On segmentation of DNA copy number profiles

Jean-Philippe Vert

(joint work with Toby Hocking, Gudrun Schleiermacher, Isabelle Janoueix-Lerosey and Francis Bach)

13th International Workshop on Bioinformatics and Systems Biology (IBSB 2013), Kyoto University, August 1st, 2013
Chromosomal aberrations in cancer
Array Comparative Genomic Hybridization (aCGH)

Motivation

- Comparative genomic hybridization (CGH) data measure the DNA copy number along the genome.
- Very useful, in particular in cancer research to observe systematically variants in DNA content.

![Graph showing log-ratio vs chromosome number]
Neuroblastoma

- Rare, but most common cancer in infants
- Arises from nervous cells, frequent metastasis
- One of the few human malignancies known to demonstrate spontaneous regression
aCGH neuroblastoma copy number data
Copy number profiles are predictive of progression in neuroblastoma

2 types of profiles:

- **Numerical**: entire chromosome amplification. *Good* outcome.
- **Segmental**: deletion 1p 3p 11q, gain 1q 2p 17q. *Bad* outcome. In this talk “breakpoints.”

*(Schleiermacher et al., J Clinical Oncology, 2010)*
- Refine classification of neuroblastoma in terms of breakpoints
- Refine prognosis based on breakpoints
- Predict metastatic locations from breakpoints

We need to automatically identify breakpoints
In this talk: How to automatically identify breakpoints?

1. Learning smoothing models using expert annotation
2. Optimizing multi-parameter models
3. Fast and scalable segmentation
In this talk: How to automatically identify breakpoints?

1. Learning smoothing models using expert annotation
2. Optimizing multi-parameter models
3. Fast and scalable segmentation
Outline

1. Learning smoothing models using expert annotation
2. Optimizing multi-parameter models
3. Fast and scalable segmentation
Many models proposed to detect breakpoints:

- **GLAD**: adaptive weights smoothing (Hupé et al., 2004)
- **DNAcopy**: circular binary segmentation (Venkatraman and Olshen, 2007)
- **cghFLasso**: fused lasso signal approximator with heuristics (Tibshirani and Wang, 2007)
- **HaarSeg**: wavelet smoothing (Ben-Yaacov and Eldar, 2008)
- **flsa**: fused lasso signal approximator path algorithm (Hoefling, 2009)
- **cghseg** (Rigaill 2010) and **PELT** (Killick et al. 2012): pruned dynamic programming
- **gada**: Sparse representation and Bayesian detection (Pique-Regi et al, 2008)

Each model has a parameter to tune the degree of smoothness, and often a default parameter.

1. How to define which model is best?
2. And how to choose the degree of smoothness?
Our answer: Ask an expert!

Answer: annotate the profiles!
SegAnnDB for easy and fast partial expert annotation

https://gforge.inria.fr/scm/viewvc.php/webapp/?root=breakpoints
Table 1 Counts of annotations in two annotation data sets of the same copy number profiles

<table>
<thead>
<tr>
<th>Annotation</th>
<th>Original</th>
<th>Detailed</th>
</tr>
</thead>
<tbody>
<tr>
<td>protocol</td>
<td>Systematic</td>
<td>Any</td>
</tr>
<tr>
<td>annotated profiles</td>
<td>575</td>
<td>575</td>
</tr>
<tr>
<td>annotated chromosomes</td>
<td>3418</td>
<td>3730</td>
</tr>
<tr>
<td>annotations</td>
<td>3418</td>
<td>4359</td>
</tr>
<tr>
<td>&gt;0breakpoints</td>
<td>2845</td>
<td>3395</td>
</tr>
<tr>
<td>1breakpoint</td>
<td>0</td>
<td>521</td>
</tr>
<tr>
<td>&gt;0breakpoints</td>
<td>573</td>
<td>443</td>
</tr>
</tbody>
</table>
Testing a model: Mostly over-segmented

cghseg.k smoothing for lambda=10^{-3.48}

breakpoint

annotation
red breakpoint
gray normal

position on chromosome (mega base pairs)
Testing a model: Mostly under-segmented

cghseg.k smoothing for lambda=10^−1.58

- Red: breakpoint
- Gray: normal

Position on chromosome (mega base pairs):

100 200 100 200 100 200 100 200 100 200 100 200 100 200
Testing a model: Not too bad

cghseg.k smoothing for lambda=10^{-1.96}
Choose the smoothing model that minimizes error with respect to breakpoint annotations

number of annotations inconsistent with model smoothing

log10(smoothness parameter)
Global error = same parameter for all profiles
Local error = parameter optimized for each profile

<table>
<thead>
<tr>
<th>cghseg.k, pelt.n</th>
<th>flsa.norm</th>
<th>dnacopy.sd</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2</td>
<td>4.8</td>
<td>11.5</td>
</tr>
</tbody>
</table>

percent incorrectly predicted annotations in training set

log10(smoothing parameter)

fewer breakpoints →

JP Vert (ParisTech)
DNA segmentation
IBSB 2013
Global error of 17 segmentation methods

- Optimization-based models
- Approximate optimization
- Glad

- True positive rate = probability(predict breakpoint | at least 1 breakpoint)
- False positive rate = probability(predict too many breakpoints | 0 or 1 breakpoint)
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Original Error</th>
<th>FP</th>
<th>FN</th>
<th>Error</th>
<th>FP</th>
<th>FN</th>
</tr>
</thead>
<tbody>
<tr>
<td>pelt.n</td>
<td>0.3</td>
<td>0.7</td>
<td>0.2</td>
<td>2.1</td>
<td>5.3</td>
<td>1.0</td>
</tr>
<tr>
<td>cghseg.k</td>
<td>0.3</td>
<td>0.7</td>
<td>0.2</td>
<td>2.3</td>
<td>5.6</td>
<td>1.1</td>
</tr>
<tr>
<td>gada</td>
<td>0.6</td>
<td>1.6</td>
<td>0.5</td>
<td>2.5</td>
<td>6.3</td>
<td>1.2</td>
</tr>
<tr>
<td>dnacopy.sd</td>
<td>2.5</td>
<td>7.5</td>
<td>1.5</td>
<td>5.1</td>
<td>14.1</td>
<td>2.2</td>
</tr>
<tr>
<td>glad.lambdabreak</td>
<td>6.4</td>
<td>2.1</td>
<td>7.3</td>
<td>8.0</td>
<td>7.1</td>
<td>7.2</td>
</tr>
<tr>
<td>flsa.norm</td>
<td>1.2</td>
<td>3.0</td>
<td>0.8</td>
<td>8.7</td>
<td>19.5</td>
<td>4.9</td>
</tr>
<tr>
<td>flsa</td>
<td>1.3</td>
<td>1.4</td>
<td>1.3</td>
<td>8.9</td>
<td>20.6</td>
<td>4.9</td>
</tr>
<tr>
<td>glad.haarseg</td>
<td>9.0</td>
<td>1.6</td>
<td>10.5</td>
<td>9.5</td>
<td>6.0</td>
<td>9.0</td>
</tr>
<tr>
<td>pelt.default</td>
<td>8.0</td>
<td>42.2</td>
<td>1.1</td>
<td>13.9</td>
<td>59.0</td>
<td>1.0</td>
</tr>
<tr>
<td>dnacopy.alpha</td>
<td>17.9</td>
<td>1.4</td>
<td>21.2</td>
<td>16.8</td>
<td>7.2</td>
<td>16.9</td>
</tr>
<tr>
<td>glad.MinBkpWeight</td>
<td>19.7</td>
<td>0.7</td>
<td>23.6</td>
<td>18.4</td>
<td>4.6</td>
<td>19.4</td>
</tr>
<tr>
<td>dnacopy.prune</td>
<td>25.9</td>
<td>2.8</td>
<td>30.5</td>
<td>23.6</td>
<td>8.9</td>
<td>24.1</td>
</tr>
<tr>
<td>glad.default</td>
<td>27.4</td>
<td>1.6</td>
<td>32.7</td>
<td>26.0</td>
<td>5.0</td>
<td>27.7</td>
</tr>
<tr>
<td>dnacopy.default</td>
<td>40.5</td>
<td>0.7</td>
<td>48.5</td>
<td>38.0</td>
<td>4.8</td>
<td>41.1</td>
</tr>
<tr>
<td>cghseg.mBIC</td>
<td>41.0</td>
<td>0.0</td>
<td>49.2</td>
<td>38.5</td>
<td>2.0</td>
<td>42.3</td>
</tr>
<tr>
<td>gada.default</td>
<td>80.7</td>
<td>0.0</td>
<td>96.9</td>
<td>82.7</td>
<td>0.1</td>
<td>92.1</td>
</tr>
<tr>
<td>cghFLasso</td>
<td>80.9</td>
<td>0.0</td>
<td>97.2</td>
<td>83.8</td>
<td>0.8</td>
<td>93.1</td>
</tr>
</tbody>
</table>
Leave-one-out error: Global models are more robust

<table>
<thead>
<tr>
<th>annotations: original</th>
<th>annotations: detailed</th>
</tr>
</thead>
<tbody>
<tr>
<td>pelt.n</td>
<td>pelt.default</td>
</tr>
<tr>
<td>cghseg.k</td>
<td>gada</td>
</tr>
<tr>
<td>flsa.norm</td>
<td>pelt.default</td>
</tr>
<tr>
<td>gada</td>
<td>glad.haarseg</td>
</tr>
<tr>
<td>pelt.default</td>
<td>flsa</td>
</tr>
<tr>
<td>glad.haarseg</td>
<td>dnacopy.sd</td>
</tr>
<tr>
<td>flsa</td>
<td>glad.lambdabreak</td>
</tr>
<tr>
<td>dnacopy.sd</td>
<td>dnacopy.alpha</td>
</tr>
<tr>
<td>glad.lambdabreak</td>
<td>glad.MinBkpWeight</td>
</tr>
<tr>
<td>dnacopy.alpha</td>
<td>glad.default</td>
</tr>
<tr>
<td>glad.MinBkpWeight</td>
<td>dnacopy.prune</td>
</tr>
<tr>
<td>glad.default</td>
<td>dnacopy.default</td>
</tr>
<tr>
<td>dnacopy.prune</td>
<td>cghegse.mBIC</td>
</tr>
<tr>
<td>dnacopy.default</td>
<td>gada.default</td>
</tr>
<tr>
<td>cghegse.mBIC</td>
<td>cgheFLasso</td>
</tr>
</tbody>
</table>

Leave-one-out test error (mean ± 1 standard deviation)

algorithm

model
- global
- local
Globally-optimized parameters are better than default parameters on new profiles.

- >0breakpoints
- 1breakpoint
- 0breakpoints
Only 10 annotated samples are sufficient to learn parameters, which are robust across annotators.
Generalization error for models trained on 10 profiles

<table>
<thead>
<tr>
<th>algorithm</th>
<th>error</th>
<th>sd</th>
<th>fn</th>
<th>sd</th>
<th>fp</th>
<th>sd</th>
<th>Timings</th>
</tr>
</thead>
<tbody>
<tr>
<td>pelt.n</td>
<td>7.7</td>
<td>1.8</td>
<td>17.9</td>
<td>9.8</td>
<td>4.1</td>
<td>3.4</td>
<td>9.49</td>
</tr>
<tr>
<td>cghseg.k</td>
<td>7.8</td>
<td>1.8</td>
<td>17.8</td>
<td>9.7</td>
<td>4.3</td>
<td>3.2</td>
<td>2.79</td>
</tr>
<tr>
<td>gada</td>
<td>9.5</td>
<td>1.5</td>
<td>28.2</td>
<td>12.6</td>
<td>3.6</td>
<td>2.8</td>
<td>7.54</td>
</tr>
<tr>
<td>glad.haarseg</td>
<td>13.2</td>
<td>1.4</td>
<td>12.2</td>
<td>1.2</td>
<td>11.7</td>
<td>1.8</td>
<td>32.62</td>
</tr>
<tr>
<td>pelt.default</td>
<td>13.9</td>
<td>0.1</td>
<td>59.0</td>
<td>0.3</td>
<td>1.0</td>
<td>0.0</td>
<td>0.08</td>
</tr>
<tr>
<td>flsa.norm</td>
<td>14.6</td>
<td>1.3</td>
<td>39.3</td>
<td>13.1</td>
<td>6.5</td>
<td>3.6</td>
<td>0.12</td>
</tr>
<tr>
<td>dnacopy.sd</td>
<td>15.8</td>
<td>2.9</td>
<td>42.8</td>
<td>24.2</td>
<td>7.1</td>
<td>5.6</td>
<td>61.90</td>
</tr>
<tr>
<td>glad.lambdabreak</td>
<td>17.4</td>
<td>1.9</td>
<td>25.4</td>
<td>15.9</td>
<td>13.1</td>
<td>4.4</td>
<td>17.02</td>
</tr>
<tr>
<td>dnacopy.alpha</td>
<td>17.8</td>
<td>0.8</td>
<td>8.1</td>
<td>0.2</td>
<td>17.8</td>
<td>0.9</td>
<td>29.38</td>
</tr>
<tr>
<td>flsa</td>
<td>20.1</td>
<td>1.2</td>
<td>56.2</td>
<td>25.6</td>
<td>8.5</td>
<td>5.8</td>
<td>0.06</td>
</tr>
<tr>
<td>glad.MinBkpWeight</td>
<td>25.5</td>
<td>1.0</td>
<td>7.8</td>
<td>3.0</td>
<td>26.5</td>
<td>1.4</td>
<td>42.39</td>
</tr>
<tr>
<td>glad.default</td>
<td>26.0</td>
<td>0.1</td>
<td>5.0</td>
<td>0.2</td>
<td>27.7</td>
<td>0.1</td>
<td>1.34</td>
</tr>
<tr>
<td>dnacopy.prune</td>
<td>26.7</td>
<td>1.0</td>
<td>19.5</td>
<td>4.8</td>
<td>24.9</td>
<td>2.0</td>
<td>41.34</td>
</tr>
<tr>
<td>dnacopy.default</td>
<td>38.0</td>
<td>0.2</td>
<td>4.8</td>
<td>0.1</td>
<td>41.1</td>
<td>0.2</td>
<td>2.02</td>
</tr>
<tr>
<td>cghseg.mBIC</td>
<td>38.5</td>
<td>0.1</td>
<td>2.0</td>
<td>0.1</td>
<td>42.3</td>
<td>0.1</td>
<td>1.81</td>
</tr>
<tr>
<td>gada.default</td>
<td>82.7</td>
<td>0.1</td>
<td>0.1</td>
<td>0.0</td>
<td>92.1</td>
<td>0.1</td>
<td>0.20</td>
</tr>
<tr>
<td>cghFLasso</td>
<td>83.8</td>
<td>0.1</td>
<td>0.8</td>
<td>0.1</td>
<td>93.1</td>
<td>0.1</td>
<td>0.18</td>
</tr>
</tbody>
</table>
Best model are cghseg.k and pelt.n: implement a Gaussian maximum-likelihood piecewise constant smoothing model:

$$\min_{k, \mu^k} \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu_i)^2 + \lambda k$$

where $\mu^k$ has at most $k$ change-points

$\lambda$ is optimized on 10 expert-annotated profiles.

Better than default parameters

Robust across annotators

Outline

1. Learning smoothing models using expert annotation

2. Optimizing multi-parameter models

3. Fast and scalable segmentation
The model is:

$$\min_{k, \mu^k} \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu_i)^2 + \lambda k$$

where $\mu^k$ has at most $k$ change-points.

$\lambda = 1$ by default, but better to use $\lambda$ optimized to maximize agreement with a database of breakpoint annotations.

Why this particular penalty? What about taking into account other properties of the signal, such as its length or variance?
Error curve for 2 annotated signals, $L = \log(\lambda)$

- Signal $i = 4.3$
- Signal $i = 10.1$

Segments $z^*_i(L)$

Error $E_i(L)$

Penalty exponent $L$
Target intervals $[L_i, \bar{L}_i]$, as a function of estimated variance for 2 signals
Target interval \([L_i, \overline{L}_i]\) for all signals

![Graph showing the relationship between variance estimate \(\log \hat{s}_i\) and penalty exponent \(L\). The graph has two sets of points, one for \(\overline{L}_i\) and the other for \(L_i\).]

Limit

- \(\overline{L}_i\)
- \(L_i\)
Limit point representation

penalty exponent $L$

variance estimate $\log \hat{s}_i$

limit

$\bar{L}_i$

$\hat{L}_i$
Max margin regression line

\begin{align*}
\text{variance estimate } \log \hat{s}_i \\
\text{penalty exponent } L
\end{align*}
Learning the penalty function

\[
\min_{k,\mu^k} \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu_i)^2 + \lambda(\hat{\sigma})k
\]

- For every signal \( i \), estimate the variance \( \hat{\sigma}_i \)
- Parametrize the penalty as
  \[
  \log \lambda_i = \beta + w \log \hat{\sigma}_i
  \]
- Equivalent to learning a penalty function

\[
\min_{k,\mu^k} \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu_i)^2 + e^{\beta \hat{\sigma}^w} k
\]
Real data can usually not be separated...
Convex surrogate loss for the annotation error

Error and relaxation for 4 annotated signals

penalty exponent \( L \)

error/loss
curve
annotation error \( E_i(L) \)
surrogate loss \( l_i(L) \)
Learning the parameters

\[
\arg\min_{\beta, w} \frac{1}{N} \sum_{i=1}^{N} \ell_i(\beta + w^T x_i) + \gamma \| w \|_1
\]

coefficients

percent ann. error

Model complexity \(\log_{10}(\gamma)\)

-log \(\hat{s}_i\)
-log \(\hat{d}_i\)
-log \(\hat{b}_i\)
-\(\hat{\beta}\)

train
test

DNA segmentation
Optimal model complexity depends on variance
Optimal model complexity depends on variance

![Diagram showing the relationship between penalty exponent L and log(variance estimate) for original and detailed low density DNA segmentation data. The graph includes points marked as min and max, with a green line indicating a limit.]
Optimal model complexity depends on number of points.
Optimal model complexity depends on number of points.
Learned coefficients

Model with two features:

- Variance estimate log $\hat{s}_i$.
- Number of points sampled log $d_i$.
- Learned model complexity

\[ f(x_i) = w_1 \log \hat{s}_i + w_2 \log d_i + \beta. \]

- Learned penalty function

\[ z_i^*[f(x_i)] = \arg \min_k ||y_i - \hat{y}_i^k||_2^2 + \hat{s}_i^{w_1} d_i^{w_2} e^\beta k. \]

<table>
<thead>
<tr>
<th>Annotation data set</th>
<th>Variance $w_1$</th>
<th>Points $w_2$</th>
<th>Intercept $\beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>original</td>
<td>1.01</td>
<td>0.96</td>
<td>-2.66</td>
</tr>
<tr>
<td></td>
<td>±0.03</td>
<td>±0.02</td>
<td>±0.10</td>
</tr>
<tr>
<td>detailed.low.density</td>
<td>1.30</td>
<td>0.93</td>
<td>-2.00</td>
</tr>
<tr>
<td></td>
<td>±0.02</td>
<td>±0.02</td>
<td>±0.13</td>
</tr>
</tbody>
</table>

Mean ± 1 standard deviation over 10 folds.
Error estimated using 10-fold cross-validation

\[ f(x_i) = w_1 \log \hat{s}_i + w_2 \log d_i + \beta \]

- \text{cghseg.k: } w_1 = 0, w_2 = 1, learn \beta \text{ using grid search to minimize the annotation error } E_i.
- \text{log.s.log.d: } learn \beta, w_1, w_2 \text{ by minimizing the un-regularized } (\gamma = 0) \text{ surrogate loss } l_i.
- \text{L1-reg: variance estimate, signal size, model error, chromosome indicator features } x_i \in \mathbb{R}^{117}, \text{ CV to choose the degree of } \ell_1 \text{ regularization } \gamma.

<table>
<thead>
<tr>
<th>model</th>
<th>variables</th>
<th>original</th>
<th>detailed.low.density</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIC</td>
<td>0</td>
<td>7.99 ± 0.00</td>
<td>13.64 ± 0.00</td>
</tr>
<tr>
<td>mBIC</td>
<td>0</td>
<td>40.99 ± 0.00</td>
<td>36.88 ± 0.00</td>
</tr>
<tr>
<td>cghseg.k</td>
<td>0</td>
<td>2.19 ± 0.82</td>
<td>6.49 ± 1.16</td>
</tr>
<tr>
<td>log.s.log.d</td>
<td>2</td>
<td>1.90 ± 0.77</td>
<td>4.72 ± 0.54</td>
</tr>
<tr>
<td>L1-reg</td>
<td>117</td>
<td>1.81 ± 0.58</td>
<td>4.70 ± 0.88</td>
</tr>
</tbody>
</table>
Complex penalties with multiple parameters can be optimized
Equivalent to interval regression problem, which we solve with convex optimization
Optimized penalties are better than default or simple penalties
Outline

1. Learning smoothing models using expert annotation

2. Optimizing multi-parameter models

3. Fast and scalable segmentation
How to scale to \( p = 10^7 \sim 10^9 \)?

<table>
<thead>
<tr>
<th>algorithm</th>
<th>error</th>
<th>sd</th>
<th>fn</th>
<th>sd</th>
<th>fp</th>
<th>sd</th>
<th>Timings</th>
</tr>
</thead>
<tbody>
<tr>
<td>pelt.n</td>
<td>7.7</td>
<td>1.8</td>
<td>17.9</td>
<td>9.8</td>
<td>4.1</td>
<td>3.4</td>
<td>9.49</td>
</tr>
<tr>
<td>cghseg.k</td>
<td>7.8</td>
<td>1.8</td>
<td>17.8</td>
<td>9.7</td>
<td>4.3</td>
<td>3.2</td>
<td>2.79</td>
</tr>
<tr>
<td>gada</td>
<td>9.5</td>
<td>1.5</td>
<td>28.2</td>
<td>12.6</td>
<td>3.6</td>
<td>2.8</td>
<td>7.54</td>
</tr>
<tr>
<td>glad.haarseg</td>
<td>13.2</td>
<td>1.4</td>
<td>12.2</td>
<td>1.2</td>
<td>11.7</td>
<td>1.8</td>
<td>32.62</td>
</tr>
<tr>
<td>pelt.default</td>
<td>13.9</td>
<td>0.1</td>
<td>59.0</td>
<td>0.3</td>
<td>1.0</td>
<td>0.0</td>
<td>0.08</td>
</tr>
<tr>
<td>flsa.norm</td>
<td>14.6</td>
<td>1.3</td>
<td>39.3</td>
<td>13.1</td>
<td>6.5</td>
<td>3.6</td>
<td>0.12</td>
</tr>
<tr>
<td>dnacopy.sd</td>
<td>15.8</td>
<td>2.9</td>
<td>42.8</td>
<td>24.2</td>
<td>7.1</td>
<td>5.6</td>
<td>61.90</td>
</tr>
<tr>
<td>glad.lambdabreak</td>
<td>17.4</td>
<td>1.9</td>
<td>25.4</td>
<td>15.9</td>
<td>13.1</td>
<td>4.4</td>
<td>17.02</td>
</tr>
<tr>
<td>dnacopy.alpha</td>
<td>17.8</td>
<td>0.8</td>
<td>8.1</td>
<td>0.2</td>
<td>17.8</td>
<td>0.9</td>
<td>29.38</td>
</tr>
<tr>
<td>flsa</td>
<td>20.1</td>
<td>1.2</td>
<td>56.2</td>
<td>25.6</td>
<td>8.5</td>
<td>5.8</td>
<td>0.06</td>
</tr>
<tr>
<td>glad.MinBkpWeight</td>
<td>25.5</td>
<td>1.0</td>
<td>7.8</td>
<td>3.0</td>
<td>26.5</td>
<td>1.4</td>
<td>42.39</td>
</tr>
<tr>
<td>glad.default</td>
<td>26.0</td>
<td>0.1</td>
<td>5.0</td>
<td>0.2</td>
<td>27.7</td>
<td>0.1</td>
<td>1.34</td>
</tr>
<tr>
<td>dnacopy.prune</td>
<td>26.7</td>
<td>1.0</td>
<td>19.5</td>
<td>4.8</td>
<td>24.9</td>
<td>2.0</td>
<td>41.34</td>
</tr>
<tr>
<td>dnacopy.default</td>
<td>38.0</td>
<td>0.2</td>
<td>4.8</td>
<td>0.1</td>
<td>41.1</td>
<td>0.2</td>
<td>2.02</td>
</tr>
<tr>
<td>cghseg.mBIC</td>
<td>38.5</td>
<td>0.1</td>
<td>2.0</td>
<td>0.1</td>
<td>42.3</td>
<td>0.1</td>
<td>1.81</td>
</tr>
<tr>
<td>gada.default</td>
<td>82.7</td>
<td>0.1</td>
<td>0.1</td>
<td>0.0</td>
<td>92.1</td>
<td>0.1</td>
<td>0.20</td>
</tr>
<tr>
<td>cghFLasso</td>
<td>83.8</td>
<td>0.1</td>
<td>0.8</td>
<td>0.1</td>
<td>93.1</td>
<td>0.1</td>
<td>0.18</td>
</tr>
</tbody>
</table>
Promoting sparsity with the \( \ell_1 \) penalty

The \( \ell_1 \) penalty (Tibshirani, 1996; Chen et al., 1998)

If \( R(\beta) \) is convex and "smooth", the solution of

\[
\min_{\beta \in \mathbb{R}^p} R(\beta) + \lambda \sum_{i=1}^{p} |\beta_i|
\]

is usually sparse.

Geometric interpretation with \( p = 2 \)
The total variation / variable fusion penalty

If \( R(\beta) \) is convex and "smooth", the solution of

\[
\min_{\beta \in \mathbb{R}^p} R(\beta) + \lambda \sum_{i=1}^{p-1} |\beta_{i+1} - \beta_i|
\]

is usually piecewise constant (Rudin et al., 1992; Land and Friedman, 1996).

Proof:

- Change of variable \( u_i = \beta_{i+1} - \beta_i, u_0 = \beta_1 \)
- We obtain a Lasso problem in \( u \in \mathbb{R}^{p-1} \)
- \( u \) sparse means \( \beta \) piecewise constant
TV signal approximator (=FLSA)

\[
\min_{\beta \in \mathbb{R}^p} \| Y - \beta \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} |\beta_{i+1} - \beta_i| \leq \mu
\]

Adding additional constraints does not change the change-points:

- \( \sum_{i=1}^p |\beta_i| \leq \nu \) (Tibshirani et al., 2005; Tibshirani and Wang, 2008)
- \( \sum_{i=1}^p \beta_i^2 \leq \nu \) (Mairal et al. 2010)
Solving TV signal approximator

\[
\min_{\beta \in \mathbb{R}^p} \| Y - \beta \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} |\beta_{i+1} - \beta_i| \leq \mu
\]

- QP with sparse linear constraints in \(O(p^2)\) \(\rightarrow 135\) min for \(p = 10^5\) (Tibshirani and Wang, 2008)
- Coordinate descent-like method \(O(p)\)? \(\rightarrow 3s\) s for \(p = 10^5\) (Friedman et al., 2007)
- For all \(\mu\) with the LARS in \(O(pK)\) (Harchaoui and Levy-Leduc, 2008)
- For all \(\mu\) in \(O(p \ln p)\) (Hoefling, 2009)
- For the first \(K\) change-points in \(O(p \ln K)\) (Bleakley and V., 2010)
Let $Y = (Y_1, \ldots, Y_n) \in \mathbb{R}^n$ a signal that we wish to approximate by a piecewise-constant signal $\mu$. For $i = 1$ to $k$ do

4: $I^* \leftarrow \arg \max_{I \in \mathcal{P}} \gamma(I^*)$
5: $\mathcal{P} \leftarrow \mathcal{P} \setminus \{I^*\}$
6: $\mathcal{P} \leftarrow \mathcal{P} \cup \{I_L(I^*), I_R(I^*)\}$
7: end for
8: return $\mathcal{P}$

**Algorithm 1** Greedy dichotomic segmentation

**Theorem (V. and Bleakley, 2010; see also Hoefling, 2009)**

TV signal approximator performs "greedy" dichotomic segmentation

Apparently greedy algorithm finds the global optimum!
**Algorithm 1** Greedy dichotomic segmentation

**Require:** $k$ number of intervals, $\gamma(I)$ gain function to split an interval $I$ into $I_L(I), I_R(I)$

1: $I_0$ represents the interval $[1, n]$
2: $\mathcal{P} = \{I_0\}$
3: **for** $i = 1$ to $k$ **do**
4: $I^* \leftarrow \arg \max_{I \in \mathcal{P}} \gamma(I^*)$
5: $\mathcal{P} \leftarrow \mathcal{P} \setminus \{I^*\}$
6: $\mathcal{P} \leftarrow \mathcal{P} \cup \{I_L(I^*), I_R(I^*)\}$
7: **end for**
8: return $\mathcal{P}$

---

**Theorem (V. and Bleakley, 2010; see also Hoefling, 2009)**

TV signal approximator performs "greedy" dichotomic segmentation

Apparently greedy algorithm finds the global optimum!
Speed trial: 2 s. for $K = 100$, $p = 10^7$
Control-free calling of copy number alterations in deep-sequencing data using GC-content normalization

Valentina Boeva\textsuperscript{1,2,3,4,*}, Andrei Zinovyev\textsuperscript{1,2,3}, Kevin Bleakley\textsuperscript{1,2,3}, Jean-Philippe Vert\textsuperscript{1,2,3}, Isabelle Janoueix-Lerosey\textsuperscript{1,4}, Olivier Delattre\textsuperscript{1,4} and Emmanuel Barillot\textsuperscript{1,2,3}

\textsuperscript{1}Institut Curie, \textsuperscript{2}INSERM, U900, Paris, F-75248, \textsuperscript{3}Mines ParisTech, Fontainebleau, F-77300 and \textsuperscript{4}INSERM, U830, Paris, F-75248 France
Extension: finding multiple change points shared by several profiles
Extension: finding multiple change points shared by several profiles
Define the "optimal" piecewise constant approximation $\hat{U} \in \mathbb{R}^{p \times n}$ of $Y$ as the solution of

$$\min_{U \in \mathbb{R}^{p \times n}} \| Y - U \|^2 \text{ such that } \sum_{i=1}^{p-1} 1(U_{i+1, \bullet} \neq U_{i, \bullet}) \leq k$$

DP finds the solution in $O(p^2kn)$ in time and $O(p^2)$ in memory.

But: does not scale to $p = 10^6 \sim 10^9$...
Selecting pre-defined groups of variables

Group lasso (Yuan & Lin, 2006)

If groups of covariates are likely to be selected together, the $\ell_1/\ell_2$-norm induces sparse solutions at the group level:

$$\Omega_{\text{group}}(w) = \sum_g \| w_g \|_2$$

$$\Omega(w_1, w_2, w_3) = \|(w_1, w_2)\|_2 + \|w_3\|_2$$

$$= \sqrt{w_1^2 + w_2^2 + \sqrt{w_3^2}}$$
GFLseg (Bleakley and V., 2011)

Replace

\[
\min_{U \in \mathbb{R}^{p \times n}} \| Y - U \|^2 \text{ such that } \sum_{i=1}^{p-1} 1(U_{i+1,\cdot} \neq U_{i,\cdot}) \leq k
\]

by

\[
\min_{U \in \mathbb{R}^{p \times n}} \| Y - U \|^2 \text{ such that } \sum_{i=1}^{p-1} w_i \| U_{i+1,\cdot} - U_{i,\cdot} \| \leq \mu
\]

GFLseg = Group Fused Lasso segmentation

Questions

- Practice: can we solve it efficiently?
- Theory: does it recover the correct segmentation?
GFLseg (Bleakley and V., 2011)

Replace

\[
\min_{U \in \mathbb{R}^{p \times n}} \| Y - U \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} 1(U_{i+1,\bullet} \neq U_{i,\bullet}) \leq k
\]

by

\[
\min_{U \in \mathbb{R}^{p \times n}} \| Y - U \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} w_i \| U_{i+1,\bullet} - U_{i,\bullet} \| \leq \mu
\]

GFLseg = Group Fused Lasso segmentation

Questions

- **Practice**: can we solve it efficiently?
- **Theory**: does it recover the correct segmentation?
TV approximator implementation

\[
\min_{U \in \mathbb{R}^{p \times n}} \| Y - U \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} w_i \| U_{i+1} - U_i \| \leq \mu
\]

**Theorem**

The TV approximator can be solved efficiently:

- **approximately** with the group LARS in \( O(npk) \) in time and \( O(np) \) in memory
- **exactly** with a block coordinate descent + active set method in \( O(np) \) in memory
Figure 2: Speed trials for group fused LARS (top row) and Lasso (bottom row). Left column: varying \( n \), with fixed \( p = 10 \) and \( k = 10 \); center column: varying \( p \), with fixed \( n = 1000 \) and \( k = 10 \); right column: varying \( k \), with fixed \( n = 1000 \) and \( p = 10 \). Figure axes are log-log. Results are averaged over 100 trials.
Consistency

Suppose a single change-point:

- at position $u = \alpha p$
- with increments $(\beta_i)_{i=1,\ldots,n}$ s.t. $\bar{\beta}^2 = \lim_{k \to \infty} \frac{1}{n} \sum_{i=1}^{n} \beta_i^2$
- corrupted by i.i.d. Gaussian noise of variance $\sigma^2$

Does the TV approximator correctly estimate the first change-point as $p$ increases?
Consistency of the weighted TV approximator

\[
\min_{U \in \mathbb{R}^{p \times n}} \| Y - U \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} w_i \| U_{i+1,\bullet} - U_{i,\bullet} \| \leq \mu
\]

**Theorem**

The weighted TV approximator with weights

\[
\forall i \in [1, p - 1], \quad w_i = \sqrt{\frac{i(p - i)}{p}}
\]

correctly finds the first change-point with probability tending to 1 as \( n \rightarrow +\infty \).

- we see the benefit of increasing \( n \)
- we see the benefit of adding weights to the TV penalty
Figure 3: **Single change-point accuracy for the group fused Lasso.** Accuracy as a function of the number of profiles $p$ when the change-point is placed in a variety of positions $u = 50$ to $u = 90$ (left and centre plots, resp. unweighted and weighted group fused Lasso), or: $u = 50\pm2$ to $u = 90\pm2$ (right plot, weighted with varying change-point location), for a signal of length 100.
Figure 4: Multiple change-point accuracy. Accuracy as a function of the number of profiles $p$ when change-points are placed at the nine positions $\{10, 20, \ldots, 90\}$ and the variance $\sigma^2$ of the centered Gaussian noise is either 0.05 (left), 0.2 (center) and 1 (right). The profile length is 100.
Application: detection of frequent abnormalities
Conclusion

- Partial expert annotation can be done efficiently to benchmark and optimize breakpoint detection methods
- Popular methods and default parameters are often not very good
- Multiparametric optimization can be formulated as interval regression
- Fast segmentation method for long, multiple signals
Acknowledgements

Toby Hocking (Tokodai), Gudrun Schleiermacher, Isabelle Janoueix and Valentina Boeva (Institut Curie), Francis Bach and Kevin Bleakley (INRIA)