

# Structural Break Detection in Time Series

Richard A. Davis

Thomas Lee

Gabriel Rodriguez-Yam

Colorado State University

(<http://www.stat.colostate.edu/~rdavis/lectures>)

This research supported in part by an IBM faculty award.

- Introduction
  - Background
  - Setup for AR models
- Model Selection Using Minimum Description Length (MDL)
  - General principles
  - Application to AR models with breaks
- Optimization using Genetic Algorithms
  - Basics
  - Implementation for structural break estimation
- Simulation examples
  - Piecewise autoregressions
  - Slowly varying autoregressions
- Applications
  - Multivariate time series
  - EEG example
- Application to nonlinear models (parameter-driven state-space models)
  - Poisson model
  - Stochastic volatility model

# Introduction

## Structural breaks:

### Kitagawa and Akaike (1978)

- fitting locally stationary autoregressive models using AIC
- computations facilitated by the use of the Householder transformation

### Davis, Huang, and Yao (1995)

- likelihood ratio test for testing a change in the parameters and/or order of an AR process.

### Kitagawa, Takanami, and Matsumoto (2001)

- signal extraction in seismology-estimate the arrival time of a seismic signal.

### Ombao, Raz, von Sachs, and Malow (2001)

- orthogonal complex-valued transforms that are localized in time and frequency- smooth localized complex exponential (SLEX) transform.
- applications to EEG time series and speech data.

## Introduction (cont)

### Locally stationary:

Dahlhaus (1997, 2000,...)

- locally stationary processes
- estimation

Adak (1998)

- piecewise stationary
- applications to seismology and biomedical signal processing

### MDL and coding theory:

Lee (2001, 2002)

- estimation of discontinuous regression functions

Hansen and Yu (2001)

- model selection

## Introduction (cont)

**Time Series:**  $y_1, \dots, y_n$

**Piecewise AR model:**

$$Y_t = \gamma_j + \phi_{j1}Y_{t-1} + \dots + \phi_{jp_j}Y_{t-p_j} + \sigma_j\varepsilon_t, \quad \text{if } \tau_{j-1} \leq t < \tau_j,$$

where  $\tau_0 = 1 < \tau_1 < \dots < \tau_{m-1} < \tau_m = n + 1$ , and  $\{\varepsilon_t\}$  is IID(0,1).

**Goal:** Estimate

$m$  = number of segments

$\tau_j$  = location of  $j^{\text{th}}$  break point

$\gamma_j$  = level in  $j^{\text{th}}$  epoch

$p_j$  = order of AR process in  $j^{\text{th}}$  epoch

$(\phi_{j1}, \dots, \phi_{jp_j})$  = AR coefficients in  $j^{\text{th}}$  epoch

$\sigma_j$  = scale in  $j^{\text{th}}$  epoch

## Introduction (cont)

Motivation for using piecewise AR models:

Piecewise AR is a special case of a *piecewise stationary process* (see Adak 1998),

$$\tilde{Y}_{t,n} = \sum_{j=1}^m Y_t^j I_{[\tau_{j-1}, \tau_j)}(t/n),$$

where  $\{Y_t^j\}, j = 1, \dots, m$  is a sequence of stationary processes. It is argued in Ombao et al. (2001), that if  $\{Y_{t,n}\}$  is a locally stationary process (in the sense of Dahlhaus), then there exists a piecewise stationary process  $\{\tilde{Y}_{t,n}\}$  with

$$m_n \rightarrow \infty \quad \text{with} \quad m_n/n \rightarrow 0, \quad \text{as} \quad n \rightarrow \infty,$$

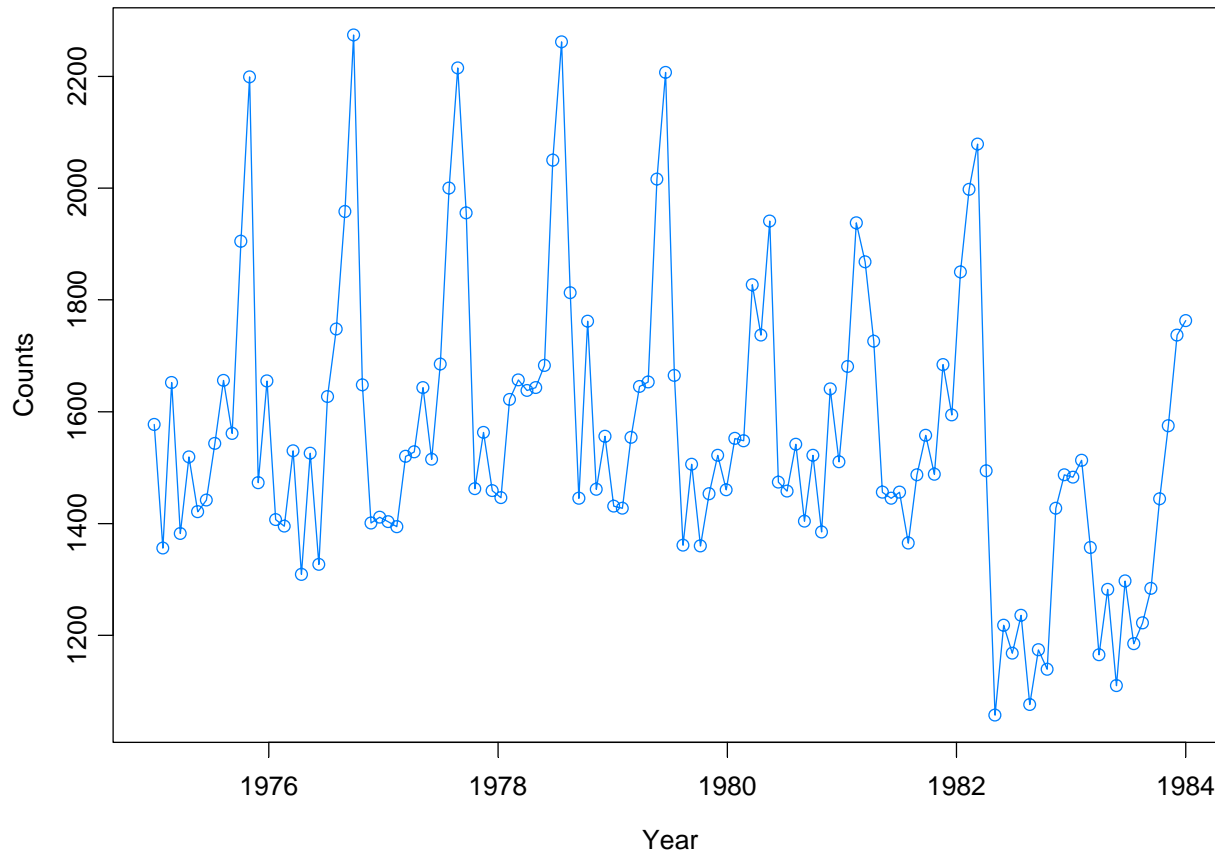
that approximates  $\{Y_{t,n}\}$  (in average mean square).

**Roughly speaking:**  $\{Y_{t,n}\}$  is a locally stationary process if it has a time-varying spectrum that is approximately  $|A(t/n, \omega)|^2$ , where  $A(u, \omega)$  is a continuous function in  $u$ .

## Example--Monthly Deaths & Serious Injuries, UK

**Data:**  $y_t$  = number of monthly deaths and serious injuries in UK, Jan '75 – Dec '84, ( $t = 1, \dots, 120$ )

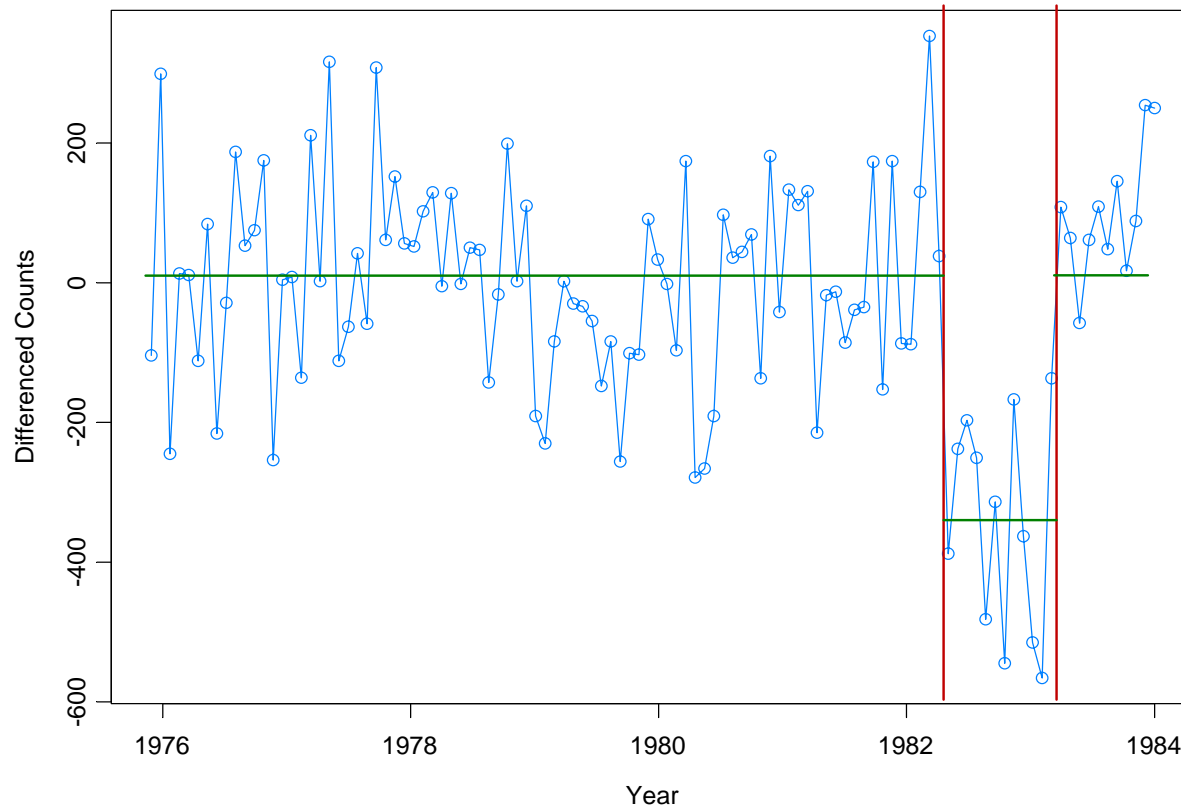
**Remark:** Seat belt legislation introduced in Feb '83 ( $t = 99$ ).



## Example -- Monthly Deaths & Serious Injuries, UK (cont)

**Data:**  $x_t$  = number of monthly deaths and serious injuries in UK, differenced at lag 12; Jan `75 – Dec `84, ( $t = 13, \dots, 120$ )

**Remark:** Seat belt legislation introduced in Feb `83 ( $t = 99$ ).



### Traditional regression analysis:

$$Y_t = a + bf(t) + W_t,$$

$$f(t) = \begin{cases} 0, & \text{if } 1 \leq t \leq 98, \\ 1, & \text{if } 98 < t \leq 120. \end{cases}$$

$$\begin{aligned} X_t &= Y_t - Y_{t-12} \\ &= bg(t) + N_t \end{aligned}$$

$$g(t) = \begin{cases} 1, & \text{if } 99 \leq t \leq 110, \\ 0, & \text{otherwise.} \end{cases}$$

**Model:**  $b = -373.4$ ,  $\{N_t\} \sim \text{AR}(13)$ .



# Model Selection Using Minimum Description Length

## Basics of MDL:

Choose the model which maximizes the compression of the data or, equivalently, select the model that minimizes the code length of the data (i.e., amount of memory required to encode the data).

$M$  = class of operating models for  $y = (y_1, \dots, y_n)$

$L_F(y)$  = code length of  $y$  relative to  $F \in M$

Typically, this term can be decomposed into two pieces (**two-part code**),

$$L_F(y) = L(\hat{F}/y) + L(\hat{e} | \hat{F}),$$

where

$L(\hat{F}/y)$  = code length of the fitted model for  $F$

$L(\hat{e} | \hat{F})$  = code length of the residuals based on the fitted model

## Model Selection Using Minimum Description Length (cont)

Applied to the segmented AR model:

$$Y_t = \gamma_j + \phi_{j1}Y_{t-1} + \cdots + \phi_{jp_j}Y_{t-p_j} + \sigma_j\varepsilon_t, \quad \text{if } \tau_{j-1} \leq t < \tau_j,$$

**First term**  $L(\hat{\mathbf{F}}/y)$  : Let  $n_j = \tau_j - \tau_{j-1}$  and  $\psi_j = (\gamma_j, \phi_{j1}, \dots, \phi_{jp_j}, \sigma_j)$  denote the length of the  $j^{\text{th}}$  segment and the parameter vector of the  $j^{\text{th}}$  AR process, respectively. Then

$$\begin{aligned} L(\hat{\mathbf{F}}/y) &= L(m) + L(\tau_1, \dots, \tau_m) + L(p_1, \dots, p_m) + L(\hat{\psi}_1 | y) + \cdots + L(\hat{\psi}_m | y) \\ &= L(m) + L(n_1, \dots, n_m) + L(p_1, \dots, p_m) + L(\hat{\psi}_1 | y) + \cdots + L(\hat{\psi}_m | y) \end{aligned}$$

**Encoding:**

**integer**  $I$  :  $\log_2 I$  bits (if  $I$  unbounded)

$\log_2 I_U$  bits (if  $I$  bounded by  $I_U$ )

**MLE**  $\hat{\theta}$  :  $\frac{1}{2} \log_2 N$  bits (where  $N$  = number of observations used to compute  $\hat{\theta}$  ;  
Rissanen (1989))

## Model Selection Using Minimum Description Length (cont)

So,

$$L(\hat{\mathbf{F}}/y) = \log_2 m + m \log_2 n + \sum_{j=1}^m \log_2 p_j + \sum_{j=1}^m \frac{p_j + 2}{2} \log_2 n_j$$

**Second term**  $L(\hat{e} | \hat{\mathbf{F}})$ : Using Shannon's classical results on information theory, Rissanen demonstrates that the code length of  $\hat{e}$  can be approximated by the **negative of the log-likelihood** of the fitted model, i.e., by

$$L(\hat{e} | \hat{\mathbf{F}}) \approx \sum_{j=1}^m \frac{n_j}{2} (\log_2 (2\pi\hat{\sigma}_j^2) + 1)$$

For fixed values of  $m, (\tau_1, p_1), \dots, (\tau_m, p_m)$ , we define the MDL as

$$MDL(m, (\tau_1, p_1), \dots, (\tau_m, p_m))$$

$$= \log_2 m + m \log_2 n + \sum_{j=1}^m \log_2 p_j + \sum_{j=1}^m \frac{p_j + 2}{2} \log_2 n_j + \sum_{j=1}^m \frac{n_j}{2} \log_2 (2\pi\hat{\sigma}_j^2) + \frac{n}{2}$$

The strategy is to find the best segmentation that minimizes  $MDL(m, \tau_1, p_1, \dots, \tau_m, p_m)$ .

To speed things up, we use **Y-W estimates** of AR parameters.

# Optimization Using Genetic Algorithms

## Basics of GA:

Class of optimization algorithms that mimic natural evolution.

- Start with an initial set of *chromosomes*, or population, of possible solutions to the optimization problem.
- Parent chromosomes are randomly selected (proportional to the rank of their objective function values), and produce offspring using *crossover* or *mutation* operations.
- After a sufficient number of offspring are produced to form a second generation, the process then *restarts to produce a third generation*.
- Based on Darwin's *theory of natural selection*, the process should produce future generations that give a *smaller (or larger)* objective function.

## Application to Structural Breaks—(cont)

**Genetic Algorithm:** Chromosome consists of  $n$  genes, each taking the value of  $-1$  (no break) or  $p$  (order of AR process). Use natural selection to find a *near* optimal solution.

Map the break points with a chromosome  $c$  via

$$(m, (\tau_1, p_1), \dots, (\tau_m, p_m)) \longleftrightarrow c = (\delta_1, \dots, \delta_n),$$

where

$$\delta_t = \begin{cases} -1, & \text{if no break point at } t, \\ p_j, & \text{if break point at time } t = \tau_{j-1} \text{ and AR order is } p_j. \end{cases}$$

For example,

$$c = (2, -1, -1, -1, -1, 0, -1, -1, -1, -1, 0, -1, -1, -1, 3, -1, -1, -1, -1, -1)$$

t: 1		6		11		15
------	--	---	--	----	--	----

would correspond to a process as follows:

$$\text{AR}(2), t=1:5; \text{AR}(0), t=6:10; \text{AR}(0), t=11:14; \text{AR}(3), t=15:20$$

## Implementation of Genetic Algorithm—(cont)

**Generation 0:** Start with  $L$  (200) randomly generated chromosomes,  $c_1, \dots, c_L$  with associated MDL values,  $MDL(c_1), \dots, MDL(c_L)$ .

**Generation 1:** A new child in the next generation is formed from the chromosomes  $c_1, \dots, c_L$  of the previous generation as follows:

- with probability  $\pi_c$ , *crossover* occurs.
  - two parent chromosomes  $c_i$  and  $c_j$  are selected at random with probabilities proportional to the ranks of  $MDL(c_i)$ .
  - $k^{th}$  gene of child is  $\delta_k = \delta_{i,k}$  w.p.  $1/2$  and  $\delta_{j,k}$  w.p.  $1/2$
- with probability  $1 - \pi_c$ , *mutation* occurs.
  - a parent chromosome  $c_i$  is selected
  - $k^{th}$  gene of child is  $\delta_k = \delta_{i,k}$  w.p.  $\pi_1$  ;  $-1$  w.p.  $\pi_2$  ; and  $p$  w.p.  $1 - \pi_1 - \pi_2$ .

## Implementation of Genetic Algorithm—(cont)

Execution of GA: Run GA until *convergence* or until a *maximum number of generations* has been reached. .

Various Strategies:

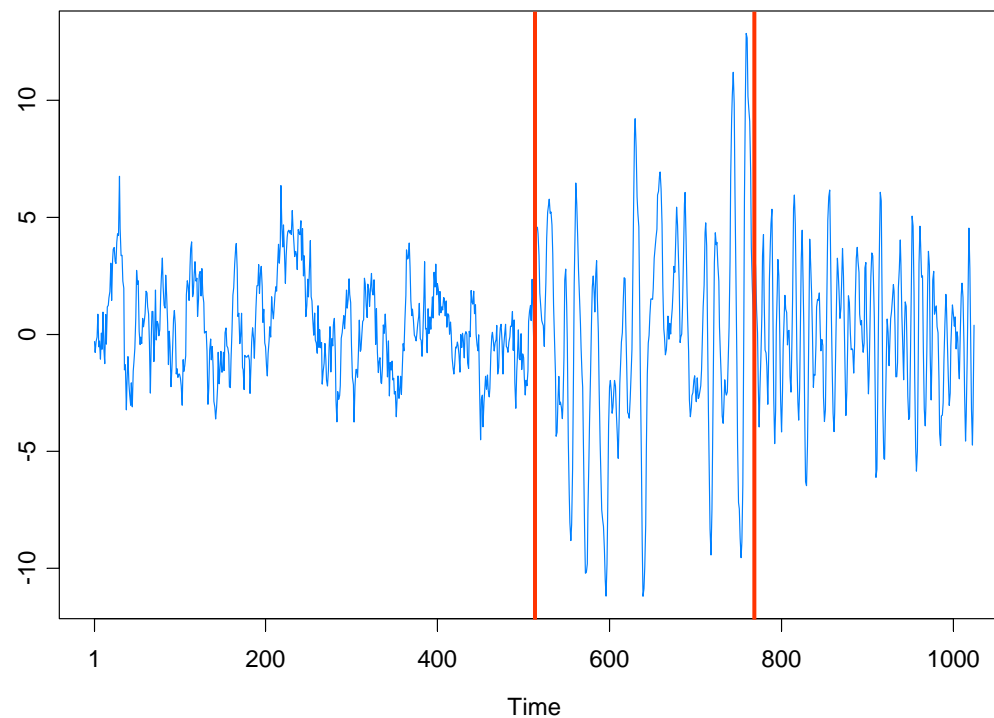
- include the *top ten* chromosomes from last generation in next generation.
- use multiple *islands*, in which populations run independently, and then allow *migration* after a fixed number of generations. This implementation is amenable to *parallel computing*.

## Simulation Examples-based on Ombao et al. (2001) test cases

1. **Piecewise stationary:** Consider a time series following the model,

$$Y_t = \begin{cases} .9Y_{t-1} + \varepsilon_t, & \text{if } 1 \leq t < 513, \\ 1.69Y_{t-1} - .81Y_{t-2} + \varepsilon_t, & \text{if } 513 \leq t < 769, \\ 1.32Y_{t-1} - .81Y_{t-2} + \varepsilon_t, & \text{if } 769 \leq t \leq 1024, \end{cases}$$

where  $\{\varepsilon_t\} \sim \text{IID } N(0,1)$ .



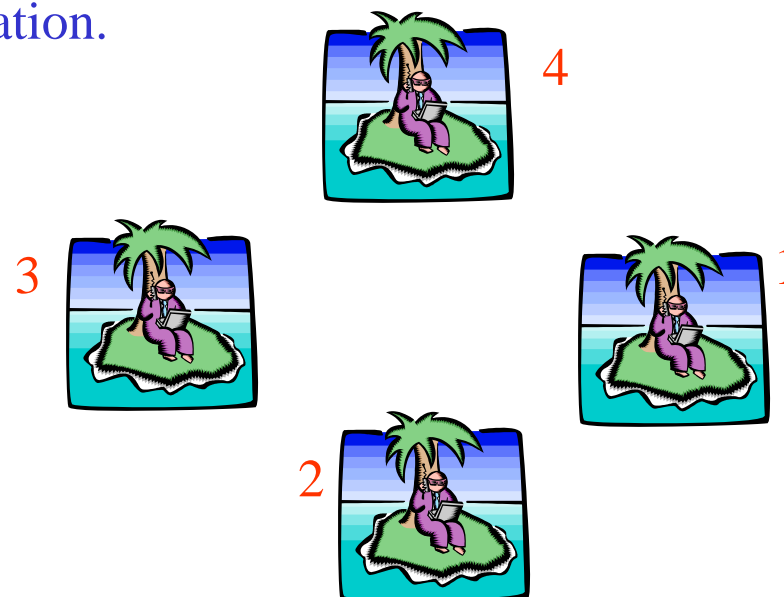


## 1. Piecewise stat (cont)

**Implementation:** Start with  $NI = 50$  islands, each with population size  $L = 200$ .

After every  $Mi = 5$  generations, allow migration.

Replace worst 2 in Island 2  
with best 2 from Island 4.



**Stopping rule:** Stop when the max MDL does not change for 10 consecutive migrations or after 100 migrations.

**Span configuration for model selection:** Max AR order  $K = 10$ ,

$p$	0	1	2	3	4	5	6-10
$m_p$	25	25	30	35	40	45	50
$\pi_p$	.08	.1	.1	.09	.09	.09	.09

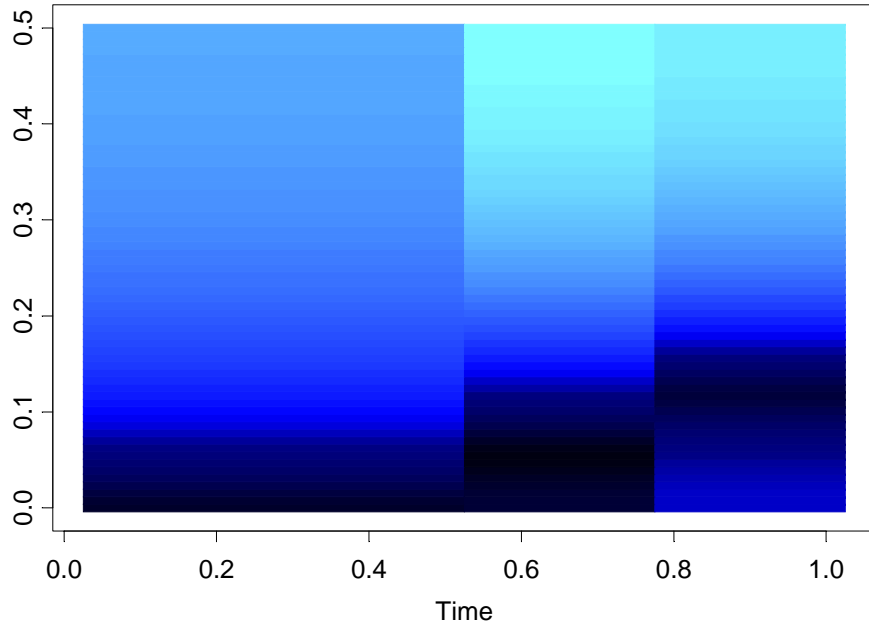
## 1. Piecewise stat (cont)

GA results: 3 pieces with breaks at  $\tau_1=513$  and  $\tau_2=769$ . Total run time 16.31 secs

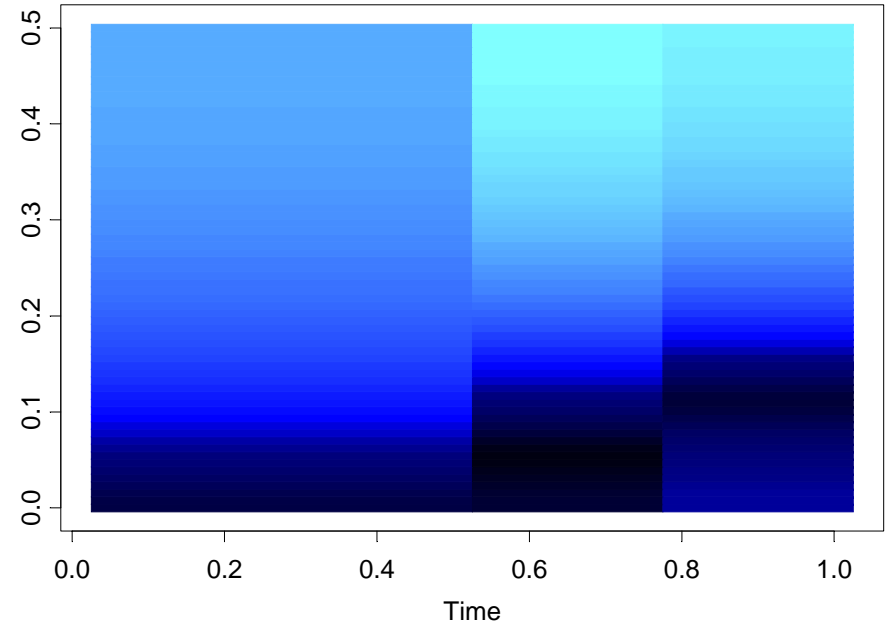
Fitted model:

	$\phi_1$	$\phi_2$	$\sigma^2$
1- 512:	.857		.9945
513- 768:	1.68	-0.801	1.1134
769-1024:	1.36	-0.801	1.1300

True Model



Fitted Model



## 1. Piecewise stat (cont)

**Simulation:** 200 replicates of time series of length 1024 were generated. (SLEX results from Ombao et al.)

$$ASE = n^{-1} (1/33) \sum_{t=1}^n \sum_{j=0}^{32} \{ \log \hat{f}(t/n, \omega_j) - \log f(t/n, \omega_j) \}^2, \quad \omega_j = 2\pi j / 64.$$

# of segments	Auto-SLEX			GA			
	%	Change Points	ASE	%	mean	std	ASE
2	0	1/2	-	0			
3	60.0	1/4 , 3/4	<b>25.42</b> (4.56)	82.0	.500 .749	.006 .006	<b>3.64</b> (.13)
4	34.0	1/4, 2/4, 3/4	<b>34.09</b> (6.74)	17.5	.476 .616 .761	.080 .110 .037	<b>3.73</b> (.13)
5	5.0	2/8, 4/8, 5/8, 6/8, 7/8	<b>32.77</b> (5.20)	0			
≥ 6	1.0		<b>50.01</b> (6.25)	0.5			<b>3.83</b>

## 1. Piecewise stat (cont)

Simulation (cont):

True model:

$$Y_t = \begin{cases} .9Y_{t-1} + \varepsilon_t, & \text{if } 1 \leq t < 513, \\ 1.69Y_{t-1} - .81Y_{t-2} + \varepsilon_t, & \text{if } 513 \leq t < 769, \\ 1.32Y_{t-1} - .81Y_{t-2} + \varepsilon_t, & \text{if } 769 \leq t \leq 1024, \end{cases}$$

AR orders selected (percent):

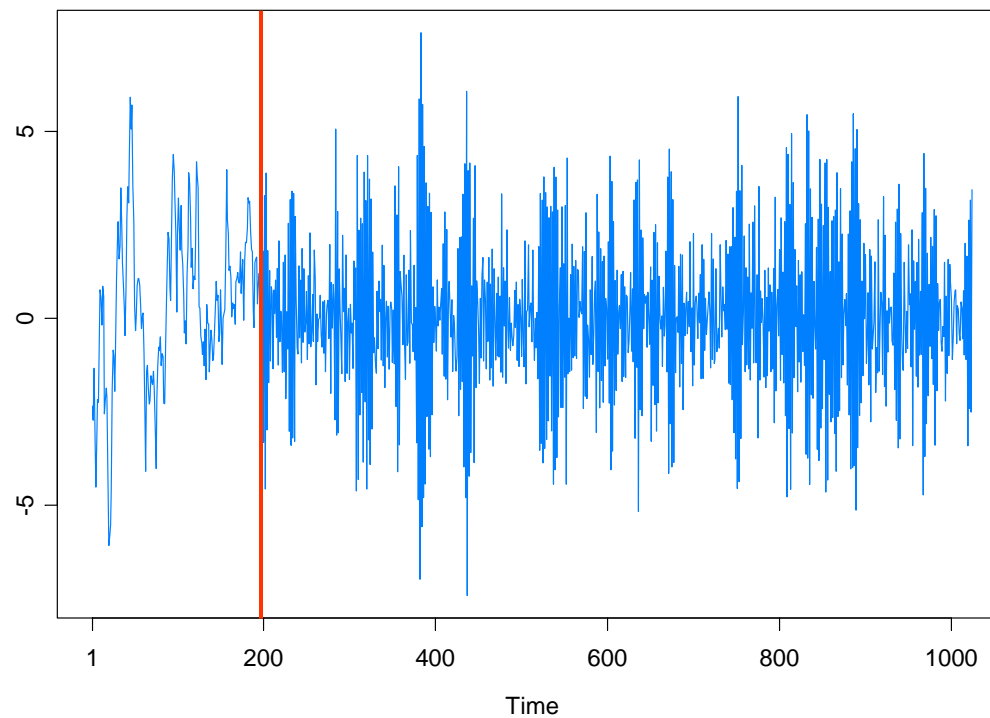
Order	0	1	2	3	4	5	$\geq 6$
$p_1$	0	<b>99.4</b>	0.60	0	0	0	0
$p_2$	0	0	<b>86.0</b>	11.6	1.8	0.6	0
$p_3$	0	0	<b>89.0</b>	10.4	0.6	0	0

## Simulation Examples (cont)

### 2. Piecewise stationary:

$$Y_t = \begin{cases} .9Y_{t-1} + \varepsilon_t, & \text{if } 1 \leq t < 198 \\ -.9Y_{t-1} + \varepsilon_t, & \text{if } 198 \leq t \leq 1024 \end{cases}$$

where  $\{\varepsilon_t\} \sim \text{IID } N(0,1)$ .



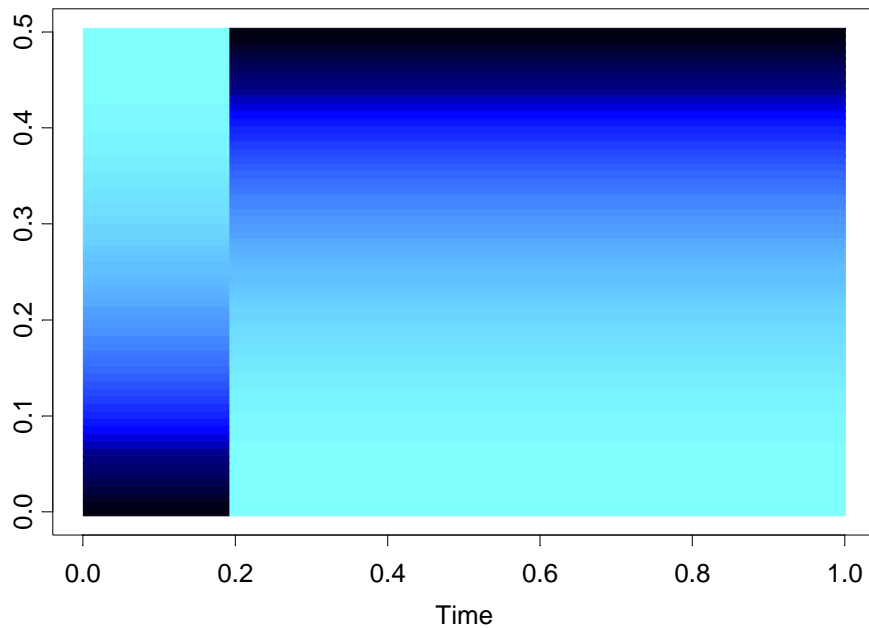
## 2. Piecewise stationary (cont)

GA results: 2 pieces with break at  $\tau_1=196$ . Total run time 11.96 secs

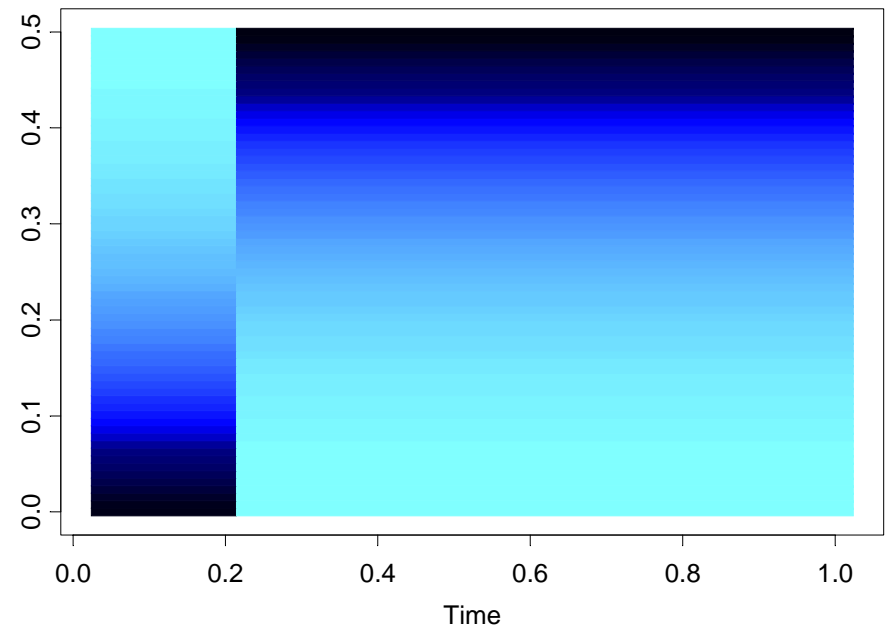
Fitted model:

	$\phi_1$	$\sigma^2$
1- 195:	.872	1.081
196 - 1024:	-.883	1.078

True Model



Fitted Model



## 2. Piecewise stationary (cont)

**Simulation:** 200 replicates of time series of length 1024 were generated. AR(1) models with break at  $196/1024 = .191$ .

# of segments	Auto-SLEX	
	%	ASE
$\leq 4$	0	-
5	44.0	<b>1.81</b> (.039)
6	37.0	<b>.220</b> (.056)
7	6.0	<b>.211</b> (.047)
8	12.0	<b>.290</b> (.096)
$\geq 9$	1.0	<b>.255</b> (.100)

# of segmnts	change points			
	%	mean	std	ASE
1	0			
2	97.0	.192	.002	<b>.017</b> (.013)
3	3.0	.154 .323	.023 .320	<b>.060</b> (.044)
$\geq 4$	0			

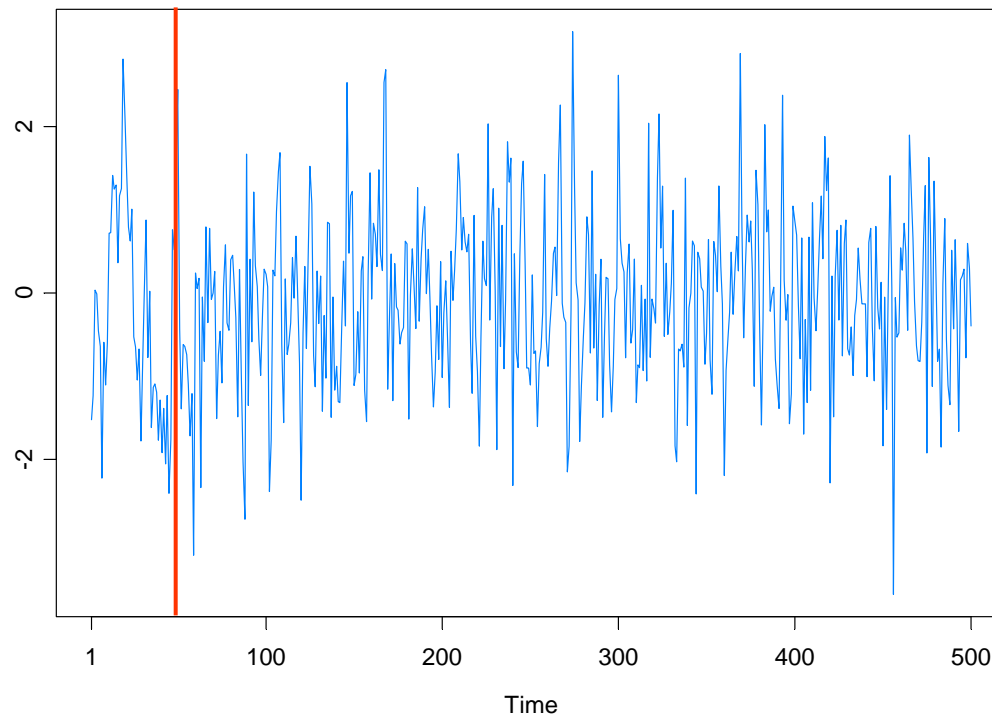
## Simulation Examples (cont)

### 3. Piecewise stationary:

$$Y_t = \begin{cases} .9Y_{t-1} + \varepsilon_t, & \text{if } 1 \leq t < 51 \\ .25Y_{t-1} + \varepsilon_t, & \text{if } 50 \leq t \leq 500 \end{cases}$$

where  $\{\varepsilon_t\} \sim \text{IID } N(0,1)$ .

GA results: 2 pieces with break at  $\tau_1=47$





### 3. Piecewise stationary (cont)

Simulation results: Change occurred at time  $\tau_1 = 51$ ;  $51/500=.1$

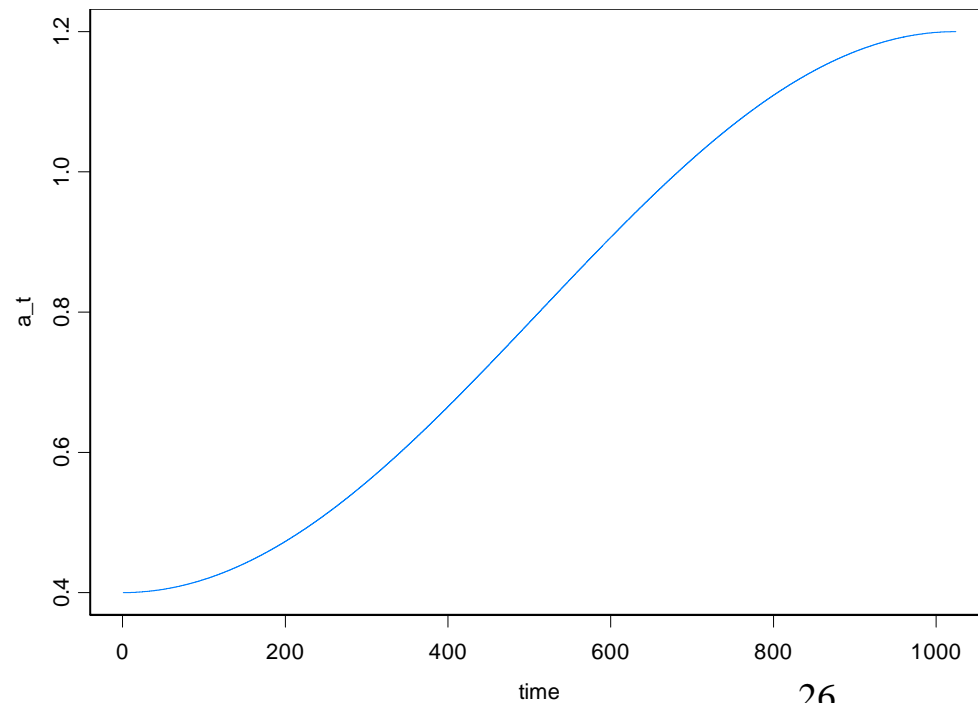
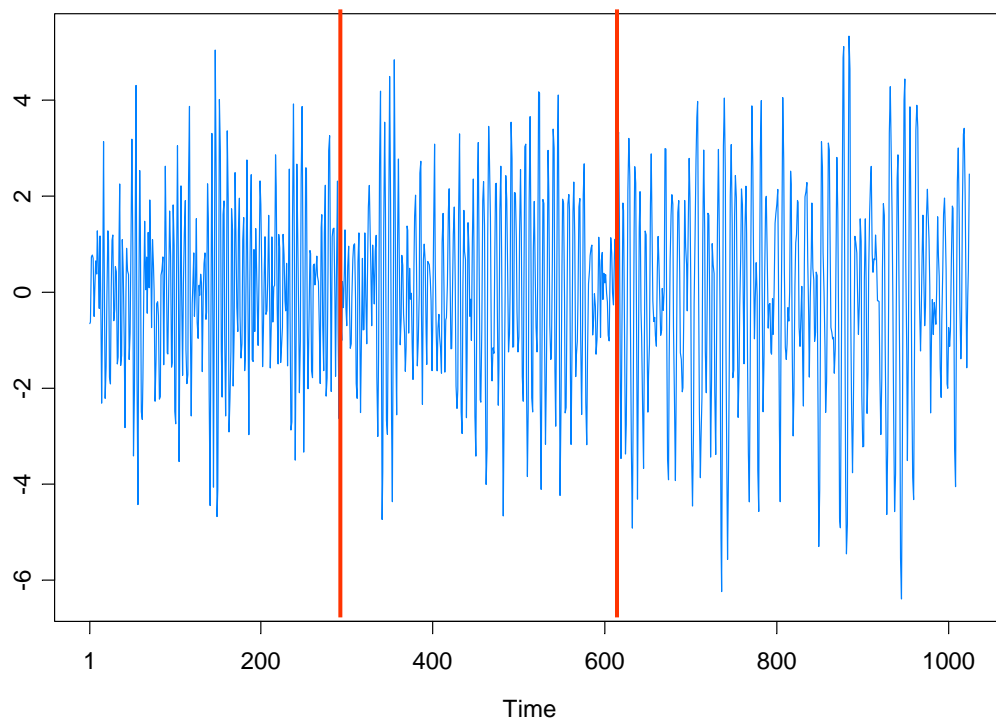
# of segments	change points		
	%	mean	std
1	9.0		.
2	89.0	.096	.017
3	2.0	.048 .092	.020 .011
$\geq 4$	0		

## Simulation Examples (cont)

### 4. Slowly varying AR(2) model:

$$Y_t = a_t Y_{t-1} - .81 Y_{t-2} + \varepsilon_t \quad \text{if } 1 \leq t \leq 1024$$

where  $a_t = .8[1 - 0.5 \cos(\pi t / 1024)]$ , and  $\{\varepsilon_t\} \sim \text{IID } N(0,1)$ .



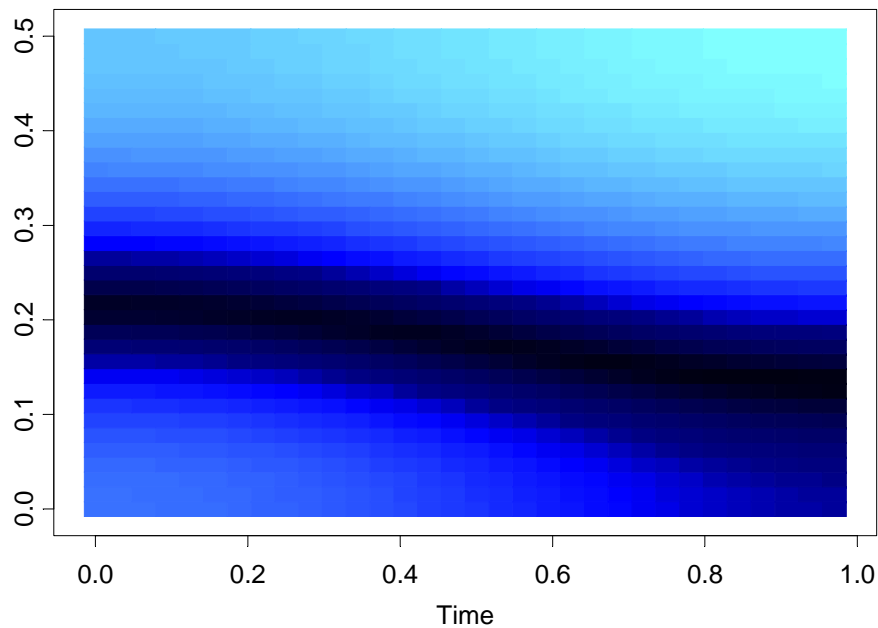
## 4. Slowly varying AR(2) (cont)

GA results: 3 pieces with breaks at  $\tau_1=293$  and  $\tau_2=615$ . Total run time 27.45 secs

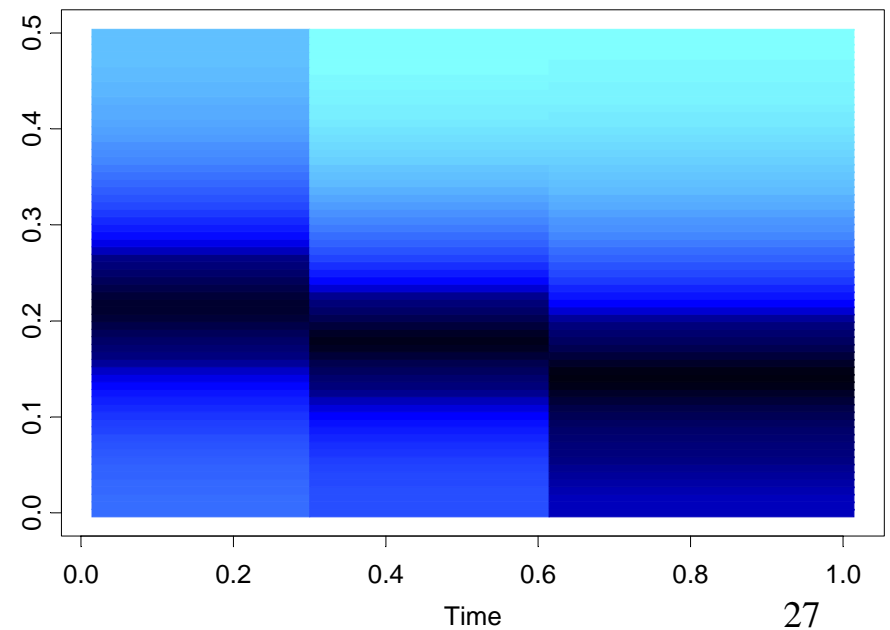
Fitted model:

	$\phi_1$	$\phi_2$	$\sigma^2$
1- 292:	.365	-0.753	1.149
293- 614:	.821	-0.790	1.176
615-1024:	1.084	-0.760	0.960

True Model



Fitted Model



## 4. Slowly varying AR(2) (cont)

**Simulation:** 200 replicates of time series of length 1024 were generated.

# of segments	Auto-SLEX	
	%	ASE
$\leq 4$	14.0	<b>.238</b> (.030)
5	27.0	<b>.228</b> (.025)
6	35.0	<b>.232</b> (.029)
7	18.0	<b>.243</b> (.033)
8	15.0	<b>.269</b> (.040)
$\geq 9$	1.0	<b>.308</b>

# of segments	change points			
	%	mean	std	ASE
1	0			
2	36.0	.502	.044	<b>.127</b> (.014)
3	63.0	.258	.071	<b>.080</b> (.016)
		.661	.075	
4	1.0	.309		
		.550		
		.860		
$\geq 5$	0			

## 4. Slowly varying AR(2) (cont)

Simulation (cont):

True model:  $Y_t = a_t Y_{t-1} - .81 Y_{t-2} + \varepsilon_t$  if  $1 \leq t \leq 1024$

AR orders selected (percent): (2 segment realizations)

Order	0	1	2	3	4	$\geq 5$
$p_1$	0	0	<b>97.2</b>	1.4	1.4	0
$p_2$	0	0	<b>97.2</b>	2.8	0	0

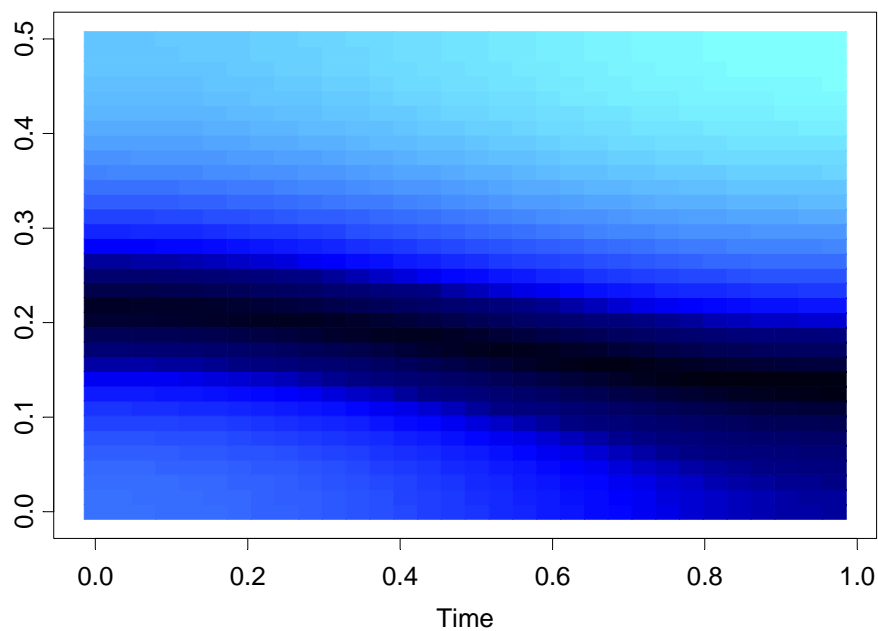
AR orders selected (percent): (3 segment realizations)

Order	0	1	2	3	4	$\geq 5$
$p_1$	0	0	<b>100</b>	0	0	0
$p_2$	0	0	<b>98.4</b>	1.6	0	0
$p_3$	0	0	<b>97.6</b>	1.6	0.8	0

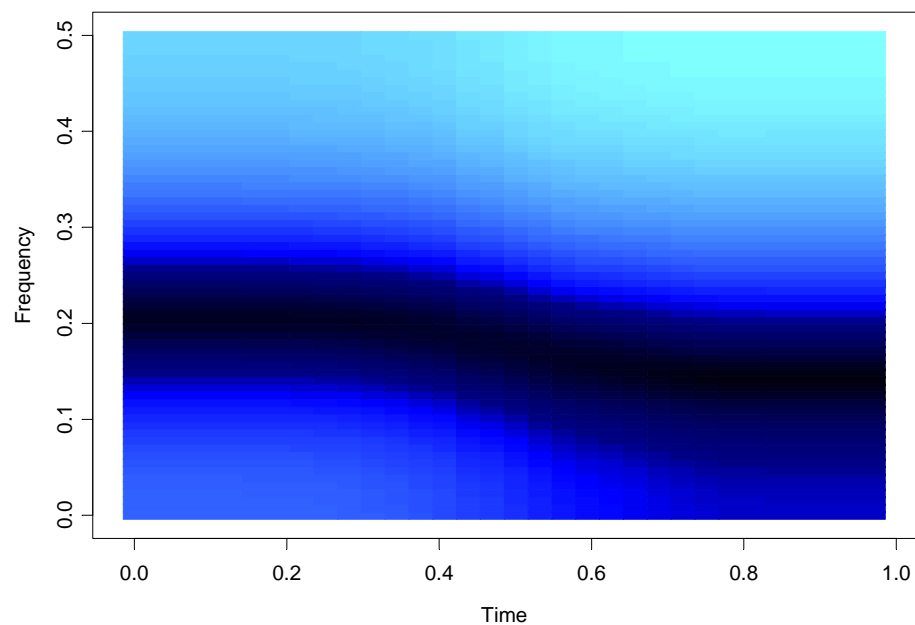
## 4. Slowly varying AR(2) (cont)

In the graph below right, we average the spectrogram over the *GA fitted models* generated from each of the 200 simulated realizations.

True Model



Average Model

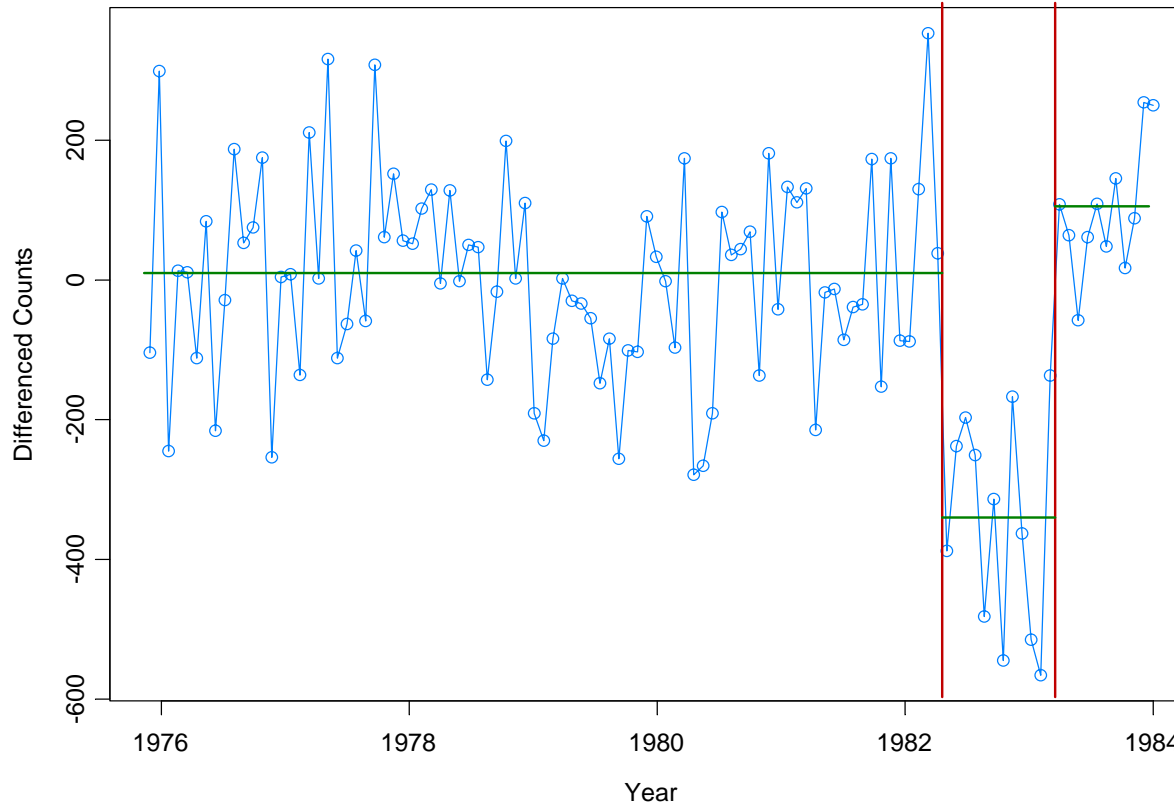




## Example: Monthly Deaths & Serious Injuries, UK

**Data:**  $Y_t$  = number of monthly deaths and serious injuries in UK, Jan '75 – Dec '84, ( $t = 1, \dots, 120$ )

**Remark:** Seat belt legislation introduced in Feb '83 ( $t = 99$ ).



**Results from GA:** 3 pieces; time = 4.4secs

**Piece 1:** ( $t=1, \dots, 98$ ) IID; **Piece 2:** ( $t=99, \dots, 108$ ) IID; **Piece 3:**  $t=109, \dots, 120$  AR(1)



## Application to Multivariate Time Series

Multivariate time series (d-dimensional):  $\mathbf{y}_1, \dots, \mathbf{y}_n$

Piecewise AR model:

$$\mathbf{Y}_t = \boldsymbol{\gamma}_j + \Phi_{j1} \mathbf{Y}_{t-1} + \dots + \Phi_{jp_j} \mathbf{Y}_{t-p_j} + \Sigma_j^{1/2} \mathbf{Z}_t, \quad \text{if } \tau_{j-1} \leq t < \tau_j,$$

where  $\tau_0 = 1 < \tau_1 < \dots < \tau_{m-1} < \tau_m = n + 1$ , and  $\{\mathbf{Z}_t\}$  is IID( $\mathbf{0}, I_d$ ).

In this case,

$$\begin{aligned} MDL(m, (\tau_1, p_1), \dots, (\tau_m, p_m)) &= \log m + m \log n + \sum_{j=1}^m \log p_j \\ &+ \sum_{j=1}^m \frac{p_j d^2 + d + d(d+1)/2}{2} \log n_j + \sum_{j=1}^m \frac{1}{2} \sum_{t=\tau_{j-1}}^{\tau_j-1} \left( \log(|\hat{V}_t|) + (\mathbf{Y}_t - \hat{\mathbf{Y}}_t)^T \hat{V}_t^{-1} (\mathbf{Y}_t - \hat{\mathbf{Y}}_t) \right), \end{aligned}$$

where  $\hat{\mathbf{Y}}_t = E(\mathbf{Y}_t | \mathbf{Y}_1, \dots, \mathbf{Y}_{t-1})$  and  $\hat{V}_t = E(\mathbf{Y}_t - \hat{\mathbf{Y}}_t)^2$  and the AR parameters are estimated by the multivariate Y-W equations based on Whittle's generalization of the Durbin-Levinson algorithm.

## Example: Bivariate Time Series

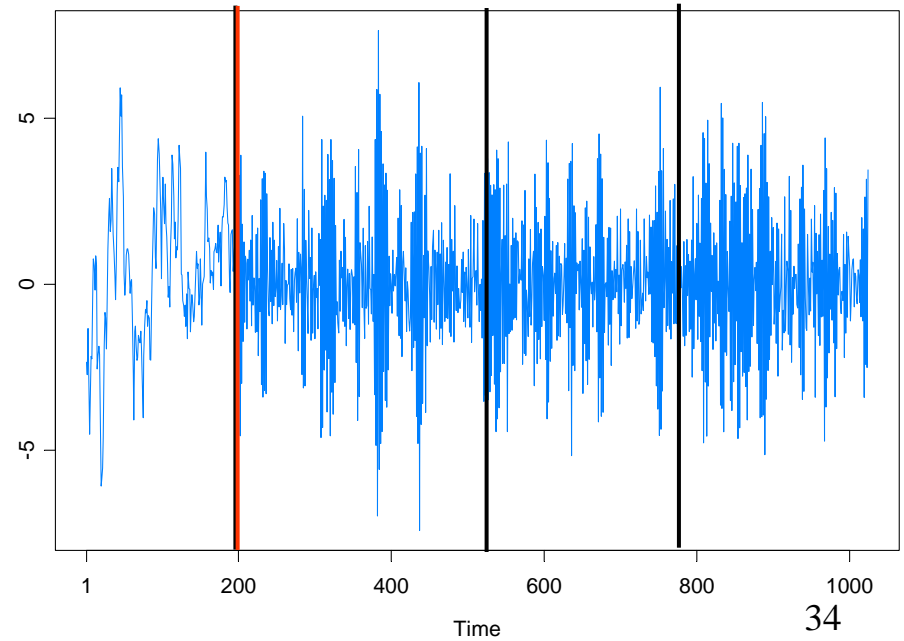
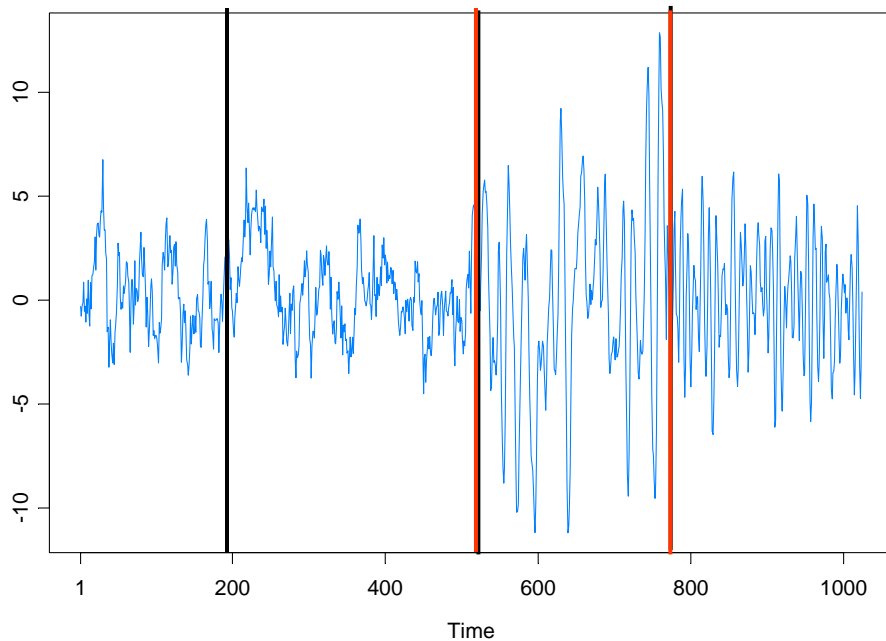
- $\{Y_{t1}\}$  same as the series in Example 2 (3 segments: AR(1), AR(3), AR(2))
- $\{Y_{t2}\}$  same as the series in Example 1 (2 segments: AR(1), AR(1))

GA results:

TS 1: 3 pieces with breaks at  $\tau_1=513$  and  $\tau_2=769$ . Total run time 16.31 secs

TS 2: 2 pieces with break at  $\tau_1=196$ . Total run time 11.96 secs

Bivariate: 4 pieces with breaks at  $\tau_1=197$ ,  $\tau_2=519$ ,  $\tau_3=769$ : AR(1), AR(1), AR(2), AR(2)  
Total run time 1126 secs

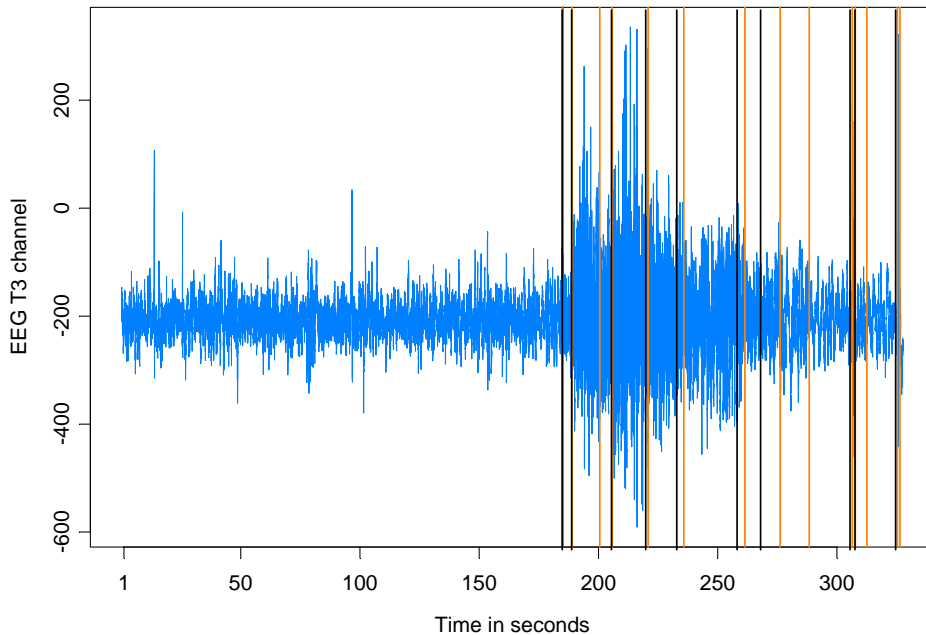


## Example: EEG Time series

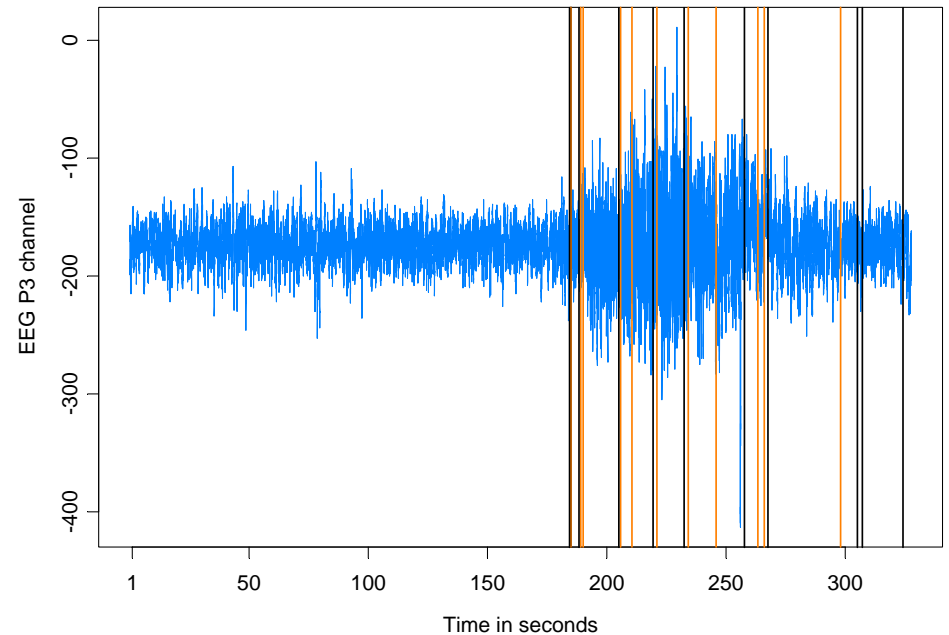
**Data:** Bivariate EEG time series at channels T3 (left temporal) and P3 (left parietal). Female subject was diagnosed with left temporal lobe epilepsy. Data collected by Dr. Beth Malow and analyzed in [Ombao et al \(2001\)](#). (n=32,768; sampling rate of 100Hz). Seizure started at about 1.85 seconds.

**GA bivariate results:** 14 break points for T3, 11 for P3, 2, 6, 15, 2, 3, 5, 4, 1

T3 Channel



P3 Channel

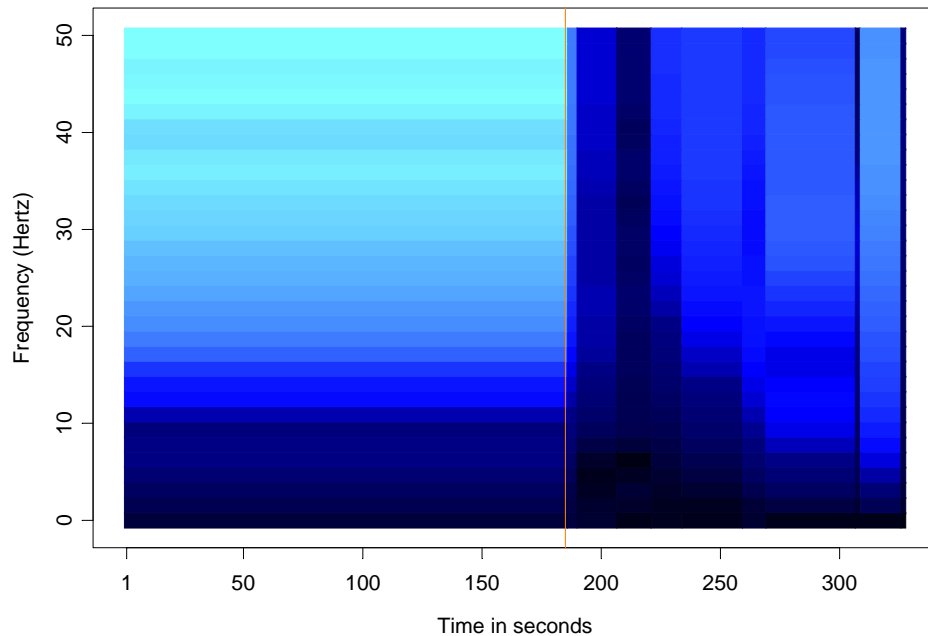


## Example: EEG Time series (cont)

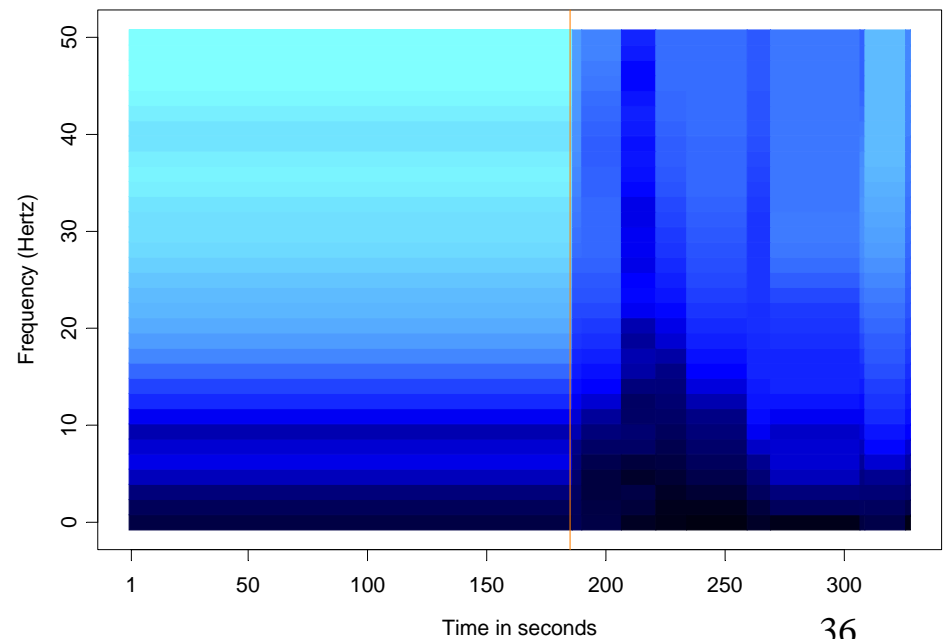
### Remarks:

- the general conclusions of this analysis are similar to those reached in Ombao et al.
- prior to seizure, power concentrated at lower frequencies and then spread to high frequencies.
- power returned to the lower frequencies at conclusion of seizure.

### T3 Channel



### P3 Channel

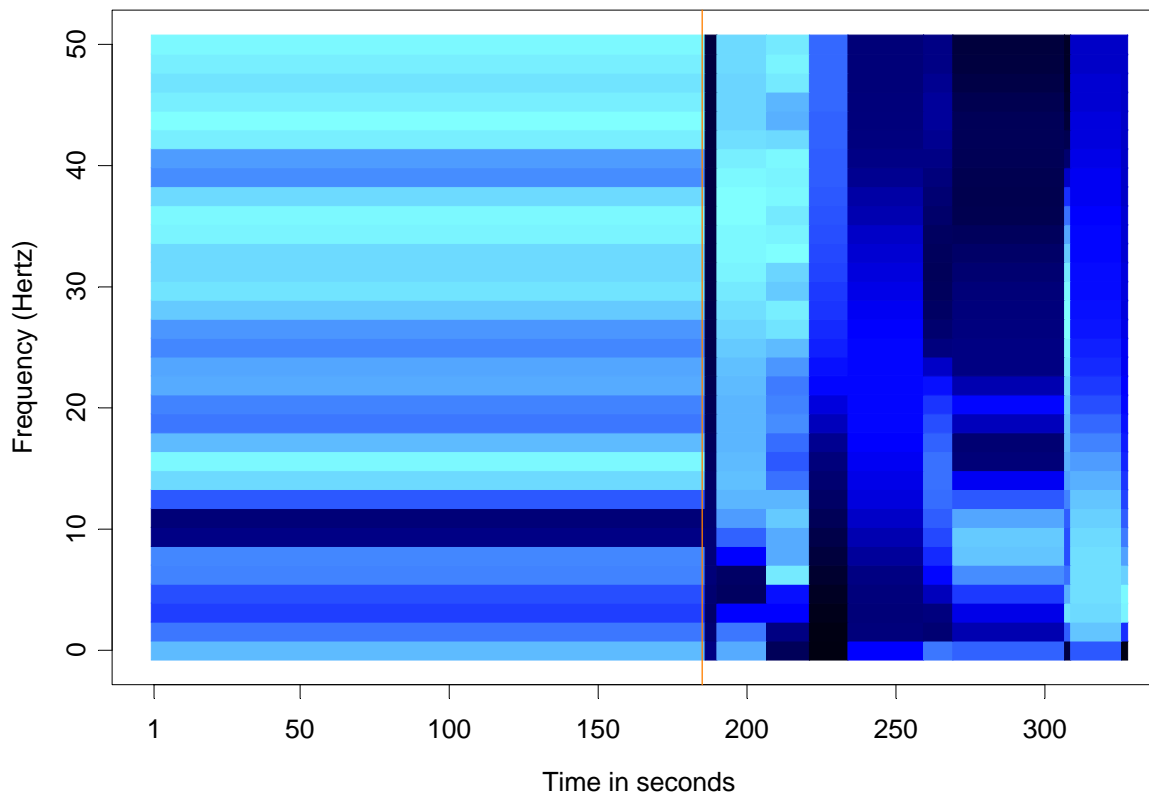


## Example: EEG Time series (cont)

### Remarks (cont):

- T3 and P3 strongly coherent at 9-12 Hz prior to seizure.
- strong coherence at low frequencies just after onset of seizure.
- strong coherence shifted to high frequencies during the seizure.

### T3/P3 Coherency



## Application to Structural Breaks (Davis, Lee, Rodriguez-Yam)

### State Space Model Setup:

#### Observation equation:

$$p(y_t | \alpha_t) = \exp\{\alpha_t y_t - b(\alpha_t) + c(y_t)\}.$$

**State equation:**  $\{\alpha_t\}$  follows the piecewise AR(1) model given by

$$\alpha_t = \gamma_k + \phi_k \alpha_{t-1} + \sigma_k \varepsilon_t, \quad \text{if } \tau_{k-1} \leq t < \tau_k,$$

where  $1 = \tau_0 < \tau_1 < \dots < \tau_m < n$ , and  $\{\varepsilon_t\} \sim \text{IID } N(0,1)$ .

#### Parameters:

$m$  = number of break points

$\tau_k$  = location of break points

$\gamma_k$  = level in  $k^{\text{th}}$  epoch

$\phi_k$  = AR coefficients  $k^{\text{th}}$  epoch

$\sigma_k$  = scale in  $k^{\text{th}}$  epoch

## Application to Structural Breaks—(cont)

**Estimation:** For  $(m, \tau_1, \dots, \tau_m)$  fixed, calculate the approximate likelihood evaluated at the “MLE”, i.e.,

$$L_a(\hat{\psi}; y_n) = \frac{|G_n|^{1/2}}{(K + G_n)^{1/2}} \exp\{y_n^T \alpha^* - 1^T \{b(\alpha^*) - c(y_n)\} - (\alpha^* - \mu)^T G_n (\alpha^* - \mu) / 2\},$$

where  $\hat{\psi} = (\hat{\gamma}_1, \dots, \hat{\gamma}_m, \hat{\phi}_1, \dots, \hat{\phi}_m, \hat{\sigma}_1^2, \dots, \hat{\sigma}_m^2)$  is the MLE.

**Goal:** Optimize an *objective function* over  $(m, \tau_1, \dots, \tau_m)$ .

- use minimum description length (MDL) as an objective function
- use genetic algorithm for optimization

## Application to Structural Breaks—(cont)

**Minimum Description Length (MDL):** Choose the model which maximizes the compression of the data or, equivalently, select the model that minimizes the code length of the data (i.e., amount of memory required to store the data).

$$\begin{aligned} \text{Code Length("data")} &= \text{CL("fitted model")} + \text{CL("data | fitted model")} \\ &\sim \text{CL("parameters")} + \text{CL("residuals")} \end{aligned}$$

$$\begin{aligned} MDL(m, \tau_1, \dots, \tau_m) \\ &= \underbrace{\log(m) + m \log(n) + 1.5 \sum_{j=1}^m \log(\tau_j - \tau_{j-1})}_{CL(\text{"Parameters"})} - \underbrace{\sum_{j=1}^m \log(L_a(\hat{\psi}_j; y_{\tau_{j-1}:\tau_j}))}_{CL(\text{"residuals"})} \end{aligned}$$

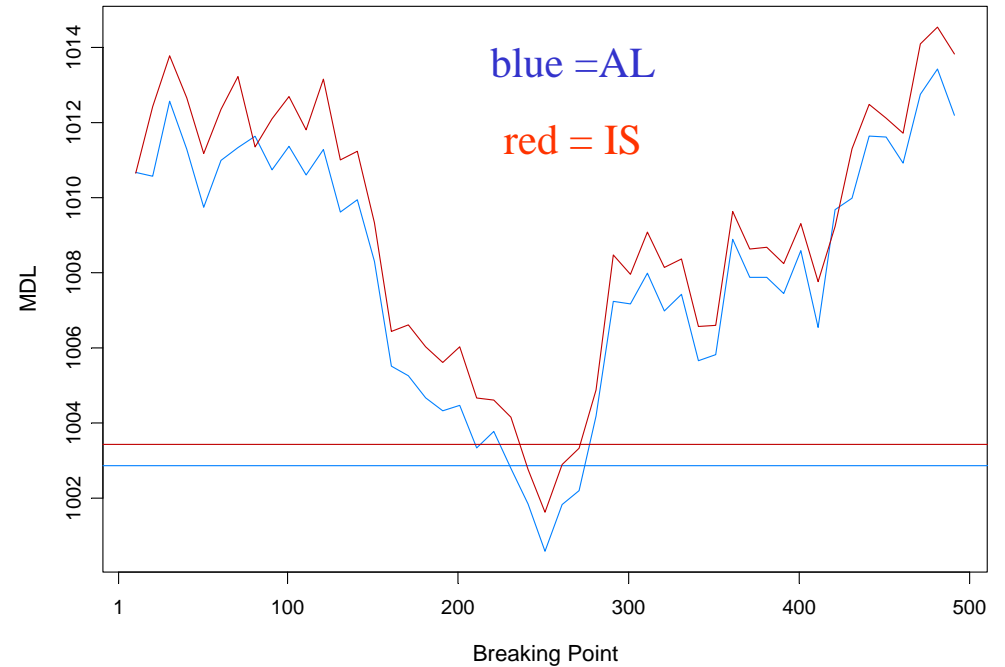
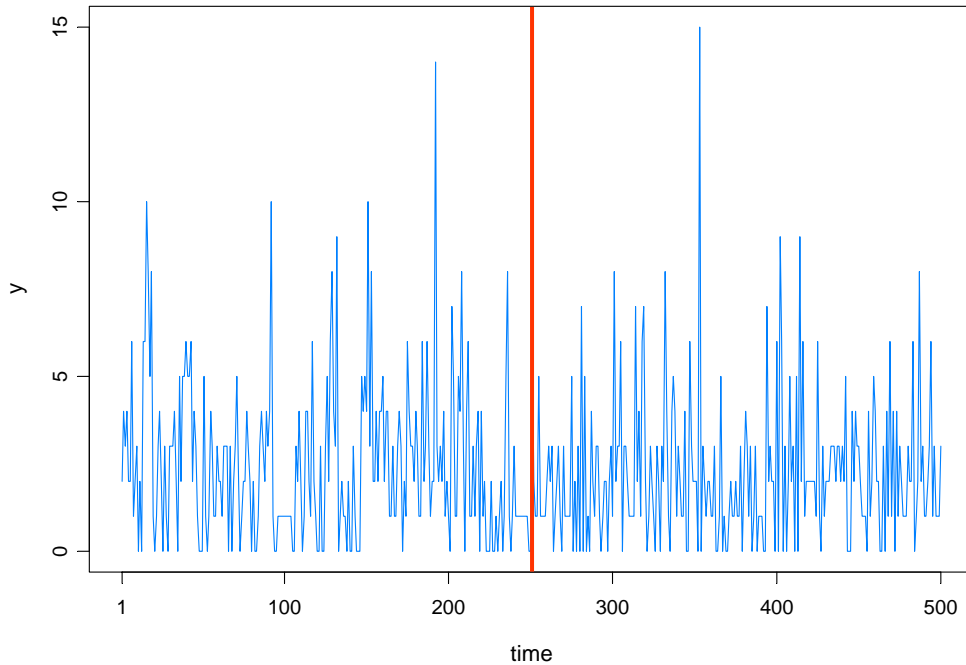
**Generalization:** AR(p) segments can have unknown order.

$$\begin{aligned} MDL(m, (\tau_1, p_1), \dots, (\tau_m, p_m)) \\ &= \log(m) + m \log(n) + 0.5 \sum_{j=1}^m (p_j + 2) \log(\tau_j - \tau_{j-1}) - \sum_{j=1}^m \log(L_a(\hat{\psi}_j; y_{\tau_{j-1}:\tau_j})) \end{aligned}$$



## Count Data Example

**Model:**  $Y_t | \alpha_t \sim \text{Pois}(\exp\{\beta + \alpha_t\})$ ,  $\alpha_t = \phi\alpha_{t-1} + \varepsilon_t$ ,  $\{\varepsilon_t\} \sim \text{IID } N(0, \sigma^2)$

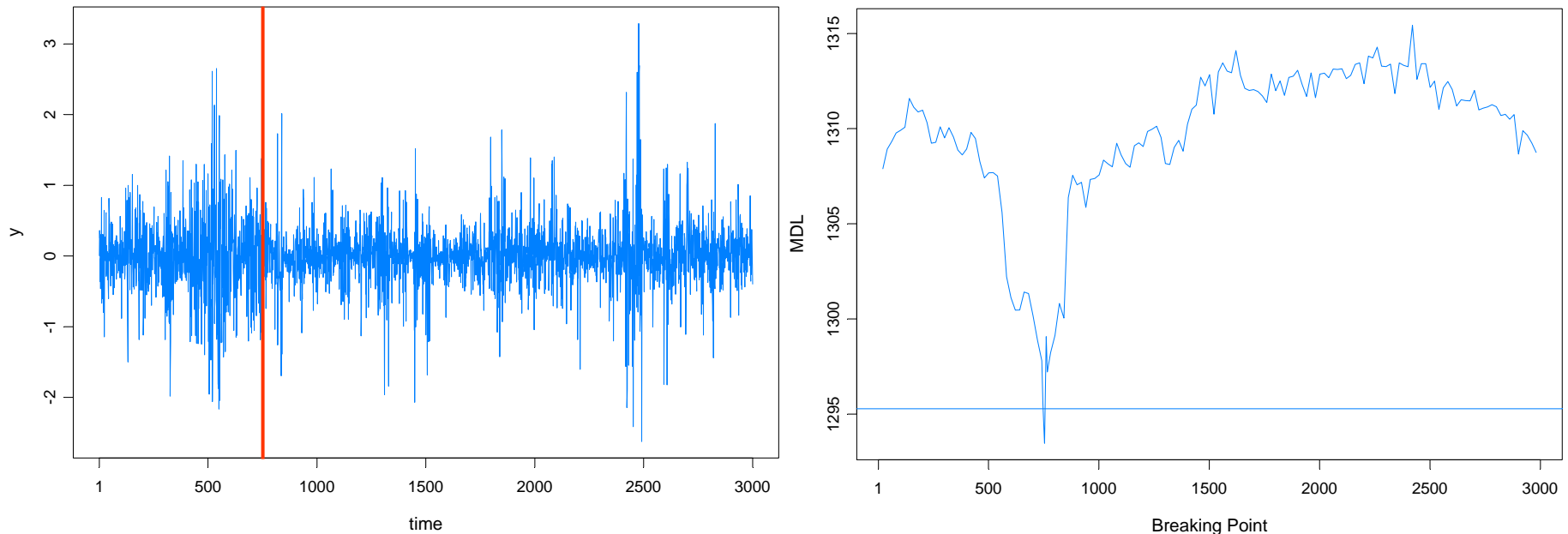


### True model:

- $Y_t | \alpha_t \sim \text{Pois}(\exp\{.7 + \alpha_t\})$ ,  $\alpha_t = .5\alpha_{t-1} + \varepsilon_t$ ,  $\{\varepsilon_t\} \sim \text{IID } N(0, .3)$ ,  $t < 250$
- $Y_t | \alpha_t \sim \text{Pois}(\exp\{.7 + \alpha_t\})$ ,  $\alpha_t = -.5\alpha_{t-1} + \varepsilon_t$ ,  $\{\varepsilon_t\} \sim \text{IID } N(0, .3)$ ,  $t > 250$ .
- GA estimate 251, time 267secs

## SV Process Example

**Model:**  $Y_t | \alpha_t \sim N(0, \exp\{\alpha_t\})$ ,  $\alpha_t = \gamma + \phi \alpha_{t-1} + \varepsilon_t$ ,  $\{\varepsilon_t\} \sim \text{IID } N(0, \sigma^2)$

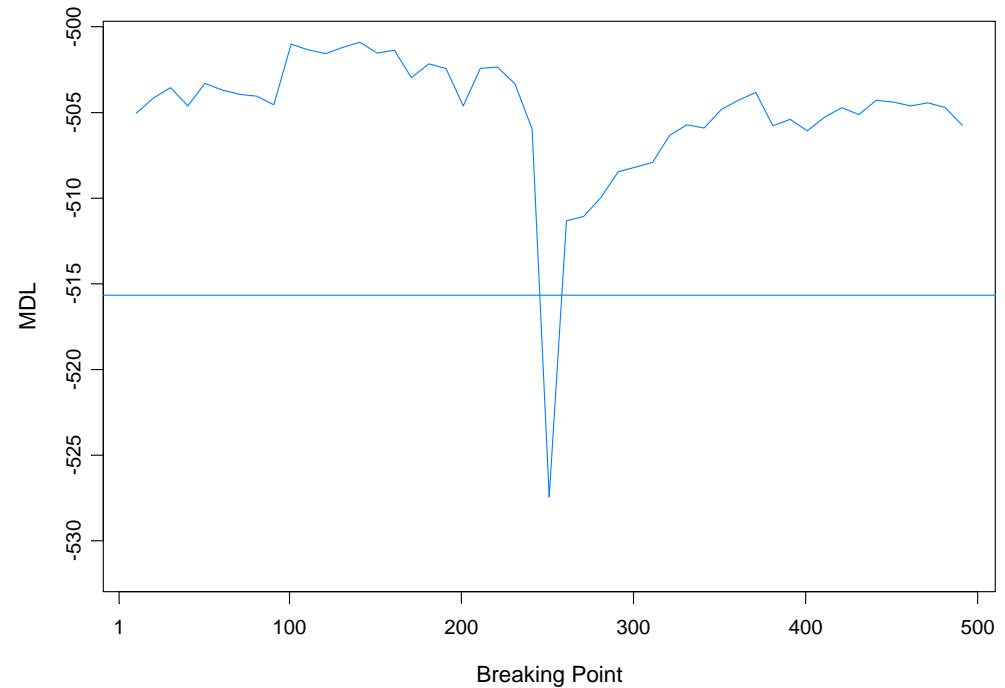
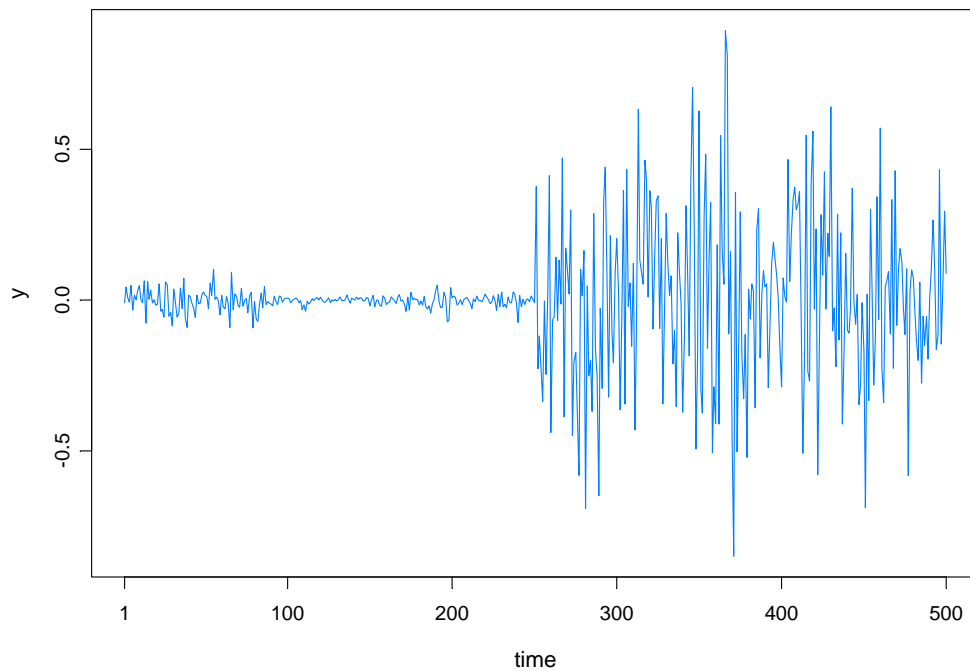


### True model:

- $Y_t | \alpha_t \sim N(0, \exp\{\alpha_t\})$ ,  $\alpha_t = -.05 + .975\alpha_{t-1} + \varepsilon_t$ ,  $\{\varepsilon_t\} \sim \text{IID } N(0, .05)$ ,  $t \leq 750$
- $Y_t | \alpha_t \sim N(0, \exp\{\alpha_t\})$ ,  $\alpha_t = -.25 + .900\alpha_{t-1} + \varepsilon_t$ ,  $\{\varepsilon_t\} \sim \text{IID } N(0, .25)$ ,  $t > 750$ .
- GA estimate 754, time 1053 secs

## SV Process Example

**Model:**  $Y_t | \alpha_t \sim N(0, \exp\{\alpha_t\})$ ,  $\alpha_t = \gamma + \phi \alpha_{t-1} + \varepsilon_t$ ,  $\{\varepsilon_t\} \sim \text{IID } N(0, \sigma^2)$



### True model:

- $Y_t | \alpha_t \sim N(0, \exp\{\alpha_t\})$ ,  $\alpha_t = -.175 + .977\alpha_{t-1} + \varepsilon_t$ ,  $\{\varepsilon_t\} \sim \text{IID } N(0, .1810)$ ,  $t \leq 250$
- $Y_t | \alpha_t \sim N(0, \exp\{\alpha_t\})$ ,  $\alpha_t = -.010 + .996\alpha_{t-1} + \varepsilon_t$ ,  $\{\varepsilon_t\} \sim \text{IID } N(0, .0089)$ ,  $t > 250$ .
- GA estimate 251, time 269s

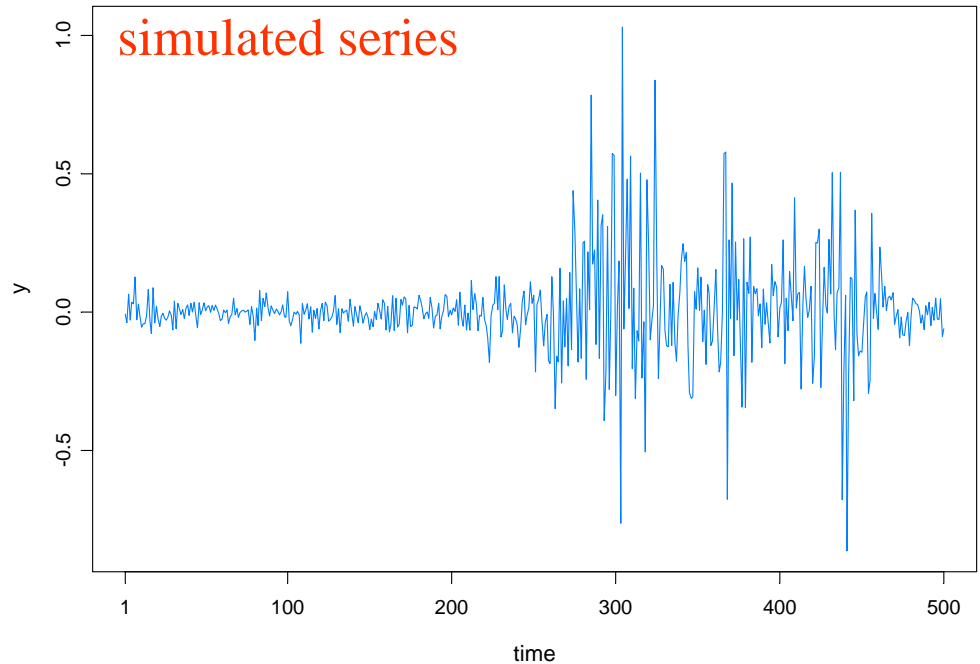
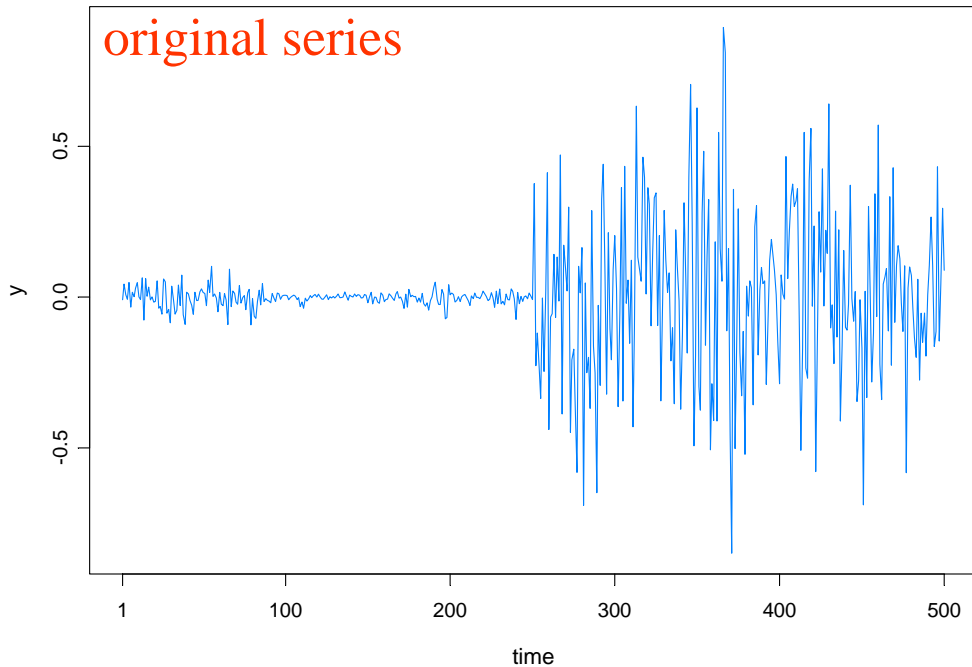
## SV Process Example-(cont)

True model:

- $Y_t | \alpha_t \sim N(0, \exp\{\alpha_t\})$ ,  $\alpha_t = -.175 + .977\alpha_{t-1} + \varepsilon_t$ ,  $\{\varepsilon_t\} \sim \text{IID } N(0, .1810)$ ,  $t \leq 250$
- $Y_t | \alpha_t \sim N(0, \exp\{\alpha_t\})$ ,  $\alpha_t = -.010 + .996\alpha_{t-1} + \varepsilon_t$ ,  $\{\varepsilon_t\} \sim \text{IID } N(0, .0089)$ ,  $t > 250$ .

Fitted model based on no structural break:

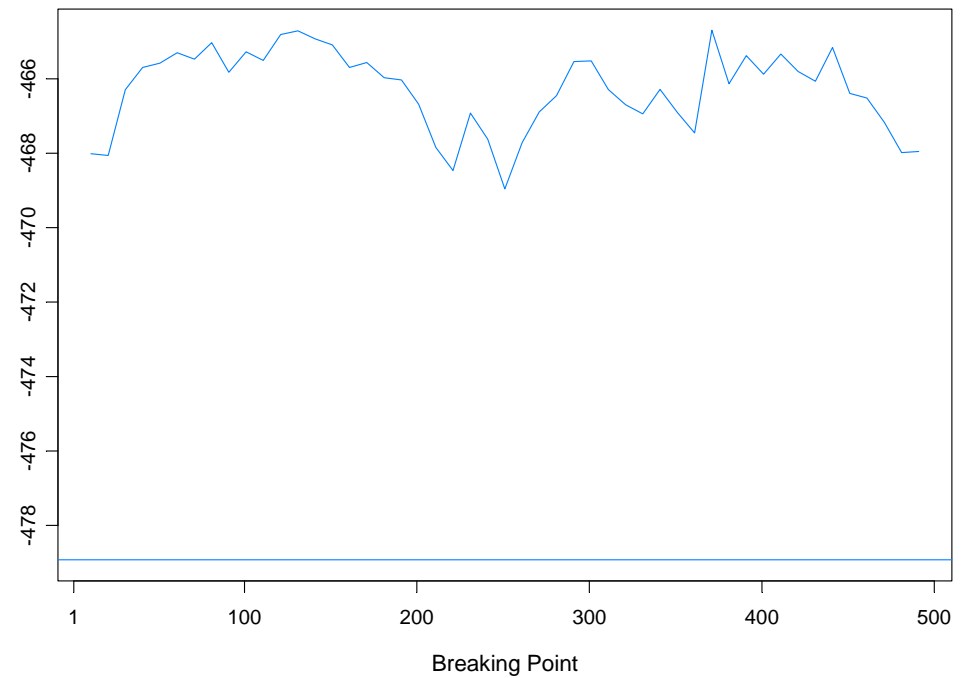
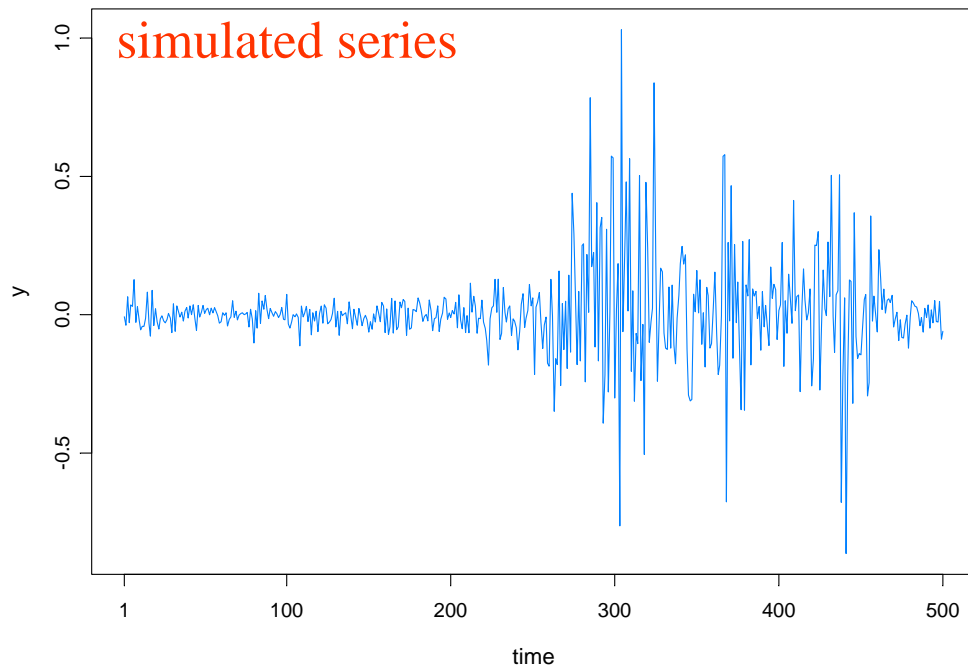
- $Y_t | \alpha_t \sim N(0, \exp\{\alpha_t\})$ ,  $\alpha_t = -.0645 + .9889\alpha_{t-1} + \varepsilon_t$ ,  $\{\varepsilon_t\} \sim \text{IID } N(0, .0935)$



## SV Process Example-(cont)

Fitted model based on no structural break:

- $Y_t | \alpha_t \sim N(0, \exp\{\alpha_t\})$ ,  $\alpha_t = -.0645 + .9889\alpha_{t-1} + \varepsilon_t$ ,  $\{\varepsilon_t\} \sim \text{IID } N(0, .0935)$



## Summary Remarks

1. *MDL* appears to be a good criterion for detecting structural breaks.
2. Optimization using a *genetic algorithm* is well suited to find a near optimal value of MDL.
3. This procedure extends easily to *multivariate* problems.
4. While estimating structural breaks for nonlinear time series models is *more challenging*, this paradigm of using *MDL together GA* holds promise for break detection in *parameter-driven* models and other nonlinear models.