

# Introduction à la détection de ruptures

Séminaire CEREGE

Charles Truong<sup>1</sup>  
(charles.doffy.net)

<sup>1</sup>Centre Borelli  
Université Paris-Saclay  
ENS Paris-Saclay, CNRS

Lundi 12 octobre



université  
PARIS-SACLAY

école  
normale  
supérieure  
paris-saclay

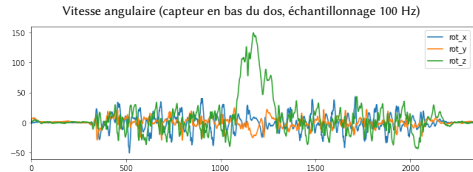
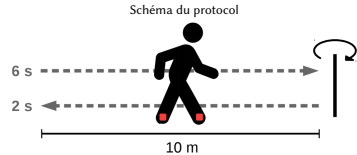


## Introduction

- ▶ La détection de ruptures est une tâche centrale pour manipuler des séries temporelles non-stationnaires.
- ▶ Exemple d'application : diagnostic automatique de patients neurologiques [Truong et al., 2019].

Des sujets sains et pathologiques suivent un protocole fixé :

- position debout,
- marche aller (10m),
- demi-tour,
- marche retour (10m),
- position debout.



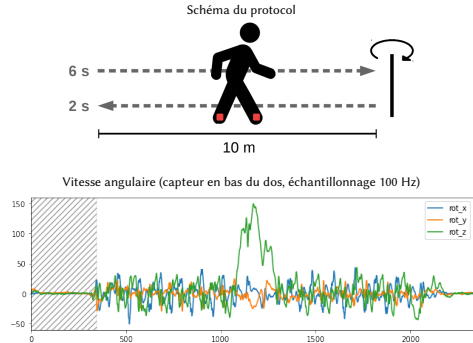
- ▶ Application en finance, contrôle qualité industriel, surveillance de la santé publique, etc. [Truong et al., 2020].

## Introduction

- ▶ La détection de ruptures est une tâche centrale pour manipuler des séries temporelles non-stationnaires.
- ▶ Exemple d'application : diagnostic automatique de patients neurologiques [Truong et al., 2019].

Des sujets sains et pathologiques suivent un protocole fixé :

- **position debout**,
- marche aller (10m),
- demi-tour,
- marche retour (10m),
- position debout.



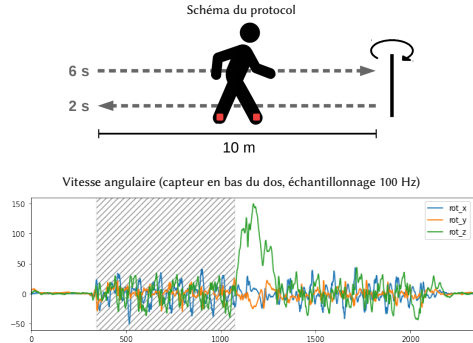
- ▶ Application en finance, contrôle qualité industriel, surveillance de la santé publique, etc. [Truong et al., 2020].

## Introduction

- ▶ La détection de ruptures est une tâche centrale pour manipuler des séries temporelles non-stationnaires.
- ▶ Exemple d'application : diagnostic automatique de patients neurologiques [Truong et al., 2019].

Des sujets sains et pathologiques suivent un protocole fixé :

- position debout,
- **marche aller (10m)**,
- demi-tour,
- marche retour (10m),
- position debout.



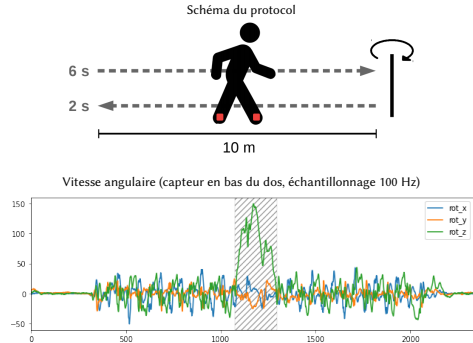
- ▶ Application en finance, contrôle qualité industriel, surveillance de la santé publique, etc. [Truong et al., 2020].

# Introduction

- ▶ La détection de ruptures est une tâche centrale pour manipuler des séries temporelles non-stationnaires.
- ▶ Exemple d'application : diagnostic automatique de patients neurologiques [Truong et al., 2019].

Des sujets sains et pathologiques suivent un protocole fixé :

- position debout,
- marche aller (10m),
- **demi-tour**,
- marche retour (10m),
- position debout.



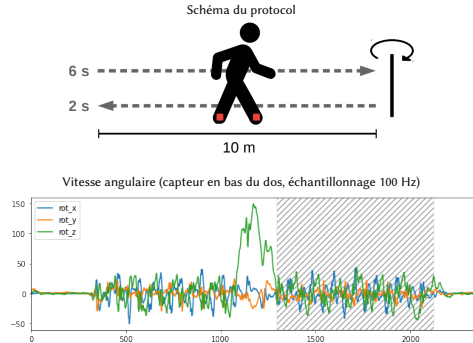
- ▶ Application en finance, contrôle qualité industriel, surveillance de la santé publique, etc. [Truong et al., 2020].

## Introduction

- ▶ La détection de ruptures est une tâche centrale pour manipuler des séries temporelles non-stationnaires.
- ▶ Exemple d'application : diagnostic automatique de patients neurologiques [Truong et al., 2019].

Des sujets sains et pathologiques suivent un protocole fixé :

- position debout,
- marche aller (10m),
- demi-tour,
- **marche retour (10m),**
- position debout.



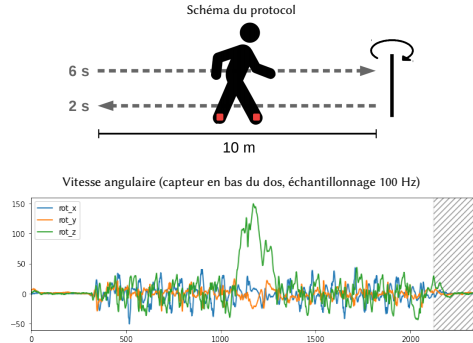
- ▶ Application en finance, contrôle qualité industriel, surveillance de la santé publique, etc. [Truong et al., 2020].

## Introduction

- ▶ La détection de ruptures est une tâche centrale pour manipuler des séries temporelles non-stationnaires.
- ▶ Exemple d'application : diagnostic automatique de patients neurologiques [Truong et al., 2019].

Des sujets sains et pathologiques suivent un protocole fixé :

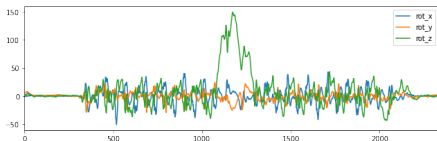
- position debout,
- marche aller (10m),
- demi-tour,
- marche retour (10m),
- **position debout.**



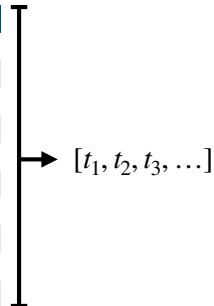
- ▶ Application en finance, contrôle qualité industriel, surveillance de la santé publique, etc. [Truong et al., 2020].

## Qu'est-ce que la détection de rupture ?

- ▶ La détection de ruptures consiste à trouver les limites temporelles entre deux périodes hétérogènes.
- ▶ Grossièrement : "signal multivarié  $\longrightarrow$  liste d'indices de ruptures"



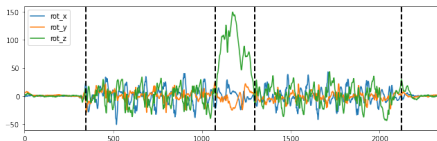
x	y	z
8.139	8.337	-21.055
6.964	9.881	-20.693
4.317	9.993	-19.309
1.752	8.950	-16.941
-0.305	7.356	-13.143
-2.320	7.384	-7.361
-3.312	8.467	-2.530
-3.697	10.891	2.523
-2.622	13.363	5.863
-2.728	11.761	4.473





## Qu'est-ce que la détection de rupture ?

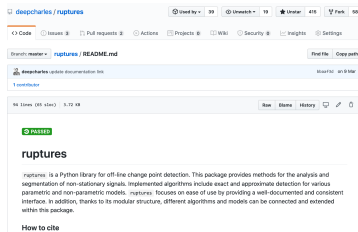
- ▶ La détection de ruptures consiste à trouver les limites temporelles entre deux périodes hétérogènes.
- ▶ Grossièrement : "signal multivarié  $\longrightarrow$  liste d'indices de ruptures"



x	y	z
8.139	8.337	-21.055
6.964	9.881	-20.693
4.317	9.993	-19.309
1.752	8.950	-16.941
-0.305	7.356	-13.143
-2.320	7.384	-7.361
-3.312	8.467	-2.530
-3.697	10.891	2.523
-2.622	13.363	5.863
-2.728	11.761	4.473

$\longrightarrow [t_1, t_2, t_3, \dots]$

# ruptures: détection en Python



deepcharles / ruptures

Used by 39 · Unwatched 19 · Watched 416 · Forked 58

Code · Issues · Pull requests · Actions · Projects · Wiki · Security · Insights · Settings

Branch: master · ruptures / README.md

Final file · Copy path

deepcharles update documentation link · 1 contributor

64 lines (45 sloc) · 3.72 KB

Run · Edit · History

## ruptures

ruptures is a Python library for off-line change point detection. This package provides methods for the analysis and segmentation of non-stationary signals. Implemented algorithms include exact and approximate detection for various parametric and non-parametric models. ruptures focuses on ease of use by providing a well-documented and consistent interface. In addition, thanks to its modular structure, different algorithms and models can be connected and extended within this package.

How to cite

Page Github ([github.com/deepcharles/ruptures](https://github.com/deepcharles/ruptures))



Signal Processing  
Volume 167, February 2020, 107299

ELSEVIER

SIGNAL  
PROCESSING

Review

## Selective review of offline change point detection methods

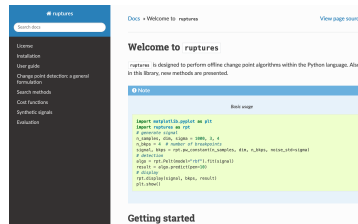
Charles Truong<sup>a</sup>, Laurent Oudre<sup>a,b</sup>, Nicolas Vayatis<sup>a</sup>

Show more

<https://doi.org/10.1016/j.sigpro.2019.107299>

Get rights and content

Publication associée [Truong et al., 2020]



ruptures

Search docs

User  
Installation  
User guide  
Change point detection, a general  
Termination  
Search methods  
Cost functions  
Synthetic signals  
Evaluation

Docs · Welcome to ruptures · View page source

## Welcome to ruptures

ruptures is designed to perform offline change point algorithms within the Python language. Also in this library, new methods are presented.

### How to use

Back stage

```
import matplotlib.pyplot as plt
import ruptures as rpt
# generate signal
n_samples, dim, signal = 1000, 3, 4
n_iter = 4 # number of bootstrap
signal, bpts = rpt.bootstrap(signal, dim, n_iter, min_size=10)
# detection
algo = rpt.HMMModel(rpt.LTICSignal)
results = algo.predict(signal)
# display
plt.plot(signal, bpts, red=True)
plt.show()
```

### Getting started

Documentation

```
(install-rpt) ~ - [pip install ruptures]
Processing ./library/caches/pip/whl/c/8d/9a/cc/e48a5d9936848a563287de7e905162549528ba5c9bba48/ruptur
es-1.0.3-py3-none-any.whl
Collecting numpy
Using cached numpy-1.18.5-cp37-cp37m-macosx_10_9_arm64.tgz (15.1 MB)
Collecting scipy
Using cached scipy-1.4.1-cp37-cp37m-macosx_10_9_arm64.tgz (28.4 MB)
Installing collected packages: numpy, scipy, ruptures
Successfully installed numpy-1.18.5 scipy-1.4.1 ruptures-1.0.3
WARNING: You are using pip version 20.0.2; however, version 20.1.1 is available.
You should consider upgrading via the '/Users/Sara/.virtualenvs/install-rpt/bin/python -m pip install --u
pgrade pip' command.
(install-rpt) ~ -
```

Comment installer

Liens et information sur la page Github.

# Table of contents

1. Introduction

2. Qu'est-ce que la détection de rupture ?

3. Principe général

4. Exemples

- Changement de moyenne et variance (1-D)

- Changement de moyenne et variance (n-D)

- Changement de distribution

- Changement de pente

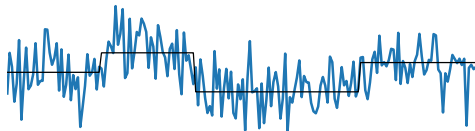
- Analyse de la marche

5. Extension au cas 2D

6. Conclusion

## Principe général

Quelle segmentation choisir ?



$$\mathcal{T} = \{t_1, t_2, t_3\}$$



$$V(\mathcal{T}) = c(y_{0..t_1}) + c(y_{t_1..t_2}) + c(y_{t_2..t_3}) + c(y_{t_3..T})$$

La "meilleure segmentation", notée  $\hat{\mathcal{T}}$ , minimise le critère  $V(\mathcal{T})$  :

$$V(\mathcal{T}) := \sum_{k=0}^K c(y_{t_k..t_{k+1}}).$$

Exemple de fonction (de coût)  $c(\cdot)$  :

$$c(y) = \sum_t (y_t - \bar{y})^2.$$

### Problème 1.

Nombre fixe  $K$  de ruptures :

$$\hat{\mathcal{T}} := \arg \min_{\mathcal{T}} V(\mathcal{T}) \quad \text{s.t. } |\mathcal{T}| = K.$$

### Problème 2.

Nombre inconnu de ruptures :

$$\hat{\mathcal{T}} := \arg \min_{\mathcal{T}} V(\mathcal{T}) + \text{pen}(\mathcal{T})$$

où  $\text{pen}(\mathcal{T})$  mesure la complexité de la segmentation  $\mathcal{T}$ .

## Principe général

Les méthodes de segmentation sont la combinaison de trois éléments [Truong et al., 2020].

Fonction de coût

(type de rupture)

Méthode de recherche

(rapidité d'exécution)

Contrainte

(nombre de rupture)

Criterion  $V(\mathcal{T})$  à minimiser : 
$$V(\mathcal{T}) := \sum_{k=0}^K c(y_{t_k \dots t_{k+1}}) .$$

### Problème 1.

Nombre  $K$  fixé de ruptures :

$$\hat{\mathcal{T}} := \arg \min_{\mathcal{T}} V(\mathcal{T}) \quad \text{s.t. } |\mathcal{T}| = K .$$

### Problème 2.

Nombre inconnu de rupture :

$$\hat{\mathcal{T}} := \arg \min_{\mathcal{T}} V(\mathcal{T}) + \text{pen}(\mathcal{T})$$

où  $\text{pen}(\mathcal{T})$  mesure la complexité de la segmentation  $\mathcal{T}$ .

# Table of contents

1. Introduction

2. Qu'est-ce que la détection de rupture ?

3. Principe général

4. Exemples

- Changement de moyenne et variance (1-D)

- Changement de moyenne et variance (n-D)

- Changement de distribution

- Changement de pente

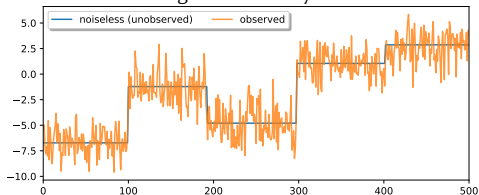
- Analyse de la marche

5. Extension au cas 2D

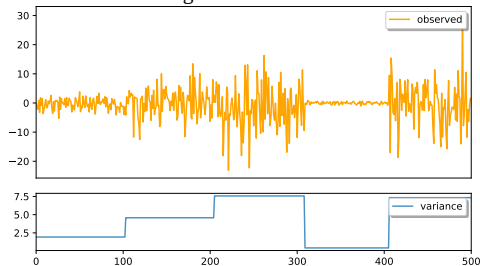
6. Conclusion

## Changement de moyenne et variance (1-D)

Changement de moyenne



Changement de variance



Fonction de coût :

$$c(y_{a..b}) = \sum_{t=a}^{b-1} (y_t - \bar{y}_{a..b})^2$$

où  $\bar{y}_{a..b}$  est la moyenne empirique du sous-segment  $y_{a..b}$ .

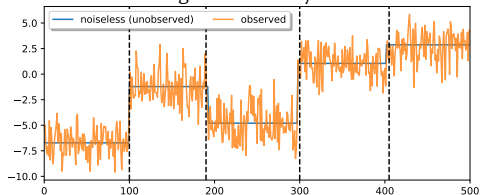
Fonction de coût :

$$c(y_{a..b}) = (b - a) \log(\hat{\sigma}_{a..b})$$

où  $\hat{\sigma}_{a..b}$  est l'écart-type empirique du sous-segment  $y_{a..b}$ .

## Changement de moyenne et variance (1-D)

Changement de moyenne



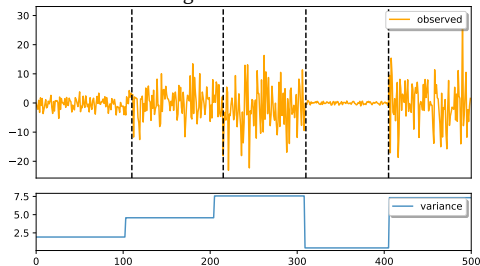
(Erreur: 3 échantillons.)

Fonction de coût :

$$c(y_{a..b}) = \sum_{t=a}^{b-1} (y_t - \bar{y}_{a..b})^2$$

où  $\bar{y}_{a..b}$  est la moyenne empirique du sous-segment  $y_{a..b}$ .

Changement de variance



(Erreur: 10 échantillons.)

Fonction de coût :

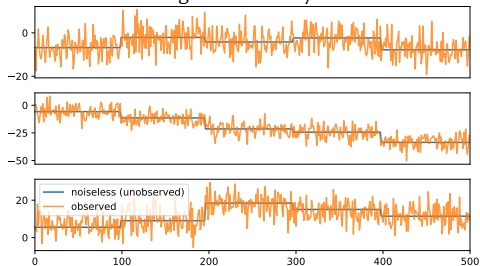
$$c(y_{a..b}) = (b - a) \log(\hat{\sigma}_{a..b})$$

où  $\hat{\sigma}_{a..b}$  est l'écart-type empirique du sous-segment  $y_{a..b}$ .

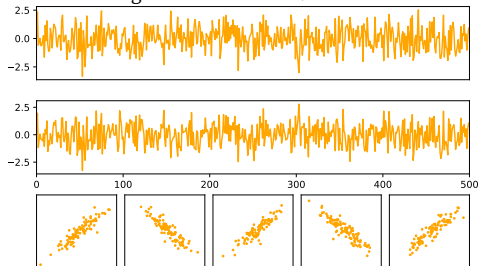


## Changement de moyenne et variance (n-D)

Changement de moyenne



Changement de variance/covariance



Fonction de coût :

$$c(y_{a..b}) = \sum_{t=a}^{b-1} \|y_t - \bar{y}_{a..b}\|^2$$

où  $\bar{y}_{a..b}$  est la moyenne empirique du sous-segment  $y_{a..b}$ .

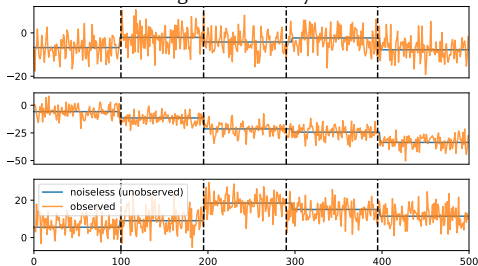
Fonction de coût :

$$c(y_{a..b}) = (b - a) \log \det \hat{\Sigma}_{a..b}$$

où  $\hat{\Sigma}_{a..b}$  est la matrice de covariance empirique du sous-segment  $y_{a..b}$ .

## Changement de moyenne et variance (n-D)

Changement de moyenne



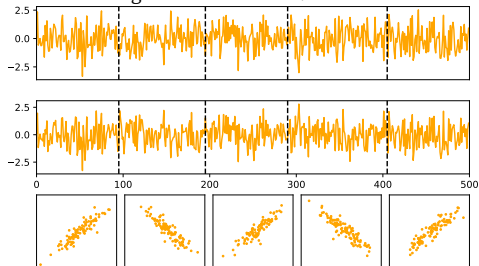
(Erreur: 7 échantillons.)

Fonction de coût :

$$c(y_{a..b}) = \sum_{t=a}^{b-1} \|y_t - \bar{y}_{a..b}\|^2$$

où  $\bar{y}_{a..b}$  est la moyenne empirique du sous-segment  $y_{a..b}$ .

Changement de variance/covariance



(Erreur: 3 échantillons.)

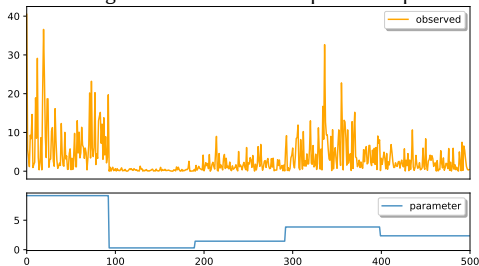
Fonction de coût :

$$c(y_{a..b}) = (b - a) \log \det \hat{\Sigma}_{a..b}$$

où  $\hat{\Sigma}_{a..b}$  est la matrice de covariance empirique du sous-segment  $y_{a..b}$ .

## Changement de distribution

Changement de distribution paramétrique



Fonction de coût :

$$c(y_{a..b}) = -\max_{\theta} \log f_{\theta}(y_{a..b})$$

où  $f_{\theta}$  est la densité de la distribution choisie, paramétrée par  $\theta$ .

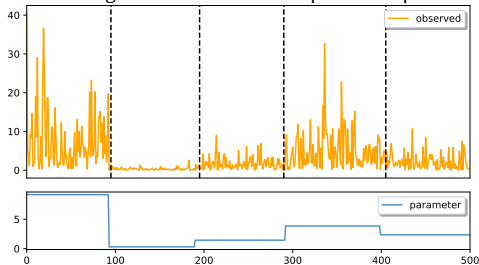
Changement de distribution non-paramétrique

Quand la distribution sous-jacente n'est pas connue:

- ▶ [Matteson and James, 2014, A nonparametric approach for multiple change point analysis of multivariate data. *Journal of the American Statistical Association*, 109(505), 334–345.]
- ▶ [Arlot et al., 2019, A kernel multiple change-point algorithm via model selection. *Journal of Machine Learning Research*, 20(162), 1–56.]
- ▶ [Ross and Adams, 2012, Two nonparametric control charts for detecting arbitrary distribution changes. *Journal of Quality Technology*, 44(2), 102–117.]

## Changement de distribution

### Changement de distribution paramétrique



(Distribution exponentielle, erreur: 6 échantillons.)

Fonction de coût :

$$c(y_{a..b}) = -\max_{\theta} \log f_{\theta}(y_{a..b})$$

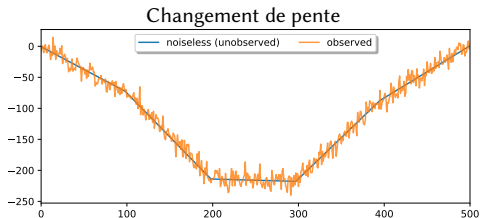
où  $f_{\theta}$  est la densité de la distribution choisie, paramétrée par  $\theta$ .

### Changement de distribution non-paramétrique

Quand la distribution sous-jacente n'est pas connue:

- ▶ [Matteson and James, 2014, A nonparametric approach for multiple change point analysis of multivariate data. *Journal of the American Statistical Association*, 109(505), 334–345.]
- ▶ [Arlot et al., 2019, A kernel multiple change-point algorithm via model selection. *Journal of Machine Learning Research*, 20(162), 1–56.]
- ▶ [Ross and Adams, 2012, Two nonparametric control charts for detecting arbitrary distribution changes. *Journal of Quality Technology*, 44(2), 102–117.]

## Changement de pente



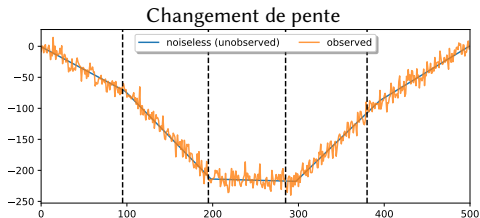
```
1 import numpy as np
2 import ruptures as rpt
3
4 # Given an observed signal `signal`.
5 n_samples = len(signal)
6 signal_with_regressors = np.c_[signal, np.arange(n_samples), np.ones(n_samples)]
7 my_bkps = rpt.Dynp(model="linear").fit(signal_with_regressors).predict(n_bkps)
```

Code associé, en utilisant ruptures.

Fonction de coût :

$$c(y_{a..b}) = \min_{a,b} (y_t - at - b)^2.$$

## Changement de pente



(Erreur: 16 échantillons.)

```
1 import numpy as np
2 import ruptures as rpt
3
4 # Given an observed signal `signal`.
5 n_samples = len(signal)
6 signal_with_regressors = np.c_[signal, np.arange(n_samples), np.ones(n_samples)]
7 my_bkps = rpt.Dynp(model="linear").fit(signal_with_regressors).predict(n_bkps)
```

Code associé, en utilisant ruptures.

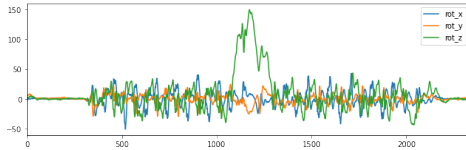
Fonction de coût :

$$c(y_{a..b}) = \min_{a,b} (y_t - at - b)^2.$$

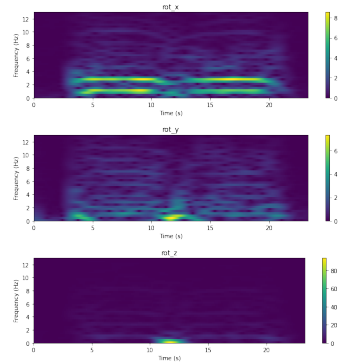
## Analyse de la marche

- Pour simplifier la détection, le signal est transformé (ici, transformée de Fourier court-terme)
- Puis, détection de rupture de moyenne.

Vitesse angulaire (capteur en bas du dos, échantillonnage 100 Hz)



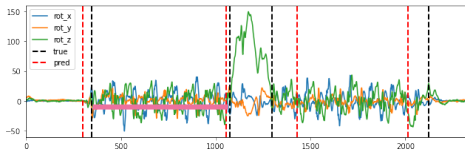
Signal transformé



## Analyse de la marche

- Pour simplifier la détection, le signal est transformé (ici, transformée de Fourier court-terme)
- Puis, détection de rupture de moyenne.

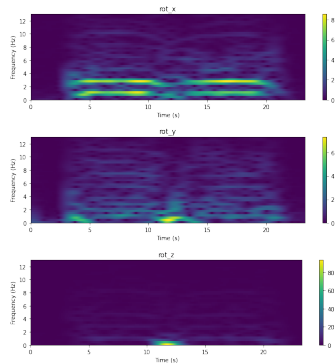
Vitesse angulaire (capteur en bas du dos, échantillonnage 100 Hz)



```
[246]: 1 # cost for multivariate mean-shifts
2 cost = rpt.costs.CostL2()
3 # search method: binary segmentation
4 algo = rpt.Binseg(custom_cost=cost)
5 # fit
6 algo.fit(X)
7 # predict
8 prediction = algo.predict(4)
9 # error
10 error = hausdorff(true_bkps, prediction)/100
11 # print results
12 print(f"True change points:\t\t{true_bkps}")
13 print(f"Predicted change points:\t{prediction}")
14 print(f"Max error: {error:.2f} sec")
```

True change points: [347, 1075, 1297, 2123, 2332]  
Predicted change points: [300, 1055, 1430, 2015, 2332]  
Max error: 1.33 sec

Signal transformé





# Table of contents

1. Introduction

2. Qu'est-ce que la détection de rupture ?

3. Principe général

4. Exemples

- Changement de moyenne et variance (1-D)

- Changement de moyenne et variance (n-D)

- Changement de distribution

- Changement de pente

- Analyse de la marche

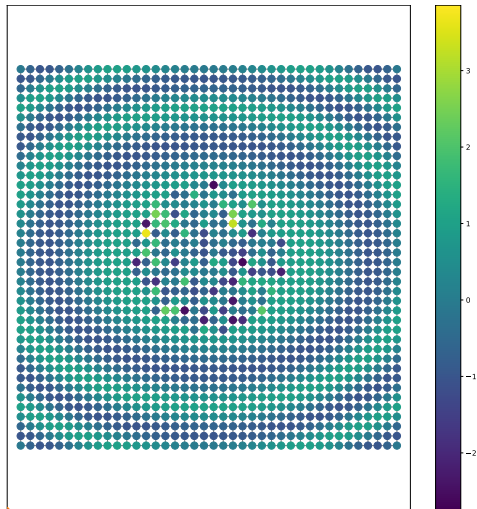
5. Extension au cas 2D

6. Conclusion

## Extension au cas 2D

### Changement de régularité

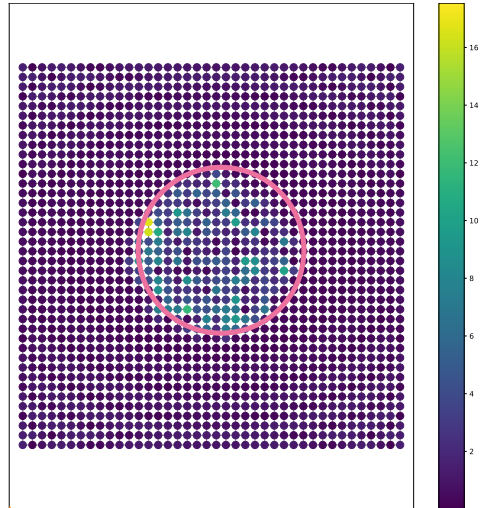
- ▶ On mesure la régularité locale.
- ▶ Le maillage n'a pas besoin d'être régulier.
- ▶ Bien d'autres descripteurs que la régularité sont disponibles.



## Extension au cas 2D

### Changement de régularité

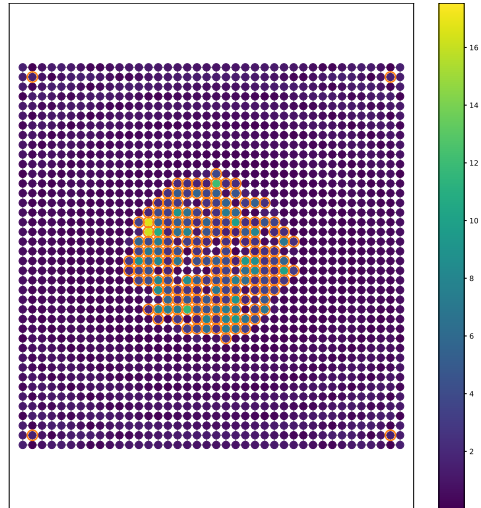
- ▶ On mesure la régularité locale.
- ▶ Le maillage n'a pas besoin d'être régulier.
- ▶ Bien d'autres descripteurs que la régularité sont disponibles.



## Extension au cas 2D

### Changement de régularité

- ▶ On mesure la régularité locale.
- ▶ Le maillage n'a pas besoin d'être régulier.
- ▶ Bien d'autres descripteurs que la régularité sont disponibles.



## Conclusion

- ▶ Etape de traitement automatique utile pour gérer des signaux longs ou nombreux.
- ▶ L'information nécessaire doit être contenue dans le signal.
- ▶ De nouvelles méthodes sont fréquemment intégrées à ruptures.
- ▶ Ouvert aux collaborations !

## References



Arlot, S., Celisse, A., and Harchaoui, Z. (2019).  
A kernel multiple change-point algorithm via model selection.  
*Journal of Machine Learning Research*, 20(162):1–56.



Matteson, D. S. and James, N. A. (2014).  
A nonparametric approach for multiple change point analysis of multivariate data.  
*Journal of the American Statistical Association*, 109(505):334–345.



Ross, G. J. and Adams, N. M. (2012).  
Two nonparametric control charts for detecting arbitrary distribution changes.  
*Journal of Quality Technology*, 44(2):102–117.



Truong, C., Barrois-Müller, R., Moreau, T., Provost, C., Vienne-Jumeau, A., Moreau, A., Vidal, P.-P., Vayatis, N., Buffat, S., Yelnik, A., Ricard, D., and Oudre, L. (2019).  
A data set for the study of human locomotion with inertial measurements units.  
*Image Processing On Line*, 9.



Truong, C., Oudre, L., and Vayatis, N. (2020).  
Selective review of offline change point detection methods.  
*Signal Processing*, 167.