Changepoint Analysis Using R

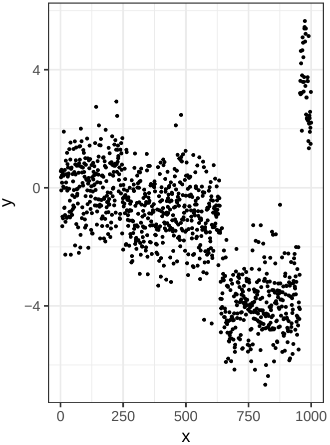
Robert Maidstone

16 October, 2018

What are Changepoints?

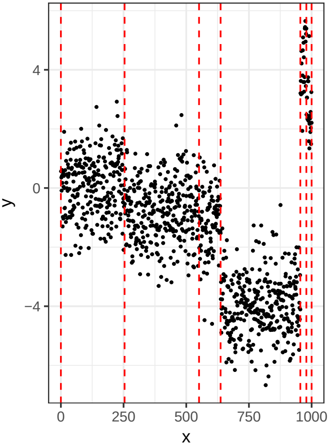
Change in mean

```r
set.seed(14)
m<-5
n<-1000
true_cps <- c(0,sort(sample(1:(n-1),m)),n)
means <- rnorm(m+1,0,4)
y<-c()
for(i in 1:(m+1)){
  j <- (true_cps[i]+1):true_cps[i+1]
  y[j]<-rnorm(length(j),means[i],1)
}
```



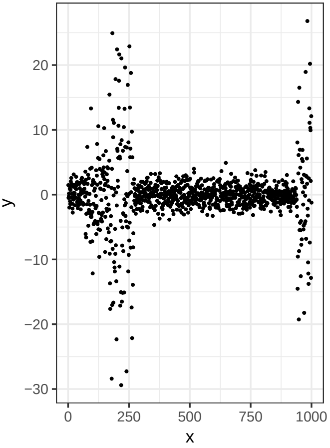
Change in mean

```r
set.seed(14)
m<-5
n<-1000
true_cps <- c(0,sort(sample(1:(n-1),m)),n)
means <- rnorm(m+1,0,4)
y<-c()
for(i in 1:(m+1)){
  j <- (true_cps[i]+1):true_cps[i+1]
  y[j]<-rnorm(length(j),means[i],1)
}
```



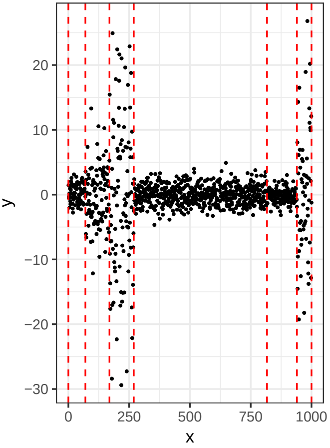
Change in variance

```r
set.seed(12)
m<-5
n<-1000
true_cps <- c(0,sort(sample(1:(n-1),m)),n)
sd <- runif(m+1,1,20)
y<-c()
for(i in 1:(m+1)){
  j <- (true_cps[i]+1):true_cps[i+1]
  y[j]<-rnorm(length(j),0,sd[i])
}
```



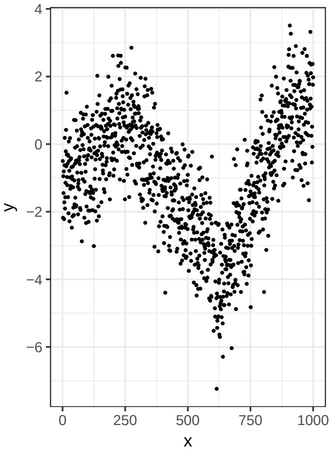
Change in variance

```
set.seed(12)
m<-5
n<-1000
true_cps <- c(0,sort(sample(1:(n-1),m)),n)
sd <- runif(m+1,1,20)
y<-c()
for(i in 1:(m+1)){
  j <- (true_cps[i]+1):true_cps[i+1]
  y[j]<-rnorm(length(j),0,sd[i])
}
```



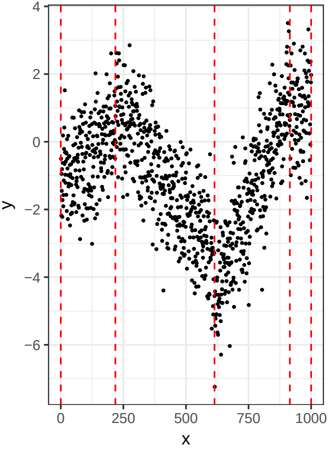
Change in trend

```r
set.seed(110)
m<-3
n<-1000
true_cps <- c(0,sort(sample(1:(n-1),m)),n)
slope <- rnorm(m+1,0,.01)
intercept <- rnorm(1,0,1)
y<-c()
for(i in 1:(m+1)){
  j <- (true_cps[i]+1):true_cps[i+1]
  if(i==1){
  for(jind in j){
    y[jind]<-intercept+(jind-true_cps[i])*
      slope[i] + rnorm(1,0,1)
  }
  }else{
    for(jind in j){
      y[jind]<-y[j[1]-1]+(jind-true_cps[i])*
        slope[i] + rnorm(1,0,1)
    }
  }
}
```



Change in trend

```r
set.seed(110)
m<-3
n<-1000
true_cps <- c(0,sort(sample(1:(n-1),m)),n)
slope <- rnorm(m+1,0,.01)
intercept <- rnorm(1,0,1)
y<-c()
for(i in 1:(m+1)){
  j <- (true_cps[i]+1):true_cps[i+1]
  if(i==1){
  for(jind in j){
    y[jind]<-intercept+(jind-true_cps[i])*
      slope[i] + rnorm(1,0,1)
  }
  }else{
    for(jind in j){
      y[jind]<-y[j[1]-1]+(jind-true_cps[i])*
        slope[i] + rnorm(1,0,1)
    }
  }
}
```



Real World Examples

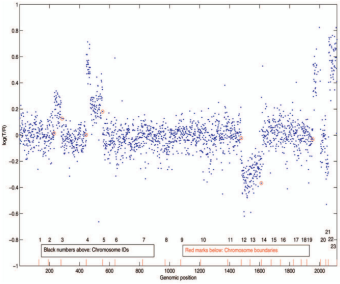
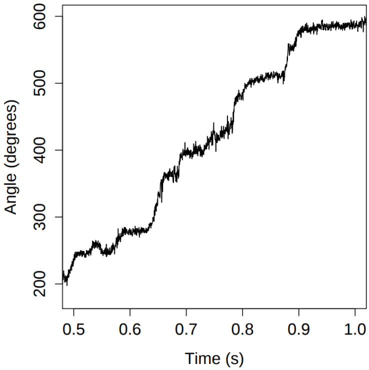


Fig. 5.
Genome of the breast tumor S1514 [39].

(a) Copy number at genomic positions in a human breast tumor sample (Chen & Wang, 2009).



(b) Rotation of a bacterial flagella motor (Maidstone, 2016).

Many changepoint methods

Many different changepoint algorithms exist

- Exhaustive Search
- Optimal Partitioning
- PELT
- FPOP (and R-FPOP)
- CROPS
- Segment Neighbourhood Search
- pDPA
- SNIP

- Binary Segmentation
- WBS
- CBS
- SMUCE
- SMOP
- ED-PELT
- E-Divisive
- ECP

Cost function representation

Most changepoint detection methods boil down to minimising the sum of some cost function, $\mathcal{C}(\cdot)$, over the segments.

$$\min_{\tau_{1:m},m}\left[\sum_{j=0}^{m+1}\mathcal{C}(\mathbf{y}_{\tau_j+1:\tau_{j+1}})\right]$$

This cost function could be a number of things:

1. Negative log-likelihood,
2. Negative posterior,
3. Minimum Description Length.

Dynamic Programming Methods

Optimal Partitioning: Optimisation based sum of optimal up to last changepoint and the cost between last changepoint and current time (plus a penalty to avoid over fitting).

$$F(\tau^*) = \min_{0 \leq \tau < \tau^*} [F(\tau) + \mathcal{C}(\mathbf{y}_{(\tau+1):\tau^*}) + \beta].$$

Segment Neighbourhood Search: Optimisation for k segments based on optimal for $k-1$ segments plus cost for new segment.

$$q^k_{1,j} = \min_{v \in \{1,\dots,j-1\}} [q^{k-1}_{1,v} + q^1_{v+1,j}].$$

Optimal Partitioning using Rigaill's Pruning

Input: A set of data of the form, $(y_1, y_2, \ldots, y_n)$ where $y_i \in \mathbb{R}$.
A measure of fit $\gamma(\cdot,\cdot)$,
A penalty β which does not depend on the number or location of changepoints.

Initialise: Let $n =$length of data, and set $F(0) = -\beta$, $cp(0) = 0$,
$LOC = \{0\}$, $D = [\min_{1\leq i\leq n}(y_i), \max_{1\leq i\leq n}(y_i)]$, $Set_0 = D$,
$Cost_0(\mu, 0) = F(0) + \beta = 0$

Iterate: for $\tau^* = 1, \ldots, n$

1. Update functions $Cost_\tau(\mu, \tau^*)$ for all $\tau \in LOC$.
2. Set $F(\tau^*) = \min_{\tau \in LOC}(\min_{\mu \in Set_\tau}[Cost_\tau(\mu, \tau^*)])$.
3. Let $\tau' = \arg\min_{\tau \in LOC}(\min_{\mu \in Set_\tau}[Cost_\tau(\mu, \tau^*)])$.
4. Set $Cost_{\tau^*}(\mu, \tau^*) = F(\tau^*) + \beta$ and $Set_{\tau^*} = D$.
5. Set $cp(\tau^*) = (cp(\tau'), \tau')$.
6. Update LOC.

Output: The changepoints recorded in $cp(n)$.

refs

- Chen, J., & Wang, Y.-P. (2009). A Statistical Change Point Model Approach for the Detection of DNA Copy Number Variations in Array CGH Data. IEEE/ACM Transactions on Computational Biology and Bioinformatics / IEEE, ACM, 6(4), 529–541. http://doi.org/10.1109/TCBB.2008.129