Statistics (and Algorithms) for Change-Point Detection in Genomics: Detection of CNV and others

S. Robin (with help from A. Cleynen)


Ecole de bioinformatique 'Traitement des données NGS', Roscoff, October 2014
Change-point detection in genomics
Change-point (or segmentation) problems raise in many fields of genomics:

- Copy number variation (CGHarrays, DNAseq),
- Gene detection (tilling arrays, RNA-seq),
- Protein-DNA interaction (ChIP-chip, CHiP-seq),
- and many others...
Change-point (or segmentation) problems raise in many fields of genomics:
- Copy number variation (CGHarrays, DNAseq),
- Gene detection (tilling arrays, RNA-seq),
- Protein-DNA interaction (ChIP-chip, CHiP-seq),
- and many others...

They are faced as soon as one deal with data
- organized along one dimension (time, genome position, ...),
- expected to display abrupt changes,
- in presence of noise.
CNV analysis with DNAseq

- DNAseq reads are mapped along a reference genome.
- Variation of their density reveals variation of the copy number.

\[ Y_t = \text{number of mapped reads starting at nucleotide } t \]

→ 'depth of coverage', 'read depth'.

Rashid et al. (2011)
CNV detection using microrray

Hupé (2008)

Data.

\[ Y_t = \log\text{-fluorescence ratio at probe } t \]
Genome re-arrangement

Zoom on CGH profile

chrom. 1

1p loss

1q gain

chrom. 17

17q gain

Karyotype

Unbalanced translocation
1p - 17q

1q gain
Gene discovery with RNA-seq

Data.

\[ Y_t = \text{number of mapped reads starting at nucleotide } t \]

RNA-seq counts

Known exons

source: genomebiology.com
Segmentation: the basic problem

Data at hand: for a given individual

- a sequence of known positions along the reference genome, labeled as
  \[ t = 1, 2, \ldots, n \]

- a signal measured at each position
  \[ Y_t = \text{signal at position } t \]
Data at hand: for a given individual

- a sequence of known positions along the reference genome, labeled as
  \[ t = 1, 2, \ldots, n \]

- a signal measured at each position
  \[ Y_t = \text{signal at position } t \]
Segmentation: the basic problem

Data at hand: for a given individual

- a sequence of known positions along the reference genome, labeled as
  \[ t = 1, 2, \ldots, n \]

- a signal measured at each position
  \[ Y_t = \text{signal at position } t \]

\[ Y_t \propto f(\text{relative copy number at position } t) \]
\[ = \text{DNAseq count, log-fluorescence, etc.} \]
Segmentation: the basic problem

Data at hand: for a given individual
- a sequence of known positions along the reference genome, labeled as
  \[ t = 1, 2, \ldots, n \]
- a signal measured at each position
  \[ Y_t = \text{signal at position } t \]

\[ Y_t \propto f(\text{relative copy number at position } t) \]
\[ = \text{DNAseq count, log-fluorescence, etc.} \]
Example: Breast cancer CGHarray

Chrom. 10 and 8 of two breast carcinomas (TNBC). Rigaill (2011)
(Statistical) questions

Biological questions:

- How many change-points?
- Where are the change-points? (How precise is the location?)
- Do the change-points have the same location in samples $A$ and $B$?
- ...

...
(Statistical) questions

Biological questions:
- How many change-points?
- Where are the change-points? (How precise is the location?)
- Do the change-points have the same location in samples $A$ and $B$?
- ...

Statistical issues:
- Statistical modeling
- Computational efficiency
- Model selection
Data specificities

Dimension.

- Whole chromosome CNV detection using DNAseq $\rightarrow n = 10^8$
- Whole chromosome CNV detection using SNParray $\rightarrow n = 10^6$
- ’Small’ gene reannotation using RNAseq $\rightarrow n = 10^4$

$\rightarrow$ Different computational burdens.

\[^1\]Are statisticians slower than computer scientists?
Data specificities

Dimension.
- Whole chromosome CNV detection using DNAsseq $\rightarrow n = 10^8$
- Whole chromosome CNV detection using SNPArray $\rightarrow n = 10^6$
- 'Small’ gene reannotation using RNAseq $\rightarrow n = 10^4$
$\rightarrow$ Different computational burdens.

Structure.
- One measurement per position (probe or nucleotide)
- Pairs of positions (pair-end) $\rightarrow$ out of my scope\(^1\)

---

\(^1\)Are statisticians slower than computer scientists?
Data specificities

Dimension.
- Whole chromosome CNV detection using DNAseq $\rightarrow n = 10^8$
- Whole chromosome CNV detection using SNPArray $\rightarrow n = 10^6$
- ‘Small’ gene reannotation using RNASeq $\rightarrow n = 10^4$
- Different computational burdens.

Structure.
- One measurement per position (probe or nucleotide)
- Pairs of positions (pair-end) $\rightarrow$ out of my scope\(^1\)

Nature.
- Microarray $\rightarrow$ fluorescence $\rightarrow$ continuous data $(\in \mathbb{R})$
- NGS $\rightarrow$ read counts $\rightarrow$ discrete data $(\in \mathbb{N})$

\(^1\)Are statisticians slower than computer scientists?
Outline

1. Segmentation methods
2. Hidden Markov model
3. Regularized approaches
Segmentation methods
A model, what for?

- To translate biological questions into mathematical equations and quantities;
- To make all hypotheses explicit;
- To set the inference of interesting parameters in a global framework, possibly accounting for other effects (covariates);
- To motivate all calculations and data processing to come.
A model, what for?

- To translate biological questions into mathematical equations and quantities;
- To make all hypotheses explicit;
- To set the inference of interesting parameters in a global framework, possibly accounting for other effects (covariates);
- To motivate all calculations and data processing to come.

Model-based approach.

- Define a model that describes the biological process as well as possible;
- Make sure that it can be handled in terms of mathematics / statistics;
- Derive an (efficient and statistically valid) inference procedure.
Segmentation model
Statistical model.

- Signal = \( f(\text{Position}) \);
Segmentation model

Statistical model.

- Signal \( = f(\text{Position}) \);
- Breakpoint positions: \( \tau_1, \tau_2, \ldots, \tau_{K-1} \);

\[
\text{if } t \in r_k, \quad Y_t
\]
Statistical model.

- Signal = $f(\text{Position})$;
- Breakpoint positions: $\tau_1, \tau_2, \ldots, \tau_{K-1}$;
- 'Mean' signal (e.g. copy number) within each interval: $\mu_1, \mu_2, \ldots, \mu_K$;

\[
\text{if } t \in r_k, \quad Y_t = \mu_k
\]
Statistical model.

- Signal = \( f(\text{Position}); \)
- Breakpoint positions: \( \tau_1, \tau_2, \ldots, \tau_{K-1}; \)
- 'Mean' signal (e.g. copy number) within each interval: \( \mu_1, \mu_2, \ldots, \mu_K; \)
- Observed signal at time \( t \): independent variables with given parameter (mean, dispersion, ...).

\[
\text{if } t \in r_k, \quad Y_t \sim F(\mu_k)
\]
Dealing with (some) data specificities

Distribution. The distribution $\mathcal{F}$ of the data is chosen according to the technology:

\[
\begin{align*}
Y_t &\sim \mathcal{N}(\mu_k, \sigma^2) \quad \text{aCGH (same variance)} \\
Y_t &\sim \mathcal{P}(\mu_k) \quad \text{NGS} \\
Y_t &\sim \mathcal{NB}(\mu_k, \phi) \quad \text{NGS (over-dispersed)}
\end{align*}
\]
Dealing with (some) data specificities

**Distribution.** The distribution $\mathcal{F}$ of the data is chosen according to the technology:

$$
\begin{align*}
Y_t &\sim \mathcal{N}(\mu_k, \sigma^2) \quad \text{aCGH (same variance)} \\
Y_t &\sim \mathcal{P}(\mu_k) \quad \text{NGS} \\
Y_t &\sim \mathcal{N}\mathcal{B}(\mu_k, \phi) \quad \text{NGS (over-dispersed)}
\end{align*}
$$

if position $t$ is in segments $r_k$:

**Normalization?** Explicit modeling allows to account for additional information, e.g. $x_t =$ GC content at position $t$:

$$
Y_t \sim \mathcal{N}(\mu_k + \lambda x_t, \sigma^2)
$$

(see CGHseg package later on).
Data transformation

- The reference distributions for NGS data are Poisson and negative binomial for which many methods are being developed.
Data transformation

- The reference distributions for NGS data are Poisson and negative binomial for which many methods are being developed.

- Many methods exist for continuous data (Gaussian, as usual, easier to handle).
Data transformation

- The reference distributions for NGS data are Poisson and negative binomial for which many methods are being developed.

- Many methods exist for continuous data (Gaussian, as usual, easier to handle).

- Data transformation could be used to apply them to NGS data, e.g.:

\[
\tilde{Y}_t = \log(1 + Y_t), \quad \tilde{Y}_t = \sqrt{Y_t},
\]
Data transformation

- The reference distributions for NGS data are Poisson and negative binomial for which many methods are being developed.

- Many methods exist for continuous data (Gaussian, as usual, easier to handle).

- Data transformation could be used to apply them to NGS data, e.g.:

  \[ \tilde{Y}_t = \log(1 + Y_t), \quad \tilde{Y}_t = \sqrt{Y_t}, \]

- Reminder: the variance stabilizing transformation for the negative binomial is

  \[ \tilde{Y}_t = \text{arg sinh} \left( \sqrt{Y_t/\phi} \right) \]
Parameter inference

Two different types of parameters of the models:

- **Segmentation (change-point locations):** $T = (\tau_1, \ldots, \tau_{K-1})$
  $\rightarrow$ discrete parameter;

- **Distribution parameters (within segment means):** $\mu = (\mu_1, \ldots, \mu_K)$
  $\rightarrow$ continuous parameter.
Parameter inference

Two different types of parameters of the models:

- Segmentation (change-point locations): $T = (\tau_1, \ldots, \tau_{K-1})$  
  → discrete parameter;
- Distribution parameters (within segment means): $\mu = (\mu_1, \ldots, \mu_K)$  
  → continuous parameter.

Fitting model to data: The estimates of $T$ and $\mu$ are expected to provide a good fit to the data.

1. Define a criterion to measure this fit
2. Find the 'optimal' values $\hat{T}$ and $\hat{\mu}$ providing the best fit.
Maximum likelihood

Most common strategy: Maximum likelihood. To get estimate of the parameters of this model \((T, \mu)\) we choose to maximize the likelihood of the observed data:

\[
p(Y; T, \mu) = \prod_k \prod_{t \in r_k} p(Y_t; \mu_k)
\]

\[
\Rightarrow \quad \log p(Y; T, \mu) = \sum_k \sum_{t \in r_k} \log p(Y_t; \mu_k).
\]
Maximum likelihood

Most common strategy: Maximum likelihood. To get estimate of the parameters of this model \((T, \mu)\) we choose to maximize the likelihood of the observed data:

\[
p(Y; T, \mu) = \prod_k \prod_{t \in r_k} p(Y_t; \mu_k) \]

\[
\Rightarrow \quad \log p(Y; T, \mu) = \sum_k \sum_{t \in r_k} \log p(Y_t; \mu_k).
\]

Array CGH. Gaussian with same variance \(N(\mu_k, \sigma^2)\):

\[
\log P(Y; T, \mu) = \text{cst} - \text{cst} \sum_k \sum_{t \in r_k} (Y_t - \mu_k)^2
\]
Maximum likelihood

Most common strategy: Maximum likelihood. To get estimate of the parameters of this model \((T, \mu)\) we choose to maximize the likelihood of the observed data:

\[
p(Y; T, \mu) = \prod_k \prod_{t \in r_k} p(Y_t; \mu_k)
\]

\[
\Rightarrow \log p(Y; T, \mu) = \sum_k \sum_{t \in r_k} \log p(Y_t; \mu_k).
\]

Array CGH. Gaussian with same variance \(\mathcal{N}(\mu_k, \sigma^2)\):

\[
\log P(Y; T, \mu) = \text{cst} - \text{cst} \sum_k \sum_{t \in r_k} (Y_t - \mu_k)^2
\]

NGS. Poisson \(\mathcal{P}(\mu_k)\):

\[
\log P(Y; T, \mu) = \text{cst} - \sum_k \sum_{t \in r_k} (\mu_k - Y_t \log \mu_k)
\]
Parameter inference

When the breakpoints are known, estimating the parameters is (generally) an easy task, e.g. for the mean

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{t \in r_k} Y_t$$
Parameter inference

When the breakpoints are known, estimating the parameters is (generally) an easy task, e.g. for the mean

\[ \hat{\mu}_k = \frac{1}{n_k} \sum_{t \in r_k} Y_t \]

Finding the breakpoints: We now have to find the change-points \( T = (\tau_1, \ldots, \tau_{K-1}) \) that maximize the log-likelihood

\[ \log p(Y; T) = \sum_k \sum_{t \in r_k} \log p(Y_t; \hat{\mu}_k). \]
Parameter inference

When the breakpoints are known, estimating the parameters is (generally) an easy task, e.g. for the mean

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{t \in r_k} Y_t$$

Finding the breakpoints: We now have to find the change-points $T = (\tau_1, \ldots, \tau_{K-1})$ that maximize the log-likelihood

$$\log p(Y; T) = \sum_{k} \sum_{t \in r_k} \log p(Y_t; \hat{\mu}_k).$$

Problem. There are $\binom{n-1}{K-1}$ possible choices for the positions of the breakpoints $\tau_1, \tau_2, \ldots , \tau_{K-1}$.

For $n = 1000, \quad K = 20 \quad \rightarrow \quad \binom{n-1}{K-1} \approx 10^{40}$

→ Impossible to explore for large $n$ and $K$
Shortest path problem

Cost of segment. Define the cost of segment \( r = [i, j] \) as

\[
C(i, j) = \sum_{t \in [i, j]} -\log p(Y_t; \hat{\mu}_{[i, j]}),
\]

finding the maximum likelihood segments can be restated as a shortest path problem, that is to find the path

- going from 1 to \( n \),
- in \( K \) steps,
- for the best possible cost

\[
C(1, \tau_1) + C(\tau_1 + 1, \tau_2) + ... + C(\tau_{K-1} + 1, n).
\]
Shortest path problem

Cost of segment. Define the cost of segment $r = [i, j]$ as

$$C(i, j) = \sum_{t \in [i, j]} -\log p(Y_t; \hat{\mu}_{[i,j]}),$$

finding the maximum likelihood segments can be restated as a shortest path problem, that is to find the path

- going from 1 to $n$,
- in $K$ steps,
- for the best possible cost

$$C(1, \tau_1) + C(\tau_1 + 1, \tau_2) + \ldots + C(\tau_{K-1} + 1, n).$$

Example: Gaussian case (CGHarray).

$$C(i, j) = \sum_{t \in [i, j]} (Y_t - \overline{Y}_{[i,j]})^2.$$
Regular DP algorithm

Principle. *Sup-paths of the optimal paths are optimal themselves.*
Regular DP algorithm

Principle. *Sup-paths of the optimal paths are optimal themselves.*

Algorithm. Denote $S_K(i, j)$ the cost of the optimal segmentation of region $[i, j]$ into $K$ segments:

$$S_2(i, j) = \min_{t \in [i, j-1]} C(i, t) + C(t + 1, j)$$
$$S_K(i, j) = \min_{t \in [i, j-1]} S_{K-1}(i, t) + C(t + 1, j)$$

A second (backward) recursion provides the boundaries of optimal segments.
Regular DP algorithm

**Principle.** *Sup-paths of the optimal paths are optimal themselves.*

**Algorithm.** Denote $S_K(i, j)$ the cost of the optimal segmentation of region $[i, j]$ into $K$ segments:

$$
S_2(i, j) = \min_{t \in [i,j-1]} C(i, t) + C(t + 1, j)
$$

$$
S_K(i, j) = \min_{t \in [i,j-1]} S_{K-1}(i, t) + C(t + 1, j)
$$

A second (backward) recursion provides the boundaries of optimal segments.

**Complexity.**

- $O(Kn^2)$ for computational time and
- $O(Kn)$ for memory space.
Fastening the DP algorithm

Quadratic complexity $O(Kn^2)$ is not acceptable for very large signals ($n = 10^8$).

\(^2\text{Not to compute parameter estimates (see slide 20) before to perform segmentation}
\rightarrow \text{only applicable for 'nice' contrasts}
Fastening the DP algorithm

**Quadratic complexity** $O(Kn^2)$ is not acceptable for very large signals ($n = 10^8$).

The computational time can be reduced to almost linear ($O(Kn \log n)$).

**Pruned DPA:** (Rigaill (2010), Cleynen et al. (2014)) relies on an algebraic trick\(^2\).

**PELT:** (Killick et al. (2011)) takes advantage of the penalty to simplify the contrast.

\(^2\)Not to compute parameter estimates (see slide 20) before to perform segmentation → only applicable for 'nice' contrasts
How many change-points?

How many segments?

- The number of segments $K$ is not known a priori.
- The fit of the segmentation improves as $K$ increases.

Penalized likelihood = most common strategy:
How many change-points?

How many segments?

- The number of segments $K$ is not known a priori.
- The fit of the segmentation improves as $K$ increases.

Penalized likelihood = most common strategy:

$$-\log p(\mathbf{Y}; K)$$
How many change-points?

How many segments?
- The number of segments $K$ is not known a priori.
- The fit of the segmentation improves as $K$ increases.

Penalized likelihood = most common strategy:

$$- \log p(Y; K) + \text{pen}(K)$$
How many change-points?

How many segments?

- The number of segments $K$ is not known a priori.
- The fit of the segmentation improves as $K$ increases.

Penalized likelihood $= \text{most common strategy:}$

$$\hat{K} = \arg \min_K -\log p(Y; K) + \text{pen}(K)$$
How many change-points?

How many segments?

- The number of segments $K$ is not known a priori.
- The fit of the segmentation improves as $K$ increases.

Penalized likelihood = most common strategy:

$$\hat{K} = \arg \min_{K} - \log p(\mathbf{Y} ; K) + \text{pen}(K)$$

An abundant literature has been developed (Lebarbier (2005), Lavielle (2005), Zhang and Siegmund (2007), ...) to insure that $\Pr\{\hat{K} = K\} \to 1$ when $n \to \infty$ or to approximate $p(K|\mathbf{Y})$. 
Penalty 'calibration' requires theoretical developments, each dedicated to a specific model

- **Gaussian homoscedastic** ([Lebarbier (2005)]):
  
  \[ pen(K) = \frac{K}{n} \left( \log \frac{n}{K} + 2.5 \right) \]

- **Negative Binomial** ([Cleynen and Lebarbier (2013)]):
  
  \[ pen(K) = \frac{K}{n} \left( 1 + 4 \sqrt{1.1 + \log \frac{n}{K}} \right)^2 \]
Illustration on RNA-seq

Detection of transcribed regions using RNA-seq. Yeast chromosome 1 Cleynen and Lebarbier (2013)

Poisson model: 106 segments
Illustration on RNA-seq

Detection of transcribed regions using RNA-seq. Yeast chromosome 1 Cleynen and Lebarbier (2013)

Negative binomial model: 103 segments
Comparative study (for CGH)

Based on manually annotated CGH profiles, ROC curves can be drawn in terms of change-point detection (Hocking (2012)).
Comparative study (for CGH)

Based on manually annotated CGH profiles, ROC curves can be drawn in terms of change-point detection (Hocking (2012)).
Comparative study (for CGH)

Based on manually annotated CGH profiles, ROC curves can be drawn in terms of change-point detection (Hocking (2012)).

- DP and PELT clearly achieve the best performances.
- These two methods exactly the (penalized) likelihood maximization.
- But the choice of the number of segments remains an issue.

Comparative study for RNA-seq: Cleynen et al. (2013)
CGH-seg package

CGHseg is an R package dedicated to the analysis of CGH profiles Picard et al. (2011).
CGH-seg package

CGHseg is an R package dedicated to the analysis of CGH profiles Picard et al. (2011).

Linear model framework:

Segmentation regression on unknown $T$: $Y = T\mu + E$

$^3$uniform correlation
CGH-seg package

CGHseg is an R package dedicated to the analysis of CGH profiles Picard et al. (2011).

Linear model framework:

Segmentation  regression on unknown $T$:  

Correlation  factor model$^3$:  

$^3$uniform correlation
CGH-seg package

CGHseg is an R package dedicated to the analysis of CGH profiles Picard et al. (2011).

Linear model framework:

Segmentation regression on unknown $T$: $Y = T\mu + E$

Correlation factor model(3): $Z$ $Y = T\mu + BZ + E$

Correction fixed covariates effect: $\beta$ $Y = T\mu + X\beta + E$

---

$^3$uniform correlation
CGH-seg package

CGHseg is an R package dedicated to the analysis of CGH profiles Picard et al. (2011).

Linear model framework:

**Segmentation**

regression on unknown $T$:

$$Y = T\mu + E$$

**Correlation**

factor model:\n
$$Y = T\mu + BZ + E$$

**Correction**

fixed covariates effect: $\beta$

$$Y = T\mu + X\beta + E$$

**Calling**

cross-tabulation table $C$:

$$Y = TCm + E$$

---

$^3$uniform correlation
# CGH-seg package

CGHseg is an R package dedicated to the analysis of CGH profiles Picard *et al.* (2011).

## Linear model framework:

<table>
<thead>
<tr>
<th>Method</th>
<th>Regression Model</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Segmentation</strong></td>
<td>Segmentation regression on unknown $T$: $Y = T\mu + E$</td>
</tr>
<tr>
<td><strong>Correlation</strong></td>
<td>Correlation factor model $(3)$: $Z$: $Y = T\mu + BZ + E$</td>
</tr>
<tr>
<td><strong>Correction</strong></td>
<td>Fixed covariates effect: $\beta$: $Y = T\mu + X\beta + E$</td>
</tr>
<tr>
<td><strong>Calling</strong></td>
<td>Cross-tabulation table $C$: $Y = TCm + E$</td>
</tr>
<tr>
<td><strong>Combinations</strong></td>
<td>Fixed + random effects: $Y = T\mu + X\beta + LZ + E$</td>
</tr>
<tr>
<td></td>
<td>Fixed effects + calling: $Y = TCm + X\beta + E$</td>
</tr>
</tbody>
</table>

---

$^{3}$uniform correlation
Another segmentation problem

HiC technology reveals physical interaction between genomic loci and provides an interaction map.

Cis-interacting regions can be detected using a segmentation approach.

(Apparently 2D problem turns out to be 1D)

Lévy-Leduc et al. (2014) + HiCseg R package
A bit further: comparing change-points

Change-point location. No method described until now gives information about the precision of the change-point location $\tau_k$ (confidence or credibility interval).
Change-point location. No method described until now gives information about the precision of the change-point location $\tau_k$ (confidence or credibility interval).

Credibility interval. A quadratic ($O(Kn^2)$) algorithm can be designed to compute the (exact) posterior distribution of the change-points Rigaill et al. (2011).
Change-point comparison

Transcription boundaries in 3 different conditions for a yeast gene with 2 exons

Conditions $A/B/C$
Change-point comparison

Transcription boundaries in 3 different conditions for a yeast gene with 2 exons

Comparison

- Posterior distribution under different conditions can be compared.
- Providing a statistical assessment of transcription starting sites variations.

Cleynen and Robin (2014)
+ EBS R package
'Systematic' study in yeast

Posterior probability for the boundary to be conserved for each of the 50 genes with exons.

Inner boundaries are more conserved across conditions that extreme boundaries → Alternative transcription of UTRs?
Hidden Markov model
Back to the the basic problem

We wanted to go from there
Back to the basic problem

We wanted to go from there

... to there.

Methods presented until now provide estimates of the change-point location and of the mean signal in each segment.
Back to the the basic problem

We wanted to go from there
... to there.

Methods presented until now provide estimates of the change-point location and of the mean signal in each segment.

But we still miss the text, that is, the 'calling'.
Another model for the same problem

- \( t = 1..n \) probes (positions) are observed.
Another model for the same problem

- $t = 1..n$ probes (positions) are observed.
- An unobserved label $Z_t$ (‘loss’, ‘normal’, ‘gain’) is associated with each probe;
Another model for the same problem

- $t = 1..n$ probes (positions) are observed.
- An unobserved label $Z_t$ (‘loss’, ‘normal’, ‘gain’) is associated with each probe;
- The distribution of the observed signal $Y_t$ depends on the label.
Another model for the same problem

- $t = 1..n$ probes (positions) are observed.
- An unobserved label $Z_t$ ('loss', 'normal', 'gain') is associated with each probe;
- The distribution of the observed signal $Y_t$ depends on the label.
- We only observe the signal;
Another model for the same problem

- $t = 1..n$ probes (positions) are observed.
- An unobserved label $Z_t$ (‘loss’, 'normal', 'gain') is associated with each probe;
- The distribution of the observed signal $Y_t$ depends on the label.
- We only observe the signal;
- And we would like to retrieve the 'truth'.
Another model for the same problem

- $t = 1..n$ probes (positions) are observed.
- An unobserved label $Z_t$ (‘loss’, ‘normal’, ‘gain’) is associated with each probe;
- The distribution of the observed signal $Y_t$ depends on the label.
- We only observe the signal;
- And we would like to retrieve the ‘truth’.

(Although we know we won’t succeed.)
Hidden labels. \((Z_t)\) is a Markov chain:

- \(Z_t \in \{1..Q\}\), e.g. \(Q = 3\) for \{'loss', 'normal', 'gain'\};
- The label of a probe depends on the label of the preceding probe:

  \[
  \pi_{q\ell} = \Pr\{Z_t = \ell | Z_{t-1} = q\}
  \]

Graphical model representation.
### Statistical model

**Hidden labels.** \((Z_t)\) is a Markov chain:
- \(Z_t \in \{1..Q\}\), e.g. \(Q = 3\) for \{'loss', 'normal', 'gain'\};
- The label of a probe depends on the label of the preceding probe:

\[
\pi_{q\ell} = \Pr\{Z_t = \ell | Z_{t-1} = q\}
\]

**Observed signal.** \((Y_t)\) are independent, given the labels \((Z_t)\):

if \(Z_t = q\) : \(Y_t \sim \mathcal{N}(\mu_q, \sigma^2(q))\) or \(Y_t \sim \mathcal{P}(\mu_q)\) or anything?

**Graphical model representation.**

![Graphical Model](image-url)
Incomplete data model. Parameter inference would be easy if the labels were known.
Parameter inference

Incomplete data model. Parameter inference would be easy if the labels were known.

E-M algorithm. The most common strategy consists in retrieving the missing information.

- **E-step:** Compute (in linear time) the probability for each probe $t$ to have label $q$ (forward-backward algorithm):

\[
\tau_{tq} = \Pr\{Z_t = q \mid Y\};
\]
Parameter inference

Incomplete data model. Parameter inference would be easy if the labels were known.

E-M algorithm. The most common strategy consists in retrieving the missing information.

- **E-step:** Compute (in linear time) the probability for each probe $t$ to have label $q$ (forward-backward algorithm):
  \[
  \tau_{tq} = \Pr\{Z_t = q \mid Y\};
  \]

- **M-step:** Estimate the parameters based on the inferred labels, e.g.
  \[
  \hat{\mu}_q = \frac{\sum_t \tau_{tq} Y_t}{\sum_t \tau_{tq}}.
  \]

Classification $=$ ‘Calling’

Probe classification. Segmentation is finally obtained by assigning a label to each probe.
Classification = 'Calling'

Probe classification. Segmentation is finally obtained by assigning a label to each probe.

MAP rule. The $\tau_{tq}$ can provide maximum a posteriori classification:

$$\hat{Z}_t = \arg \max_q \tau_{tq}.$$
Classification = 'Calling'

Probe classification. Segmentation is finally obtained by assigning a label to each probe.

MAP rule. The $\tau_{tq}$ can provide maximum a posteriori classification:

$$\hat{Z}_t = \arg \max_q \tau_{tq}.$$  

Most probable hidden path. The succession of the MAP ($\hat{Z}_t$) is not the most probable hidden path ('Viterbi path'):

$$\hat{Z} = \arg \max_Z p(Z|Y) \neq (\hat{Z}_t)$$
Emission distributions

The choice of $\phi_k$ (Poisson, neg. binomial, ...) is indeed critical.

Identifiability. HMM are identifiable under mild conditions Gassiat et al. (2014) → non-parametric distributions $p_k(y)$ can be considered.

Simulation. Resampling RNA-seq data to define regions with no/weak/middle/strong expression.
Emission distributions

The choice of $\phi_k$ (Poisson, neg. binomial, ...) is indeed critical.

Identifiability. HMM are identifiable under mild conditions Gassiat et al. (2014) → non-parametric distributions $p_k(y)$ can be considered.

Simulation. Resampling RNA-seq data to define regions with no/weak/middle/strong expression.

Distribution fit

non-parm / neg. bin.
**Emission distributions**

The choice of $\phi_k$ (Poisson, neg. binomial, ...) is indeed critical.

**Identifiability.** HMM are identifiable under mild conditions *Gassiat et al. (2014)*
→ non-parametric distributions $p_k(y)$ can be considered.

**Simulation.** Resampling RNA-seq data to define regions with no/weak/middle/strong expression.
Issues

Algorithmics.

- The conditional distribution of the labels given the observation $p(Z|Y)$ is not explicit;
- But it can be computed in a linear time $O(nK^2)$ with the forward-backward algorithm.
Hidden Markov model

Issues

Algorithmics.
- The conditional distribution of the labels given the observation $p(Z|Y)$ is not explicit;
- But it can be computed in a linear time $O(nK^2)$ with the forward-backward algorithm.

Statistics.
- The E-M algorithm aims at computing the maximum-likelihood estimates;
- But its behavior strongly depends on the starting point...
- The Markov assumption has some consequences on the distribution of the length of the segments.
Issues

Algorithmics.
- The conditional distribution of the labels given the observation $p(Z|Y)$ is not explicit;
- But it can be computed in a linear time $O(nK^2)$ with the forward-backward algorithm.

Statistics.
- The E-M algorithm aims at computing the maximum-likelihood estimates;
- But its behavior strongly depends on the starting point;
- The Markov assumption has some consequences on the distribution of the length of the segments.

Model selection.
- The number of hidden classes $Q$ is unknown;
- But is often chosen with standard criteria, such as BIC.
Regularized approaches
An optimization perspective

Segmentation problem can be rephrased as

\[ \hat{\mu}_{\text{Seg}} = \min_{\mu} \sum_{t} (Y_t - \mu_t)^2 + \lambda \sum_{t} |\mu_t - \mu_{t-1}|_0 \]

where \( | \cdot |_0 \) stands for the so-called \( \ell_0 \) norm:

\[ \sum_{t} |\mu_t - \mu_{t-1}|_0 = \text{number of changes in } \mu_t = K - 1 \]

and \( \lambda \) controls the number of segments \( K \).
An optimization perspective

Segmentation problem can be rephrased as

$$\hat{\mu}_{\text{Seg}} = \min_\mu \sum_t (Y_t - \mu_t)^2 + \lambda \sum_t |\mu_t - \mu_{t-1}|_0$$

where $| \cdot |_0$ stands for the so-called $\ell_0$ norm:

$$\sum_t |\mu_t - \mu_{t-1}|_0 = \text{number of changes in } \mu_t = K - 1$$

and $\lambda$ controls the number of segments $K$.

Remarks.

- This optimization problem turns out to be non convex.
- This prevents the use the huge set of convex optimization techniques.
- Hopefully, dynamic programming does exist.
Lasso

Lasso trick. Replacing the $\ell_0$ with the $\ell_1$ norm makes the optimization problem convex:

$$\hat{\mu}_{\text{Lasso}} = \min_{\mu} \sum_t (Y_t - \mu_t)^2 + \lambda \sum_t |\mu_t|$$

that can be solved in linear time.
Lasso

Lasso trick. Replacing the $\ell_0$ with the $\ell_1$ norm makes the optimization problem convex:

$$\hat{\mu}_{\text{Lasso}} = \min_{\mu} \sum_t (Y_t - \mu_t)^2 + \lambda \sum_t |\mu_t|$$

that can be solved in linear time.

Regularization:
Lasso

Lasso trick. Replacing the $\ell_0$ with the $\ell_1$ norm makes the optimization problem convex:

$$\hat{\mu}_{\text{Lasso}} = \min_\mu \sum_t (Y_t - \mu_t)^2 + \lambda \sum_t |\mu_t|$$

that can be solved in linear time.

Regularization:

- Due to the $\ell_1$ topology, most of the parameters $\mu_t$ are set to 0.
Lasso trick. Replacing the $\ell_0$ with the $\ell_1$ norm makes the optimization problem convex:

$$\hat{\mu}_{\text{Lasso}} = \min_{\mu} \sum_t (Y_t - \mu_t)^2 + \lambda \sum_t |\mu_t|$$

that can be solved in linear time.

Regularization:

- Due to the $\ell_1$ topology, most of the parameters $\mu_t$ are set to 0.
- The coefficient $\lambda$ controls the number of non-zero terms.
Fused Lasso

Fused Lasso. For a segmentation purpose, a second term penalizing changes among the $\mu_t$ can be added (Tibshirani et al. (2005)):

$$\hat{\mu} = \min_{\mu} \sum_t (Y_t - \mu_t)^2 + \lambda_1 \sum_t |\mu_t| + \lambda_2 \sum_t |\mu_t - \mu_{t-1}|$$

where $\sum_t |\mu_t - \mu_{t-1}| = \text{sum of the absolute changes in } \mu_t$. 
Fused Lasso

Fused Lasso. For a segmentation purpose, a second term penalizing changes among the $\mu_t$ can be added (Tibshirani et al. (2005)):

$$\hat{\mu} = \min_{\mu} \sum_t (Y_t - \mu_t)^2 + \lambda_1 \sum_t |\mu_t| + \lambda_2 \sum_t |\mu_t - \mu_{t-1}|$$

where $\sum_t |\mu_t - \mu_{t-1}|_1 = \text{sum of the absolute changes in } \mu_t$.

Interpretation.

- $\sum_t (Y_t - \mu_t)^2$: fit to the observations;
- $\lambda_1 \sum_t |\mu_t|$: controls the number of 'abnormal' positions;
- $\lambda_2 \sum_t |\mu_t - \mu_{t-1}|$: controls the number of breakpoints.
Application to CGH

Effect of $w = 2\lambda_1/\lambda_2$. Tibshirani and Wang (2008)
Application to CGH

Effect of $w = 2\lambda_1/\lambda_2$. Tibshirani and Wang (2008)
Application to CGH

Effect of $w = 2\lambda_1 / \lambda_2$. Tibshirani and Wang (2008)
R packages.

**CGHseg**: Comprehensive package for joint segmentation of multiple profile (+ between profiles correlation + bias correction + calling)

**EBS**: Exact posterior distribution for segmentation (model selection, credibility intervals, ...) for both microarray and NGS

**Segmentator3IsBack**: Pruned dynamic programming for efficient segmentation of large signals (microarray, NGS)

**HiCseg**: Segmentation method for the analysis of HiC data (cis-interaction)
To end

R packages.

**CGHseg**: Comprehensive package for joint segmentation of multiple profile (+ between profiles correlation + bias correction + calling)

**EBS**: Exact posterior distribution for segmentation (model selection, credibility intervals, ...) for both microarray and NGS

**Segmentator3IsBack**: Pruned dynamic programming for efficient segmentation of large signals (microarray, NGS)

**HiCseg**: Segmentation method for the analysis of HiC data (cis-interaction)

**Acknowledgements.** A. Cleynen, E. Gassiat, S. Huet, F. Picard, G. Rigaill, 'Statistics & Genome’ group at INRA/AgroParisTech


Data transformation

Cleynen, 2012