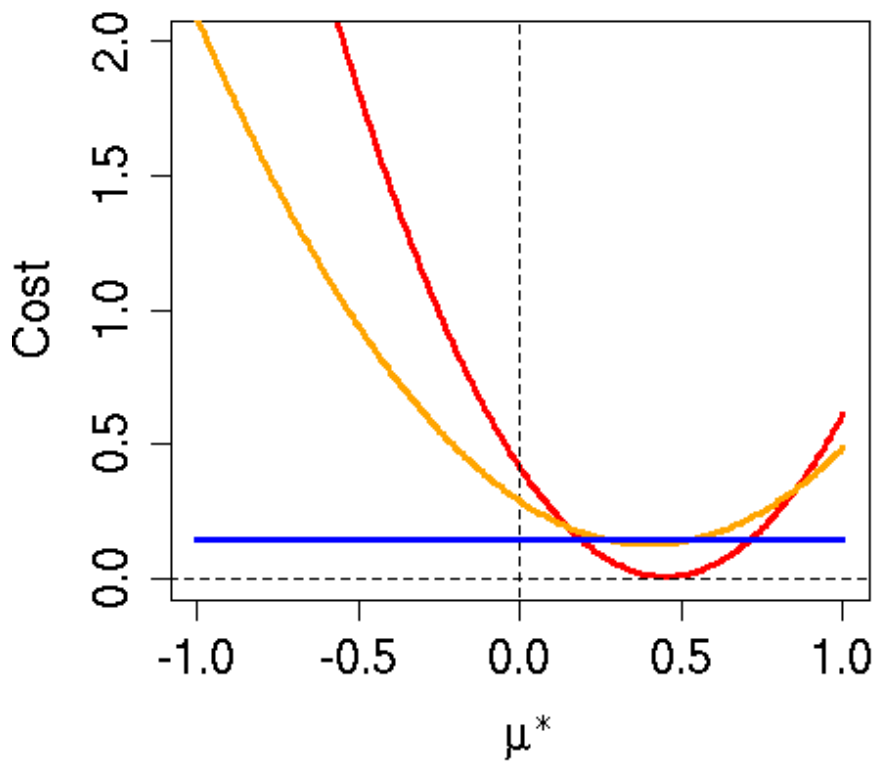


# An example

| Candidate | Cost function | Set of Intervals |
|---|---|---|
| $t' = 1$ | $Cost_{2,1} = 0.41 - 1.8\mu + 2\mu^2$ | $Set_{2,1} = [\ 0.190\ ,\ 0.5\ ]$ |
| $t' = 3$ | $Cost_{2,3} = C_{1,3} = 0.14$ | $Set_{2,3} = [-0.5, 0.190\ ]$ |



# Worst case and empirical time complexity

## Worst case

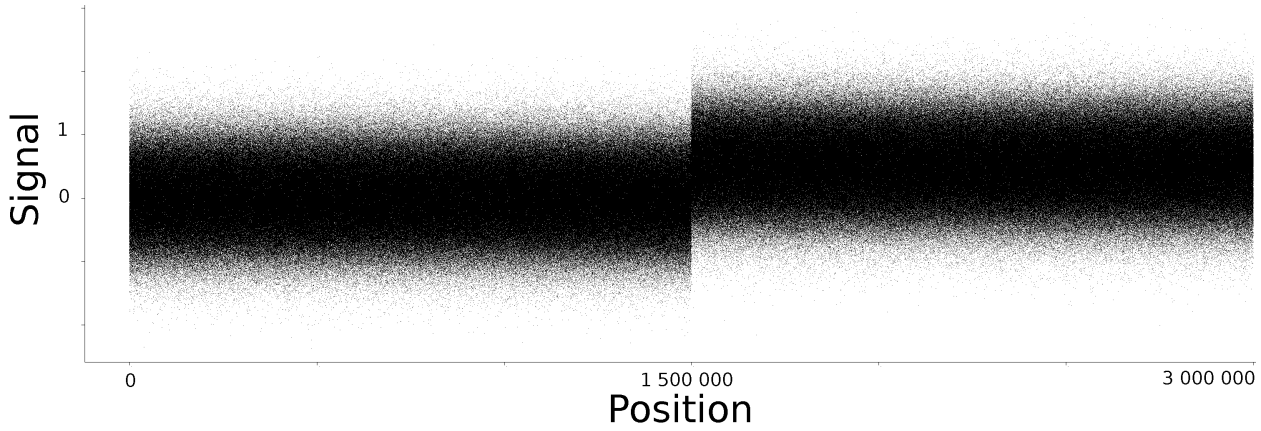
- Corresponds to a maximum number of intervals
- At worst $2n - 1$ intervals
- Worst complexity in time: $O(Kn^2)$
- Space complexity: $\Theta(Kn)$ space
- At worst equivalent to the classic DP algorithm

## Empirical complexity

- In practice very few candidates $\rightarrow$ runtime $\ll O(n^2)$

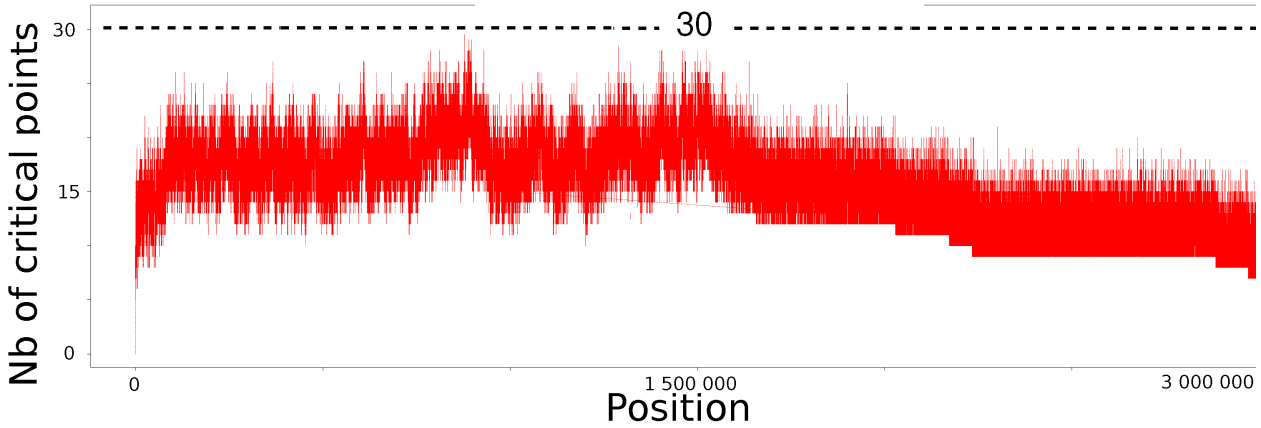
# Number of intervals stored at each step

- A simulated sequence of $3.10^6$ observations:




- Number of intervals at each step:

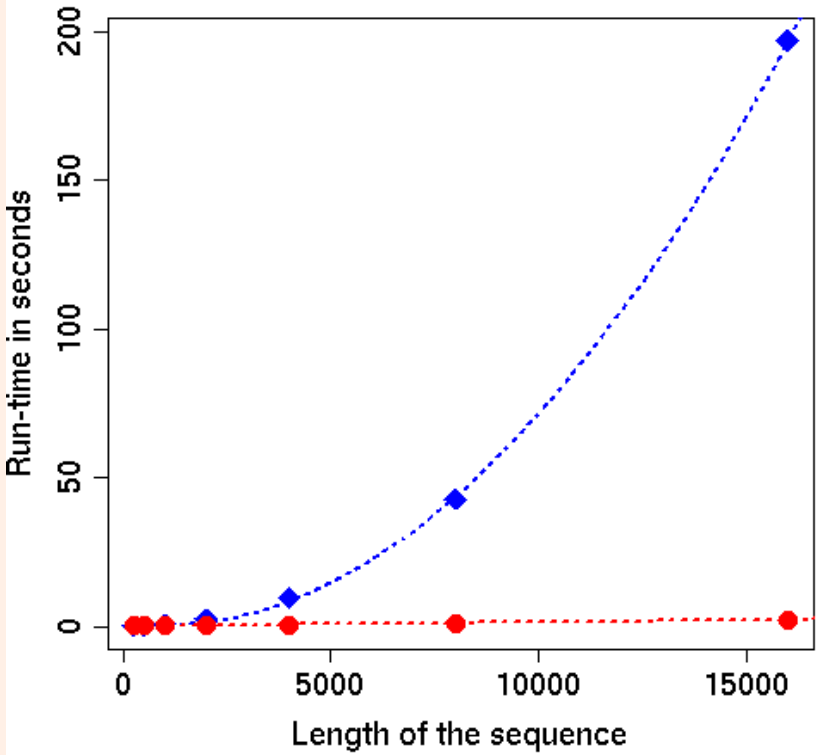
Less than 30 points compared to a worst case of $6.10^6 - 1$

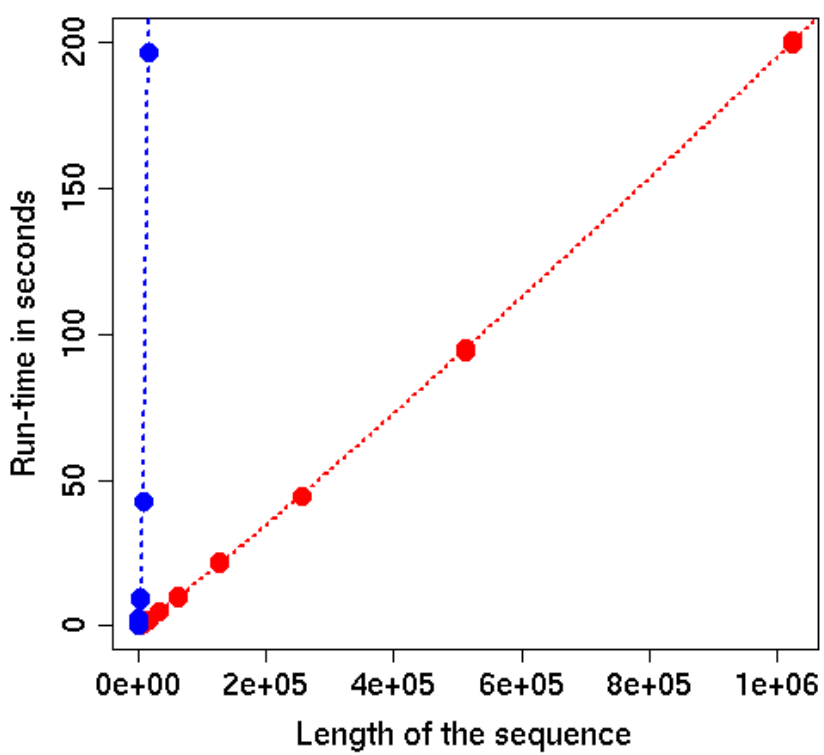


# Empirical time complexity

## Time to analyze sequences of increasing size

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

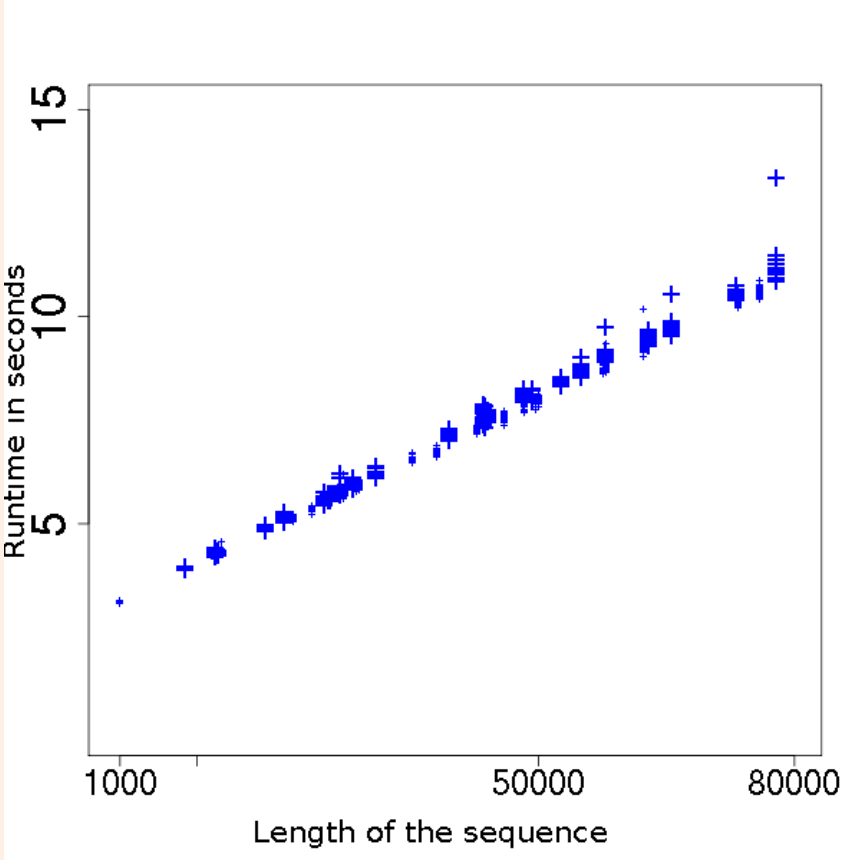


# Empirical time complexity

## Real Data

- Computer of 3.16*GHz*
- GEO GSE17359 dataset $2 \times 18 \times 24$ chromosomes



# Conclusion

- Optimal segmentation w.r.t. the quadratic loss
- At worst in $O(Kn^2)$
- In practice
  - For $n = 10^5$ and $K = 100$ several seconds
  - For $n = 10^6$ and $K = 100$ a few minutes
- Can be generalized to other losses
  - For example: Poisson model

# Thank you

Aknowledgements

- Stéphane Robin, Emilie Lebarbier, Michel Koskas, Tristan Mary-Huard
- Emmanuel Barillot, Philippe Hupé, Tatiana Popova
- Thierry Dubois, Bérengère Marty, Virginie Maire, Aurélie Dumont-Telliez, Marion Richardson