Fast and optimal changepoint detection method using nonlinear penalties and functional pruning

Liehrmann Arnaud

University of Evry Val d'Essone

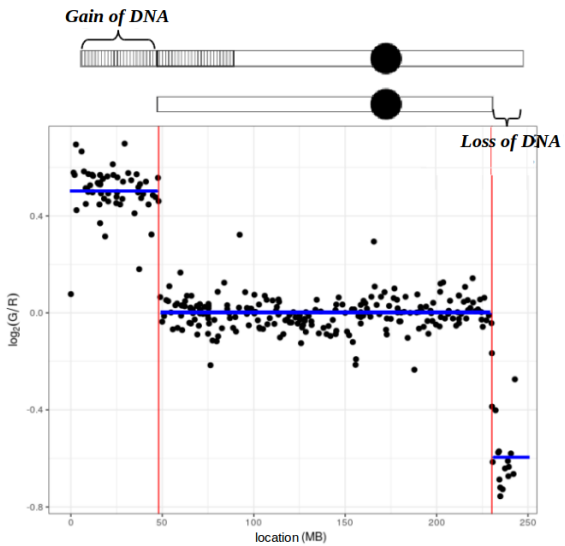
Feb 14, 2020

Summary

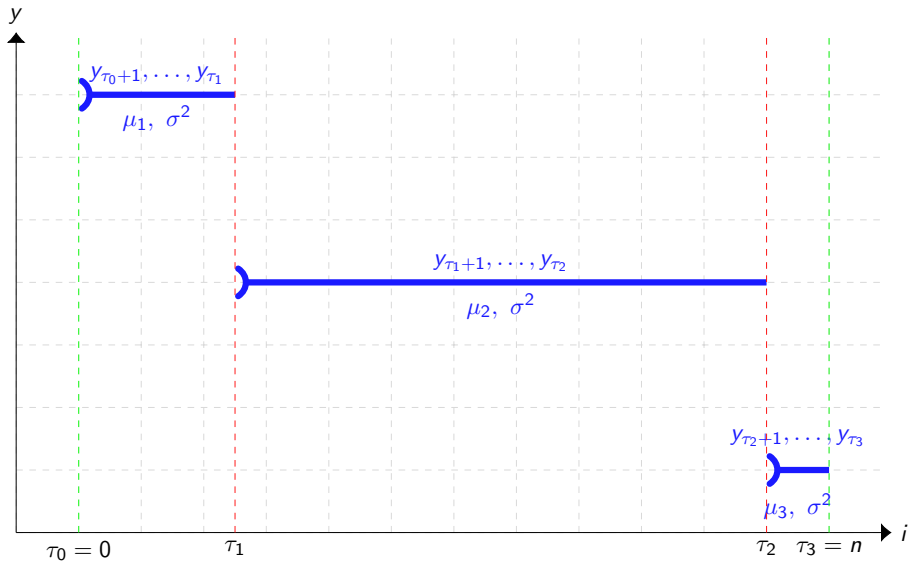
1. Introduction
2. Resolution by dynamic programming
3. Implementation of FpopPSD
4. Results
5. Conclusion & outlook

Statistical segmentation model and application

CNV profile
obtained through CGH-array



Graphical representation of
the statistical segmentation model.



Statistical segmentation model *(Picard et al. 2005)*

$$\forall i \quad | \quad \tau_{j-1} + 1 \leq i \leq \tau_j, \quad Y_i \sim \mathcal{N}(\mu_j, \sigma^2) \quad iid$$

Limit of currently used methods

Comparison of segmentation methods *(Hocking et al. 2013)*

17 methods compared on CNV profiles from neuroblastoma cells, annotated by biologists and bioinformaticians.

Results

11.6% of the changepoints are not detected by the best methods with the more generous smoothing parameters. (**FPOP** *Maidstone et al. 2017* et **PELT** *Killick et al. 2012*)

The goal

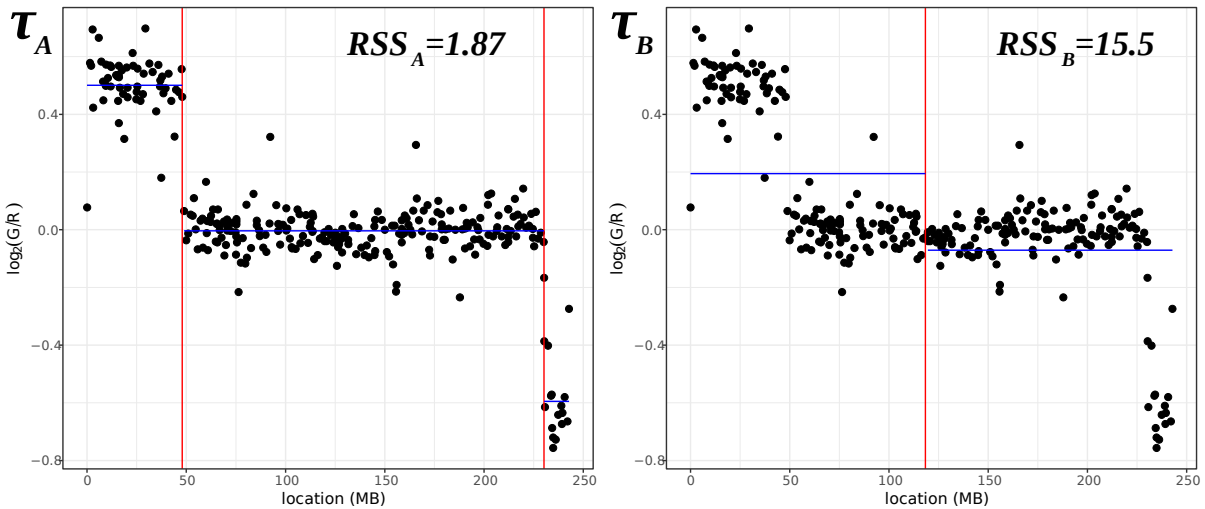
Develop a method which reduce the number of undetected changepoints.

Definition of the penalized problem

Statistical goal

Select $\boldsymbol{\tau}$ that optimizes the penalized likelihood criteria, **among 2^{n-1} solutions**:

$$F_n = \min_{\boldsymbol{\tau}} \left\{ \sum_{j=1}^{|\boldsymbol{\tau}|+1} \sum_{i=\tau_{j-1}+1}^{\tau_j} \overbrace{(y_i - \mu_j)^2}^{\text{data fitting}} + \overbrace{\text{pen}(\boldsymbol{\tau})}^{\text{promote parsimonious models}} \right\}$$



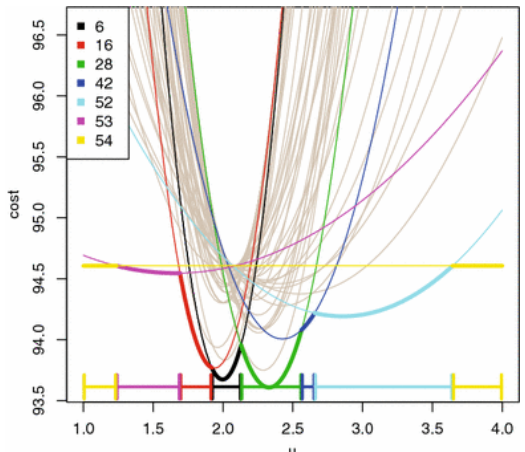
Functionalization of the penalized problem

Algorithmic goal

Given the parameter μ of the last segment:

$$\widetilde{F}_n(\mu) = \min_{\boldsymbol{\tau}} \left\{ \overbrace{\sum_{j=1}^{|\boldsymbol{\tau}|} \sum_{i=\tau_{j-1}+1}^{\tau j} (y_i - \mu_j)^2}^{\text{optimal cost up to the last segment}} + \overbrace{\sum_{i=\tau_{|\boldsymbol{\tau}|-1}+1}^{\tau_{|\boldsymbol{\tau}|}} (y_i - \mu)^2}^{\text{cost of the last segment}} + \alpha|\boldsymbol{\tau}| \right\}$$

$$F_n = \min_{\mu} \left[\widetilde{F}_n(\mu) \right]$$



Summary

1 Introduction

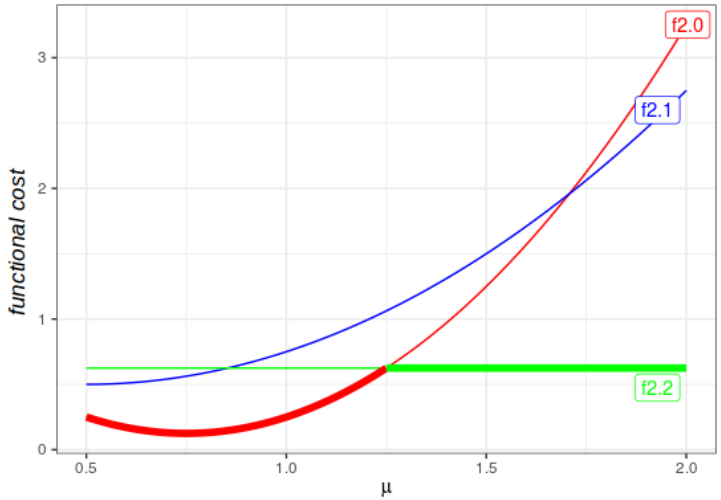
2 Resolution by dynamic programming

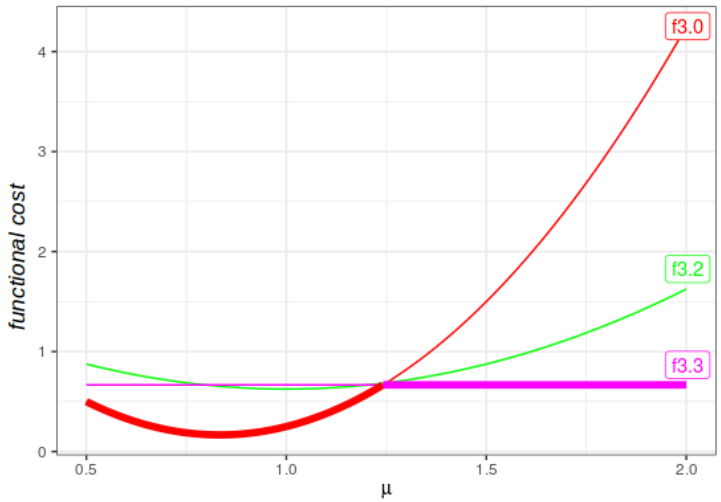
3 Implementation of FpopPSD

4 Results

5 Conclusion & outlook

Functional pruning optimal partitioning (FPOP)





Update of $\widetilde{F}_t(\mu)$ *(Maidstone et al. 2017)*

$$\widetilde{F}_t(\mu) = (y_t - \mu)^2 + \min \left\{ \overbrace{\widetilde{F}_{t-1}(\mu)}^{\text{constant state}} , \overbrace{F_{t-1} + \alpha}^{\text{jump}} \right\}$$

Representation of candidates

1. $\widetilde{f}_{t,s}(\mu) = F_s + \sum_{i=s+1}^{t} (y_i - \mu)^2 + \alpha$
2. $Z^*_{t,s} = \{\mu \mid \widetilde{f}_{t,s}(\mu) = \widetilde{F}_t(\mu)\}$ (reduced at each step)
3. $Z^*_{t,s} = \emptyset \implies Z^*_{t+1,s} = \emptyset$ **(pruning)**

Penalty on segments length

Algorithmic goal

Select $\boldsymbol{\tau}$ that optimizes the penalized likelihood criteria, conditional on the parameter μ of the last segment:

$$\widetilde{F}_n(\mu) = \min_{\boldsymbol{\tau}} \left\{ \overbrace{\sum_{j=1}^{|\boldsymbol{\tau}|-1} \sum_{i=\tau_{j-1}+1}^{\tau j} (y_i - \mu_j)^2 + \sum_{i=\tau_{|\boldsymbol{\tau}|}+1}^{\tau_{|\boldsymbol{\tau}|}} (y_i - \mu)^2 + \alpha|\boldsymbol{\tau}|}^{\text{before}} \underbrace{-\beta \sum_{j=1}^{|\boldsymbol{\tau}|} \log\left(|\tau_j - \tau_{j-1}|\right)}_{\text{disadvantages the small segments}} \right\}$$

Differences with FPOP

Representation of candidates

1. $\widetilde{f}_{t,s}(\mu) = F_s + \overbrace{\sum_{i=s+1}^{t} (y_i - \mu)^2}^{before} + \alpha - \beta log\left(|t-s|\right)$

Exact living area?

2. $Z^*_{t,s}$ (~~reduced at each step~~ can expand)
3. $\boldsymbol{Z^*_{t,s} = \emptyset \not\Longrightarrow Z^*_{t+1,s} = \emptyset}$ **(if pruned, optimality of algorithm unsecured)**

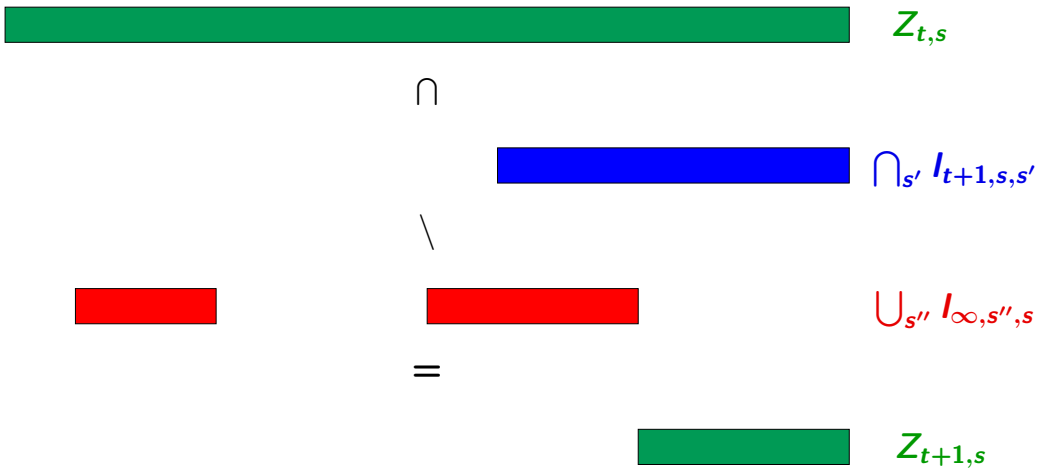
Approximation of the exact living area

2. $Z_{t,s}$ including $Z^*_{t,s}$ (reduced at each step)
3. $\boldsymbol{Z_{t+1,s} = \emptyset \Longrightarrow Z^*_{t+1,s} = \emptyset}$ **(pruning)**

Update rule for $Z_{t,s}$

Update rule for $Z_{t,s}$

$$Z_{t+1,s} = Z_{t,s} \cap \overbrace{\left(\bigcap_{s'} I_{t+1,s,s'}\right)}^{\text{comparisons with the future}} \setminus \overbrace{\left(\bigcup_{s''} I_{\infty,s'',s}\right)}^{\text{comparisons with the past}}$$



Summary

1 Introduction

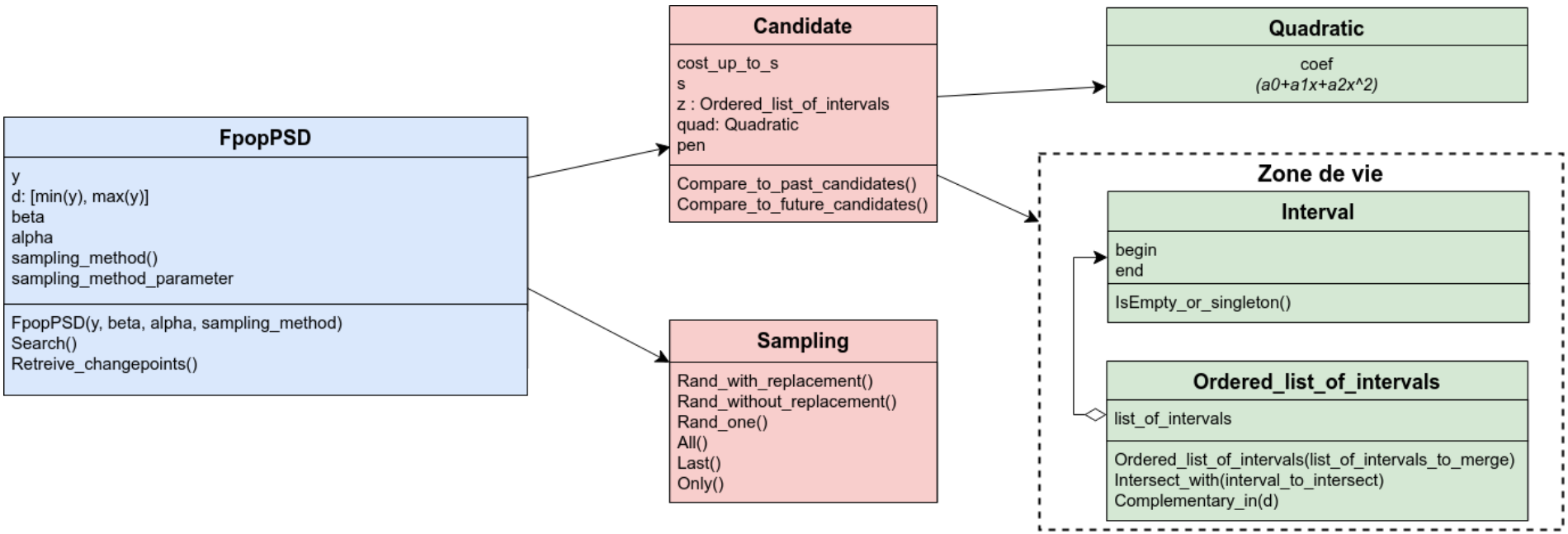
2 Resolution by dynamic programming

3 Implementation of FpopPSD

4 Results

5 Conclusion & outlook

Class diagram C++



Summary

1 Introduction

2 Resolution by dynamic programming

3 Implementation of FpopPSD

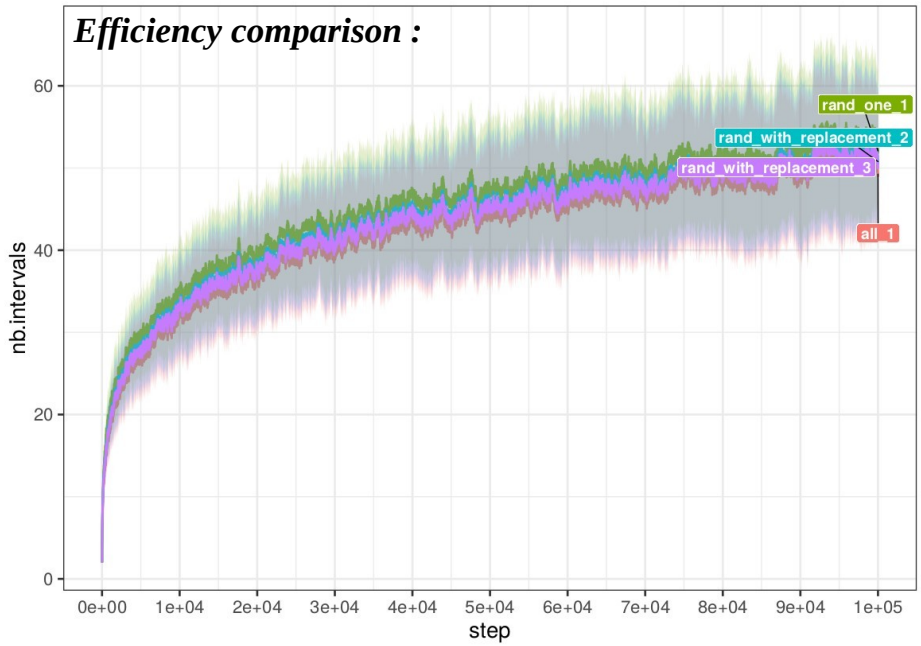
4 Results

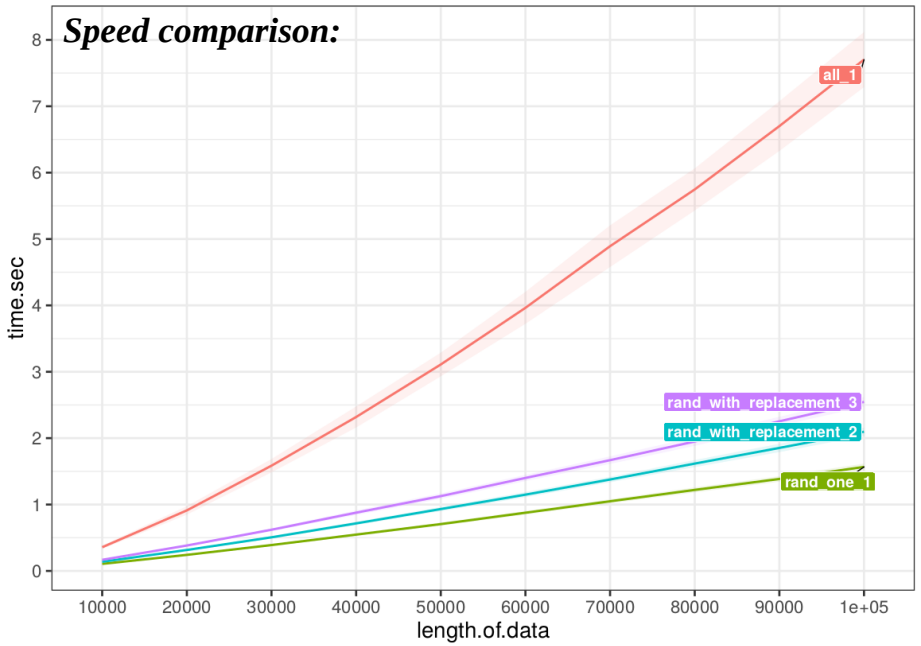
5 Conclusion & outlook

Comparison of sampling strategies

Question

Is there an efficient and fastly sampling strategy?





Package R: acnr (Pierre-Jean et al. 2015)

Data

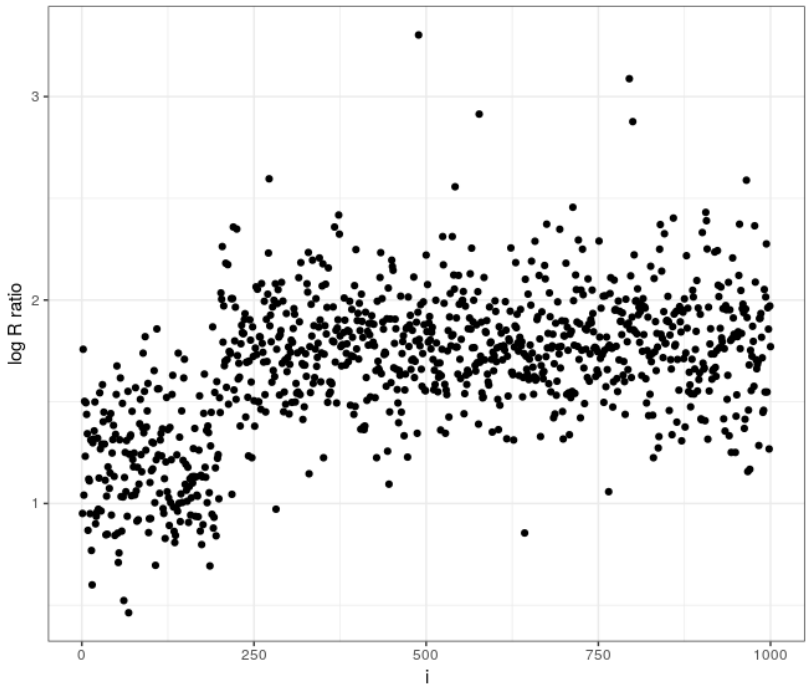
Realistic profiles of CNV for tumor cells.

Calibration of FpopPSD and FPOP smoothing parameters

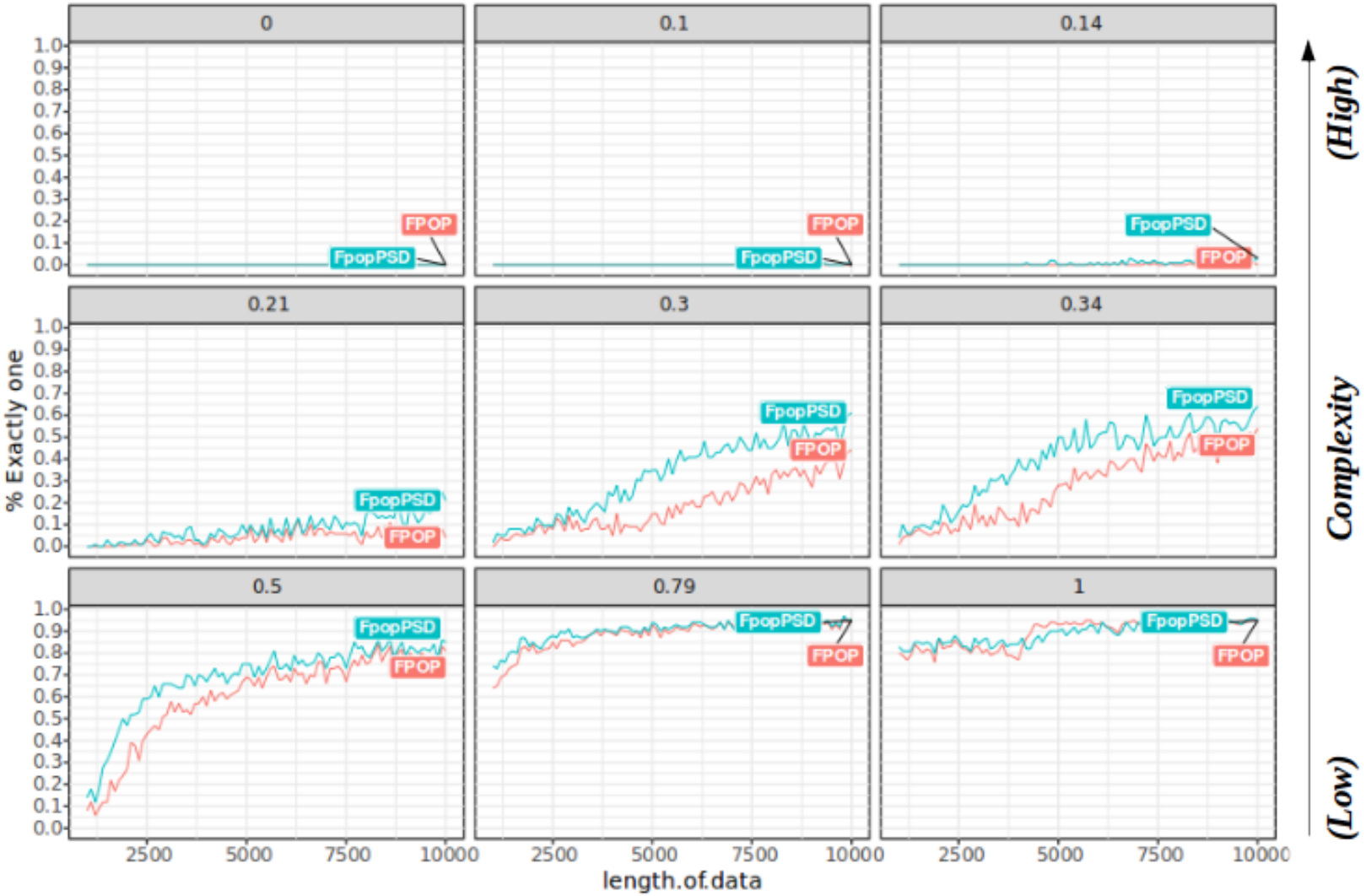
Profiles with one changepoint whose random location is known (ratio of tumor cells: 1).

Comparison criteria of FPOP/FpopPSD

Exactly one: the segmentation returns a changepoint located at + or - 20 points of the true changepoint location.



Better results than FPOP



Summary

1 Introduction

2 Resolution by dynamic programming

3 Implementation of FpopPSD

4 Results

5 Conclusion & outlook

Conclusion & outlook

- fast method ($\sim$ 1.5sec for 10^5)
- FpopPSD better than FPOP on tested profiles
- Try it on other more complex profiles or benchmark dataset (neurblastoma dataset)
- package R already available on GitHub (`https://github.com/aLiehrmann/FpopPSD`)

Thank you