SegAnnDB: interactive genomic segmentation
http://bioviz.rocq.inria.fr/

Toby Dylan Hocking, Toby.Hocking@mail.mcgill.ca

joint work with Valentina Boeva, Guillem Rigaill, Gudrun Schleiermacher, Isabelle Janoueix-Lerosey, Olivier Delattre, Wilfrid Richer, Franck Bourdeaut, Miyuki Suguro, Masao Seto, Francis Bach, and Jean-Philippe Vert.

December 1, 2015

Introduction: how to detect changes in copy number?

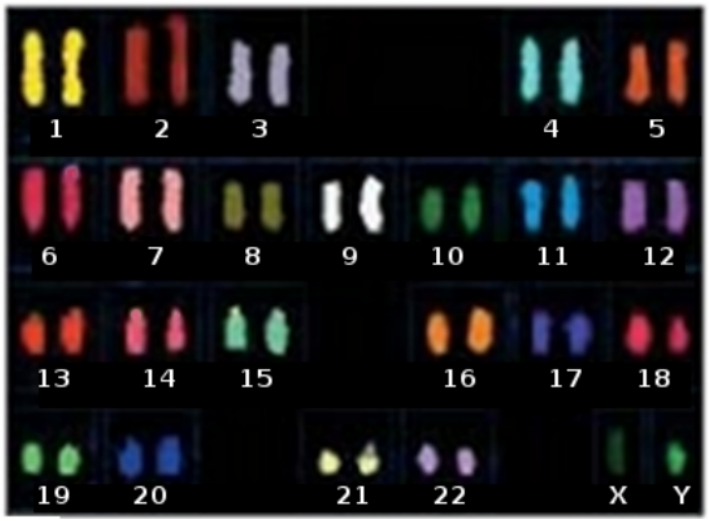
Visual breakpoint annotations

SegAnnDB: interactive genomic segmentation

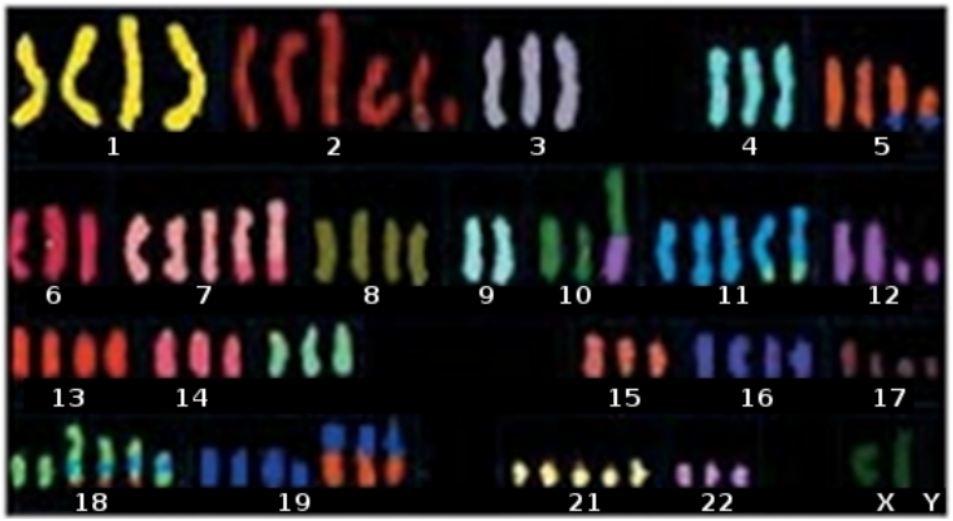
Discussion and conclusions

Cancer cells show chromosomal copy number alterations

Spectral karyotypes show the number of copies of the sex chromosomes (X,Y) and autosomes (1-22).
Source: Alberts *et al.* 2002.



Normal cell with 2 copies of each autosome.

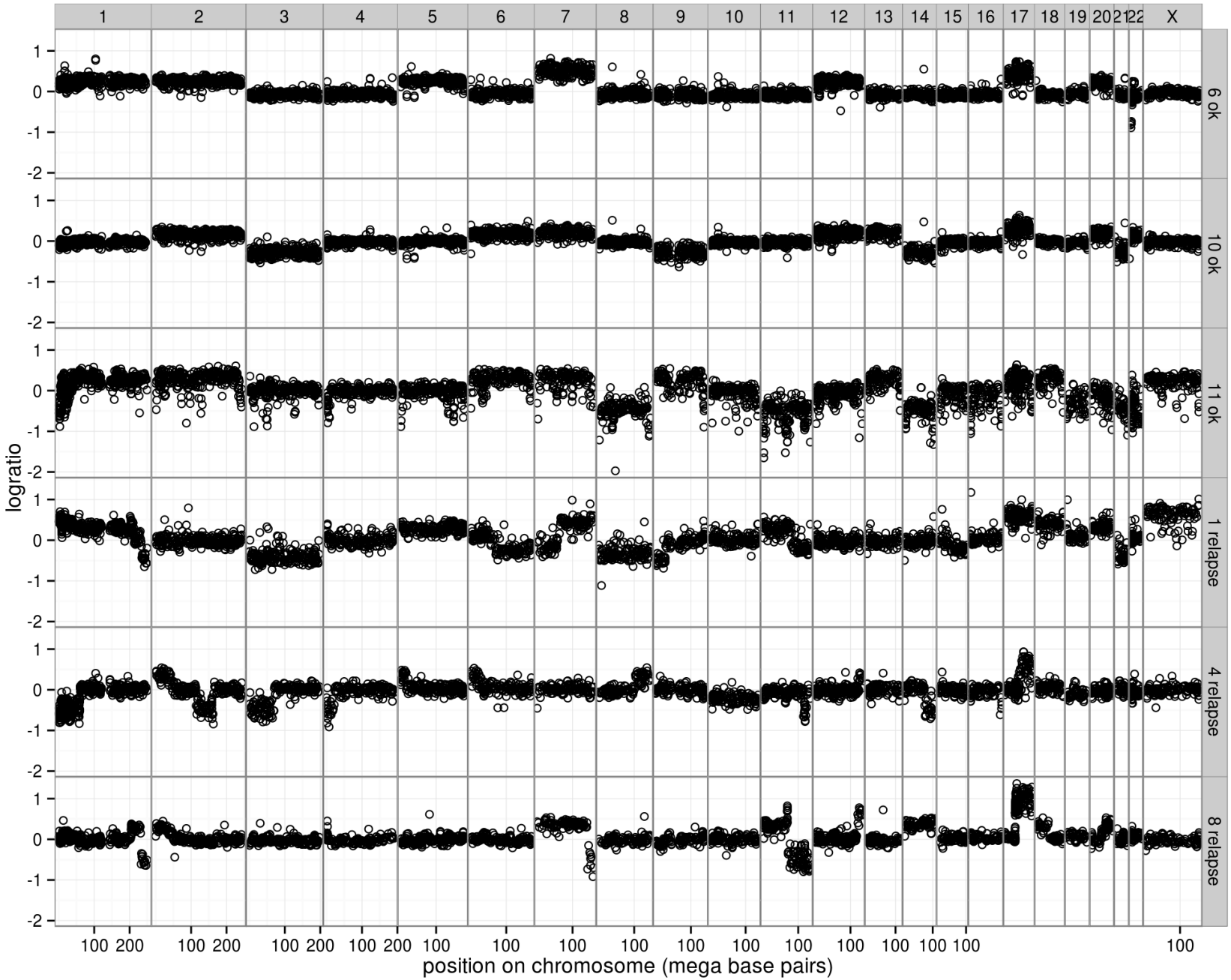


Cancer cell with many copy number alterations.

Motivation: tumor genome copy number analysis

- Comparative genomic hybridization microarrays (aCGH) allow genome-wide copy number analysis since logratio is proportional to DNA copy number (Pinkel *et al.*, 1998).
- Tumors often contain breakpoints, amplifications, and deletions at specific chromosomal locations that we would like to detect.
- Which genomic alterations are linked with good or bad patient outcome?
- To answer clinical questions like this one, we first need to accurately detect these genomic alterations.

aCGH neuroblastoma copy number data

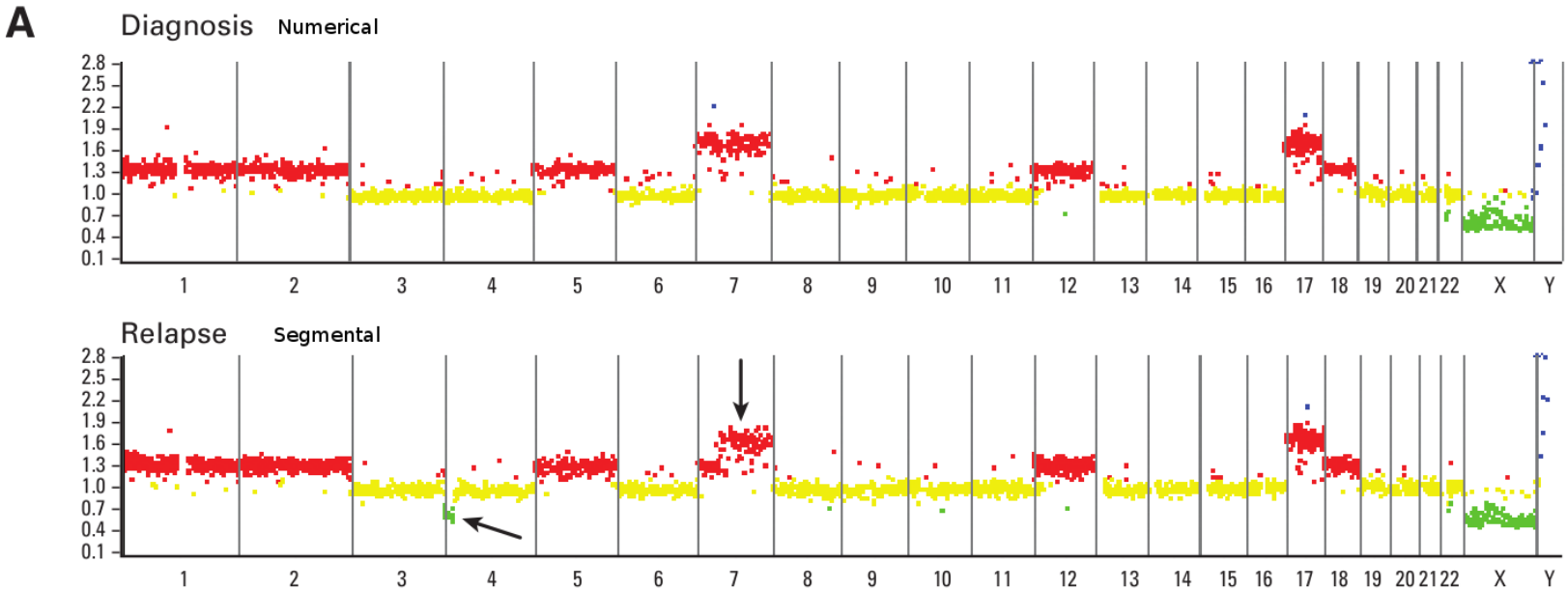


Copy number profiles are predictive of progression in neuroblastoma

Gudrun Schleiermacher, *et al.* Accumulation of Segmental Alterations Determines Progression in Neuroblastoma. J Clinical Oncology 2010.

2 types of profiles:

- Numerical: entire chromosome amplification. Good outcome.
- Segmental: deletion 1p 3p 11q, gain 1q 2p 17q. Bad outcome.



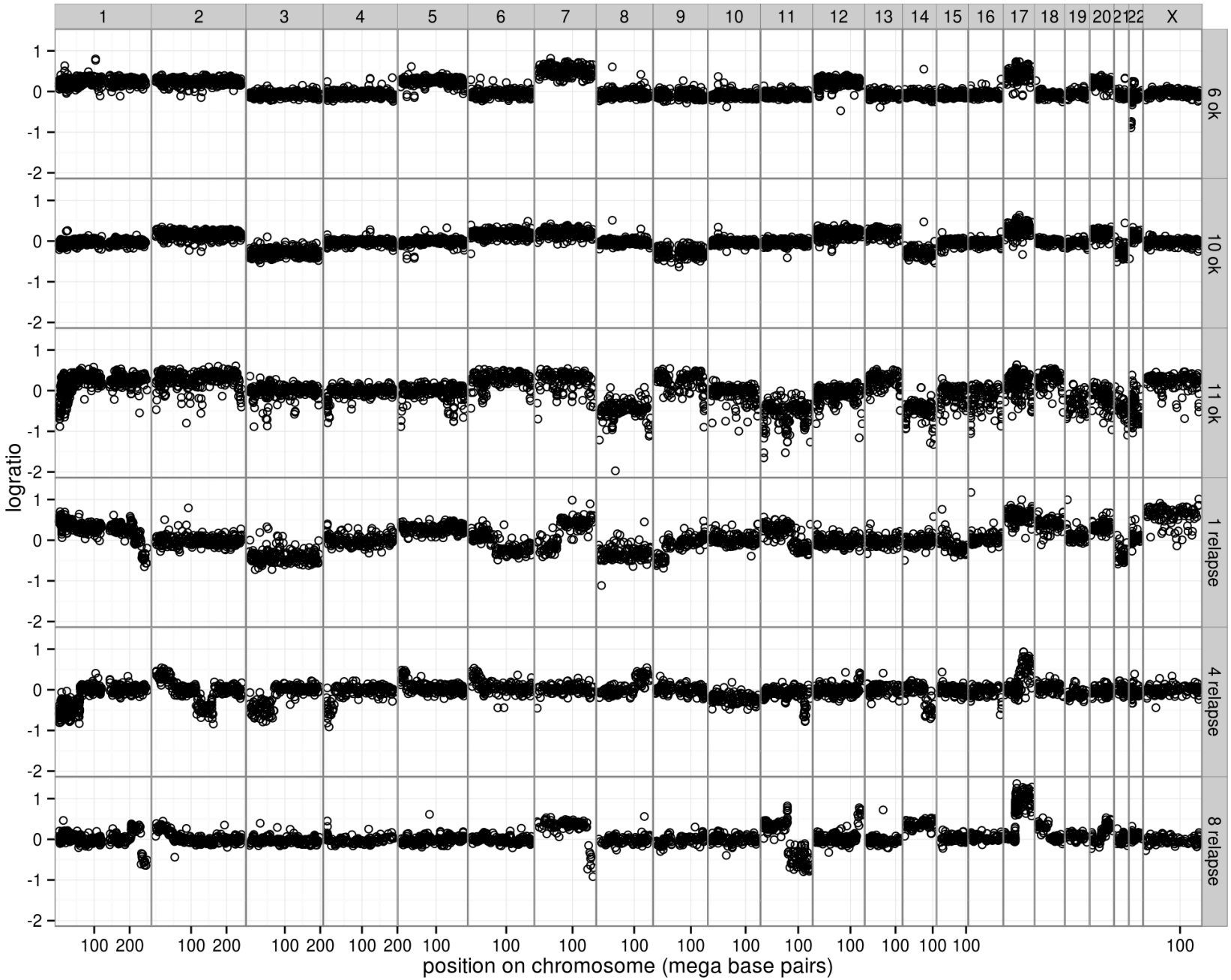
Introduction: how to detect changes in copy number?

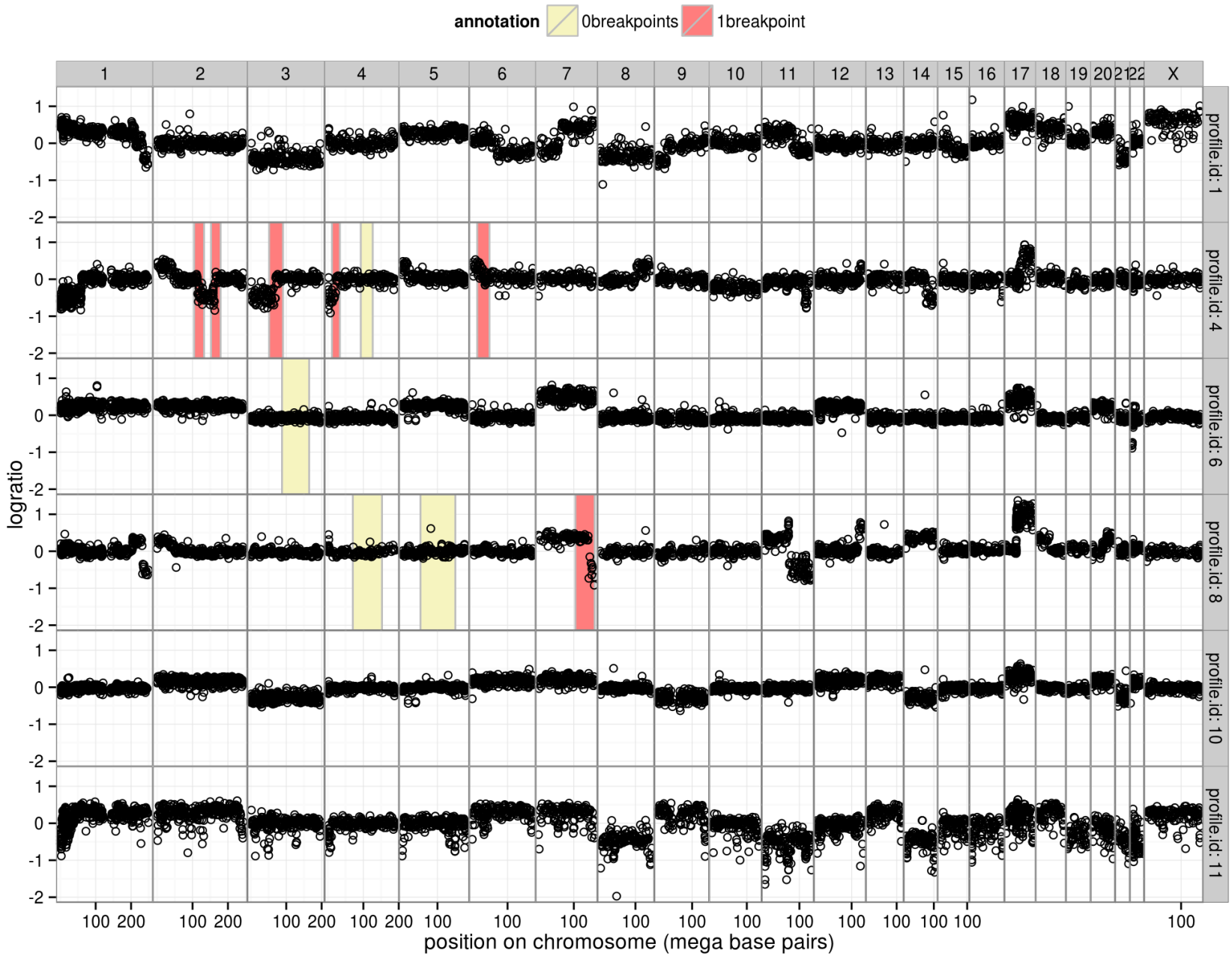
Visual breakpoint annotations

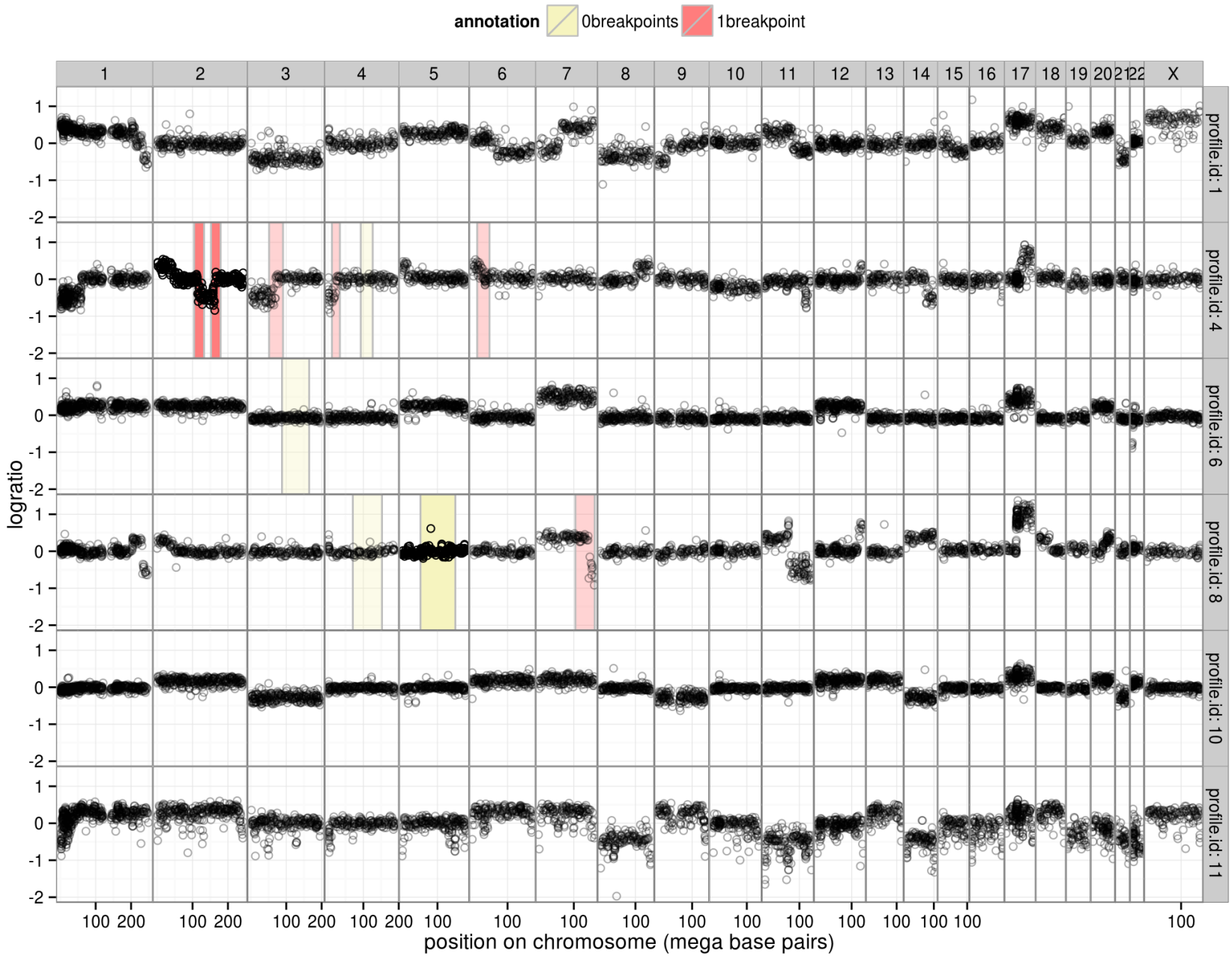
SegAnnDB: interactive genomic segmentation

Discussion and conclusions

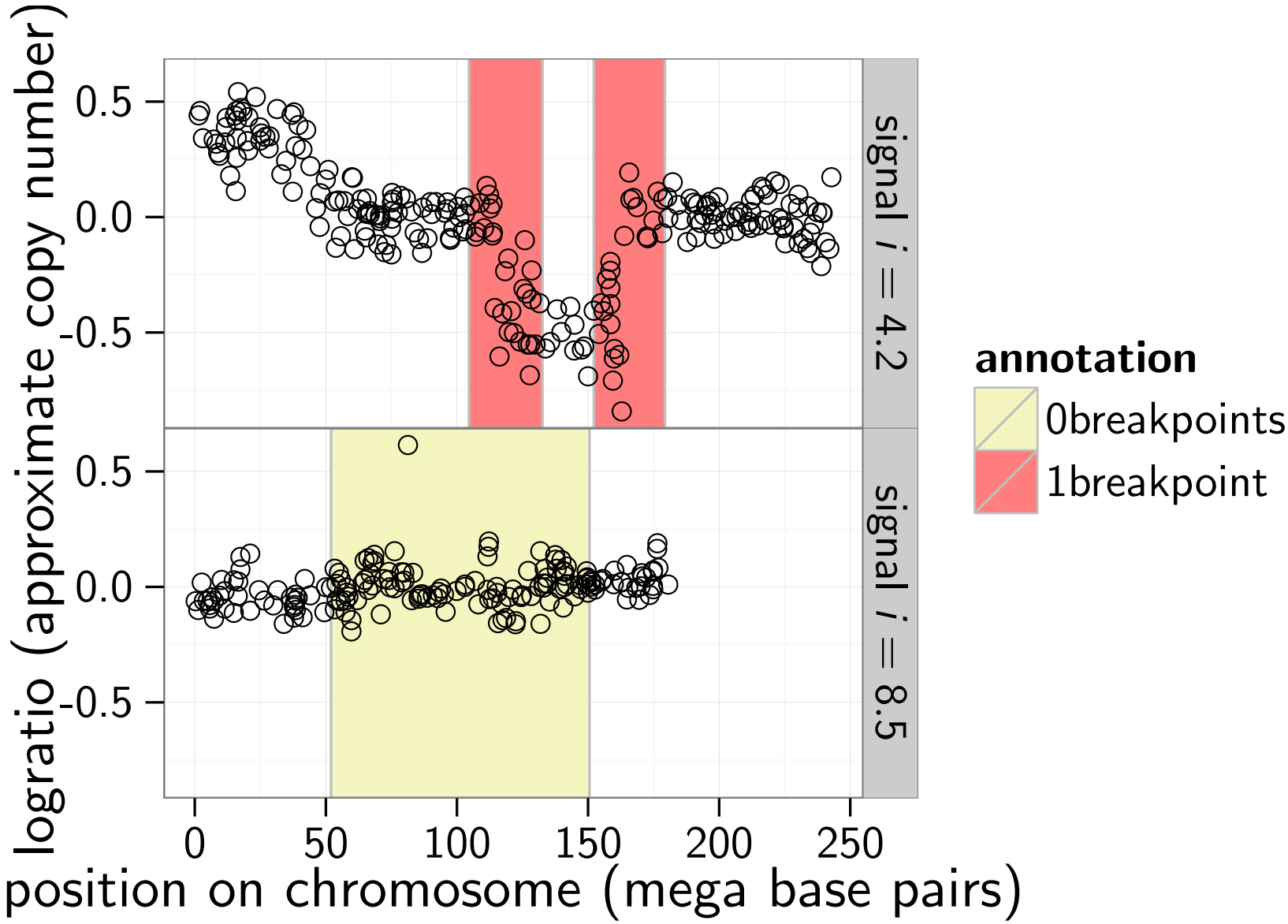
Creating breakpoint annotations (demo)



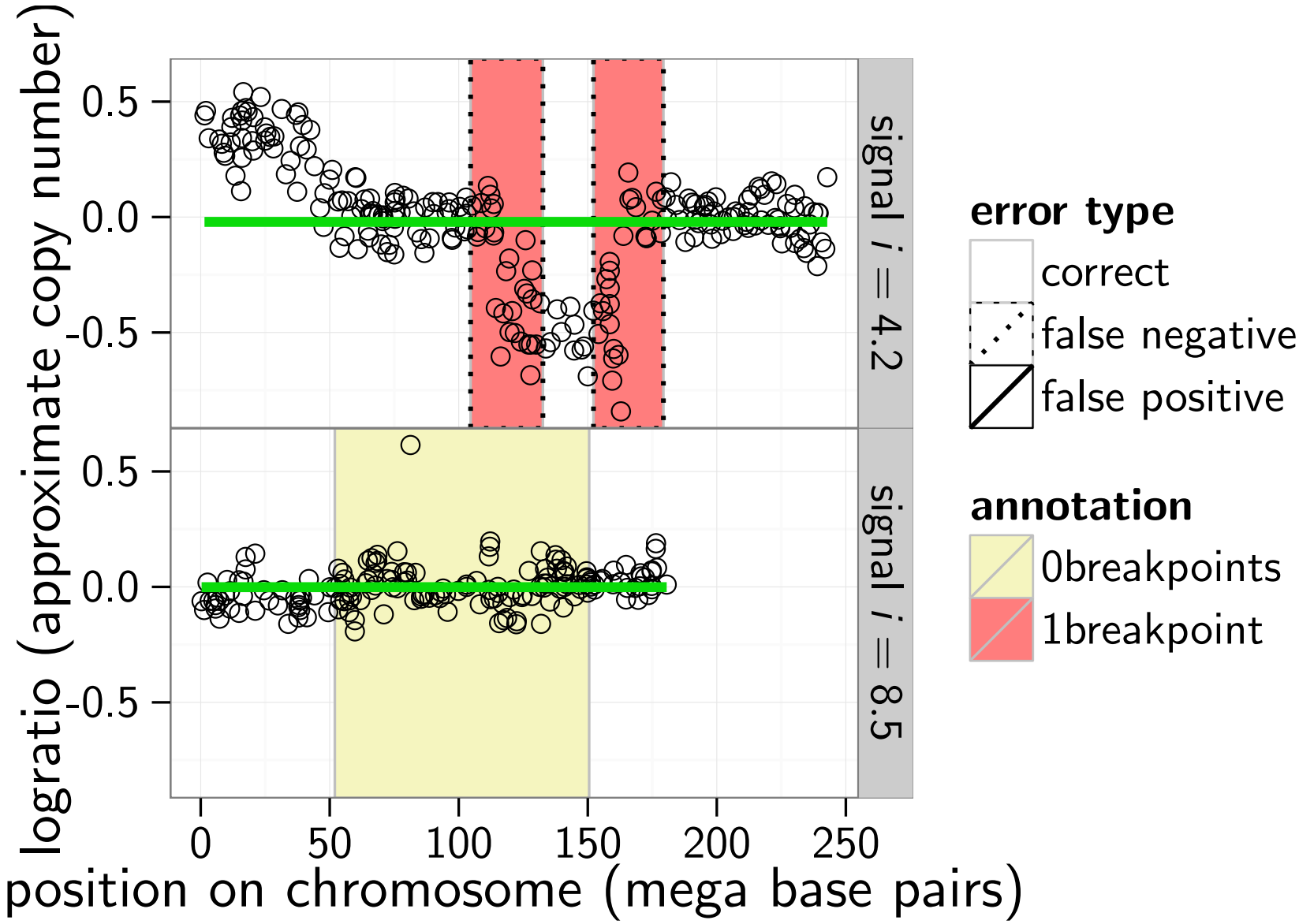

annotation
0breakpoints
1breakpoint
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
X
profile.id: 1
profile.id: 4
profile.id: 6
profile.id: 8
profile.id: 10
profile.id: 11
logratio
position on chromosome (mega base pairs)


annotation
0breakpoints
1breakpoint
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
X
profile.id: 1
profile.id: 4
profile.id: 6
profile.id: 8
profile.id: 10
profile.id: 11
logratio
position on chromosome (mega base pairs)

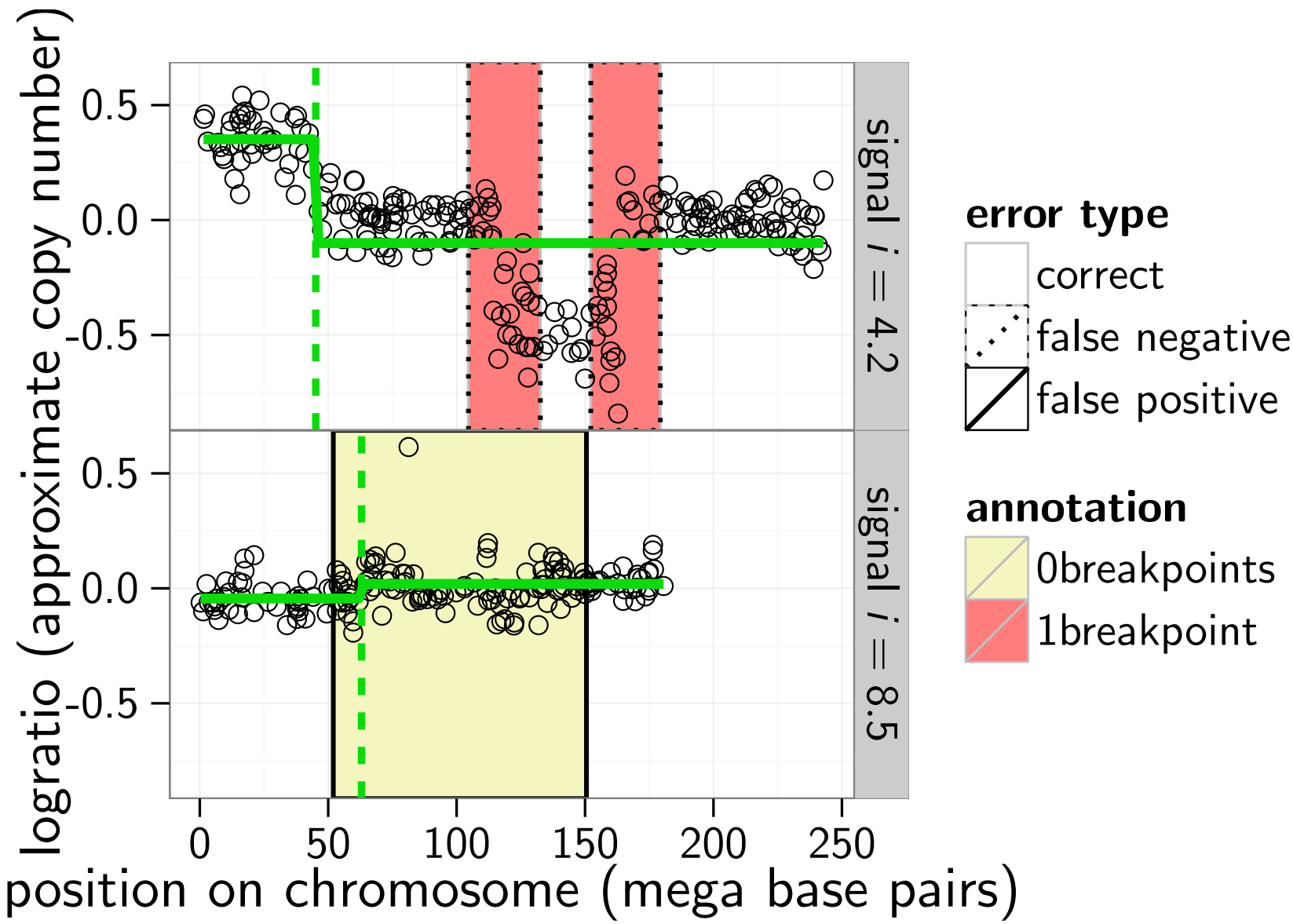
Annotations for 2 signals



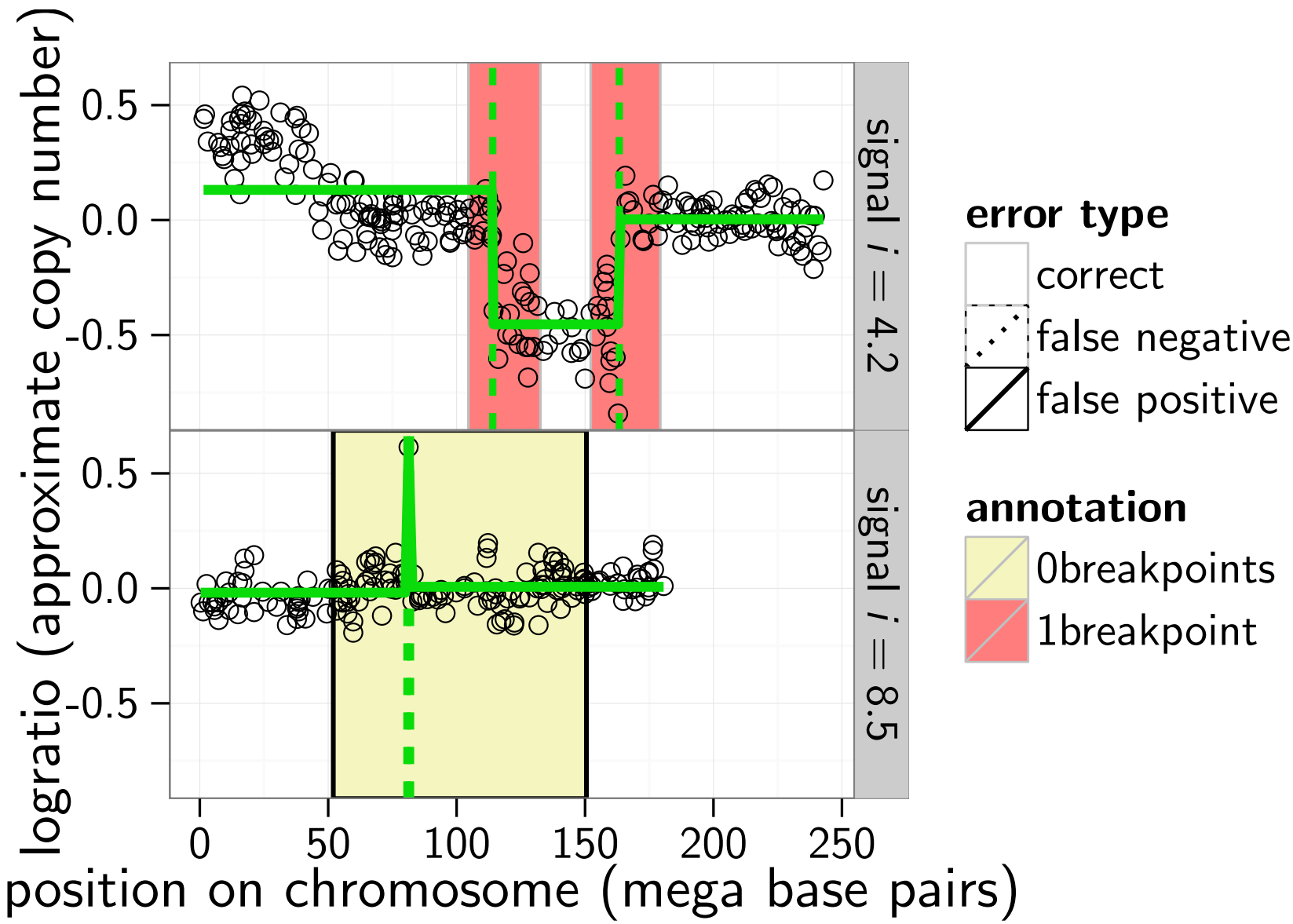
Estimated model with 1 segment



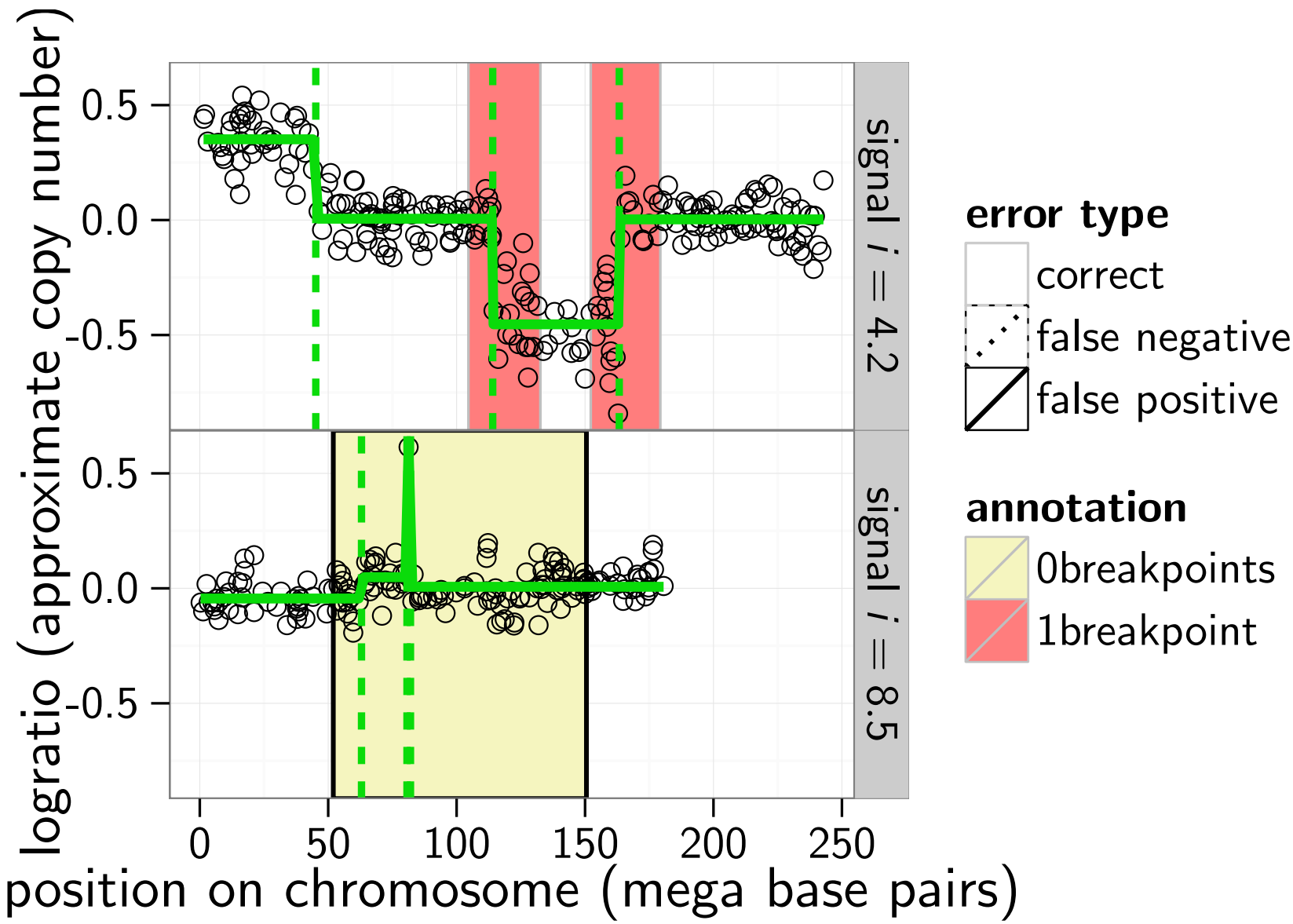
Estimated model with 2 segments



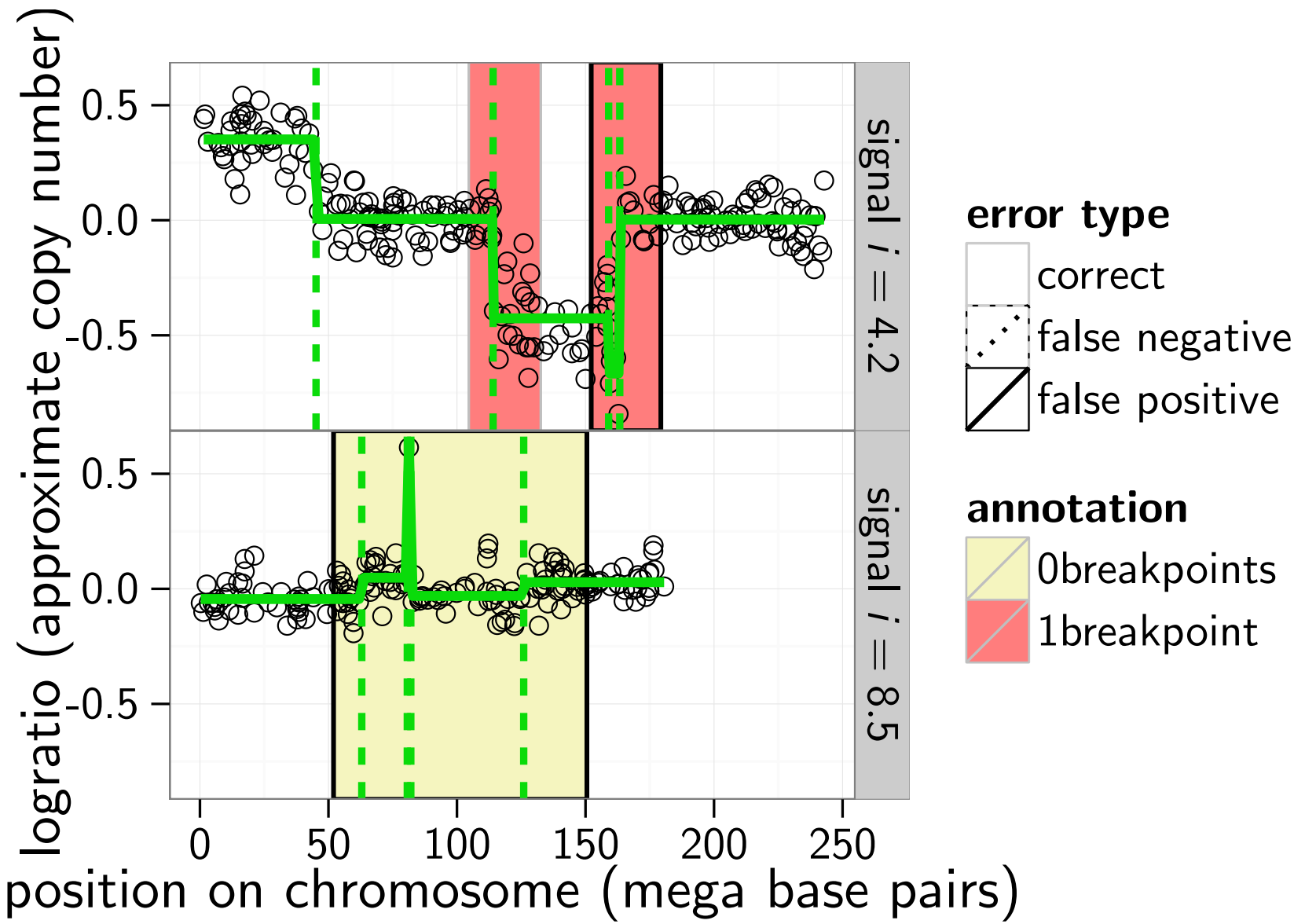
Estimated model with 3 segments



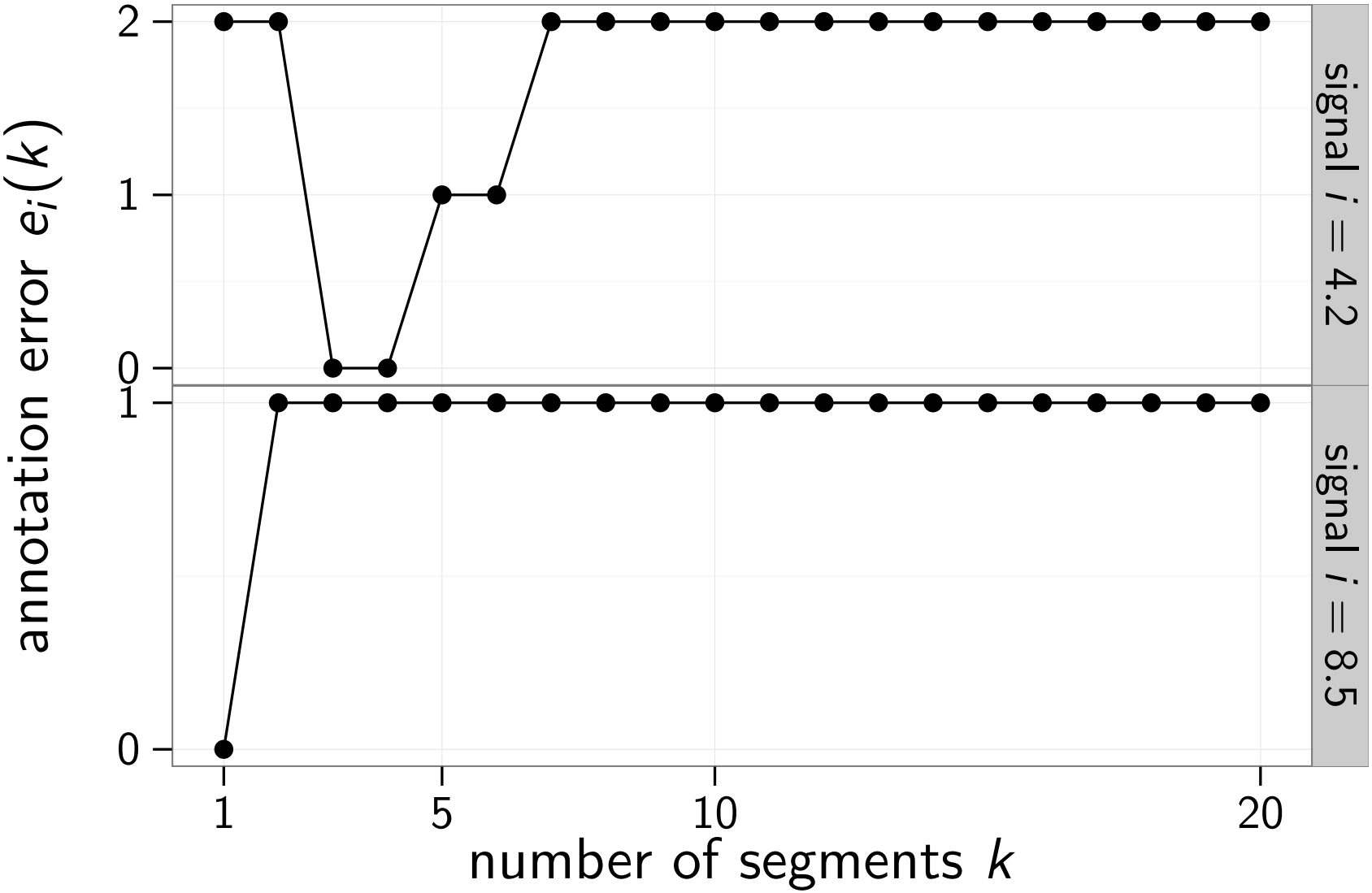
Estimated model with 4 segments



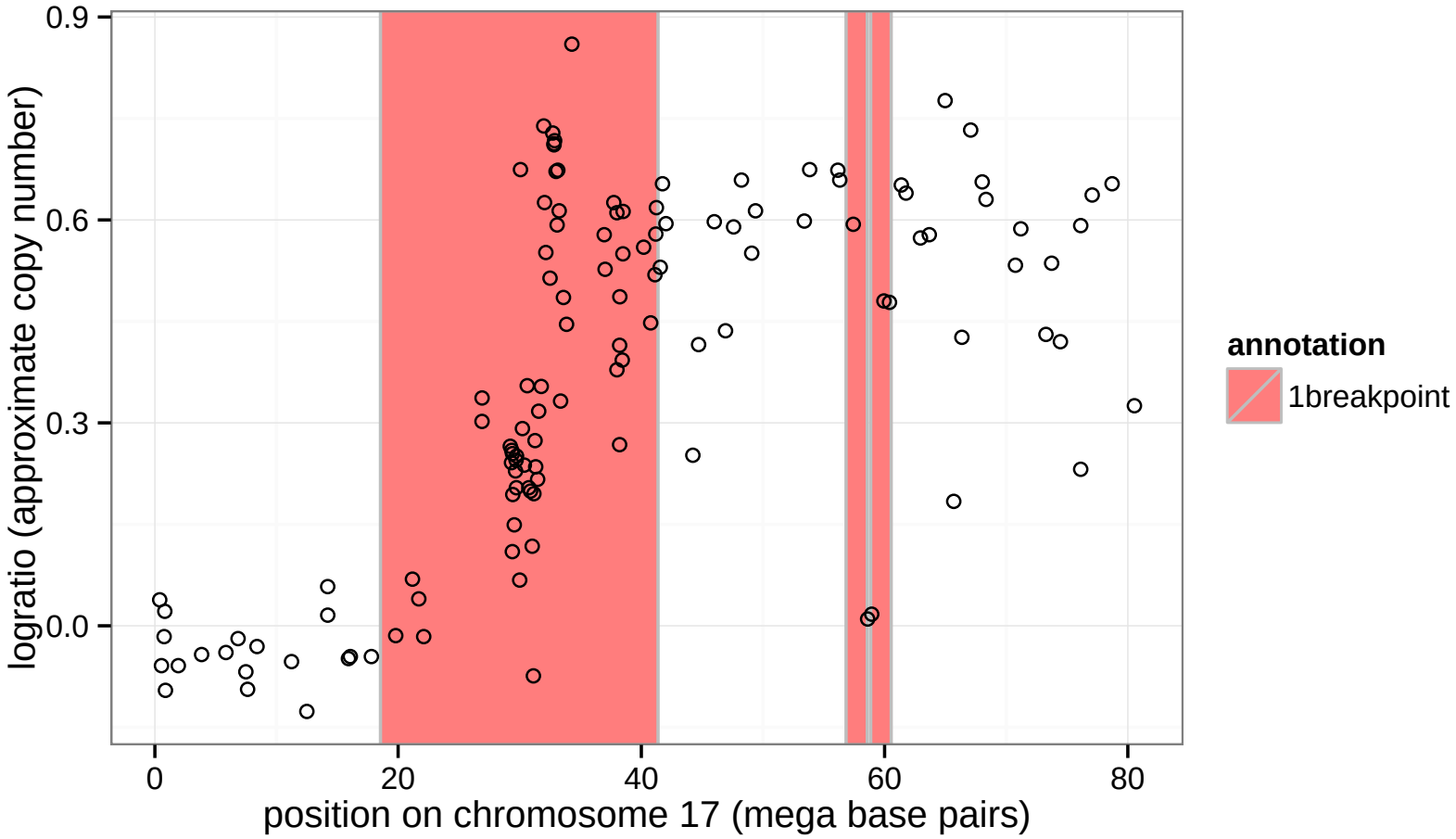
Estimated model with 5 segments



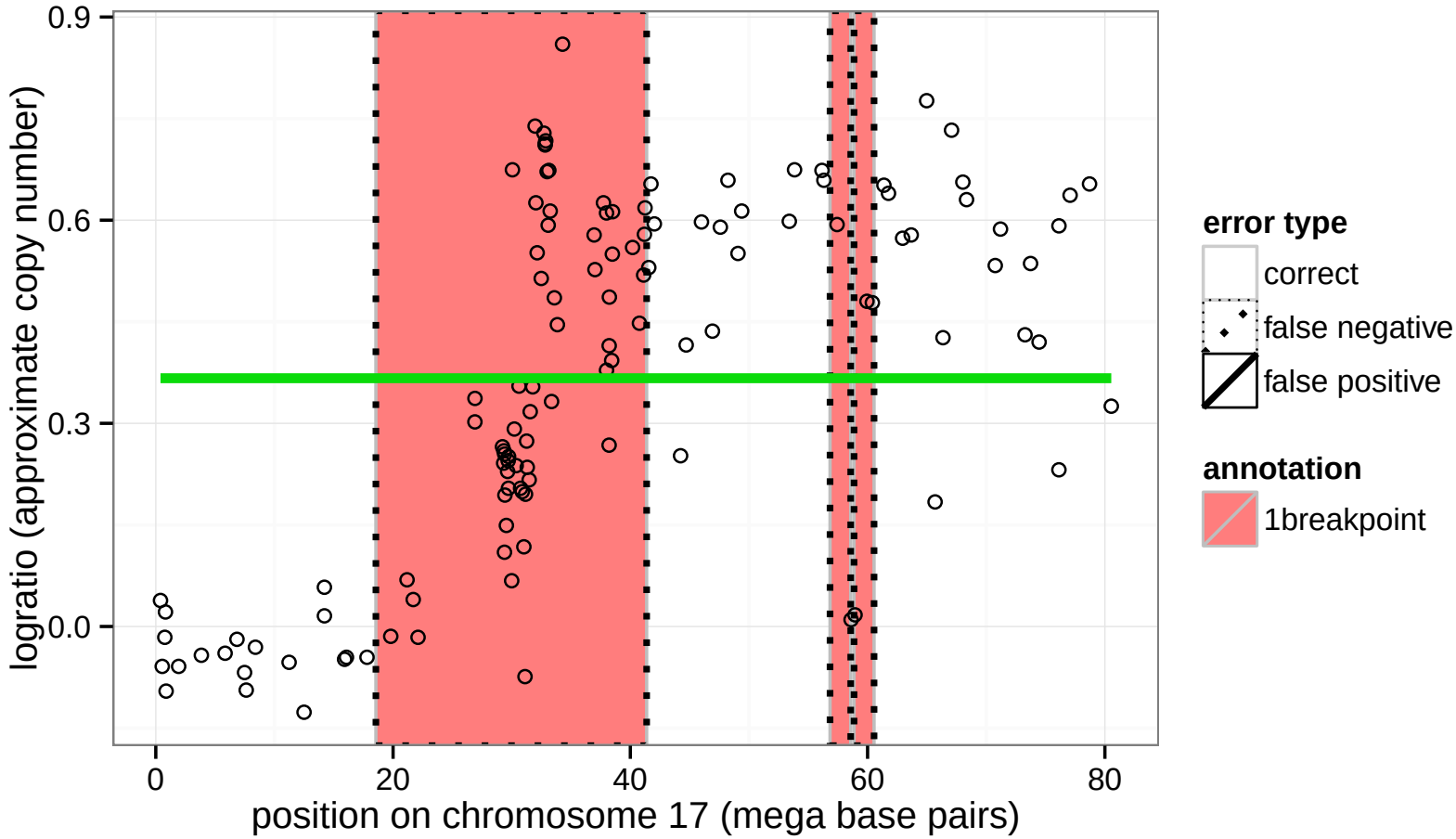
Annotation error curves for 2 signals...
which 0-error model is best?



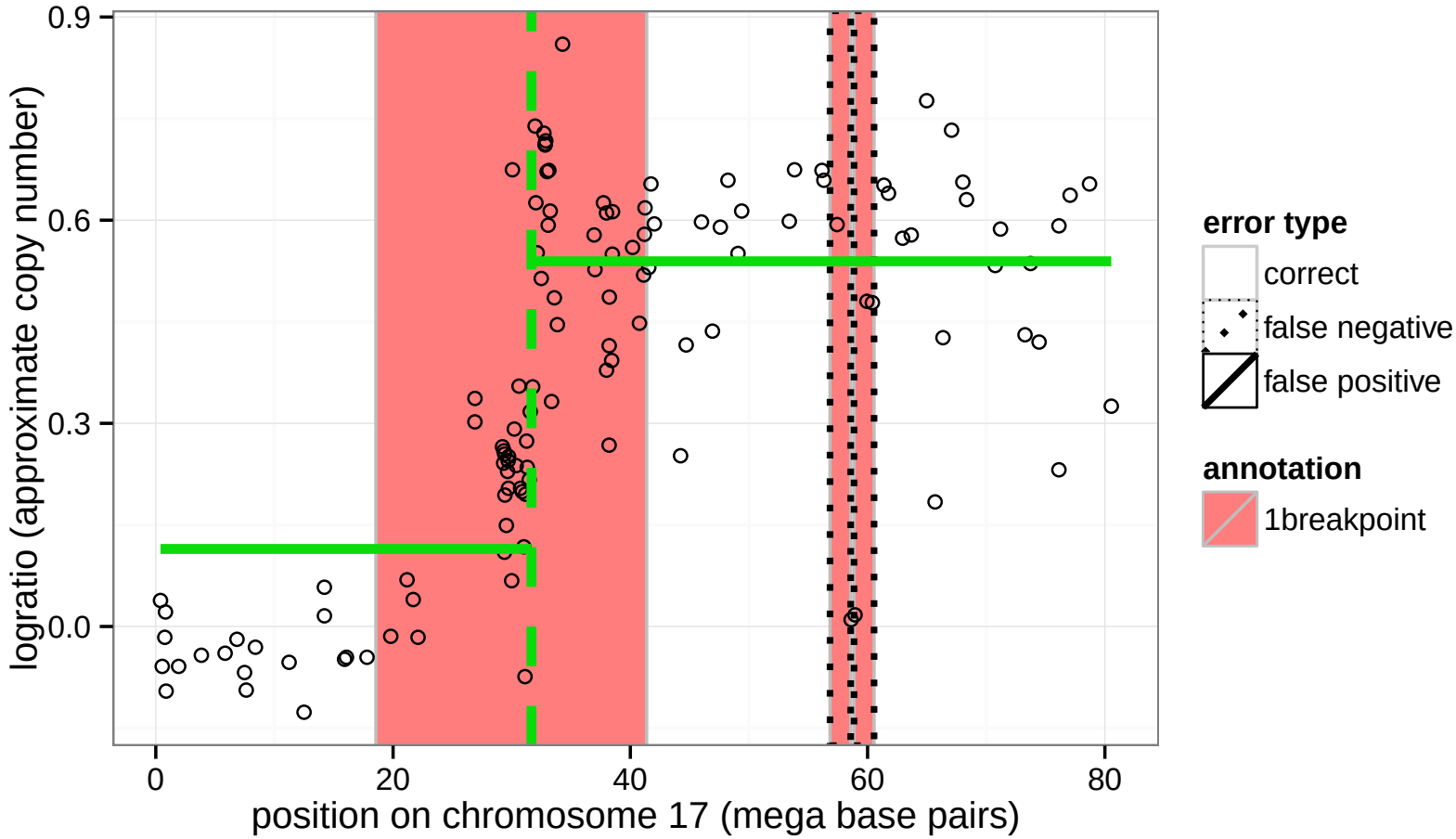
Another annotated signal



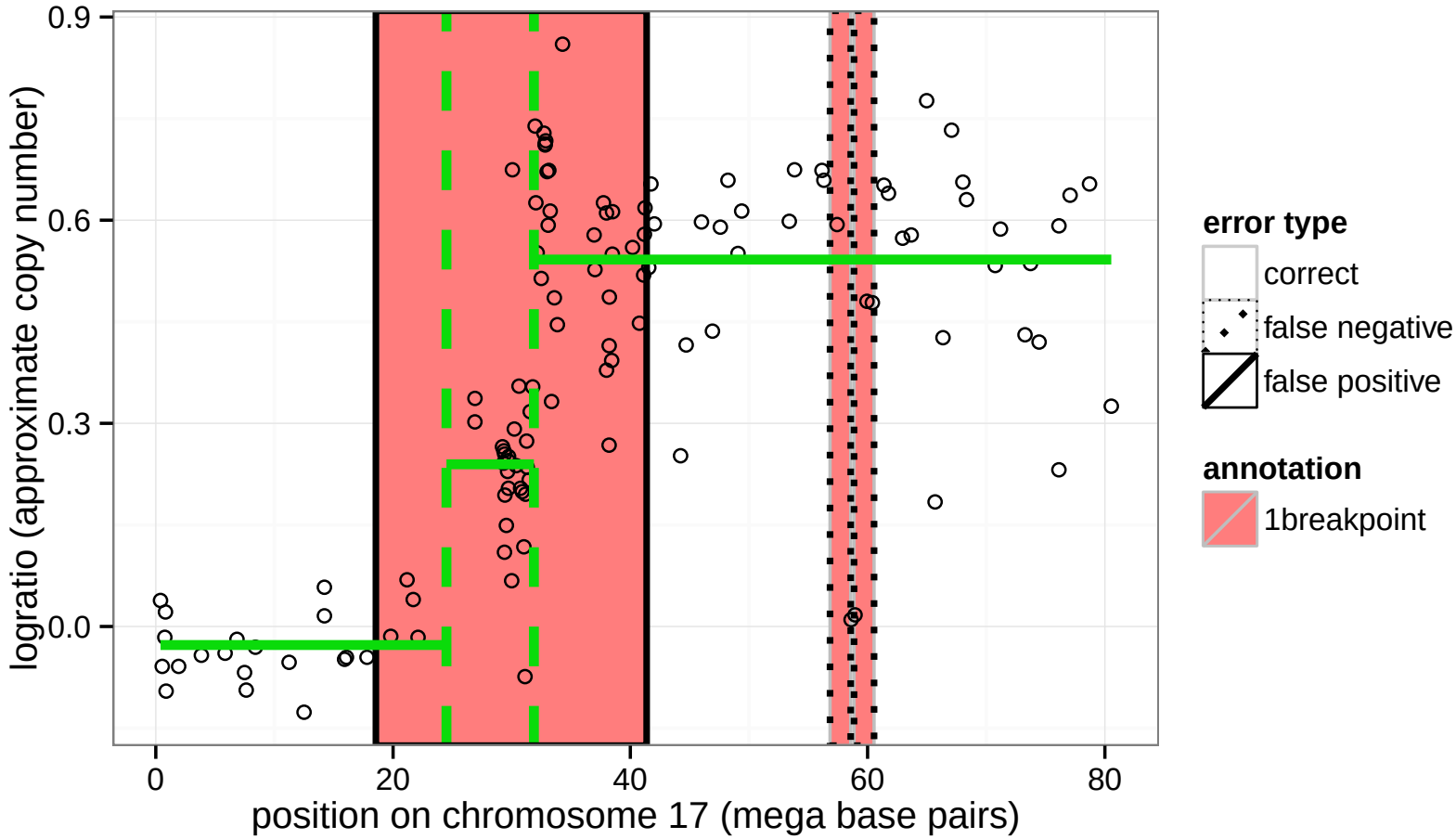
Estimated model with 1 segment



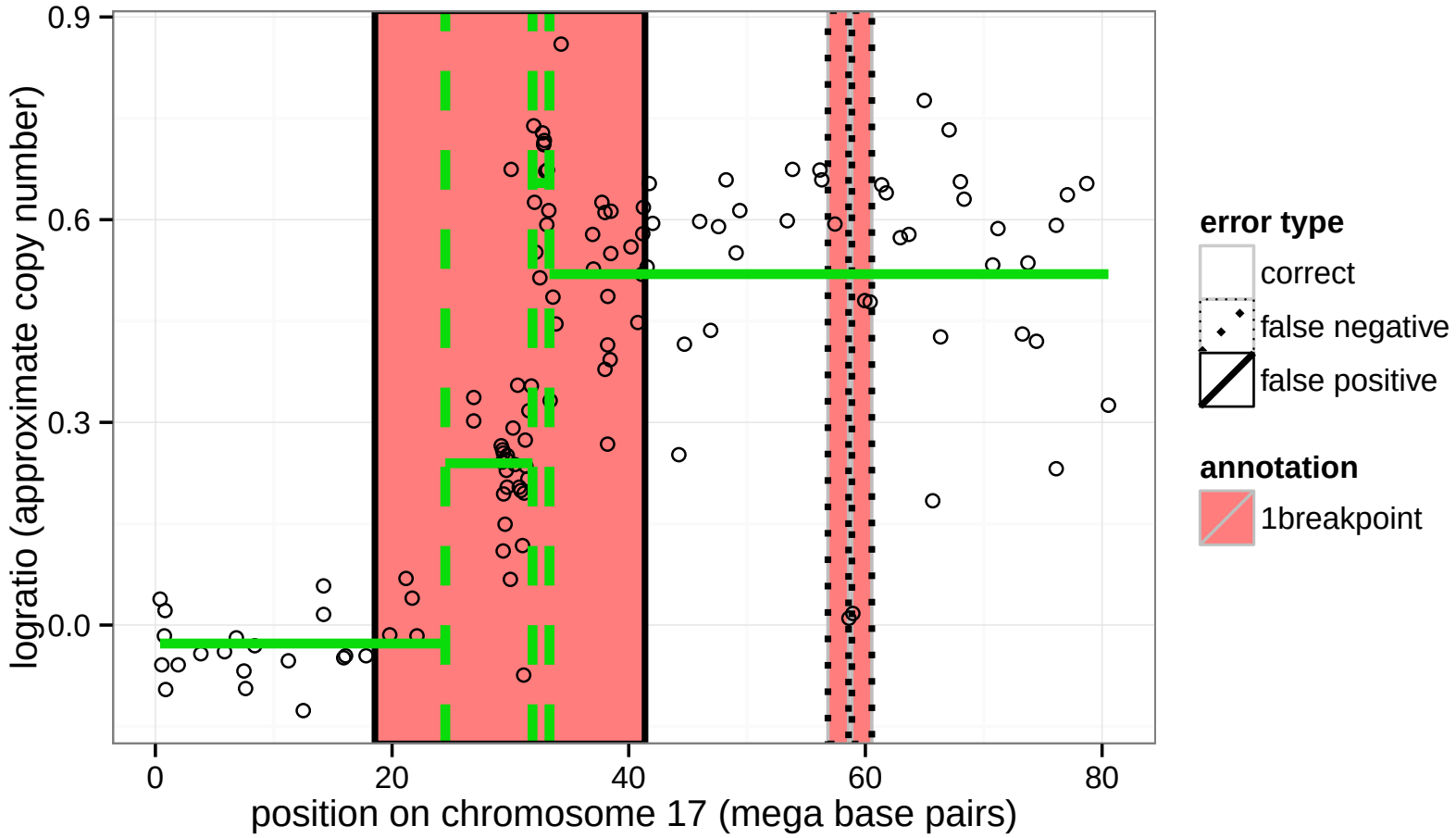
Estimated model with 2 segments



Estimated model with 3 segments

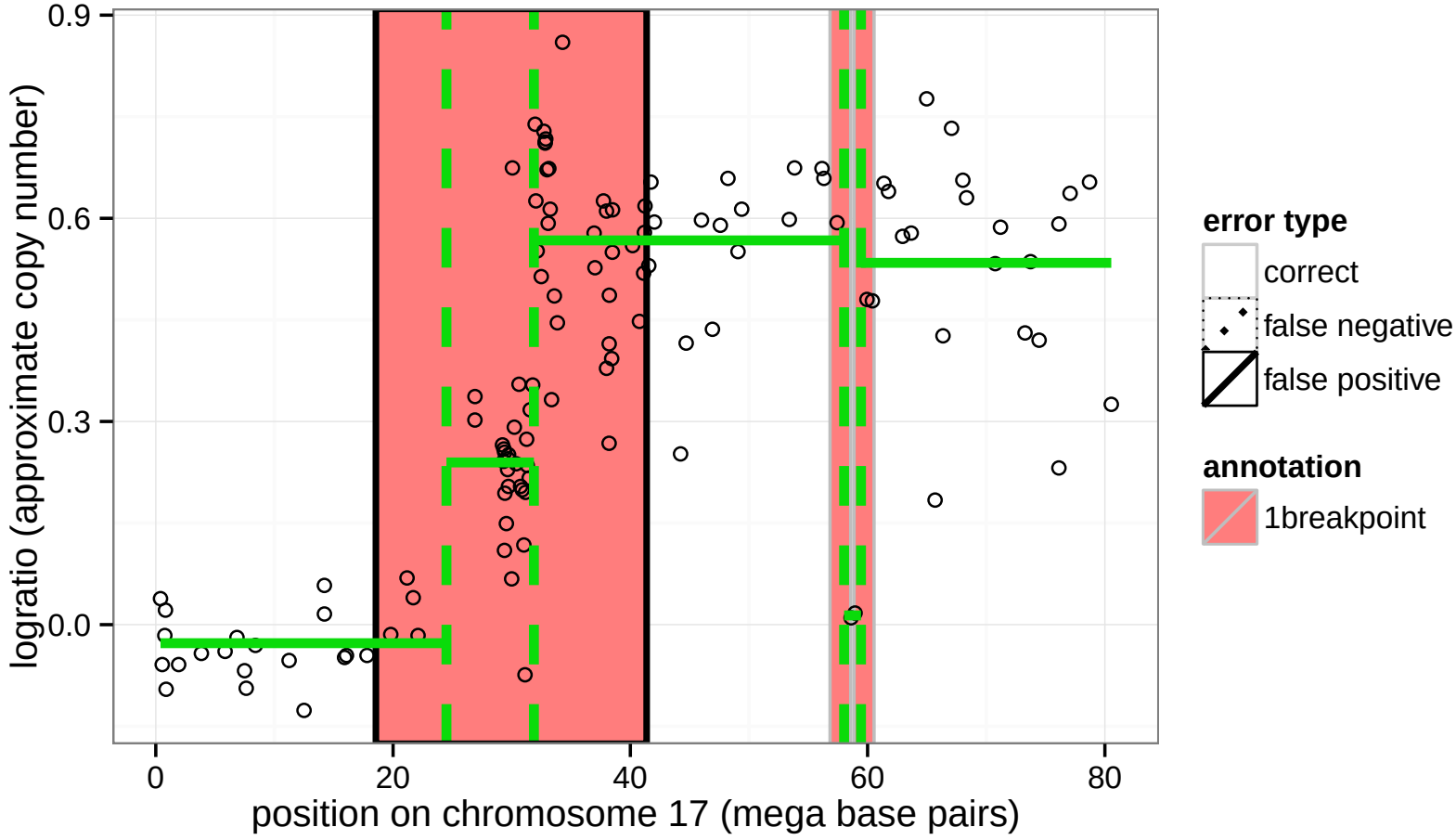


Estimated model with 4 segments



Estimated model with 5 segments

There are no consistent least squares models.



How do we find a consistent model?

Introduction: how to detect changes in copy number?

Visual breakpoint annotations

SegAnnDB: interactive genomic segmentation

Discussion and conclusions

Problems solved by SegAnnDB

- Statistical machine learning problems:

Consistent Models	Problem	Solution
1	none	
> 1	Prediction	Other annotated signals Hocking *et al.* 2013
0	Fitting	SegAnnot: constrained segmentation Hocking and Rigaill 2012

- Technical problems:
  - **Interactive scatterplots**: zooming, annotation, model updates.
  - **Storage and data export to genome browsers**: probes, user-specific annotations, models.

Optimal prediction from limited annotations

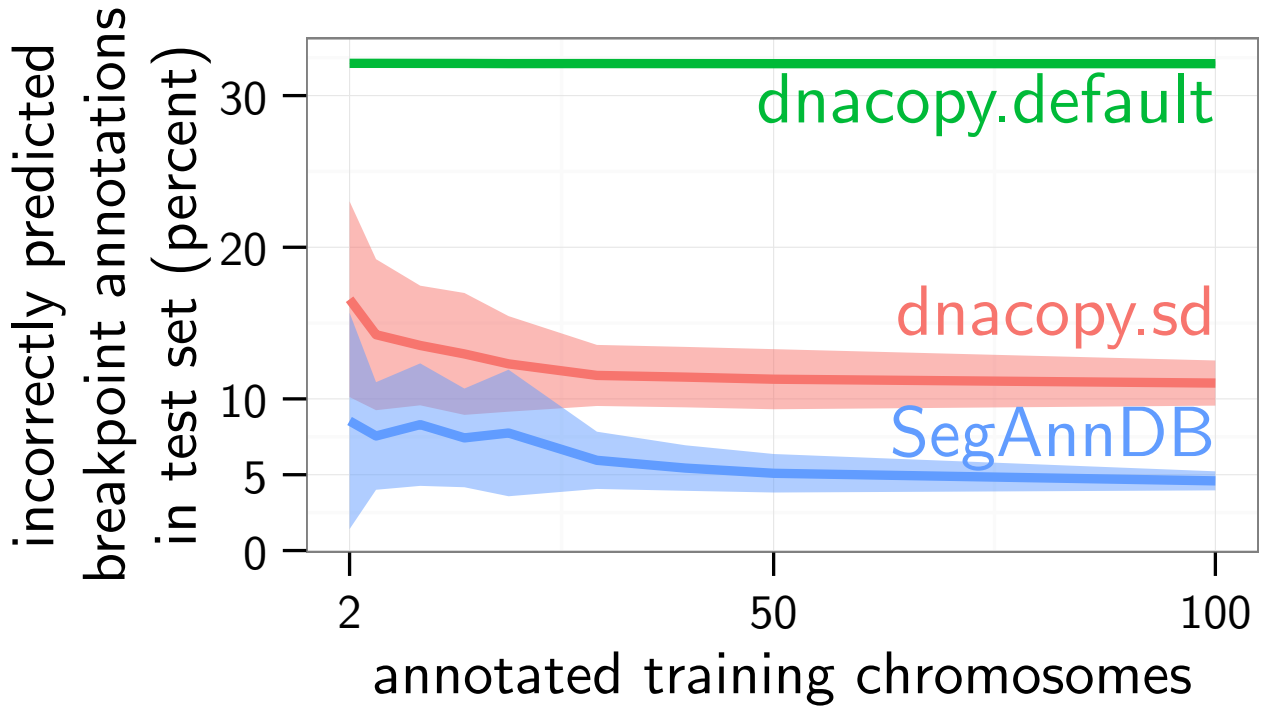
- Rigaill 2010. Pruned dynamic programming for optimal multiple change-point detection. arXiv:1004.0887.

$$\hat{y}^k = \arg\min_{\mu \in \mathbb{R}^d} ||y - \mu||_2^2$$
$$\text{such that} \quad \mu \text{ has } k-1 \text{ changes.}$$

- Pruned dynamic programming solver: 700 lines of C++.
- Python interface: 50 lines of C.
- Consistent models $\subseteq \{\hat{y}^1, \ldots, \hat{y}^{20}\}$.
- Hocking et al. ICML 2013. Choose the number of segments k using interval regression on the other annotated signals.
- Gradient descent solver: 100 lines of Python, run as a background process.

Test error decreases as models learns from more labels

- Benchmark: 3642 labeled regions across 3109 chromosomes.
- Train on a few chromosomes, test on the rest.
- Mean and SD over 60 random train set orderings.



Optimal fitting for any annotations

- Hocking and Rigaill 2012. SegAnnot: fast segmentation of annotated piecewise constant signals. HAL-00759129.

$$\begin{aligned} \arg\min_{\mu \in \mathbb{R}^d} \quad & ||y - \mu||_2^2 \\ \text{such that} \quad & \mu \text{ has 1 change in each 1breakpoint region.} \end{aligned}$$

- Dynamic programming solver: 200 lines of C.
- Python interface: 100 lines of C.

Demo of optimal fitting on `http://bioviz.rocq.inria.fr`

Interactive zoomable scatterplots

- On data upload, draw 5 sizes of PNG scatterplots using Python Imaging Library: 10Kb–1Mb for a sequence of 150,000 points.
- Test your browser-dependent image size limit `http://sugiyama-www.cs.titech.ac.jp/~toby/images/`
- For a plot, first render PNG scatterplot as the background of an SVG element.
- Then ask the server for the current regions/model, and draw with SVG.
- When client changes annotations, save on server and send model back to client.
- 500 lines of Javascript/D3.

Data storage and export

- Web server/database: 1500 lines of Python.
- Berkeley DB: small fast NoSQL database.
- DB[key] = value, key is text, value is anything.
- Pyramid web framework exports data in
  - JSON for plotted regions/model.
  - bed/bedGraph for UCSC genome browser.
  - CSV for R, etc.

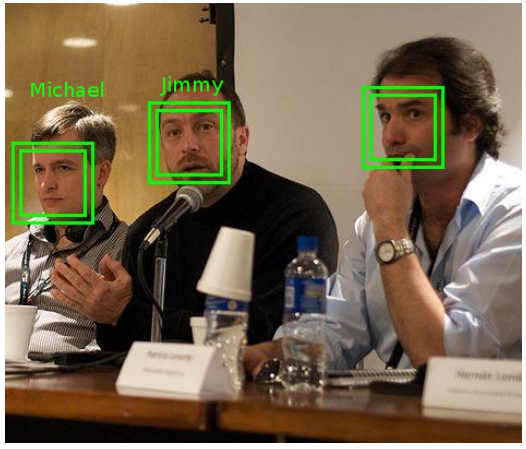
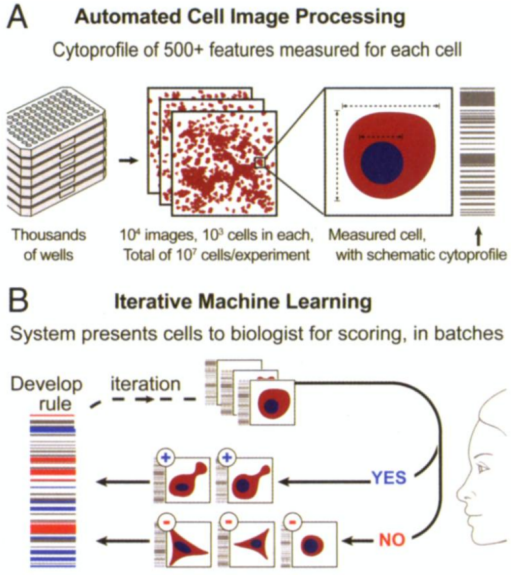
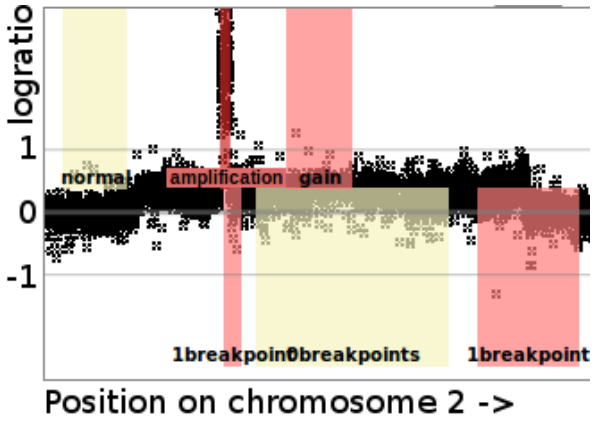
Introduction: how to detect changes in copy number?

Visual breakpoint annotations

SegAnnDB: interactive genomic segmentation

Discussion and conclusions

Discussion: SegAnnDB uses computer vision for genomic data

Photos	Cell images	Copy number profiles
		
Labels: names	phenotypes	alterations
CVPR 2013 246 papers	CellProfiler 873 citations	SegAnnDB Hocking et al, 2014.

Demo: `http://bioviz.rocq.inria.fr`

Sources: `http://en.wikipedia.org/wiki/Face_detection`

Jones et al PNAS 2009. Scoring diverse cellular morphologies in

Discussion: un-supervised versus supervised learning

- Biologists can easily locate breakpoints and noise in plots of the data.
- **Statistics/un-supervised learning**: first estimate breakpoint locations from data, then plot both to see if breakpoints (over- or under-)fit.
- **Computer vision/supervised learning**: first label regions with and without breakpoints, then predict breakpoints that minimize the number of incorrect labels.
- Exploit strong points of eyes (signal/noise) and mathematical optimization (finding the exact breakpoint).
- Advantage: supervised methods more accurate.
- Disadvantage: need time/expertise to make labels.

Conclusions/availability

- **Interactive:** SegAnnDB is the first genomic data analysis system with a model that is updated based on user-provided labels.
- **Accurate:** optimization algorithms are used to find the best model for a given set of labels.
- **Demo**: live on `http://bioviz.rocq.inria.fr/` (labeling OK, data set uploads not).
- **Available:** for analyzing your own data, install the free/open-source code on your own server/laptop: `https://github.com/tdhock/SegAnnDB`

Help: future work

- **Active learning**.
  Can we do better than random sampling?
- **Crowdsourcing**.
  If every one of you labels one profile, how do we learn a global model?

email me at Toby.Hocking@mail.mcgill.ca to collaborate!

But which model is the best?

- 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)
- GADA: sparse Bayesian learning (Pique-Regi *et al.*, 2008)
- flsa: fused lasso signal approximator path algorithm (Hoefling 2009)
- cghseg: pruned dynamic programming (Rigaill 2010)
- PELT: pruned exact linear time (Killick *et al.*, 2011)

Visual annotations indicate that maximum likelihood segmentation is the best (Hocking *et al.*, 2012).

But which model is the best?

- 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)
- GADA: sparse Bayesian learning (Pique-Regi *et al.*, 2008)
- flsa: fused lasso signal approximator path algorithm (Hoefling 2009)
- cghseg: pruned dynamic programming (Rigaill 2010)
- PELT: pruned exact linear time (Killick *et al.*, 2011)

Visual annotations indicate that maximum likelihood segmentation is the best (Hocking *et al.*, 2012).

The cghseg.k/pelt.n least squares model

For a signal $y \in \mathbb{R}^d$, the maximum likelihood model with $k \in \{1, \ldots, d\}$ segments is

$$\hat{y}^k = \underset{\mu \in \mathbb{R}^d}{\arg\min} \quad ||y - \mu||_2^2$$
$$\text{such that} \quad \mu \text{ has } k-1 \text{ changes.}$$

We select the number of segments k using **visual annotations**.