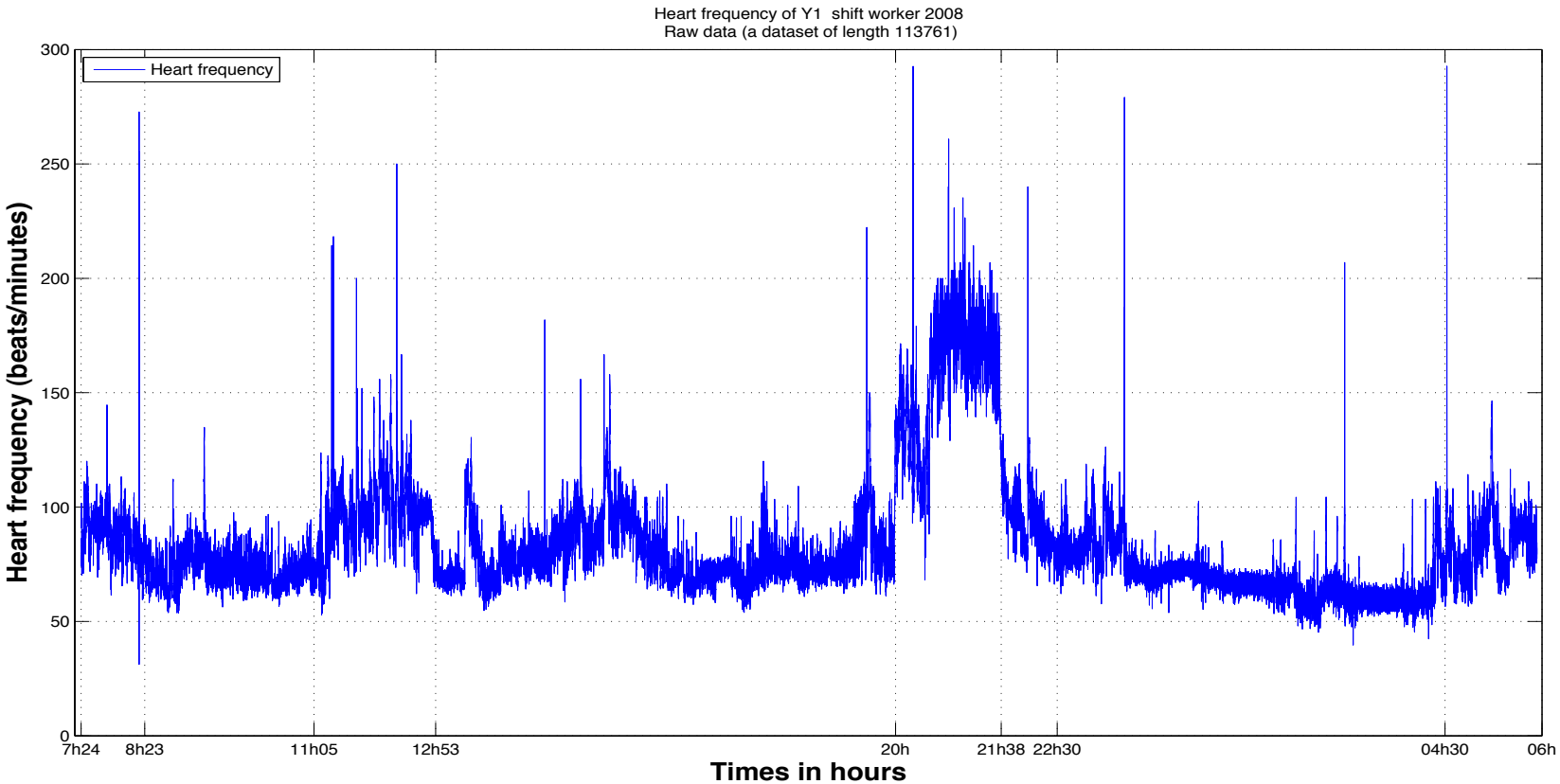
Une méthode rapide de détection de ruptures
Dérivée filtrée + p-value

Pierre R. BERTRAND

Laboratoire de Mathématiques, CNRS UMR 6620
& Université de Clermont-Ferrand 2, France.

Do Well B.
Saint-Nectaire 24-26 juin 2013

An example of large time series: Heart rate during 24 hours



- We record the duration of each heart beat.
- We have a time series of length $N = 113,700$.
- We want to test if there exist change on the mean

The problem of change detection

Model

The series X_t, for $t = 1, \ldots, n$ is a sequence of independent random variable with change on the mean. To sum up, we have:

- a configuration a K change points $\tau = (\tau_1, \ldots, \tau_K)$ enlarged by adding $\tau_0 = 0$ and $\tau_{K+1} = n$.
- the configuration of mean values $\mu = (\mu_0, \ldots, \mu_K)$ with X_t iid, $\mathbb{E}(X_t) = \mu_k$, $\mathrm{Var}(X_t) < +\infty$, for $t \in (\tau_k, \tau_{k+1}]$ and for all $k = 0, \ldots, K$.

Statistical question: Estimate the configuration of change $\tau = (\tau_1, \ldots, \tau_K)$ and the mean values $\mu = (\mu_0, \ldots, \mu_K)$.

A new problem of change detection

The data deluge

Since the beginning of 21st century, we have access to large or huge datasets, and to large or huge time series

- **A new question:**
  Change detection with a small enough computational complexity.

Penalized Least Square

State of the Art before the Data Deluge:

Penalized least square (PLS) method

- The penalized least square (PLS) method has both

$$\text{time and memory complexity :} \quad \mathcal{O}(n^2),$$

  where n denotes the size of the time series.
- In today applications $n \simeq 100,000$.
- In the near future, $n \simeq$ *millions*.

State of the Art in Change Point Analysis before Data Deluge: Penalized least square (PLS) method.

- The **penalized** contrast function can be written as follows:

$$J(K, \mathbf{T}, X) = \frac{1}{n} \sum_{k=1}^{K+1} U\left(X_{T_{k-1}+1}, \ldots, X_{T_k}\right) + \beta \mathbf{pen}(\mathbf{T})$$

  where $U\left(X_{T_{k-1}+1}, \ldots, X_{T_k}\right)$ is the contrast function for estimating θ in the segment $(T_{k-1} + 1, T_k)$.
- The change configuration is estimated by

$$(\tau_1, \ldots, \tau_K) = \arg\min_{\mathbf{T}} J(K, \mathbf{T}, X).$$

Complexity of PLS

PLS needs to compute and store the upper triangular matrix:

$$M = \left(U\left(X_{i+1}, \ldots, X_j\right)\right)_{1 \leq i \leq j \leq n}$$

Time complexity

Computation of the values $U\left(X_{i+1}, \ldots, X_j\right)$ needs time complexity of $\mathcal{O}(n^2)$.

Memory complexity

Storage of a matrix of size n $\Rightarrow$ Memory allocation: $n^2/2$.

Filtered Derivative with p-value (FDpV)

- FDpV is a two steps procedure:
- Step 1, so-called filtered derivative, selects the right change points but also wrong ones.
- Step 2 (based on p-values) is carried out to remove the false detection from the list of potential change points found in Step1.
- Since Step 1 is based on MOving SUMs, the decision function can be calculated iteratively, leading to

$$\begin{aligned} \text{memory complexity} &= N \\ \text{time complexity} &= C_1 \times N \end{aligned}$$

More details on FDpV

Definition (Basseville & Benveniste (1984))

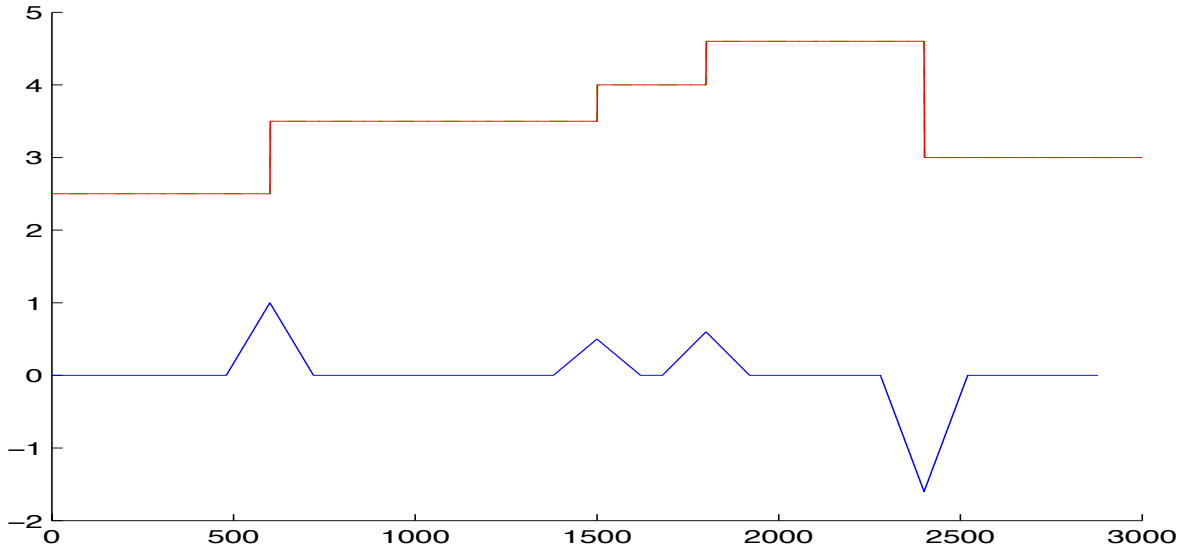
The filtered derivative function is

$$FD(t, A) = \hat{\mu}(t+1, t+A) - \hat{\mu}(t-A+1, t) \tag{1}$$

where $\hat{\mu}(t+1, t+A)$ *denote the empirical mean estimated on the box* $(t+1, t+A)$*.* $FD(t, A)$ *is defined by (1) for* $t \in [A, n-A]$ *and set to* 0 *elsewhere.*

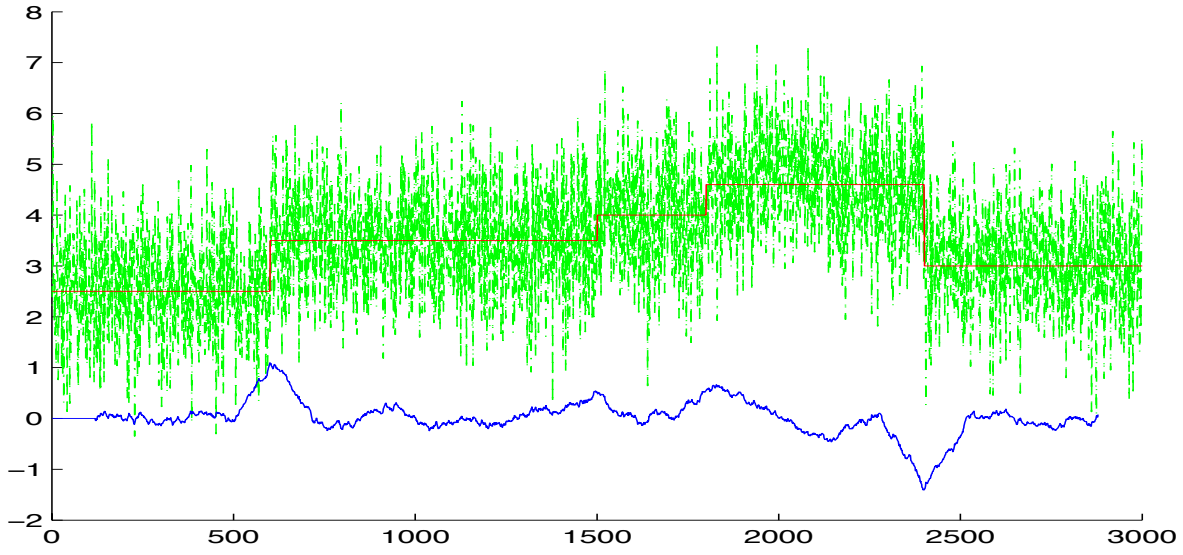
The function $FD(t, A)$ is the difference between the parameter of interest estimated on a sliding window of size A at the right of the point t and the parameter estimated on a sliding window of size A at the left of the point t.

Ex: Filtered Derivative without noise



Red: the right signal with change on the mean.
Blue: Filtered Derivative function with hat corresponding to change points.

Filtered Derivative with noise

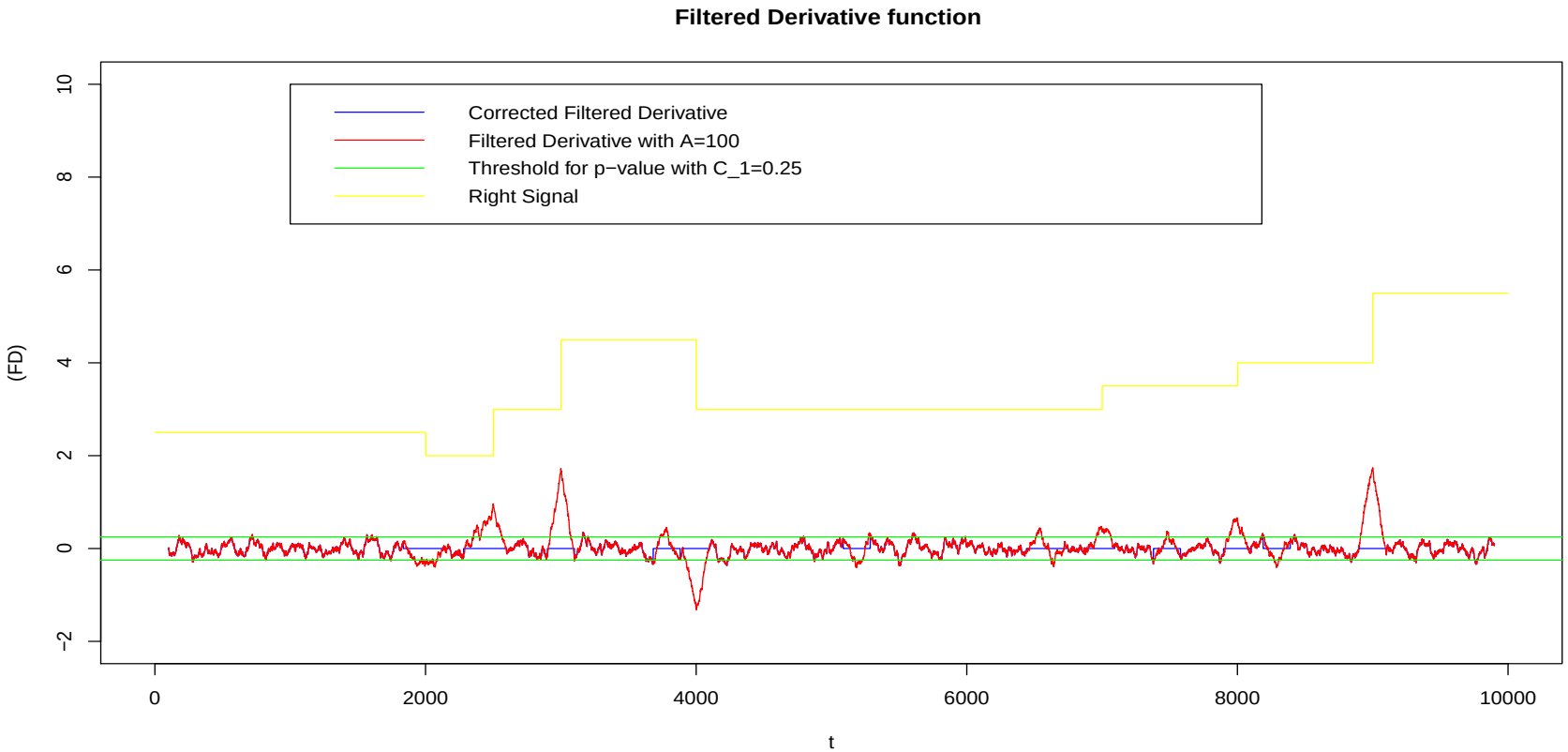


We want to estimate:

- The change points configuration: $\tau = (\tau_1, \ldots, \tau_K)$,
- The mean μ_k of each segment $(\tau_{k-1} + 1, \tau_k)$.

Detection of change points. Step 1

Step 1: Potential change points $\widetilde{\tau}_k$, for $k = 1, \ldots, \widetilde{K}$, are selected as local maxima of the absolute value of the filtered derivative $|FD(t, A)|$ where moreover $|FD(\widetilde{\tau}_k, A)|$ exceed a given threshold C_1.



Detailed algorithm

- Without noise, we get hats of width $2A$ and hight $\mu_{k+1} - \mu_k$ at each change point τ_k.
- So, we select as first potential change point $\widetilde{\tau}_1$ the global maximum of the function $|FD(t, A)|$.
- Then we set $k = 1$ and we define the function FD_{k+1} by putting to 0 a vicinity of width $2A$ of the point $\widetilde{\tau}_k$ and we iterate this algorithm while $|FD_k(\widetilde{\tau}_k, A)| > C_1$.

1. At end of the step 1, we have detected the **potential change points configuration**, denoted by $\widetilde{\tau} = \left(\widetilde{\tau}_1, \ldots, \widetilde{\tau}_{\widetilde{K}_{\max}}\right)$.

Step 2: False alarms elimination

1. For each of the potential change point $\widetilde{\tau}_k$ detected at Step 1, we test
$(H0_k) : \mu_k = \mu_{k+1}$ versus $(H1_k) : \mu_k \neq \mu_{k+1}$.
2. Under the null hypothesis,

$$\widetilde{t}_k^{\star} = \frac{\widehat{\mu}_k - \widehat{\mu}_{k-1}}{\sqrt{\frac{S_{k-1}^2}{N_{k-1}} + \frac{S_k^2}{N_k}}},$$

has a Student distribution of degree $d = N_k + N_{k-1} - 2$, which is almost Gaussian, with $N_k = \widetilde{\tau}_{k+1} - \widetilde{\tau}_k$.
3. In BFG (2011), we keep only the change points $\widetilde{\tau}_k$ such that

$$\widetilde{p}_k := 2 \times \left\{ 1 - \Phi\left(\left| \widetilde{t}_k^{\star} \right| \right) \right\} < p_2^*$$

Complexity

Time complexity

The filtered derivative function FD is computed iteratively:

$$A \times FD(t+1, A) = A \times FD(t, A) + \left[X_{t+A+1} - 2X_t + X_{t-A+1}\right]$$

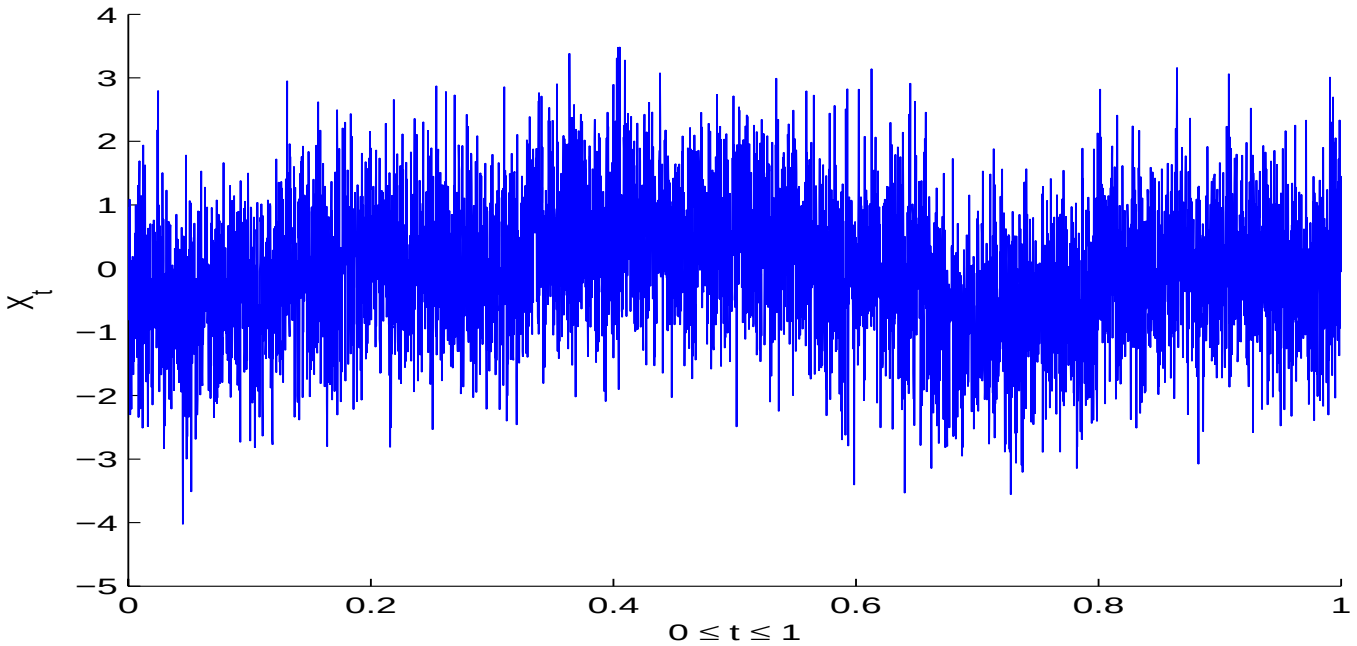
$\Rightarrow$ **Complexity of** $3n$ *operations*.

Memory complexity

Storage of a vector of size n

$\Rightarrow$ **Memory allocations:** n.

Signal to be segmented

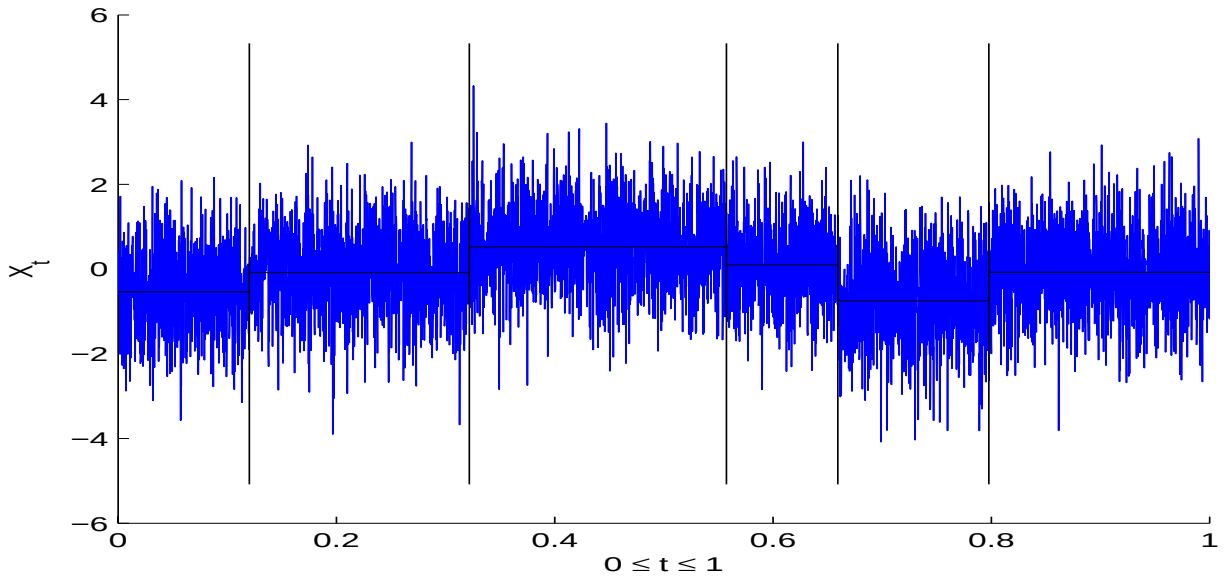


Data:

- Independent Gaussian r.v,
- $n = 5000$,
- $\tau = \{0.1294, 0.3232, 0.5532, 0.66, 0.8\}$,
- $\delta_k \in [0.5, 0.75]$.

Monte Carlo simulation

Calibration of the algorithms as given in Bertrand-Fhima-Guillin (2011)



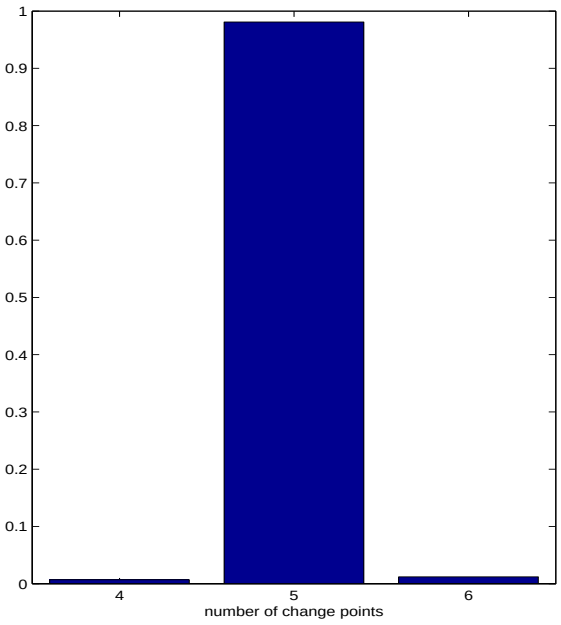
PLS method

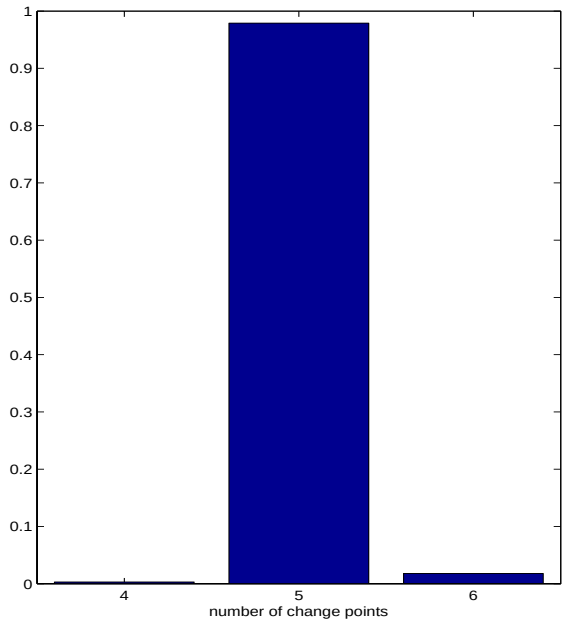
1. $K_{max} = 10$
2. p-value$= 10^{-5}$

FDp-V method

1. $C_1 = \delta_0/2$
2. $p_2^* = 10^{-5}$
3. $A = I_0/2$

Number of change points

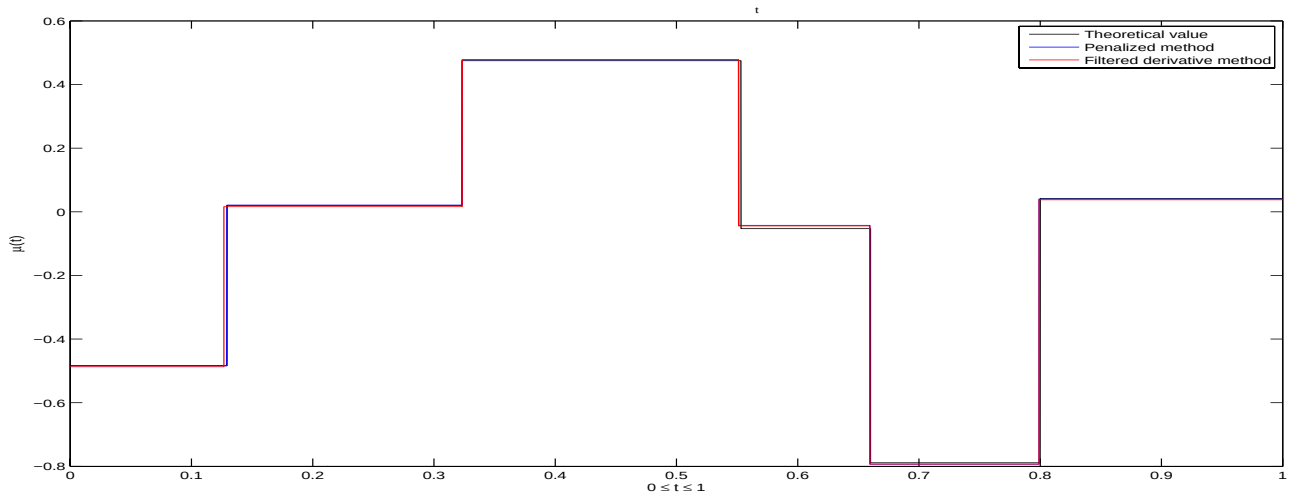




Results for M=1000 realizations:

- PLS (left) : $K = 5$ in 97.9% of all cases.
- FDp-V (right): $K = 5$ in 98.1% of all cases.

Estimation errors



	SECP	MISE
FDp-V	1.1840×10^{-4}	0.0107
PLS	1.2947×10^{-4}	0.0114

Where:

- Square Error on Change Points (SECP)= $\mathbb{E}\|\widehat{g} - g\|^2_{L^2(0,1)}$
- Mean Integrated Squared Error (MISE)= $\mathbb{E}\|\widehat{\tau} - \tau\|^2$

Complexity

	Memory allocation	CPU time
FDp-V method	0.04 MB	0.005 s
PLS method	200 MB	240 s

Conclusion (BFG, 2011)

FDp-V method:

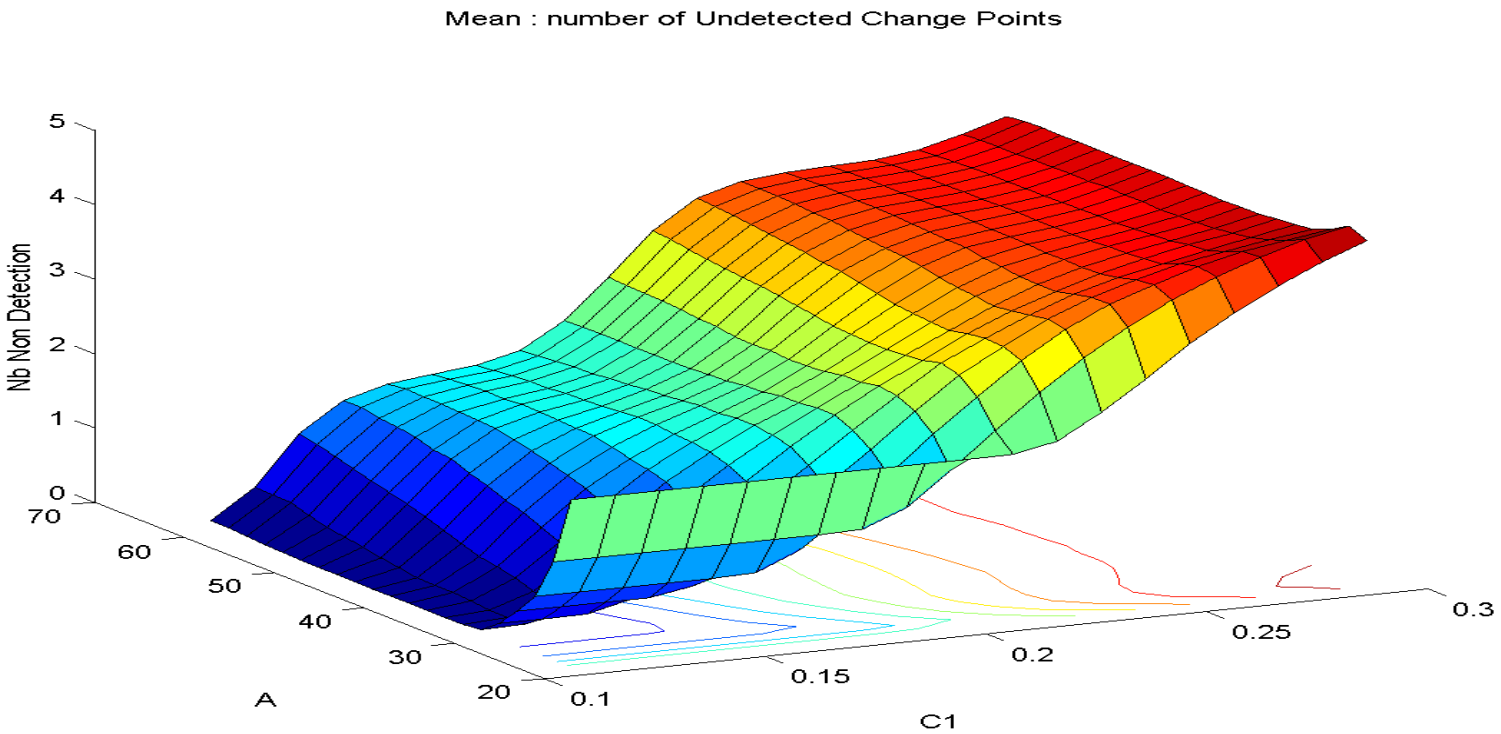
- Faster (time)
- Cheaper (memory)

Choice of the extra-parameters (2013)

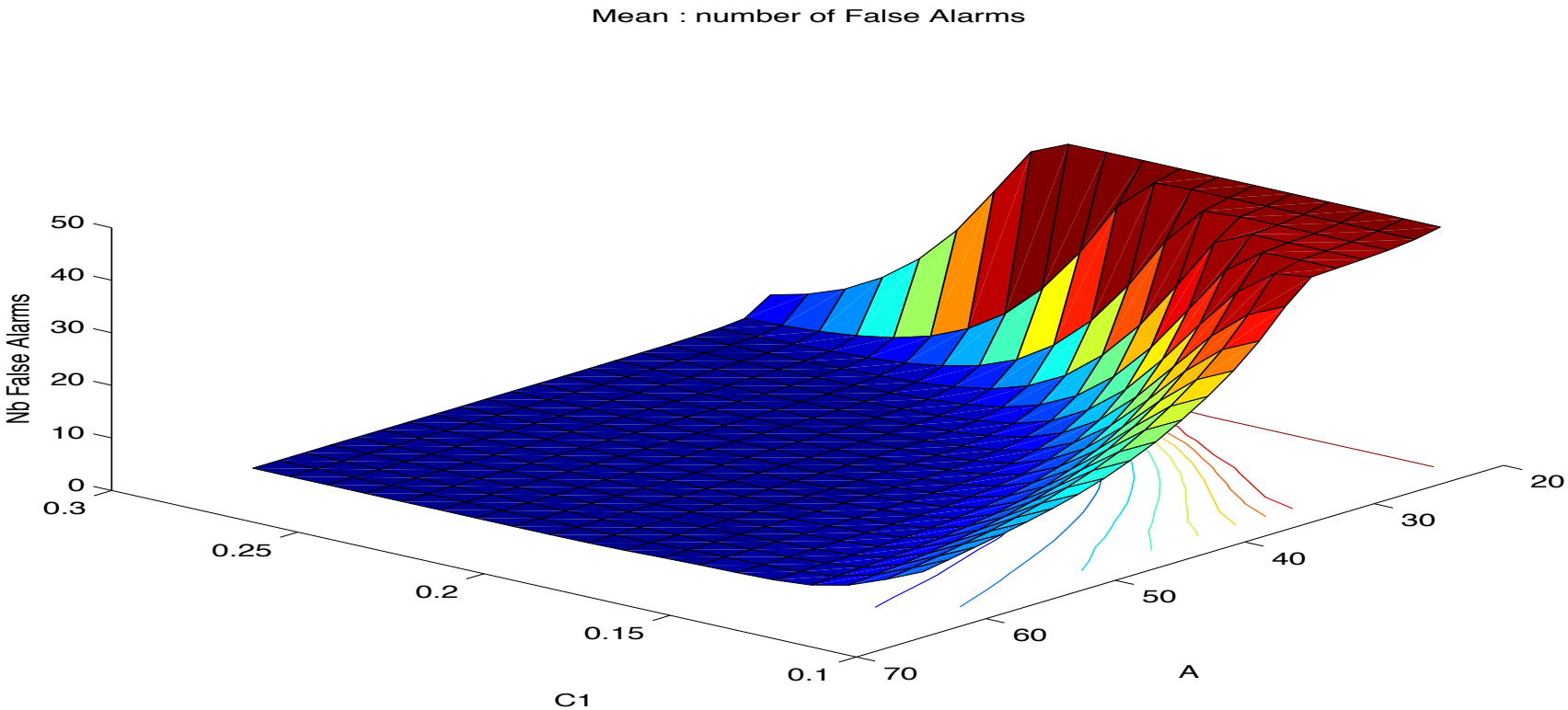
- Step 1: Non detection of right change point impact Integrate Square Error much more than false discovery of change.
- Step 2 will keep (as possible) only the right change points
- At Step 1, we want to have
  1. a small number of undetected change point
  2. and a small number of false discovery.

Choice of the extra-parameters (2013)

We have simulated 5,000 replications of a configuration with $K = 7$ change points.
We apply Step 1 (Filtered Derivative) with different value of the extra-parameters A and C_1.

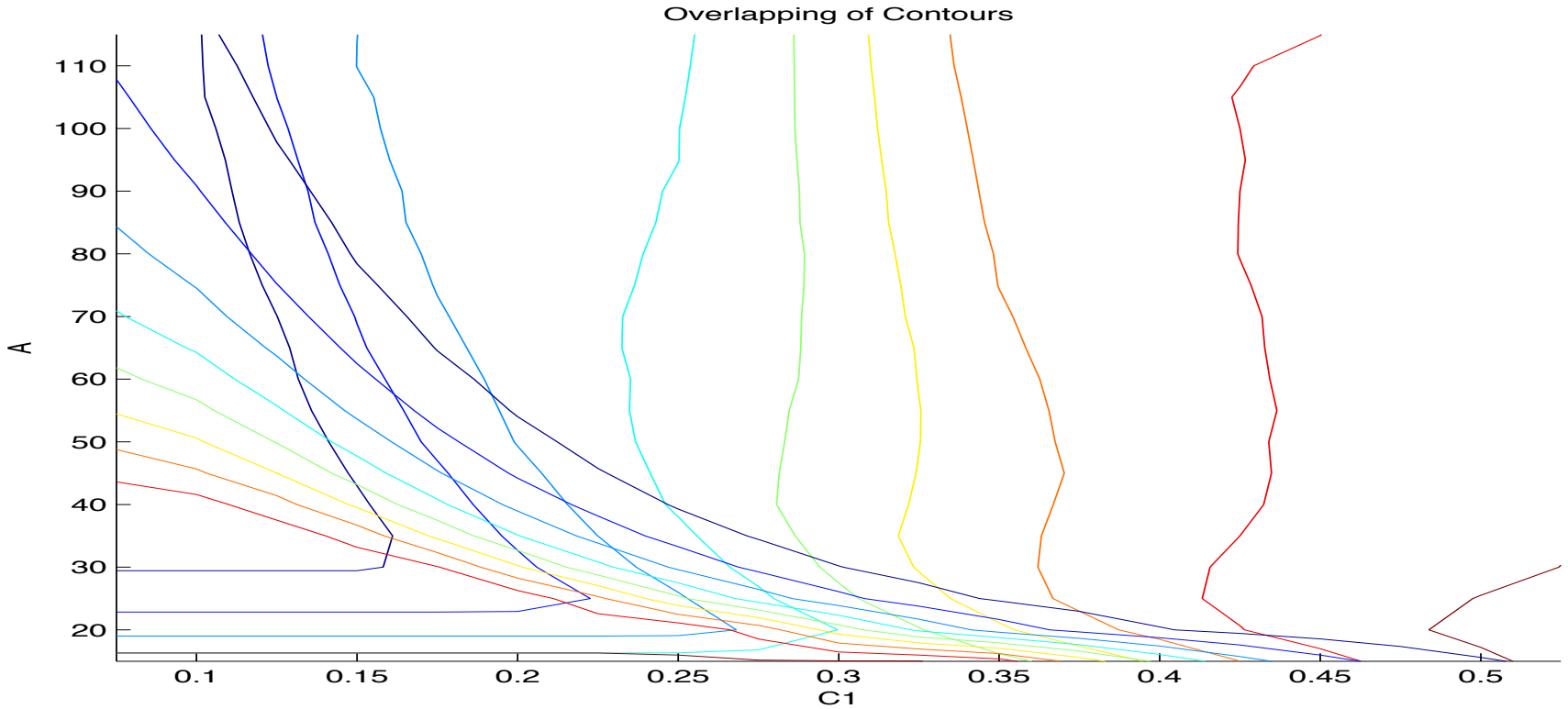


Choice of the extra-parameters (2013)

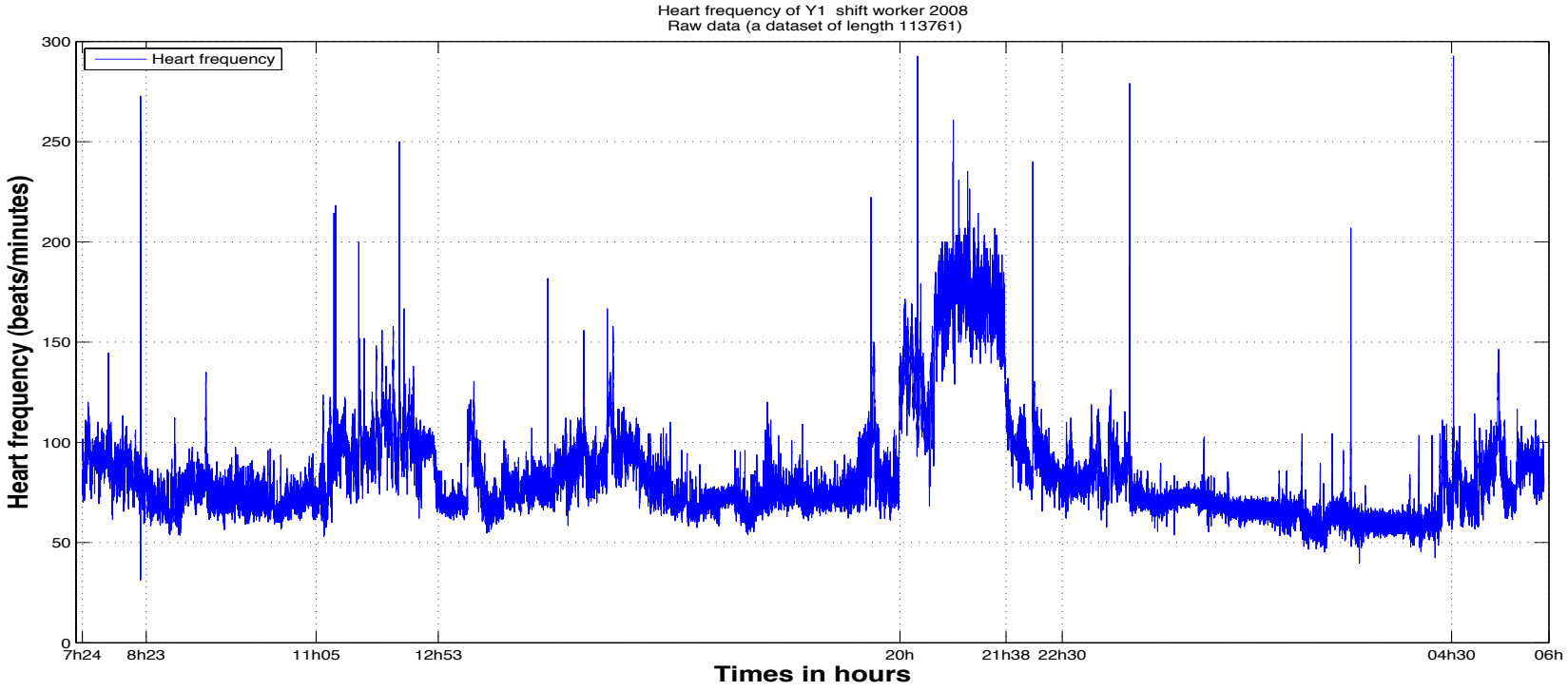


Choice of the extra-parameters (2013)

In this case, window size $A = 50 \rightarrow 200$ and thereshold $C_1 = 0.1 \rightarrow 0.15$ are good enough choices, see picture below.



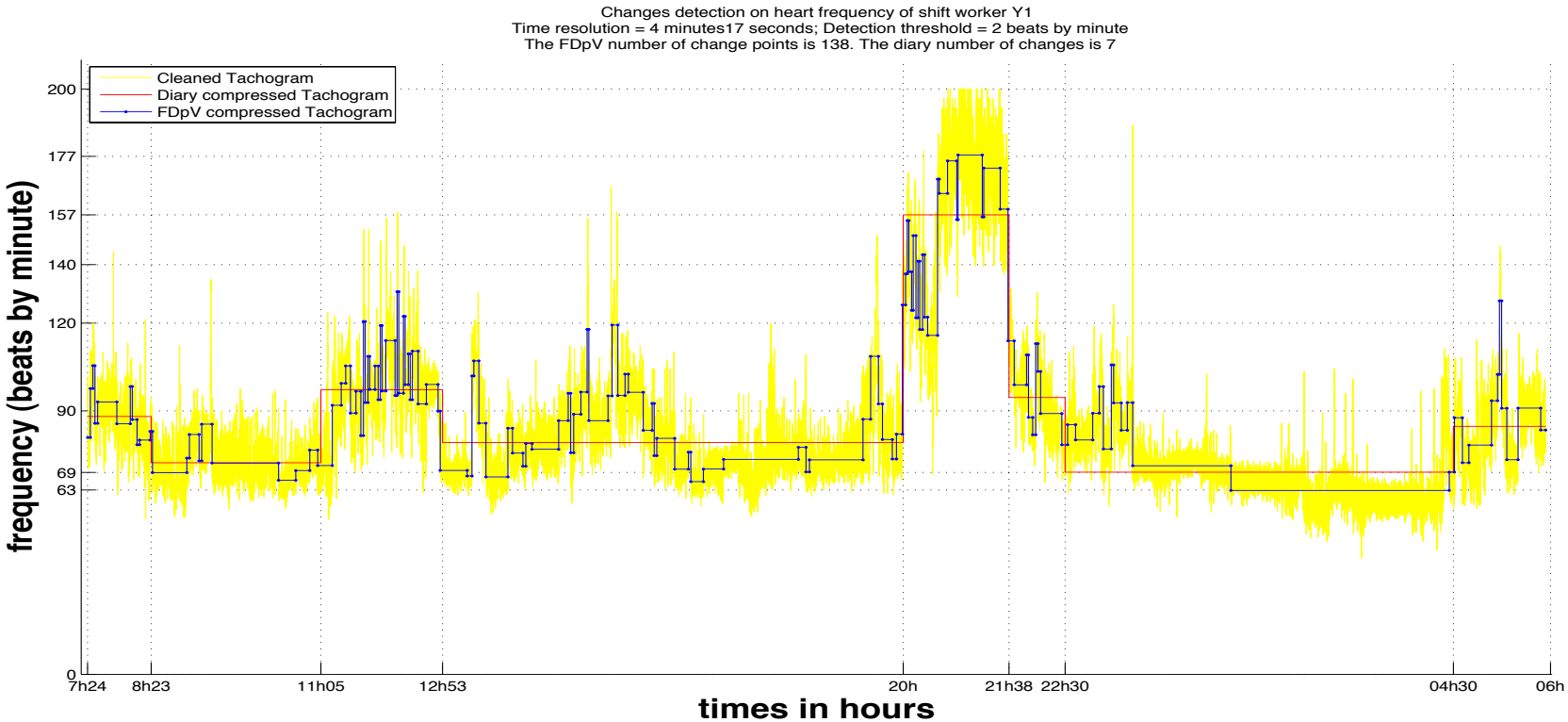
Ex. 1) Raw Tachogram of shift worker Y1.



Data provided by Professor Alain Chamoux (Clermont-Ferrand Hospital).

Fast & Light change detection

Processed by y using"Filtred Derivative with p-Value" (FDpV 2011) on the mean



Discussion

- The shift worker Y1 has manually reported changes of activity on a diary (the red line segmentation).

- This dataset is first preprocessed using "Tachogram cleaning" to avoid the maximum of artifacts.
- Then processed by "*In vivo* Tachogram Analysis (*In Vi*TA)", a demo software applying FDpV (blue line).

- It runs in 20 seconds
  Dataset of size $N = 110,000$, code written in Matlab, and 2.8 GHz processor.
- It runs in 500 mili-seconds, after translation in Java.

This is faster than manual segmentation.

Discussion

- Moreover, the FDpV segmentation is more accurate than the manual one:
- For instance, Y1 has reported *"soccer training"* from 20h to 21h38, which is supported by our analysis.
- But, with FDpV method, we can see more details:
  1. the warming-up,
  2. the time for coach's recommendations,
  3. and the soccer game with two small breaks.

Zoom on soccer game