# A linear time algorithm for constrained optimal segmentation

Toby Dylan Hocking
toby.hocking@mail.mcgill.ca
joint work with Guillem Rigaill, Paul Fearnhead, Guillaume Bourque

11 Nov 2016

Problem: optimizing ChIP-seq peak detection

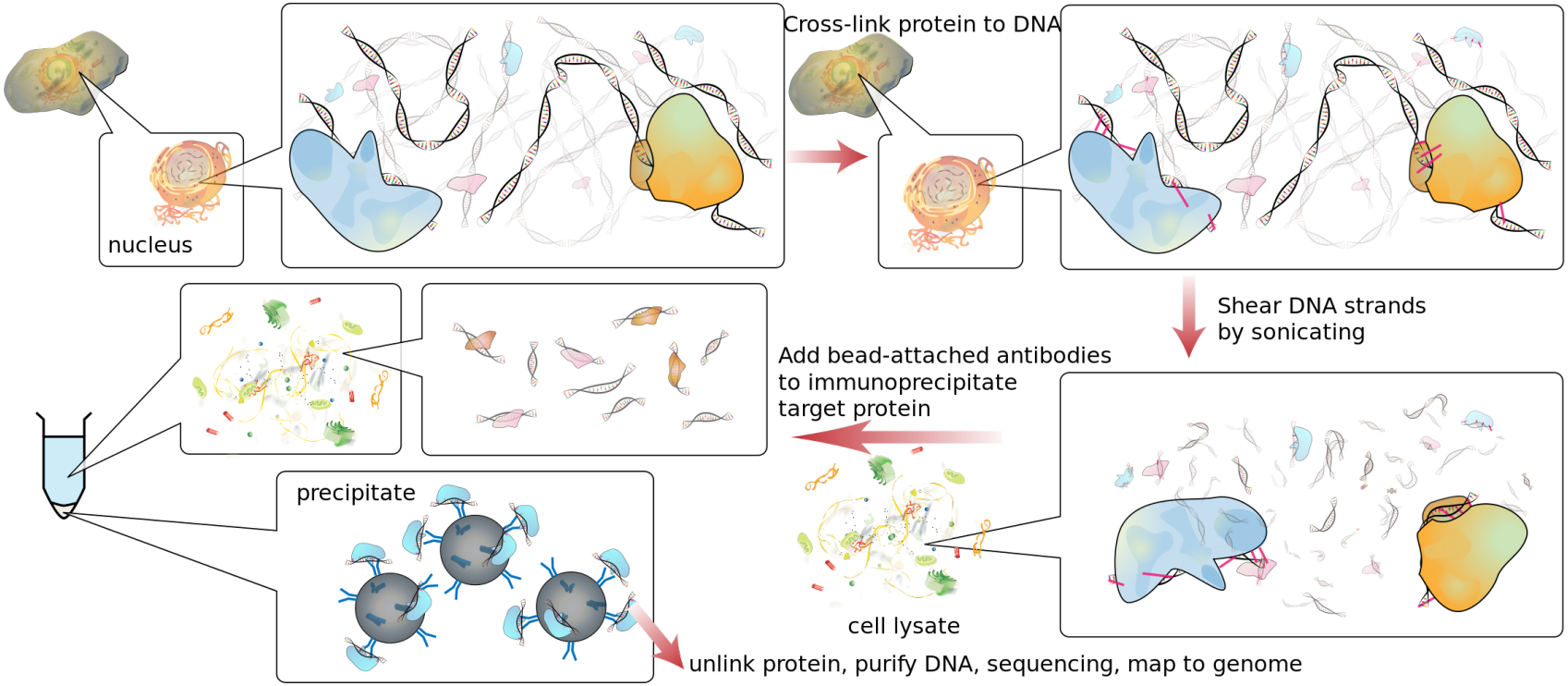
New linear time algorithm using functional pruning

Results on benchmark data sets

Conclusions and future work

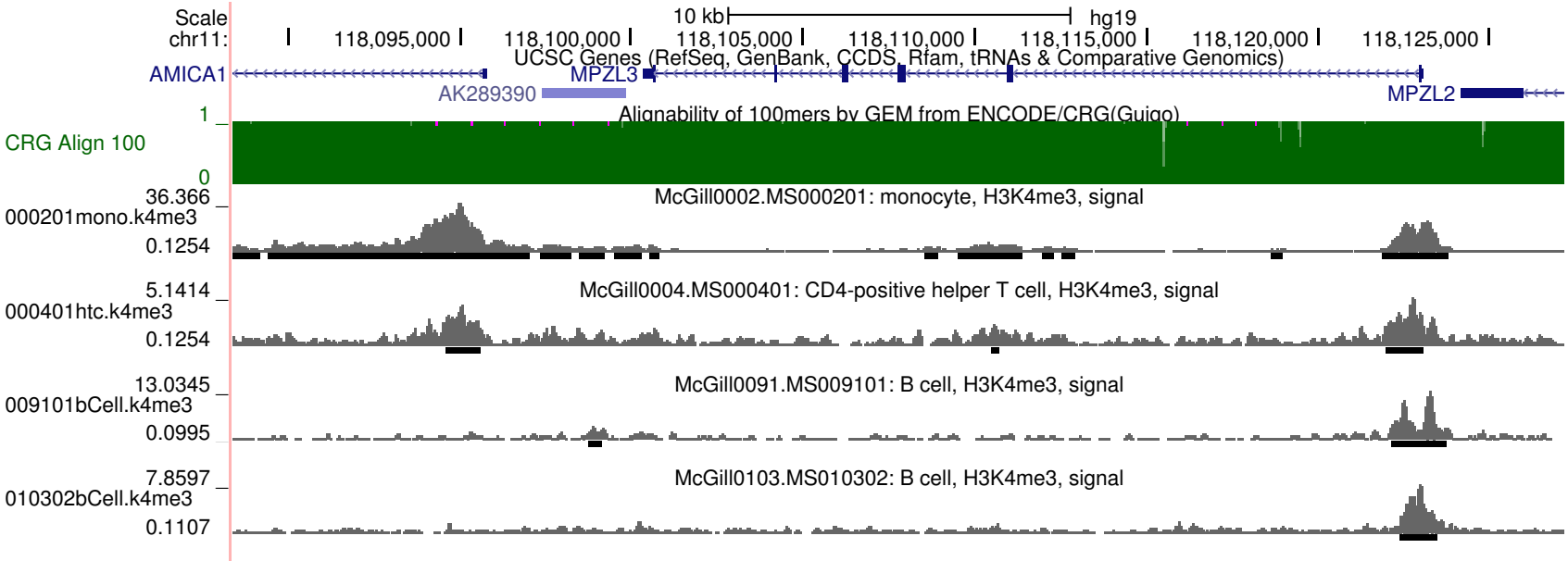
# Chromatin immunoprecipitation sequencing (ChIP-seq)

Analysis of DNA-protein interactions.




Source: "ChIP-sequencing," Wikipedia.

# Problem: find peaks in each of several samples




Grey profiles are normalized aligned read count signals.
Black bars are "peaks" called by MACS2 (Zhang et al, 2008):

- many false positives.
- overlapping peaks have different start/end positions.

# Previous work in genomic peak detection

- Model-based analysis of ChIP-Seq (MACS), Zhang et al, 2008.
- SICER, Zang et al, 2009.
- HOMER, Heinz et al, 2010.
- CCAT, Xu et al, 2010.
- RSEG, Song et al, 2011.
- Triform, Kornacker et al, 2012.
- Histone modifications in cancer (HMCan), Ashoor et al, 2013.
- PeakSeg, Hocking, Rigaill, Bourque, ICML 2015.
- PeakSegJoint Hocking and Bourque, arXiv:1506.01286.
- ... dozens of others.

Two big questions: how to choose the best...

- ...algorithm? (testing)
- ...parameters? (training)

# How to choose parameters of unsupervised peak detectors?

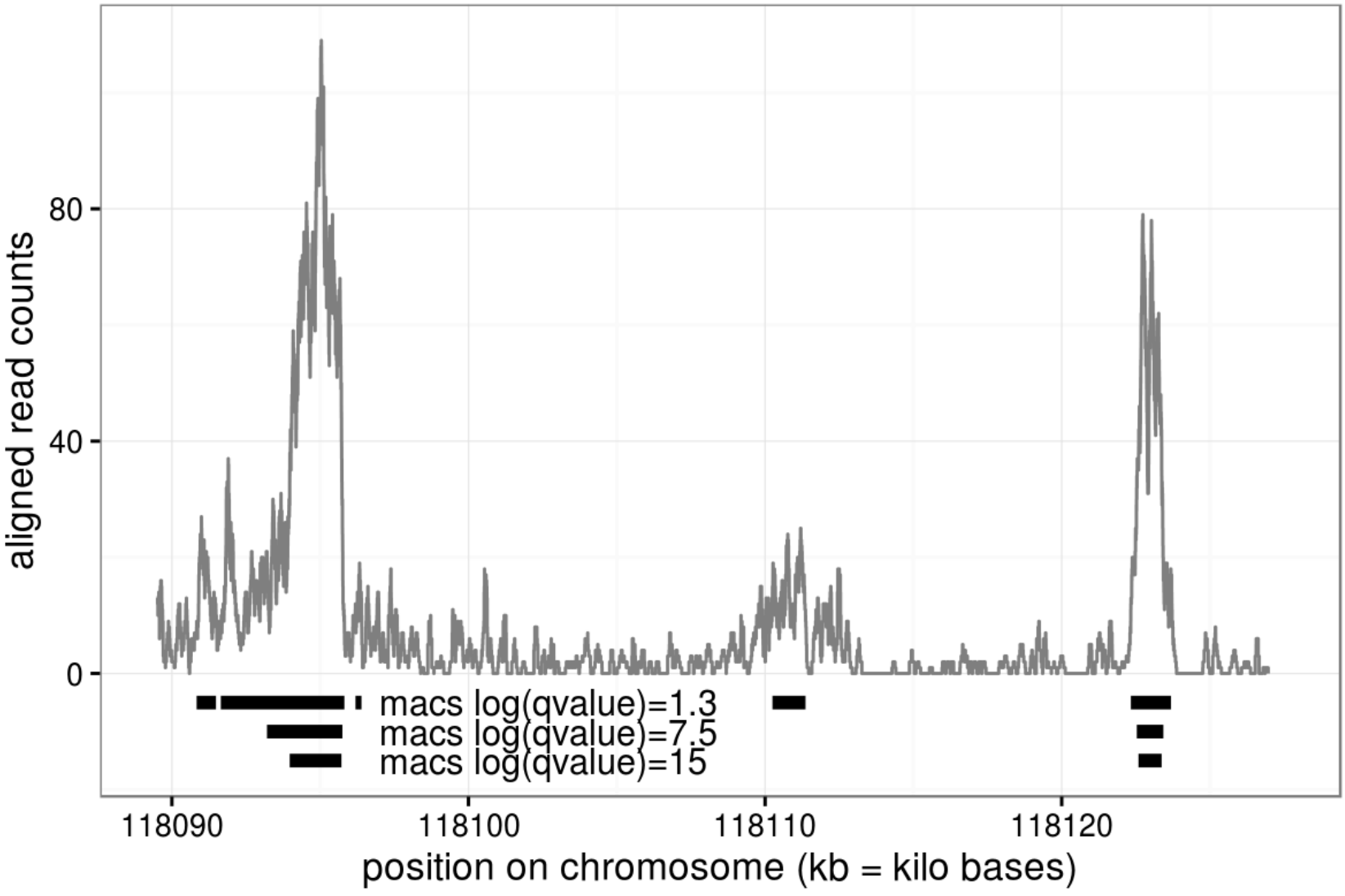
19 parameters for Model-based analysis of ChIP-Seq (MACS), Zhang et al, 2008.

```
[-g GSIZE]
[-s TSIZE] [--bw BW] [-m MFOLD MFOLD] [--fix-bimodal]
[--nomodel] [--extsize EXTSIZE | --shiftsize SHIFTSIZE]
[-q QVALUE | -p PVALUE | -F FOLDENRICHMENT] [--to-large]
[--down-sample] [--seed SEED] [--nolambda]
[--slocal SMALLLOCAL] [--llocal LARGELOCAL]
[--shift-control] [--half-ext] [--broad]
[--broad-cutoff BROADCUTOFF] [--call-summits]
```

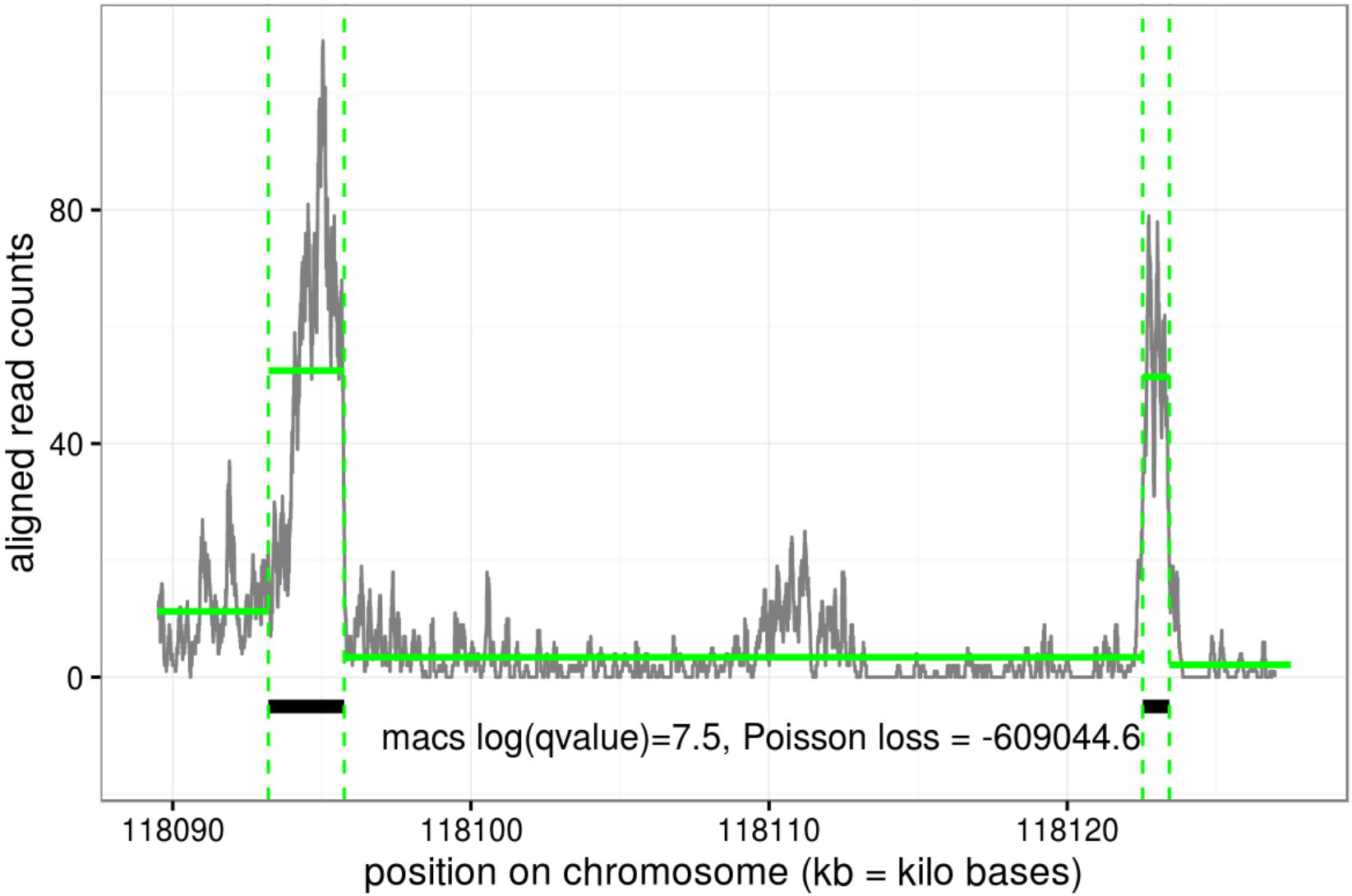
10 parameters for Histone modifications in cancer (HMCan), Ashoor et al, 2013.

```
minLength 145
medLength 150
maxLength 155
smallBinLength 50
largeBinLength 100000
pvalueThreshold 0.01
mergeDistance 200
iterationThreshold 5
finalThreshold 0
maxIter 20
```

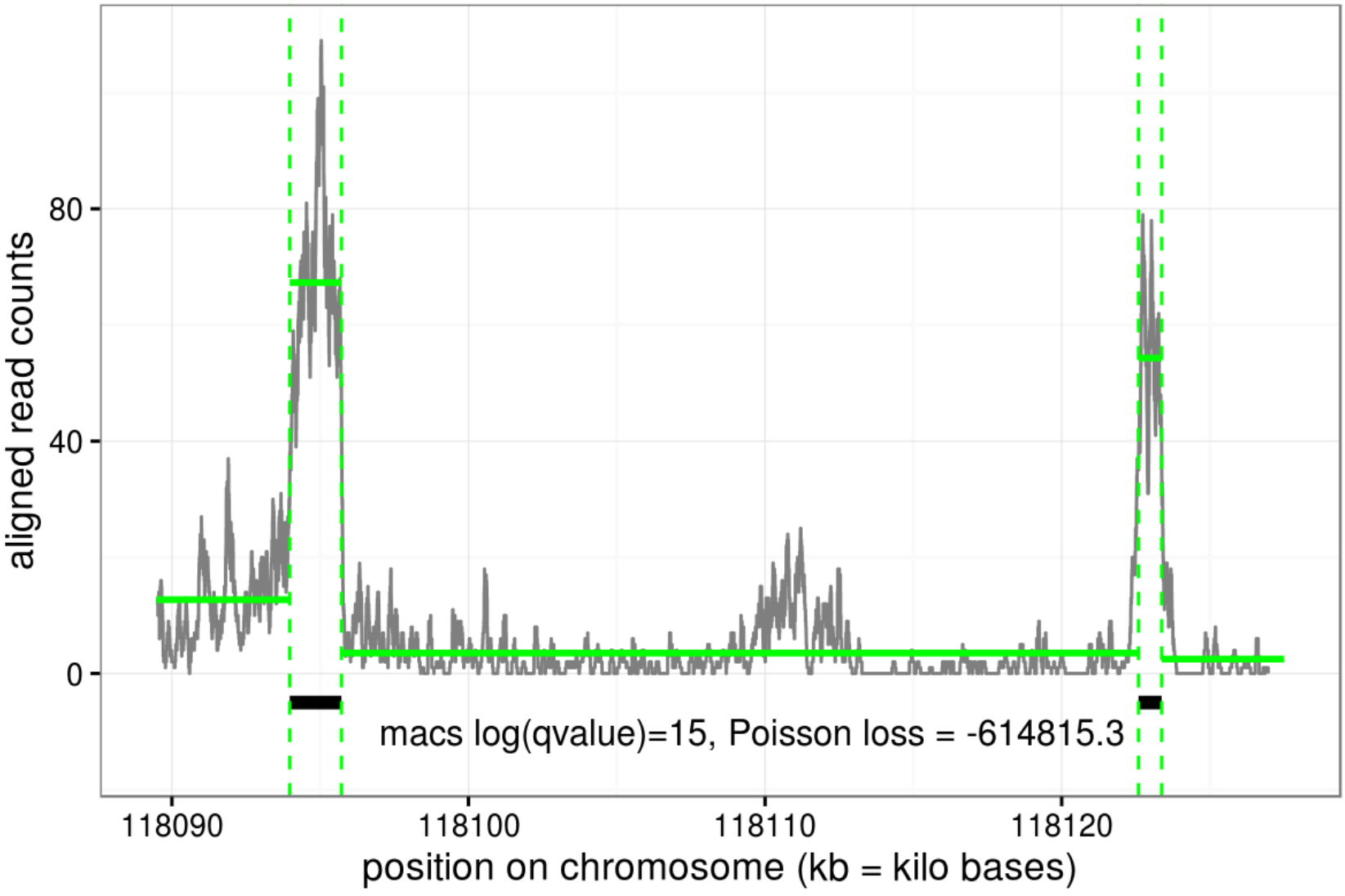
# Which macs parameter is best for these data?



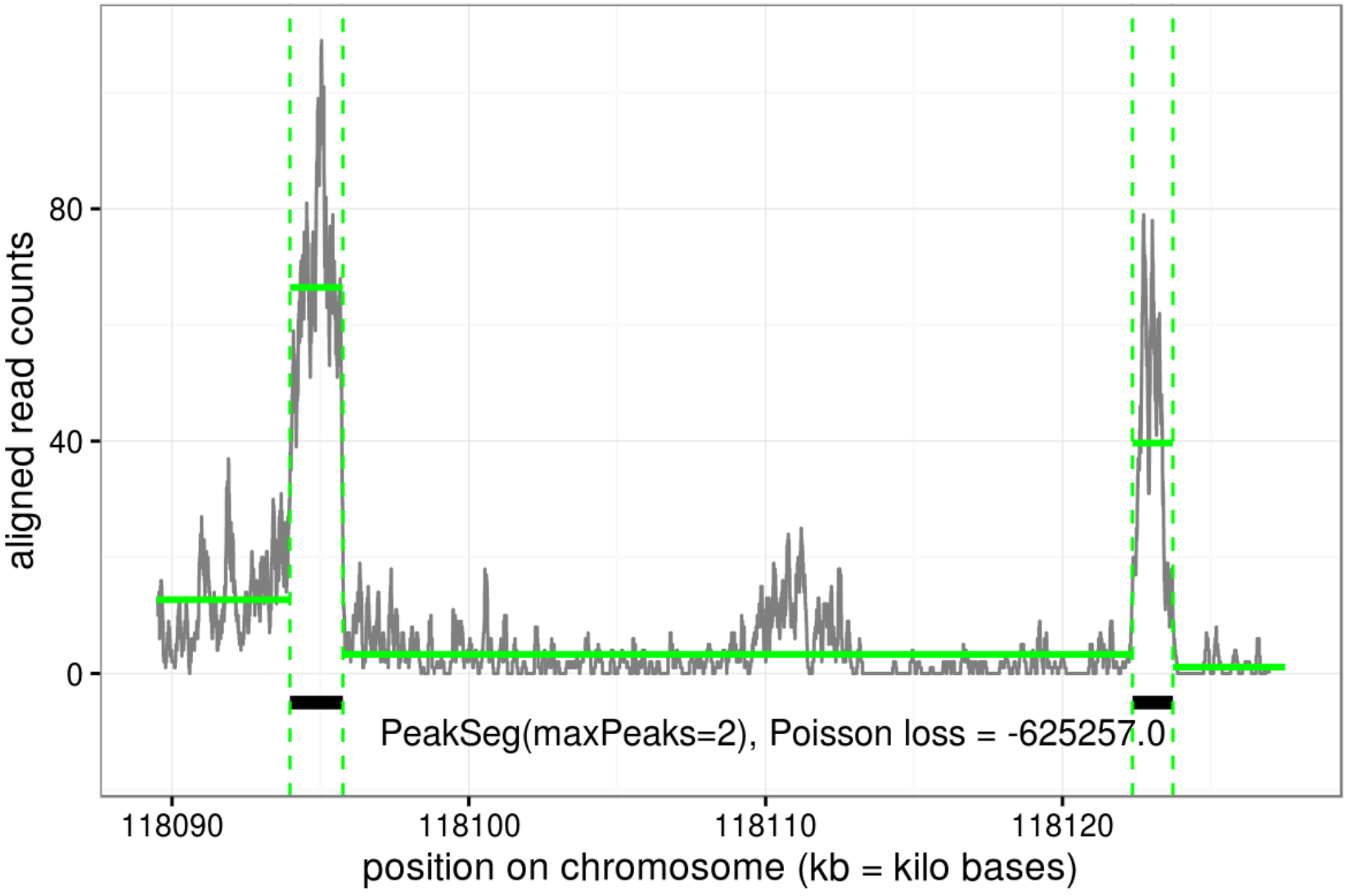
# Compute likelihood/loss of piecewise constant model



# Idea: choose the parameter with a lower loss

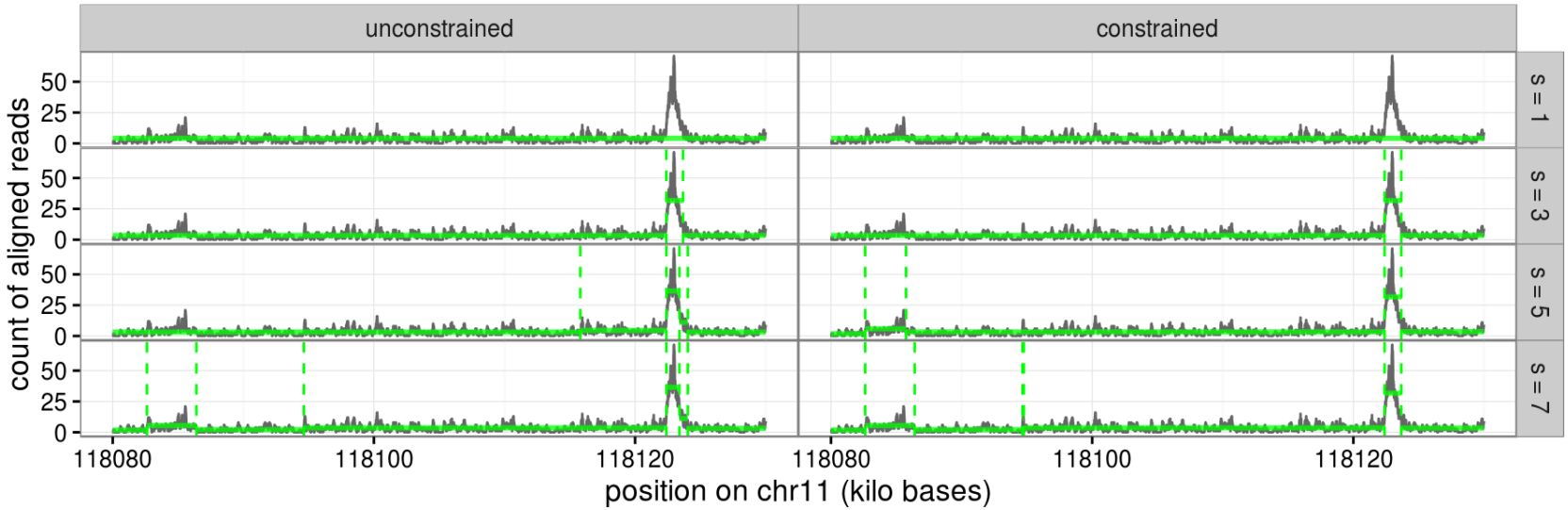


# PeakSeg: search for the peaks with lowest loss



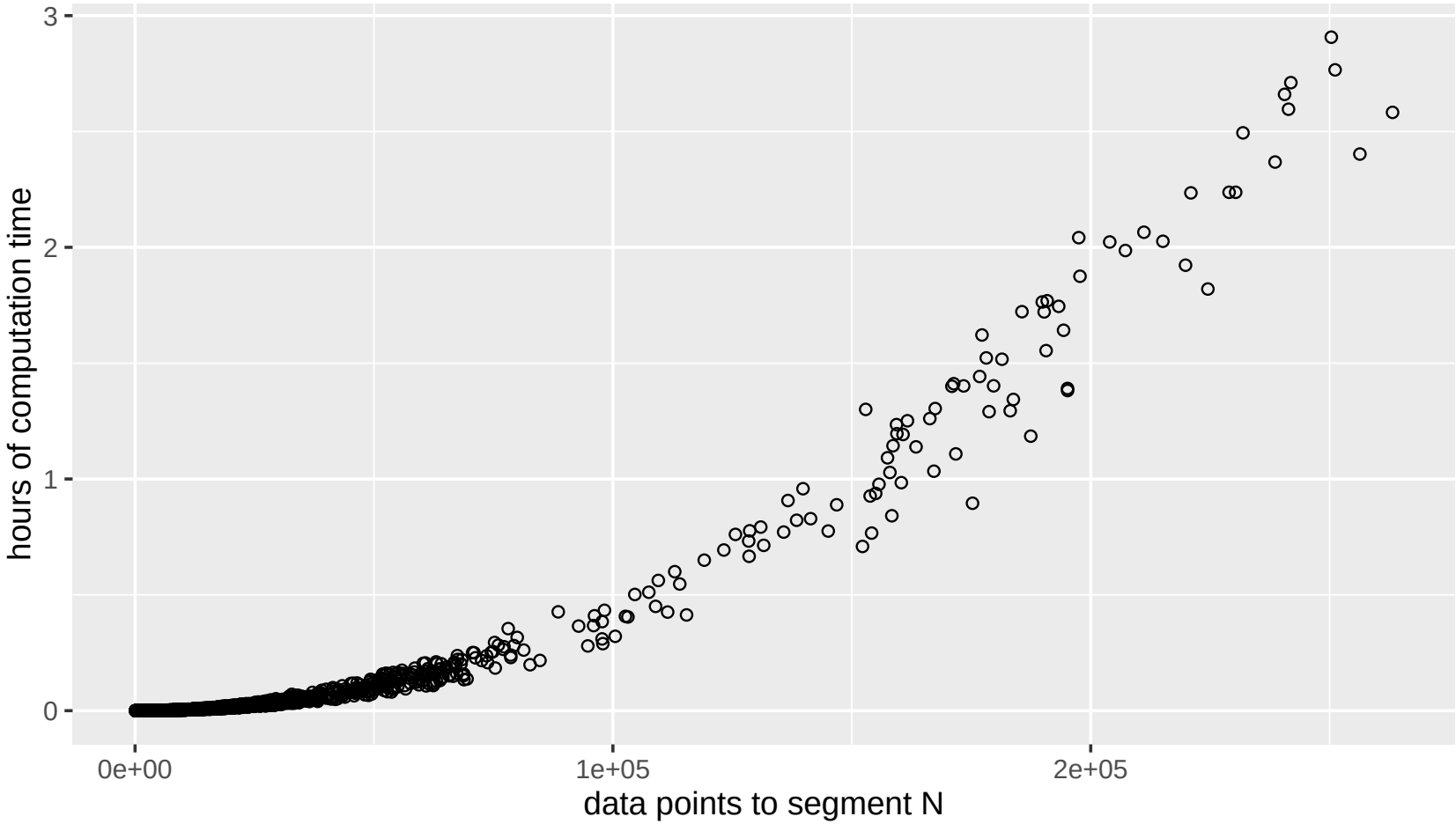

Choose the number of peaks via standard penalties (AIC, BIC, ...) or learned penalties based on visual labels (more on this later).

# Maximum likelihood Poisson segmentation models




- Previous work: unconstrained maximum likelihood mean for $s$ segments ($s-1$ changes).
- Hocking et al, ICML 2015: PeakSeg constraint enforces up, down, up, down changes (and not up, up, down).
- Odd-numbered segments are background noise, even-numbered segments are peaks.
- Constrained Dynamic Programming Algorithm, $O(N^2)$ time for $N$ data points.

# But quadratic time is not fast enough for genomic data!




- Genomic data is large, $N \geq 10^6$.
- Split into subsets? What if we split a peak in half?
- Need linear time algorithm for analyzing whole data set.

Problem: optimizing ChIP-seq peak detection

New linear time algorithm using functional pruning

Results on benchmark data sets

Conclusions and future work

# Statistical model is Poisson with change constraints

- We have $N$ count data $z_1, \ldots, z_N \in \mathbb{Z}_+$.
- Fix the number of segments $S \in \{1, 2, \ldots, N\}$.
- PeakSeg Model: $z_t \sim \text{Poisson}(m_t)$ such that $m_t$ has $S-1$ up-down changes.
- Want to find means $m_t$ which maximize the Poisson likelihood: $P(Z = z_t | m_t) = m_t^{z_t} e^{-m_t} / (z_t!)$.
- Equivalent to finding means $m_t$ which minimize the Poisson loss: $\ell(m_t, z_t) = m_t - z_t \log m_t$.
- Naive computation is $O(N^S)$, since there are $O(N^{S-1})$ possible positions for $S-1$ change-points, and it takes $O(N)$ operations to compute the mean and loss for each.
- Comparison to Hidden Markov Model:

  Likelihood Same emission terms, no transition terms.

  Constraint Number of changes rather than values.

# Relation to previous work

| | no pruning | functional pruning |
|---|---|---|
| unconstrained | Dynamic Programming | Pruned DP |
| | exact $O(N^2)$ | exact $O(N \log N)$ |
| R pkgs: | changepoint | cghseg, Segmentor |
| up-down constrained | constrained DP | **this work** |
| | inexact $O(N^2)$ | exact $O(N \log N)$ |
| R pkgs: | PeakSegDP | coseg |

- Auger and Lawrence 1989, Jackson et al 2005.
- Rigaill 2010, Johnson 2013, Cleynen et al 2014.
- Hocking, Rigaill, Bourque 2015.
- **Contribution**: new algorithm that **exactly** computes the **constrained** optimal segmentation for $N$ data points in linear $O(N \log N)$ time.

# Relation to previous work

| | no pruning | functional pruning |
|---|---|---|
| unconstrained | Dynamic Programming | Pruned DP |
| | exact $O(N^2)$ | exact $O(N \log N)$ |
| R pkgs: | changepoint | cghseg, Segmentor |
| up-down constrained | constrained DP | **this work** |
| | inexact $O(N^2)$ | exact $O(N \log N)$ |
| R pkgs: | PeakSegDP | coseg |

- Auger and Lawrence 1989, Jackson et al 2005.
- Rigaill 2010, Johnson 2013, Cleynen et al 2014.
- Hocking, Rigaill, Bourque 2015.
- **Contribution**: new algorithm that **exactly** computes the **constrained** optimal segmentation for $N$ data points in linear $O(N \log N)$ time.

# Relation to previous work

| | no pruning | functional pruning |
|---|---|---|
| unconstrained | Dynamic Programming | Pruned DP |
| | exact $O(N^2)$ | exact $O(N \log N)$ |
| R pkgs: | changepoint | cghseg, Segmentor |
| up-down constrained | constrained DP | **this work** |
| | inexact $O(N^2)$ | exact $O(N \log N)$ |
| R pkgs: | PeakSegDP | coseg |

- Auger and Lawrence 1989, Jackson et al 2005.
- Rigaill 2010, Johnson 2013, Cleynen et al 2014.
- Hocking, Rigaill, Bourque 2015.
- **Contribution**: new algorithm that **exactly** computes the **constrained** optimal segmentation for $N$ data points in linear $O(N \log N)$ time.

# Relation to previous work

| | no pruning | functional pruning |
|---|---|---|
| unconstrained | Dynamic Programming | Pruned DP |
| | exact $O(N^2)$ | exact $O(N \log N)$ |
| R pkgs: | changepoint | cghseg, Segmentor |
| up-down constrained | constrained DP | **this work** |
| | inexact $O(N^2)$ | exact $O(N \log N)$ |
| R pkgs: | PeakSegDP | coseg |

- Auger and Lawrence 1989, Jackson et al 2005.
- Rigaill 2010, Johnson 2013, Cleynen et al 2014.
- Hocking, Rigaill, Bourque 2015.
- **Contribution**: new algorithm that **exactly** computes the **constrained** optimal segmentation for $N$ data points in linear $O(N \log N)$ time.

# Dynamic programming and functional pruning

**Classical dynamic programming** (Auger and Lawrence 1989) computes the matrix of optimal loss values in $S$ segments up to $N$ data points, $O(SN^2)$

$$\begin{array}{ccc} \mathcal{L}_{1,1} & \cdots & \mathcal{L}_{1,N} \\ \vdots & & \vdots \\ \mathcal{L}_{S,1} & \cdots & \mathcal{L}_{S,N} \end{array}$$

**Dynamic programming with functional pruning** (Rigaill 2010) computes a matrix of loss **functions**, the optimal loss up to $N$ data points if segment $S$ has mean $\mu_S$, $O(SN \log N)$

$$\begin{array}{ccc} L_{1,1}(\mu_1) & \cdots & L_{1,N}(\mu_1) \\ \vdots & & \vdots \\ L_{S,1}(\mu_S) & \cdots & L_{S,N}(\mu_S), \end{array}$$

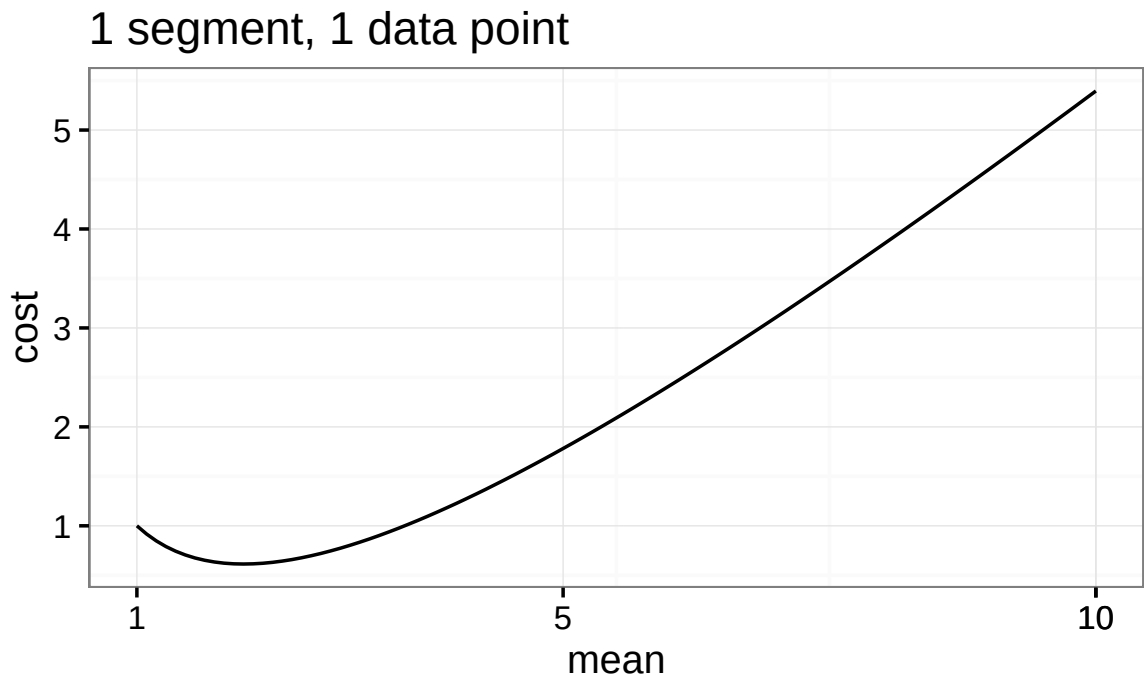
**Contribution of this work**: a new algorithm that applies the functional pruning technique to the up-down constrained model.

# First segment, first data point

- For data $z_1, \ldots, z_N \in \mathbb{Z}_+$ let

$$\gamma_t(\mu) = \ell(\mu, z_t) = \mu - z_t \log \mu$$

  be the Poisson loss for each $t \in \{1, \ldots, N\}$.
- For example $z = 2, 1, 9, 5, 10, 3$.
- Then $\gamma_1(\mu) = L_{1,1}(\mu) = 1\mu - 2 \log \mu + 0$.
- Need to store 3 coefficients (linear, log, constant).

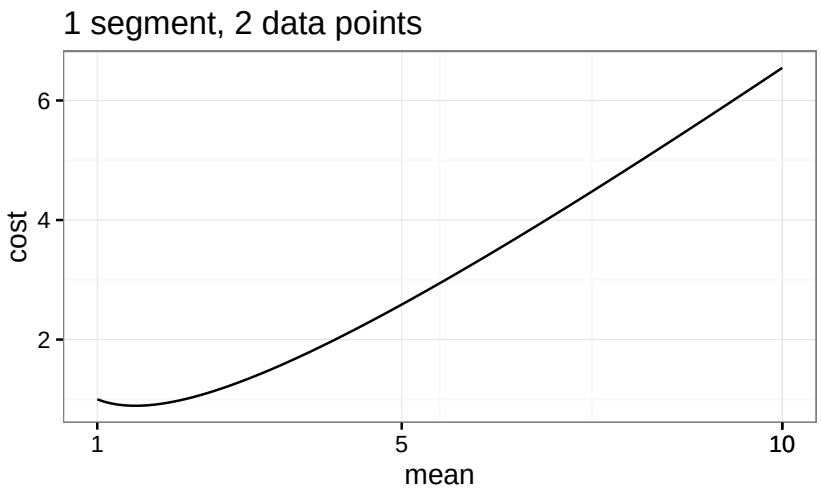


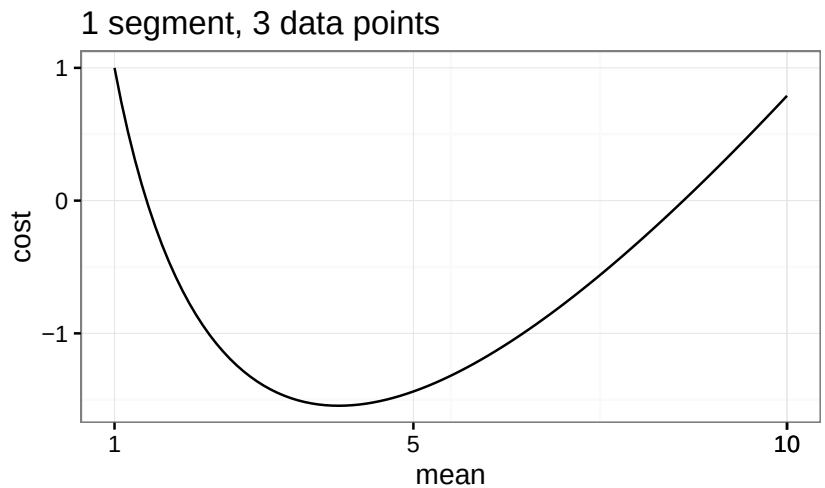
# First segment, other data points

- The loss of the first segment up to data point $t$ is

$$L_{1,t}(\mu) = \sum_{i=1}^{t} \gamma_i(\mu).$$

- For example $z = 2, 1, 9, 5, 10, 3$.
- $L_{1,2}(\mu) = 2\mu - 3\log\mu + 0$.
- $L_{1,3}(\mu) = 3\mu - 12\log\mu + 0$.
- ...



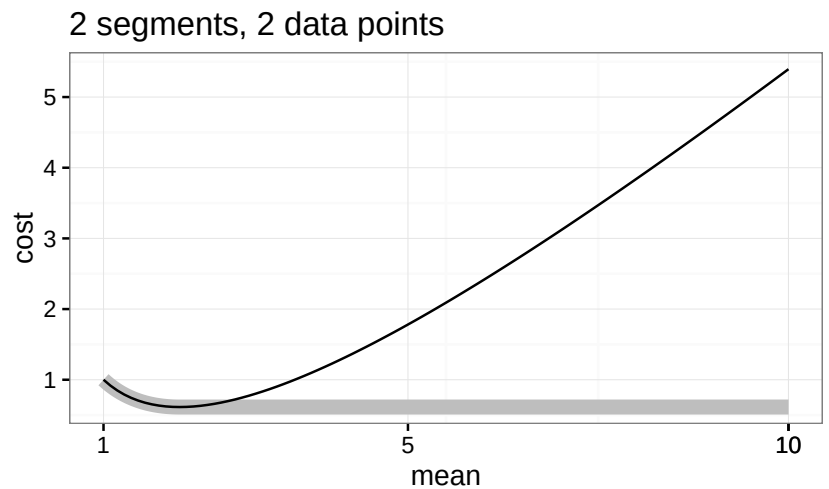



# Second segment, up to data point 2

- The mean cost in 2 segments up to data point 2 is

$$\begin{aligned} L_{2,2}(\mu_2) &= \gamma_2(\mu_2) + \min_{\mu_1 \leq \mu_2} L_{1,1}(\mu_1) \\ &= \gamma_2(\mu_2) + L^{\leq}_{1,1}(\mu_2) \end{aligned}$$

- Min-less operator is $L^{\leq}(\mu) = \min_{x \leq \mu} L(x)$,



# Comparison with unconstrained Pruned DPA

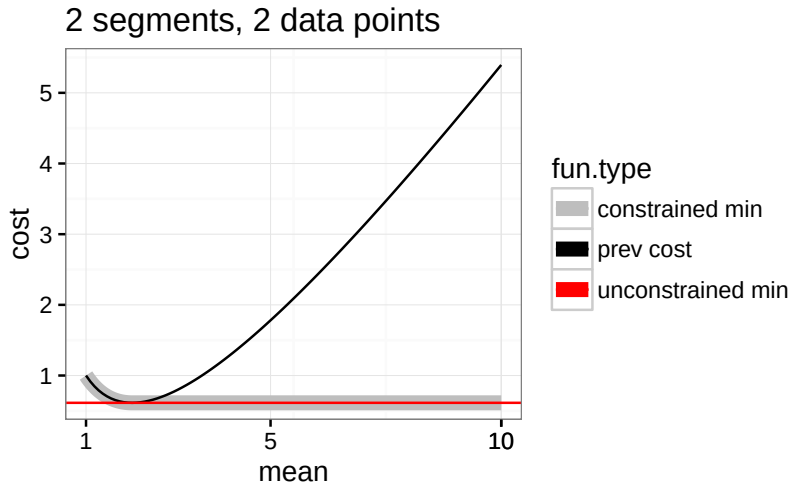
- For our constrained algorithm, the first segment mean must be less than the second, and the first segment cost is a function:

$$L_{2,2}(\mu_2) = \gamma_2(\mu_2) + \underbrace{\min_{\mu_1 \leq \mu_2} L_{1,1}(\mu_1)}_{L^{\leq}_{1,1}(\mu_2)}.$$

- For the unconstrained algorithm, it is constant:

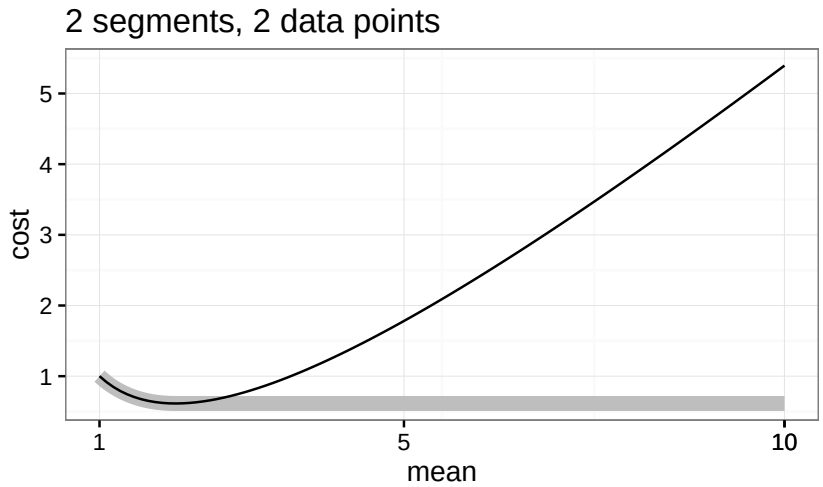
$$\widehat{L}_{2,2}(\mu_2) = \gamma_2(\mu_2) + \underbrace{\min_{\mu_1} L_{1,1}(\mu_1)}_{\mathcal{L}_{1,1}}.$$

- For example $z = 2, 1, 9, 5, 10, 3$.



# Storage as a piecewise function on intervals

- For example $z = 2, 1, 9, 5, 10, 3$.




- Storage:

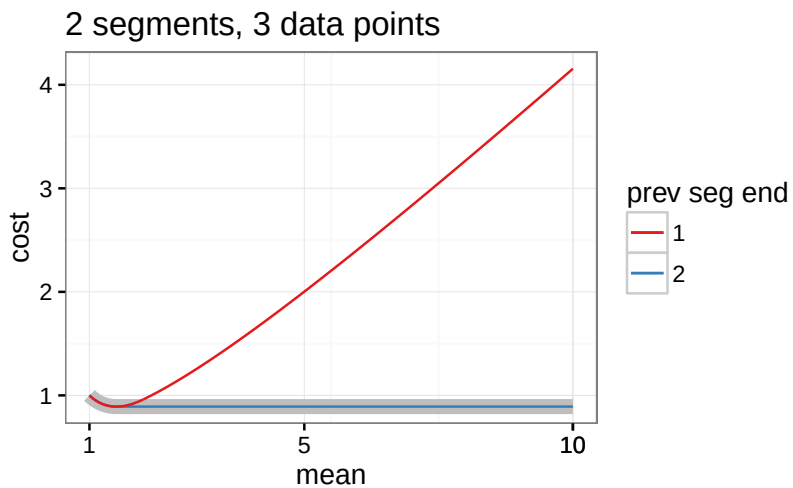
$$L_{2,2}(\mu) = \gamma_2(\mu) + \begin{cases} L_{1,1}(\mu) = 1\mu - 2\log\mu + 0 & \text{if } \mu \in [1, 2], \\ \mathcal{L}_{1,1} = 0\mu - 0\log\mu + 0.6137 & \text{if } \mu \in [2, 10]. \end{cases}$$

# Second segment, up to data point 3

- For data point 3 we need to consider two change-points:

$$L_{2,3}(\mu) = \gamma_3(\mu) + \min \begin{cases} L^{\leq}_{1,2}(\mu), & \text{change up after data point 2,} \\ L_{2,2}(\mu), & \text{change up after data point 1.} \end{cases}$$

- For $z = 2, 1, 9, 5, 10, 3$ the min operation prunes a change after data point 1.

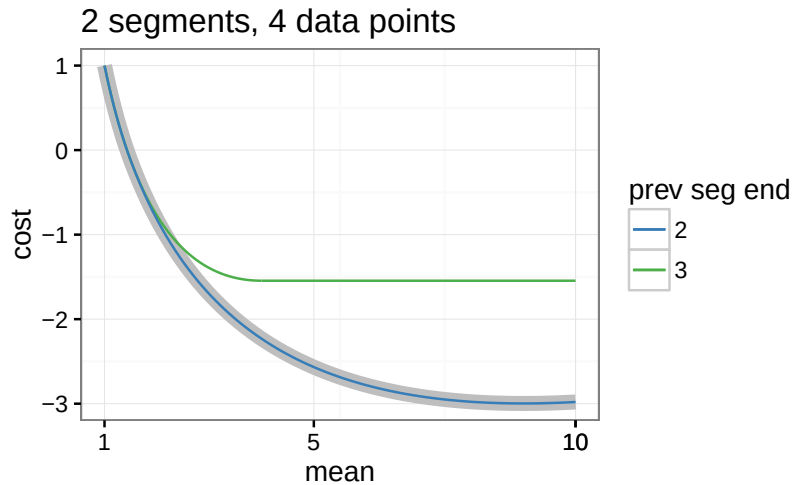


# Second segment, up to data point t

- The updates continue for every data point $t \in \{3, ..., N\}$

$$L_{2,t}(\mu) = \gamma_t(\mu) + \min \begin{cases} L^{\leq}_{1,t-1}(\mu), & \text{change up after } t-1, \\ L_{2,t-1}(\mu), & \text{change up before } t-1. \end{cases}$$

- For example for $z = 2, 1, 9, 5, 10, 3$, at data point $t = 4$ we only need to consider changes after 2 and 3 (1 has been pruned).

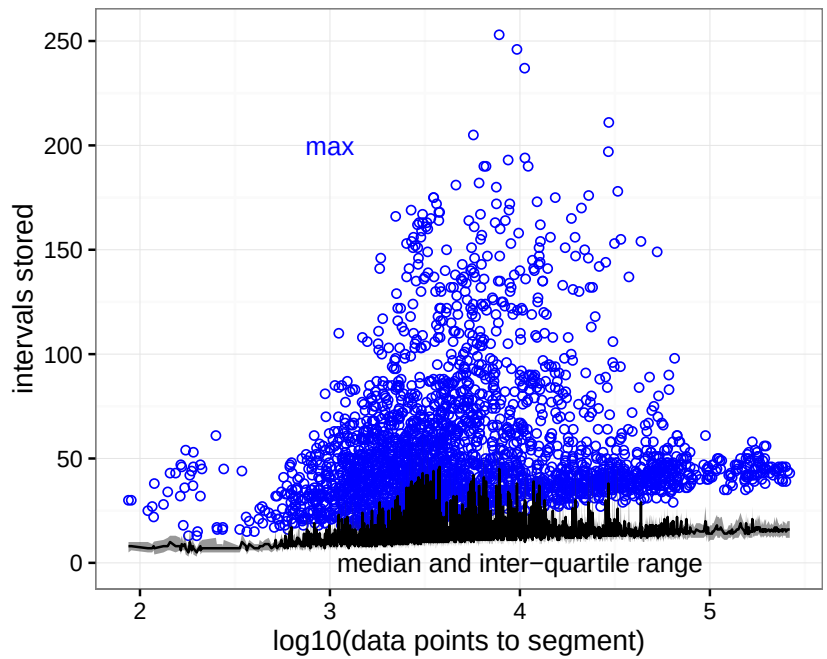


# General functional pruning equation

- The constrained cost of a mean $\mu$ for the segment $s$, up to data point $t$:

$$L_{s,t}(\mu) = \gamma_t(\mu) + \min \begin{cases} L_{s,t-1}(\mu), \\ L^*_{s-1,t-1}(\mu), \end{cases}$$

- Time complexity of min and min-less/more * is linear in the number of intervals, empirically sub-linear $O(\log N)$.




- Total time complexity: $O(SN \log N)$.

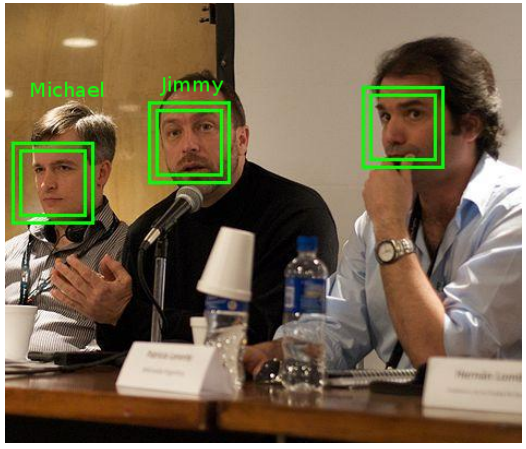
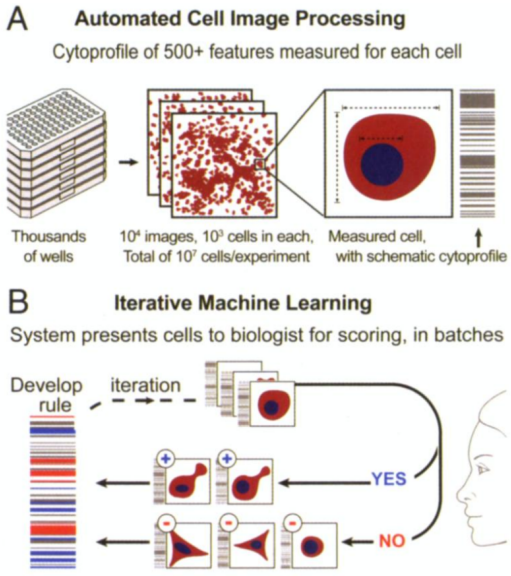
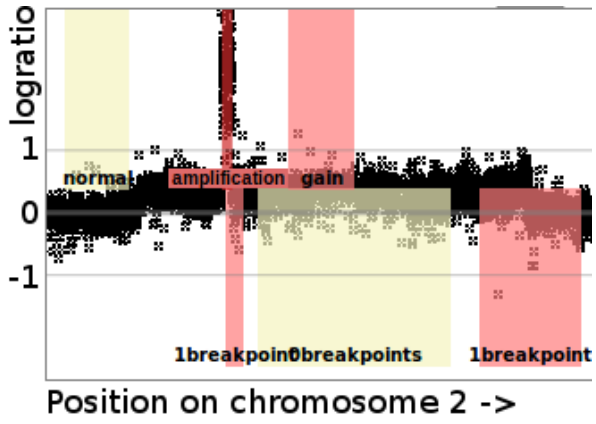
Problem: optimizing ChIP-seq peak detection

New linear time algorithm using functional pruning

Results on benchmark data sets

Conclusions and future work

# Previous work in computer vision: look and add labels to...

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

Sources: `http://en.wikipedia.org/wiki/Face_detection`
Jones et al PNAS 2009. Scoring diverse cellular morphologies in image-based screens with iterative feedback and machine learning.

# Benchmark data sets, algorithms

`http://cbio.ensmp.fr/~thocking/chip-seq-chunk-db/`

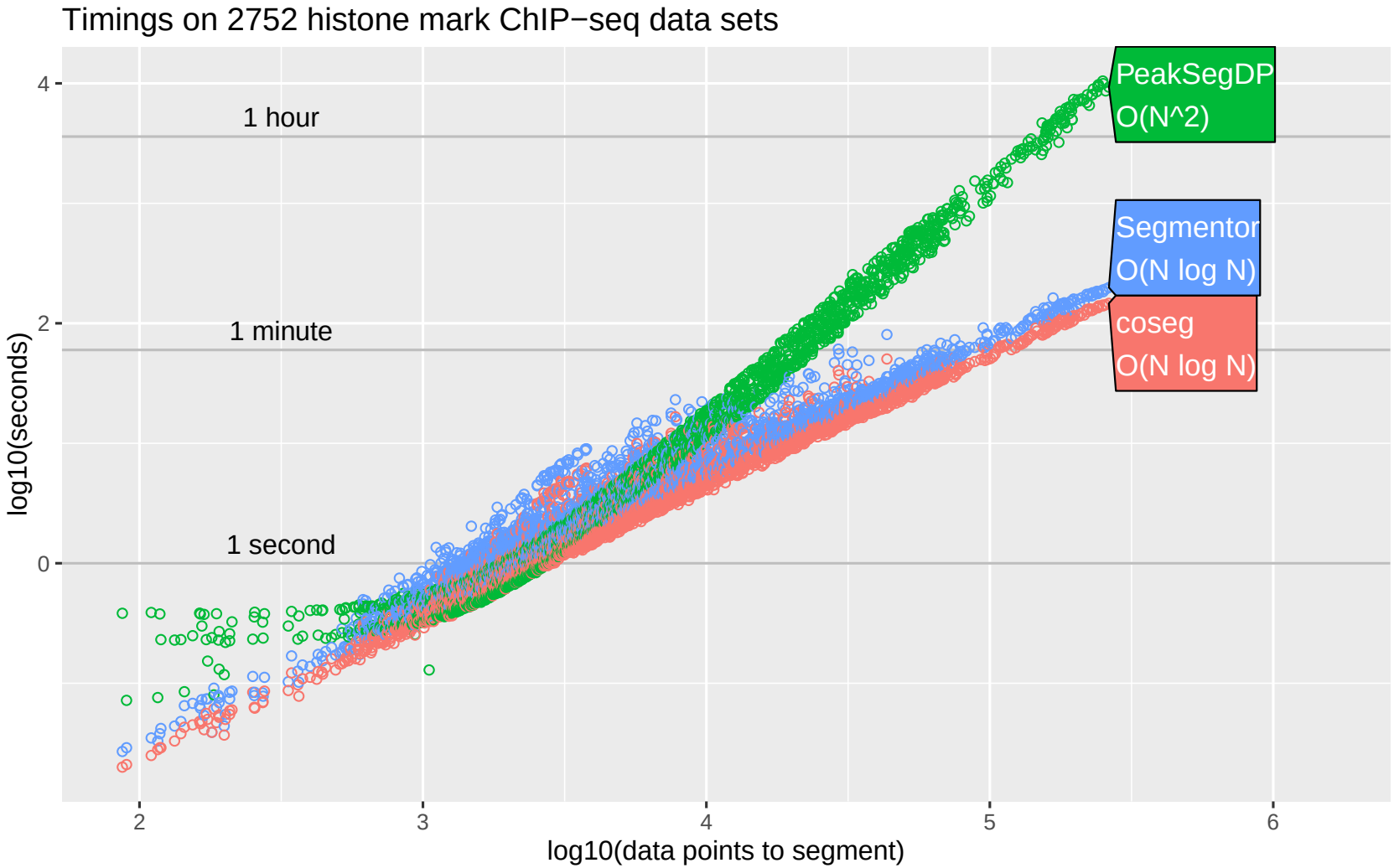
- Hocking *et al* Bioinformatics (2016). Optimizing ChIP-seq peak detectors using visual labels and supervised machine learning.
- 37 labeled H3K4me3 samples (sharp peak pattern).
- 29 labeled H3K36me3 samples (broad peak pattern).
- 12,826 labeled regions with and without peaks.
- 2,752 separate segmentation problems.

Algorithms for segmenting $N$ data points:

| package | constraint | exact? | complexity |
|---|---|---|---|
| **coseg (this work)** | $\mu_1 \leq \mu_2 \geq \mu_3 \ldots$ | yes | $O(N \log N)$ |
| PeakSegDP | $\mu_1 < \mu_2 > \mu_3 \ldots$ | no | $O(N^2)$ |
| Segmentor | none | yes | $O(N \log N)$ |

Segmentor loss $\leq$ coseg loss $\leq$ PeakSegDP loss.

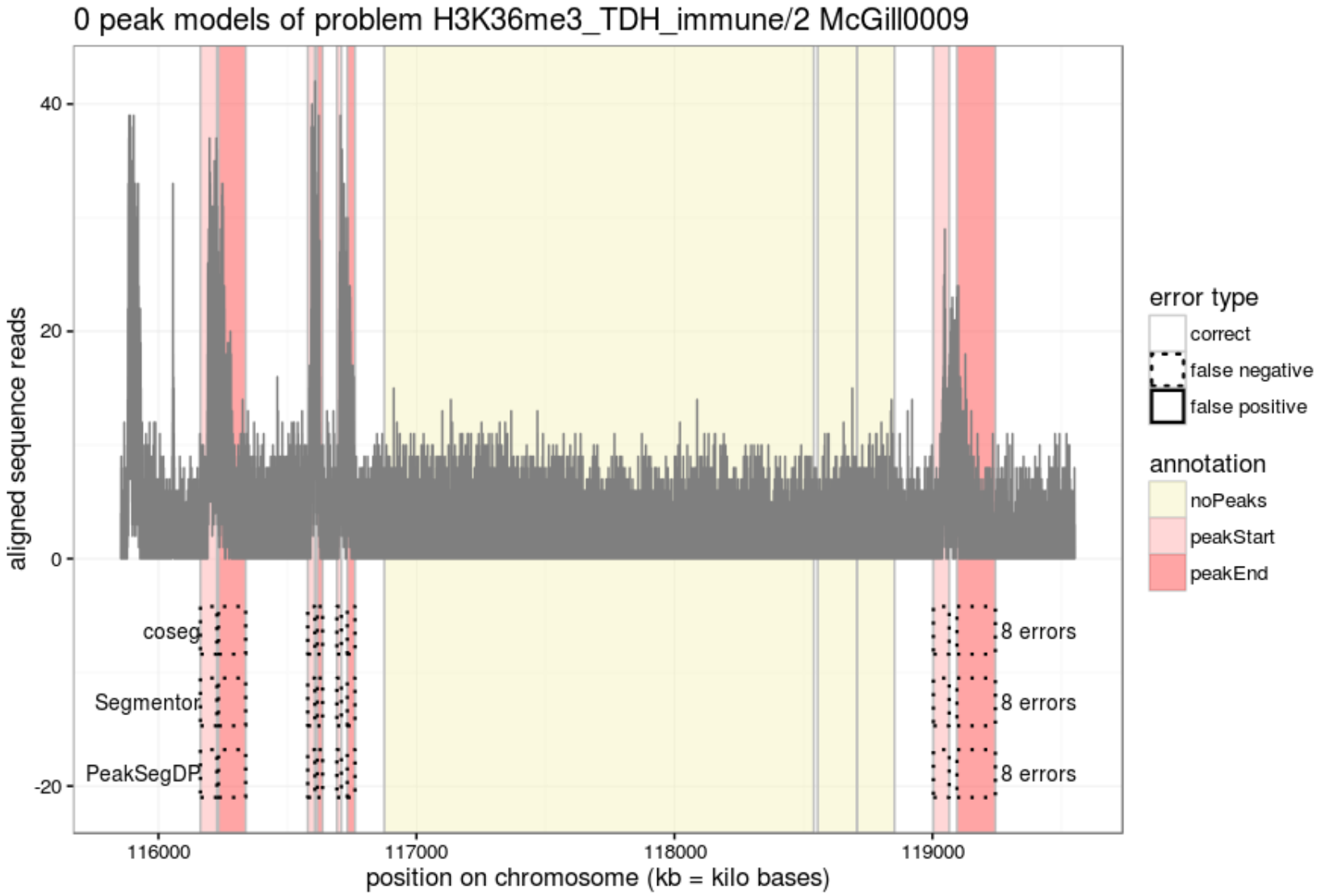
# Linear time algorithms faster for larger data sets



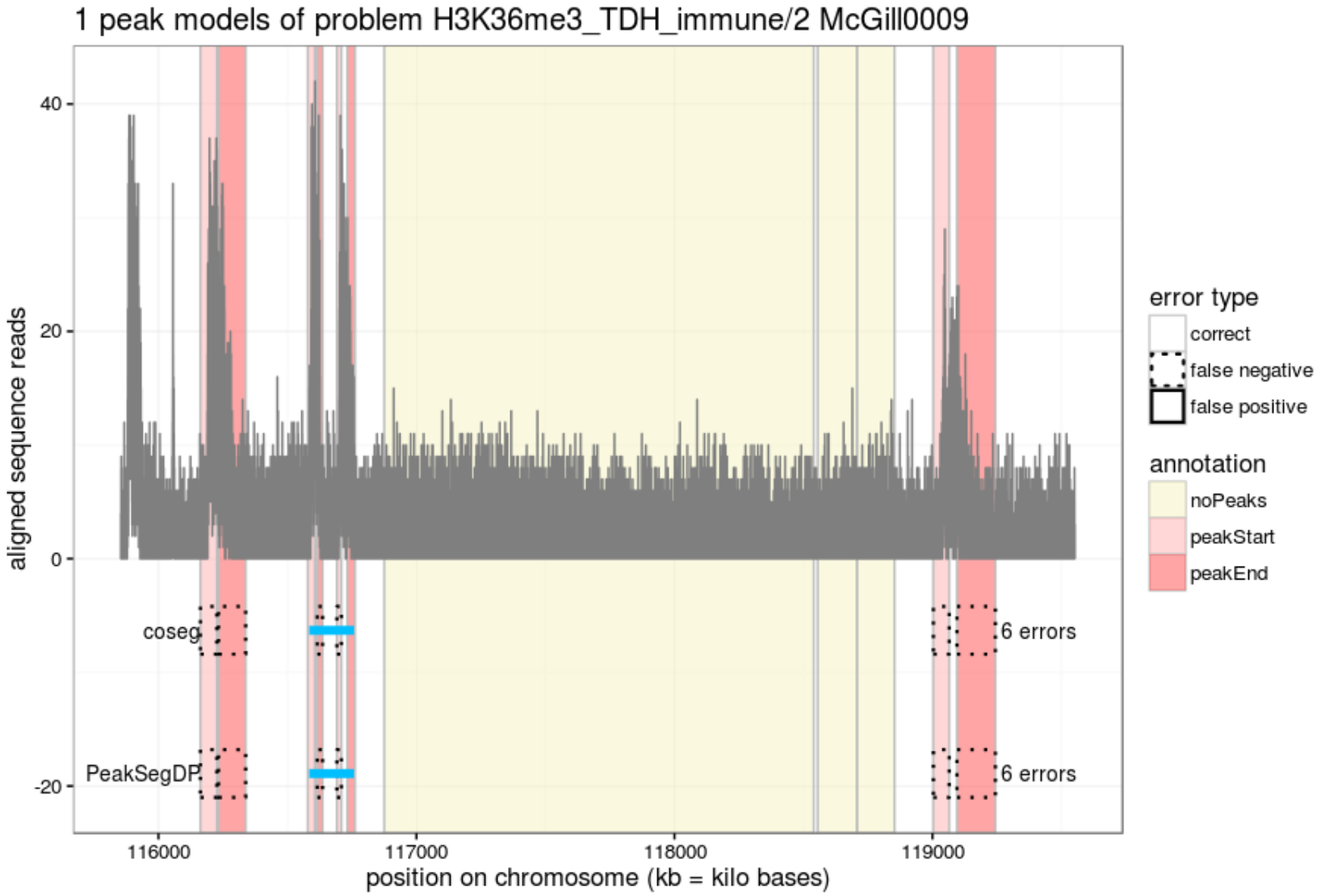

Total time to compute 10 models (0, ..., 9 peaks) for all data sets:

- PeakSegDP: 156 hours, inexact.
- coseg: 6 hours, exact.

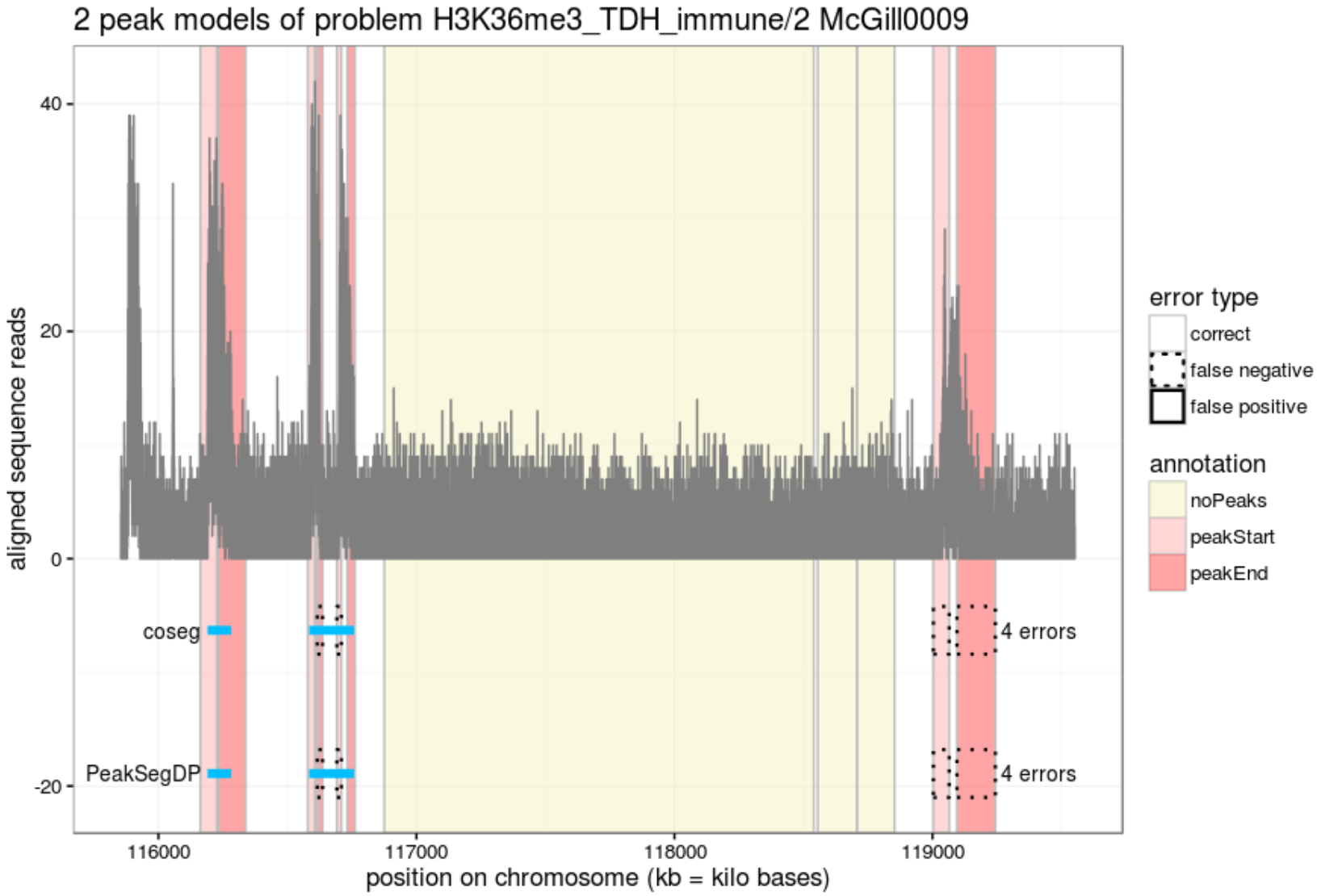
# 8 false negative labels for models with 0 peaks



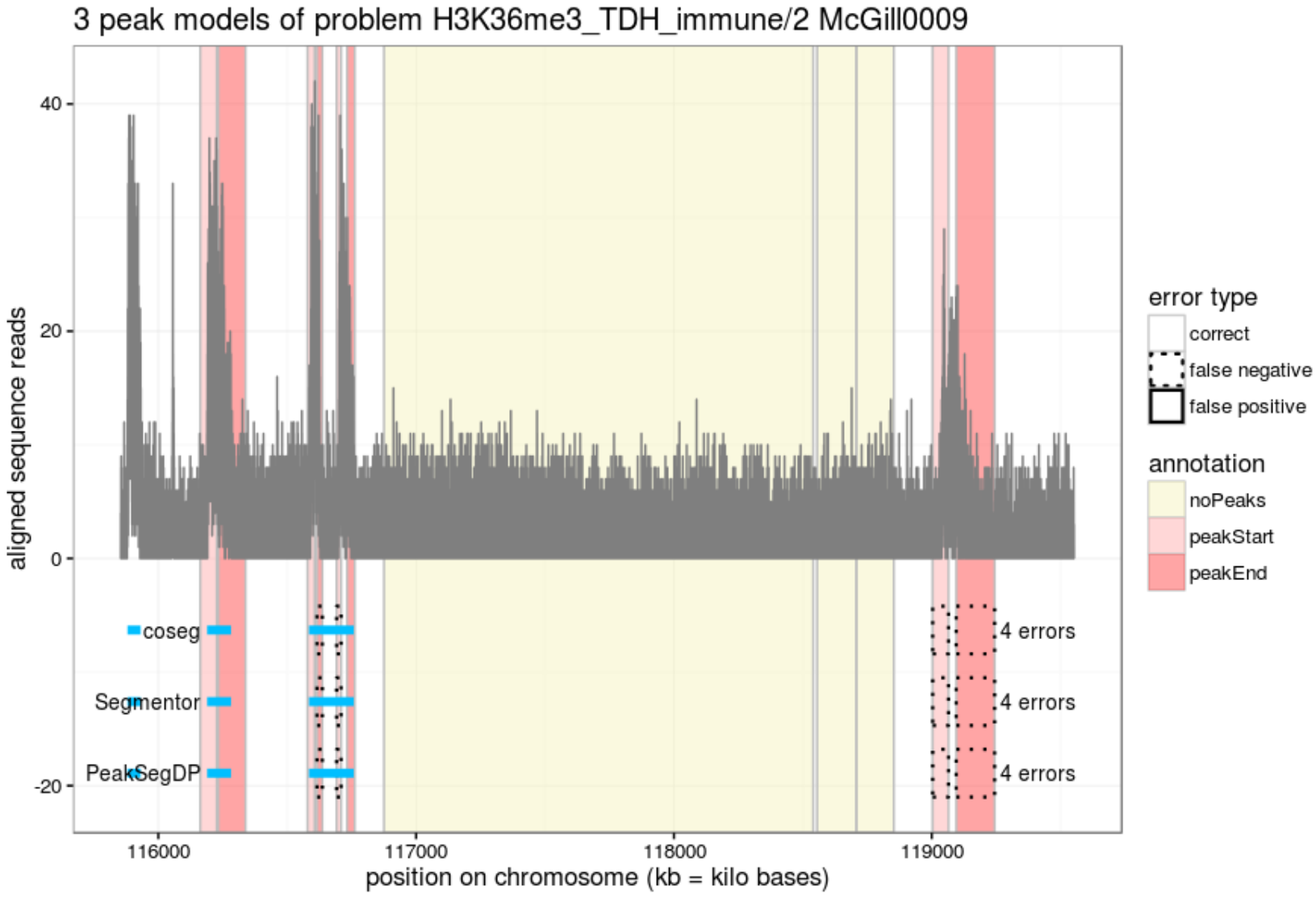
# Models with 1 peak are better (6 FN)



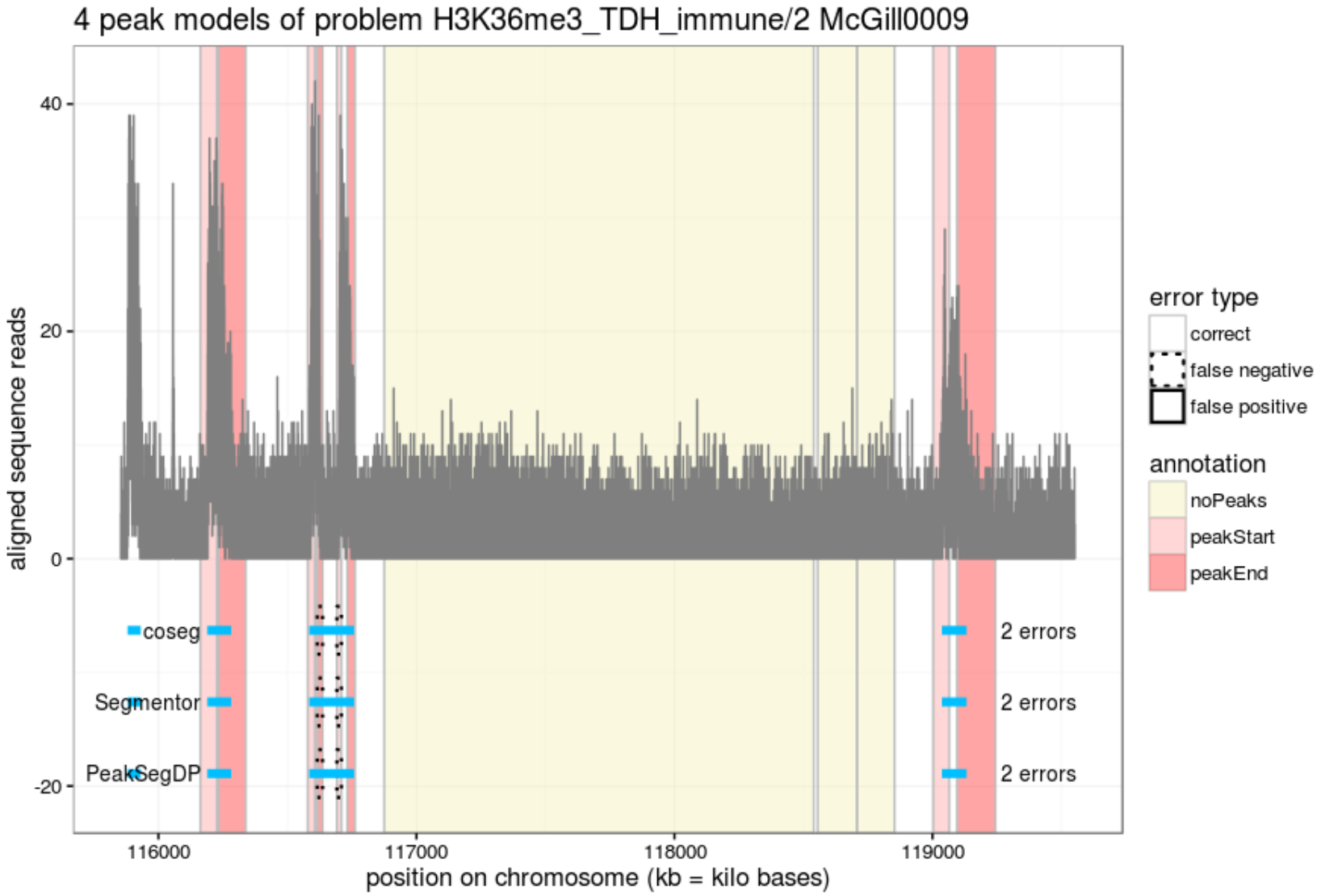
# Models with 2 peaks are better still (4 FN)



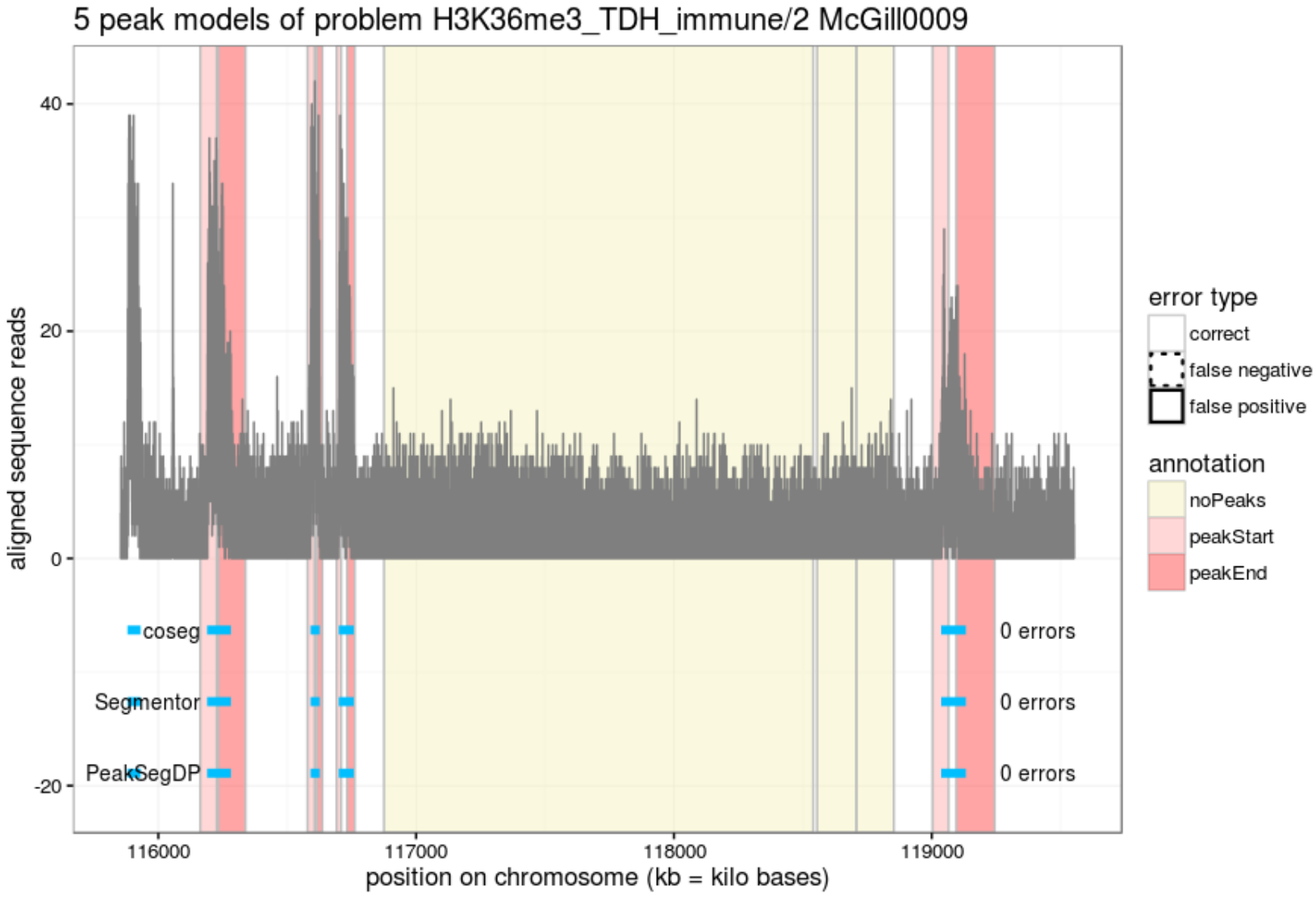
# Models with 3 peaks are the same (4 FN)



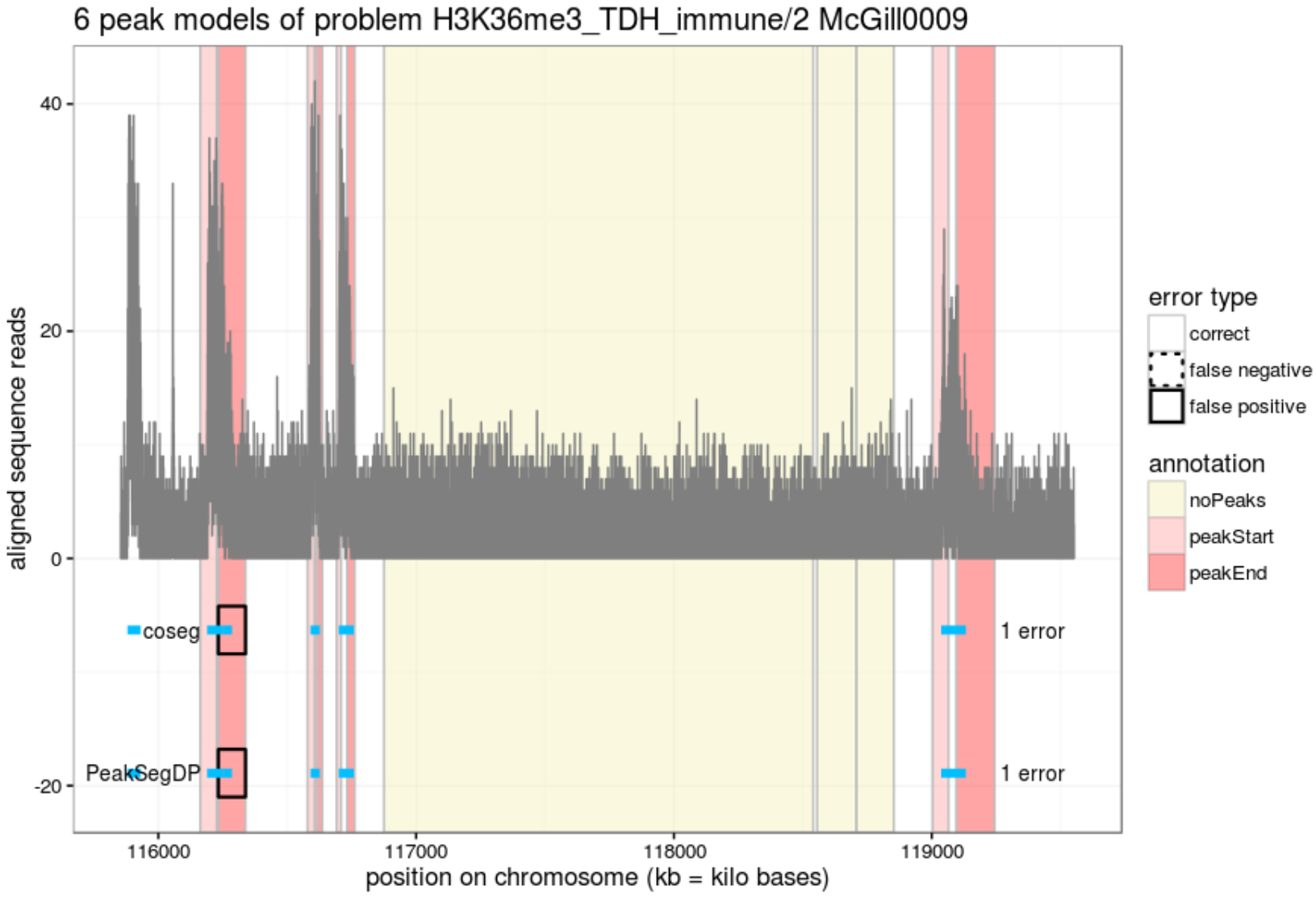
# Models with 4 peaks are better (2 FN)



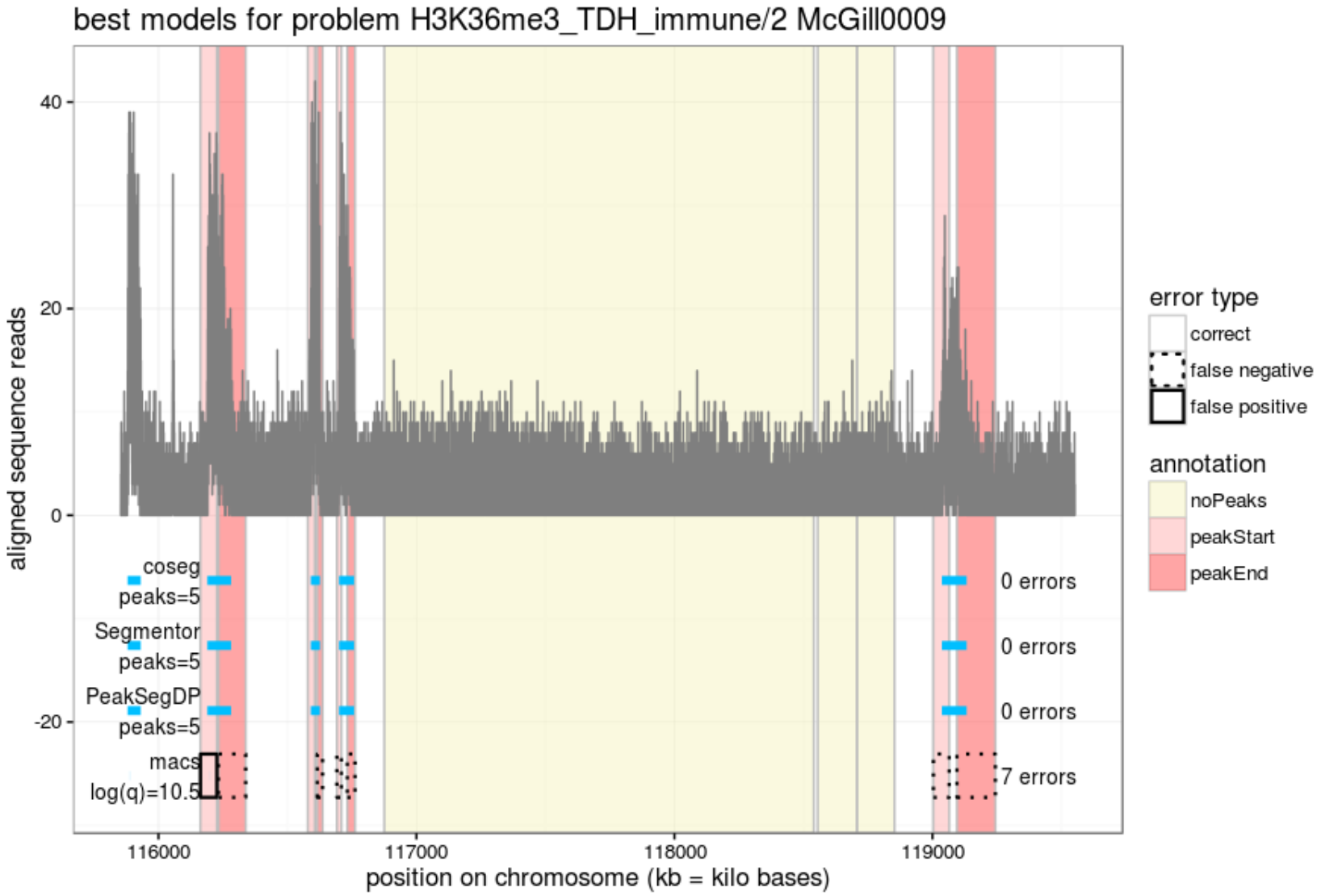
# Models with 5 peaks have no incorrect labels



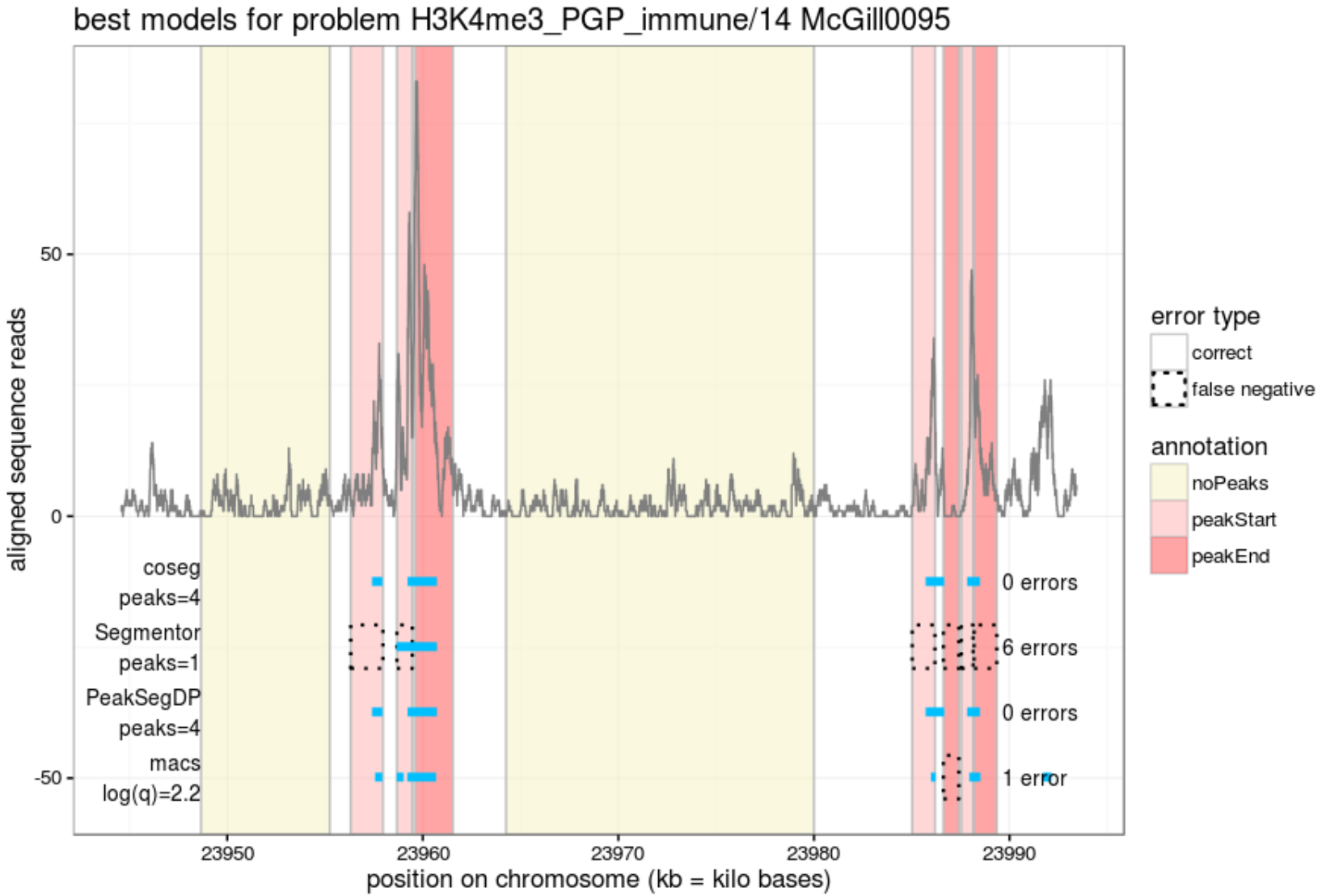
# Models with 6 peaks are worse (1 false positive)



# Constrained optimization better than macs



# 0 errors for coseg/PeakSegDP, 6 errors for Segmentor

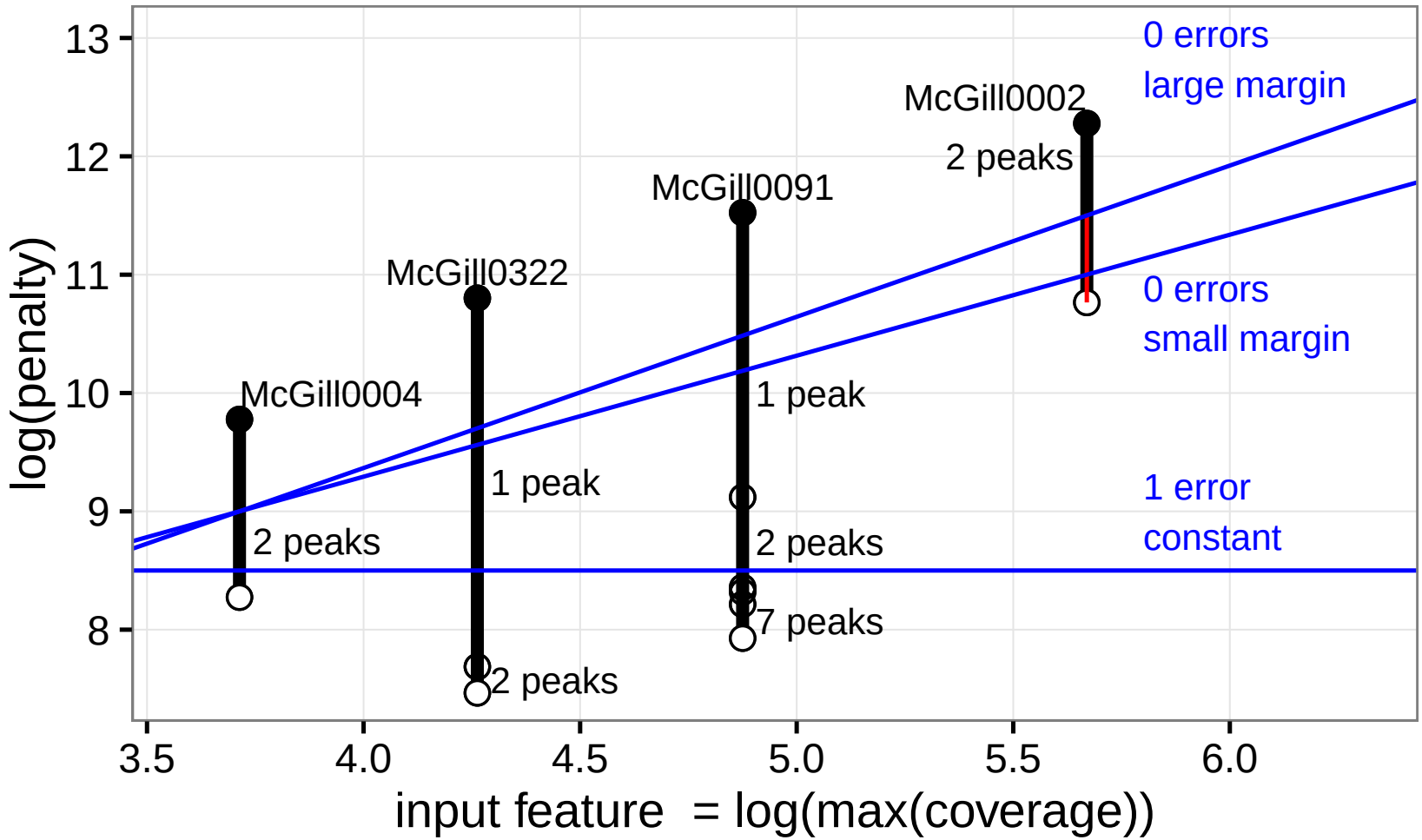


# Minimum train error in all data sets

| | errors | fp | fn | models | problems |
|---|---|---|---|---|---|
| PeakSegDP | 677 | 116 | 561 | 27469 | 2738 |
| coseg | 789 | 94 | 695 | 21278 | 1498 |
| macs | 1293 | 519 | 774 | | |
| Segmentor | 1544 | 46 | 1498 | 8106 | 4 |
| hmcan.broad | 2778 | 367 | 2411 | | |
| possible | 12826 | 11037 | 7225 | 27520 | 2752 |

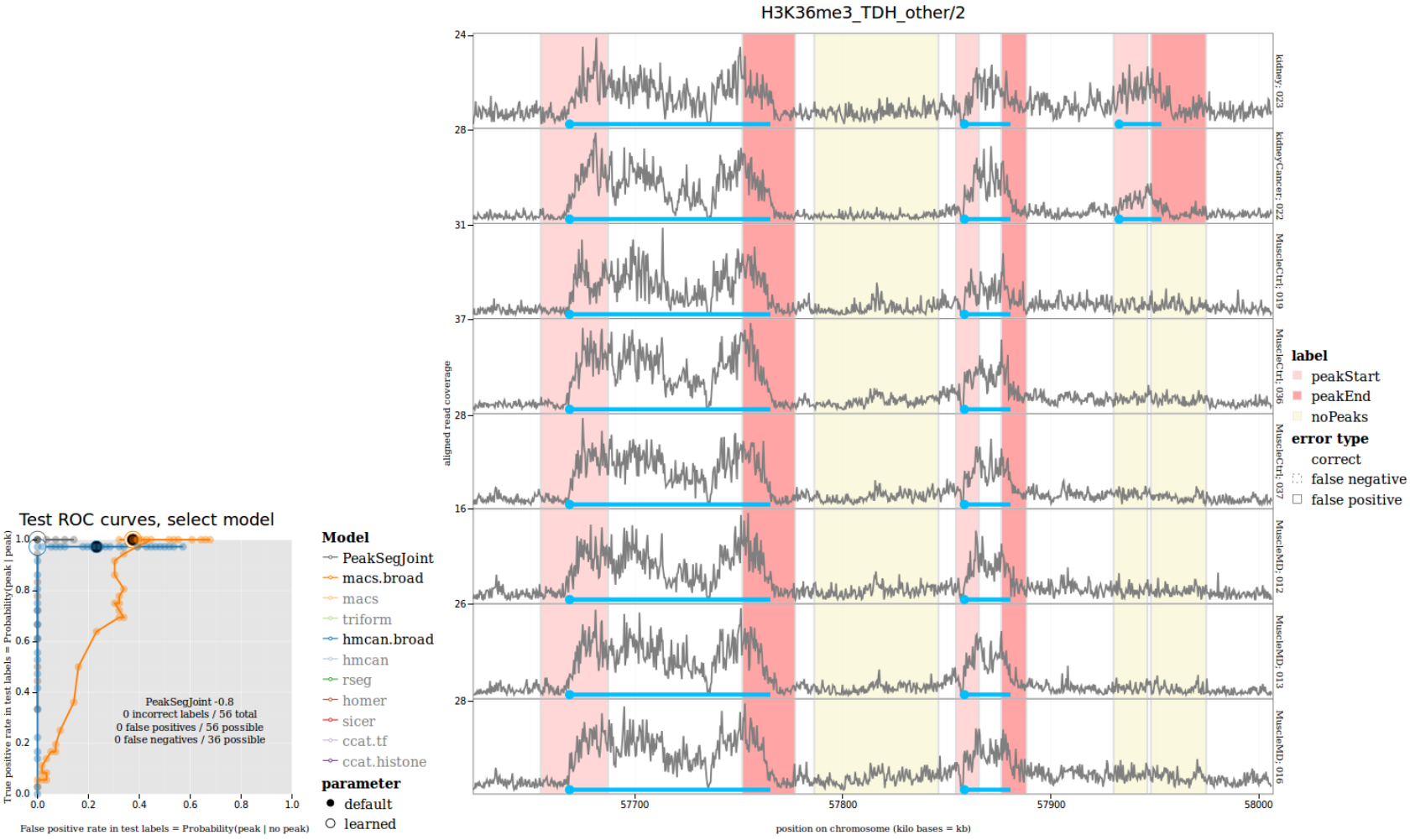
- Segmentor, PeakSegDP, coseg were used to compute up to 10 models for each problem (1,...,19 segments = 0,...,9 peaks).
- errors = fp + fn = total number of incorrect labels, after picking best parameter for each of the 2752 separate problems.
- models = number that obey the up-down constraint.
- New coseg algorithm has minimum train error almost as good as slower PeakSegDP algorithm.
- Other algorithms much less accurate.

# Max-margin penalty learning algortihm



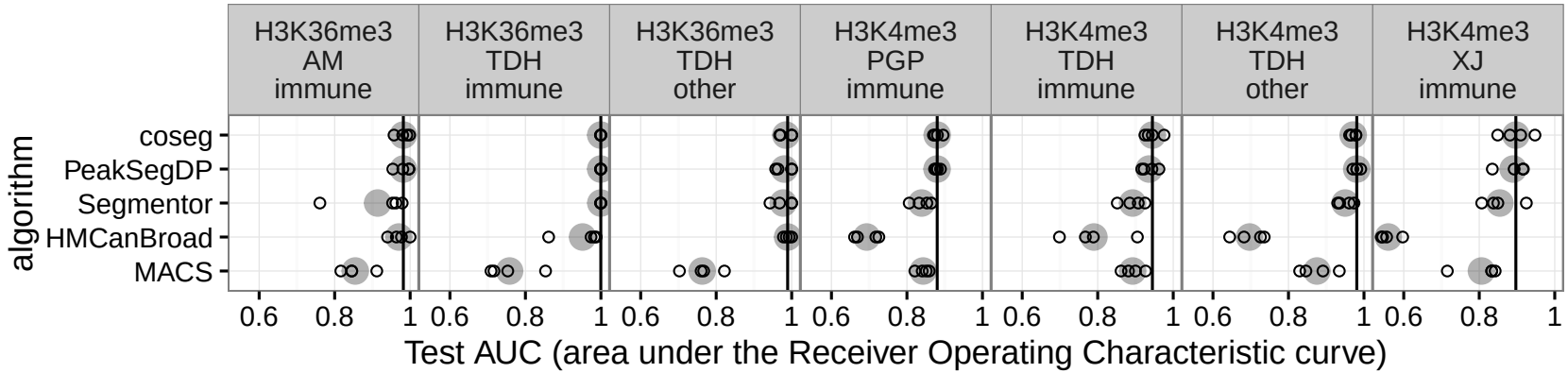

`http://bl.ocks.org/tdhock/raw/9311ca39d643d127e04a088814c81ee1/`

# ROC curves for one test fold




http://cbio.ensmp.fr/~thocking/chip-seq-chunk-db/figure-roc-test/

# Test AUC on 7 benchmark data sets




- 4-fold cross-validation: train on 3/4 of labels, test on 1/4.
- HMCanBroad is accurate for broad H3K36me3 data but not sharp H3K4me3 data.
- MACS is accurate for sharp H3K4me3 but not broad H3K36me3 data.
- Unconstrained Segmentor algorithm not as accurate as up-down constrained algorithms (coseg, PeakSegDP).
- Proposed algorithm in coseg R package yields state-of-the-art accuracy in all benchmark data sets.

`http://bl.ocks.org/tdhock/raw/886575874144c3b172ce6b7d7d770b9f/`

Problem: optimizing ChIP-seq peak detection

New linear time algorithm using functional pruning

Results on benchmark data sets

Conclusions and future work

# How to choose parameters of unsupervised peak detectors?

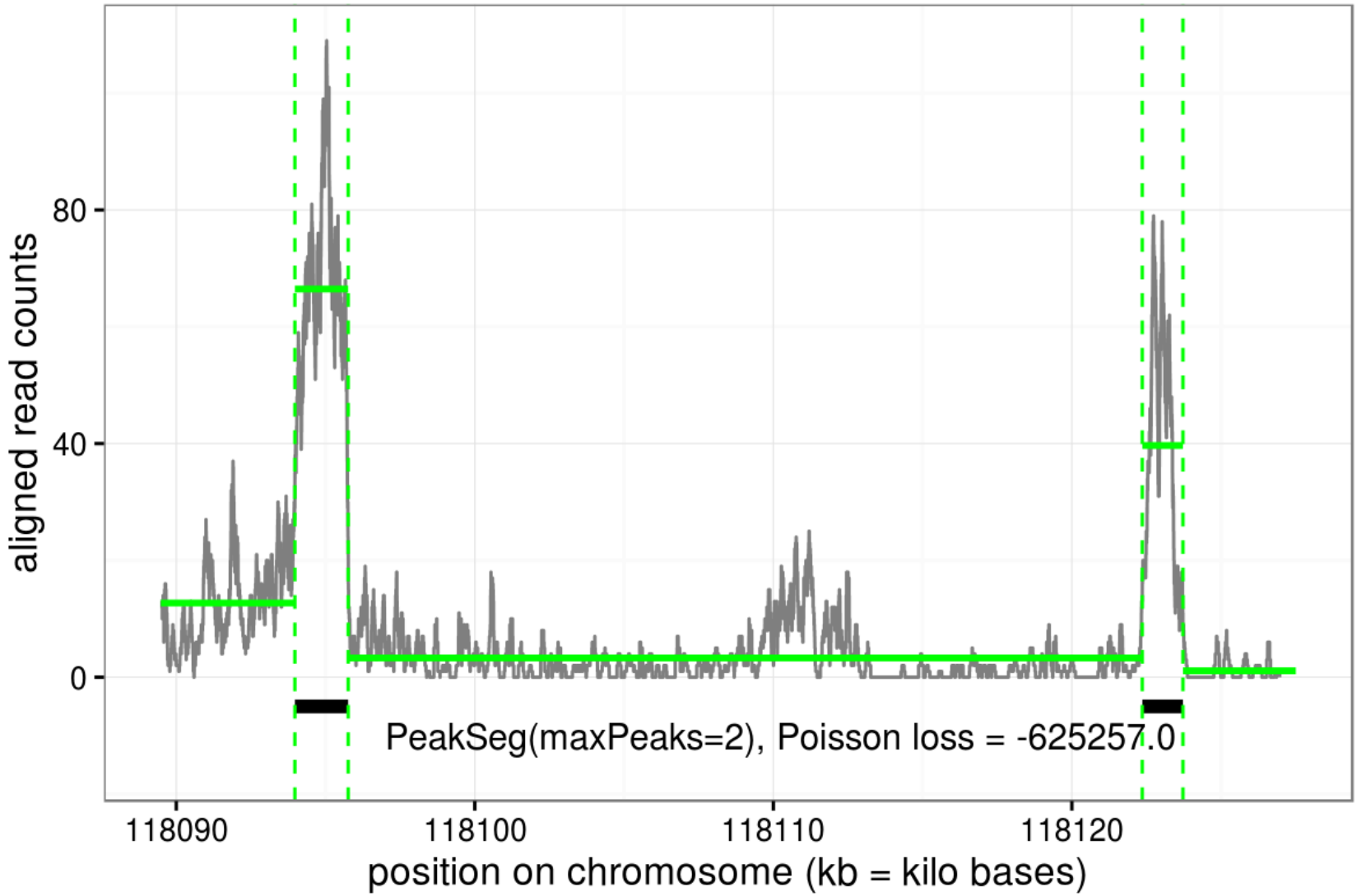
19 parameters for Model-based analysis of ChIP-Seq (MACS), Zhang et al, 2008.

```
[-g GSIZE]
[-s TSIZE] [--bw BW] [-m MFOLD MFOLD] [--fix-bimodal]
[--nomodel] [--extsize EXTSIZE | --shiftsize SHIFTSIZE]
[-q QVALUE | -p PVALUE | -F FOLDENRICHMENT] [--to-large]
[--down-sample] [--seed SEED] [--nolambda]
[--slocal SMALLLOCAL] [--llocal LARGELOCAL]
[--shift-control] [--half-ext] [--broad]
[--broad-cutoff BROADCUTOFF] [--call-summits]
```

10 parameters for Histone modifications in cancer (HMCan), Ashoor et al, 2013.

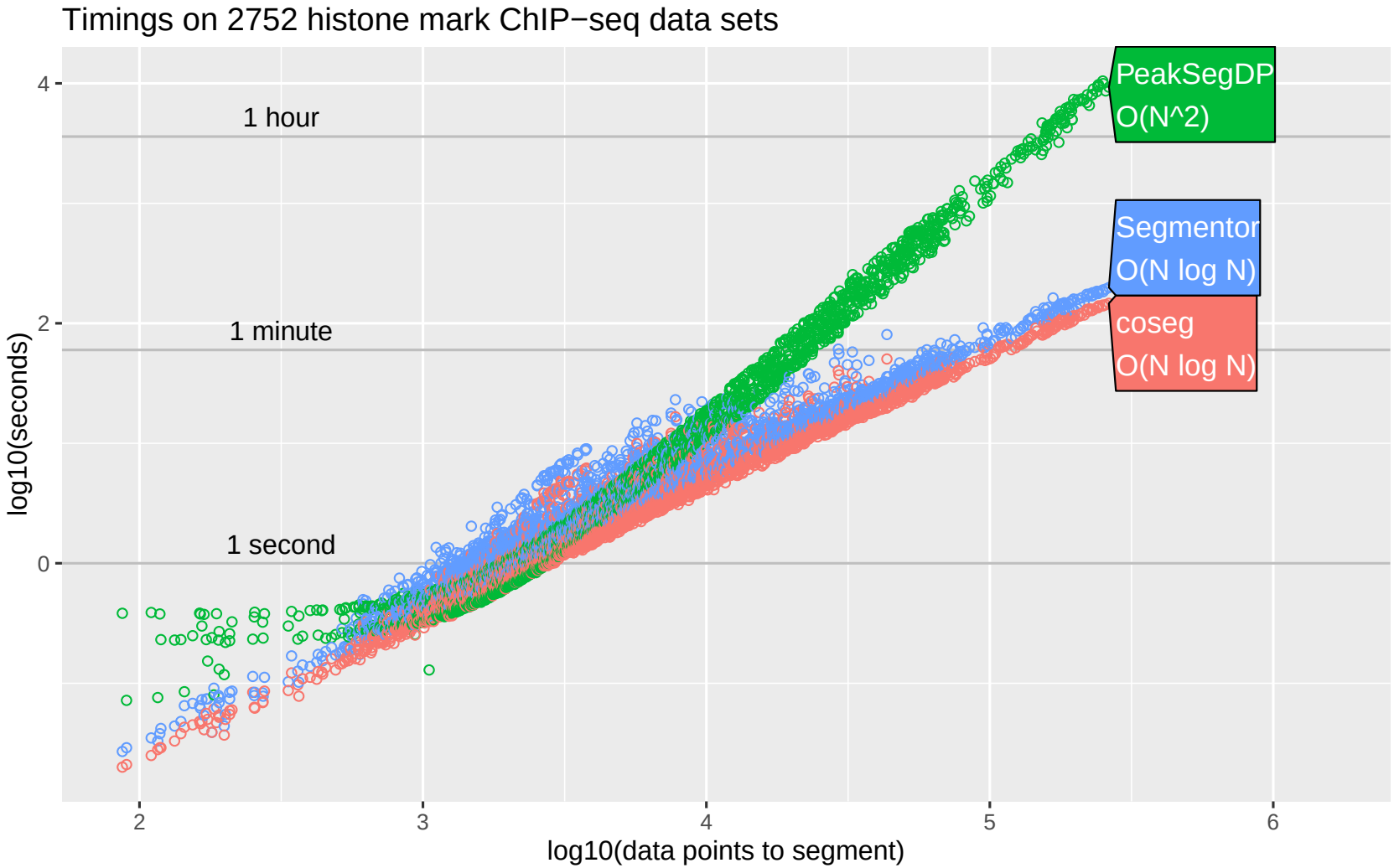
```
minLength 145
medLength 150
maxLength 155
smallBinLength 50
largeBinLength 100000
pvalueThreshold 0.01
mergeDistance 200
iterationThreshold 5
finalThreshold 0
maxIter 20
```

# PeakSeg: search for the peaks with lowest loss




Simple model with only one parameter to train (maxPeaks).

# Linear time algorithms faster for larger data sets




Total time to compute 10 models (0, ..., 9 peaks) for all data sets:

- PeakSegDP: 156 hours, inexact.
- coseg: 6 hours, exact.

# Conclusions

| | no pruning | functional pruning |
|---|---|---|
| unconstrained | Dynamic Programming | Pruned DP |
| | exact $O(N^2)$ | exact $O(N \log N)$ |
| R pkgs: | changepoint | cghseg, Segmentor |
| up-down constrained | constrained DP | **this work** |
| | inexact $O(N^2)$ | exact $O(N \log N)$ |
| R pkgs: | PeakSegDP | **coseg** |

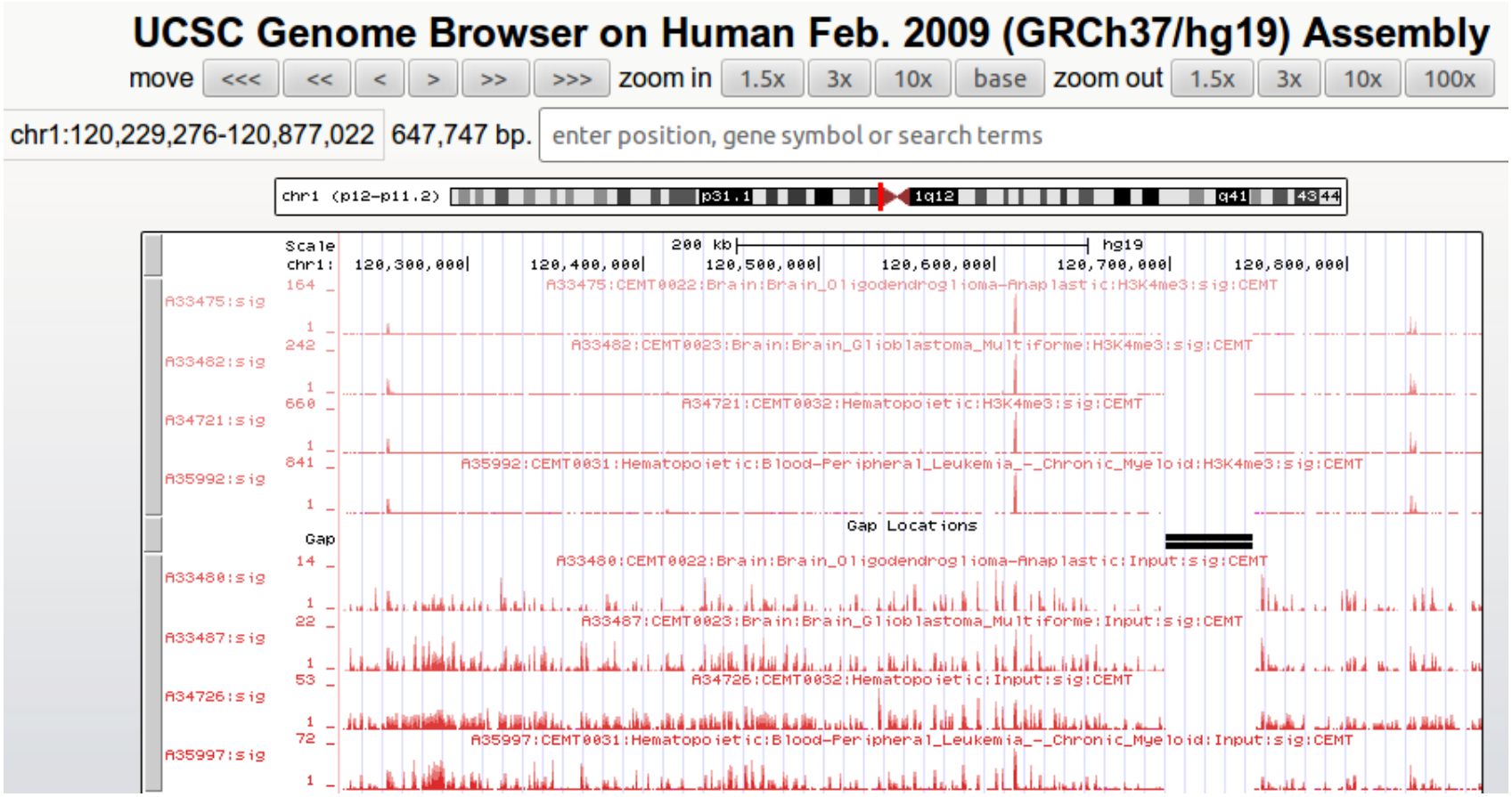
- New algorithm that **exactly** computes the **constrained** optimal change-points/peaks for $N$ data points.
- C++ code in coseg R package, $O(N \log N)$ memory `https://github.com/tdhock/coseg`
- PeakSegFPOP program for big $N > 10^6$ data, $O(\log N)$ memory and $O(N \log N)$ disk space.
- TODO: regularized isotonic regression solver.
- TODO: supervised peak calling for ENCODE, Roadmap, ...
- TODO: interactive web app for creating labels.

# Thanks for your attention!

Questions? toby.hocking@mail.mcgill.ca

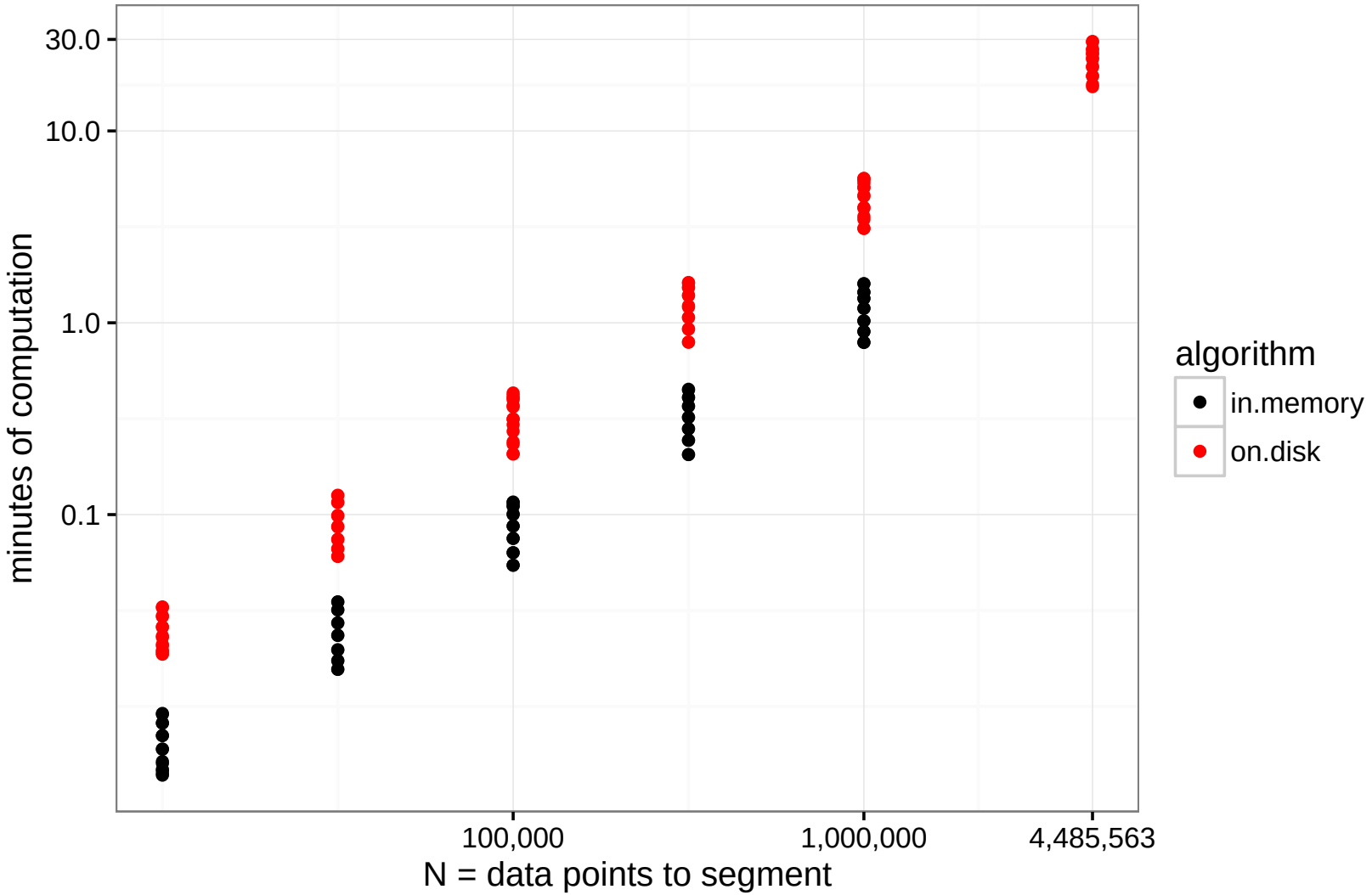
- **coseg** R package, `https://github.com/tdhock/coseg`
- **PeakSegFPOP** command line program, `https://github.com/tdhock/PeakSegFPOP`
- source code for these slides: `https://github.com/tdhock/PeakSegFPOP-paper`

# Segmenting whole chromosomes?



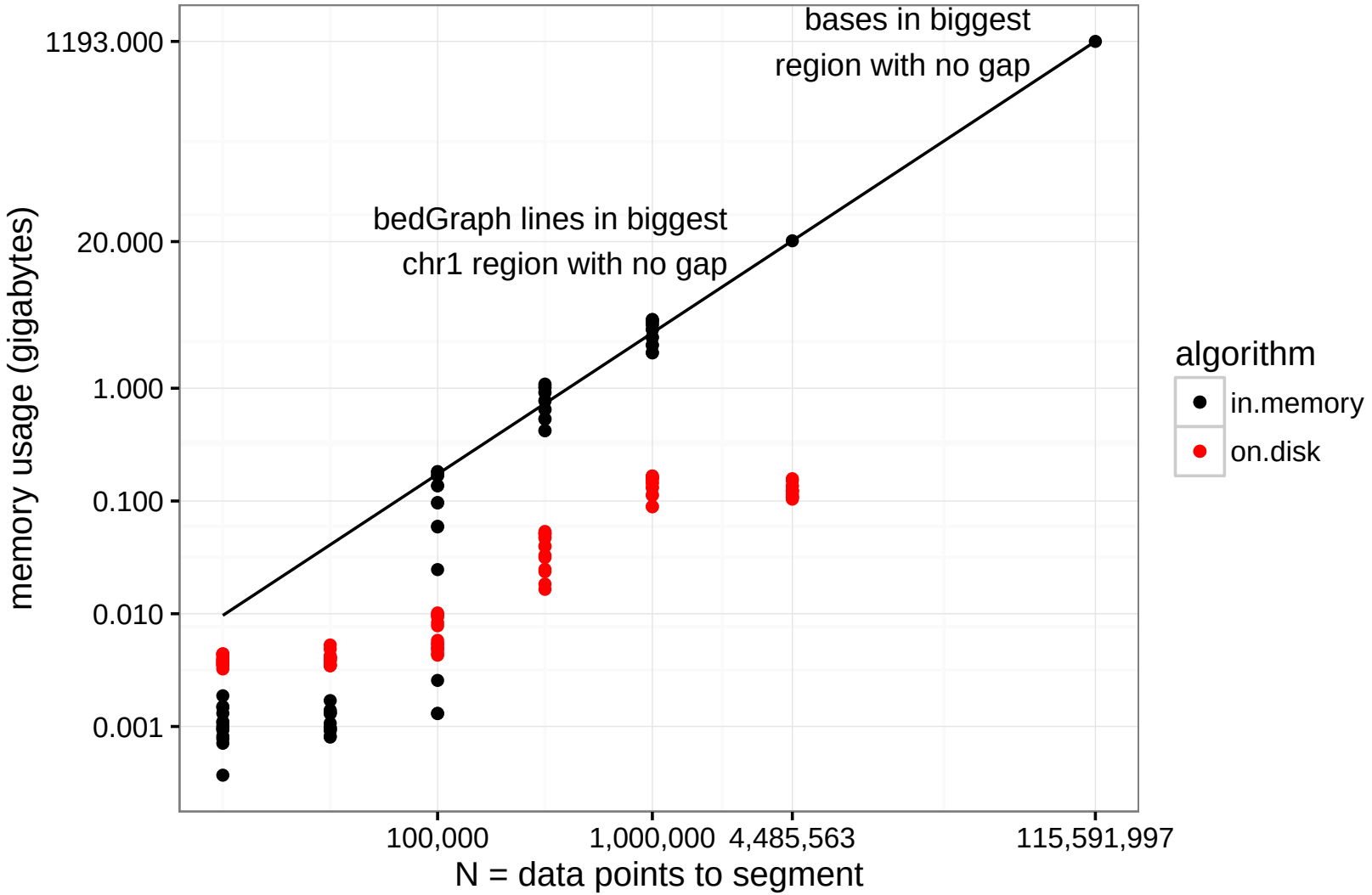

- 365 regions with no gaps in hg19.
- 272 regions with no gaps on chr1-22, X, Y.
- Smallest: 31,833 bases (chr6:157,609,467-157,641,300).
- Largest: 115,591,997 bases (chr4:75,452,279-191,044,276).

# Reasonable time to segment biggest region on chr1



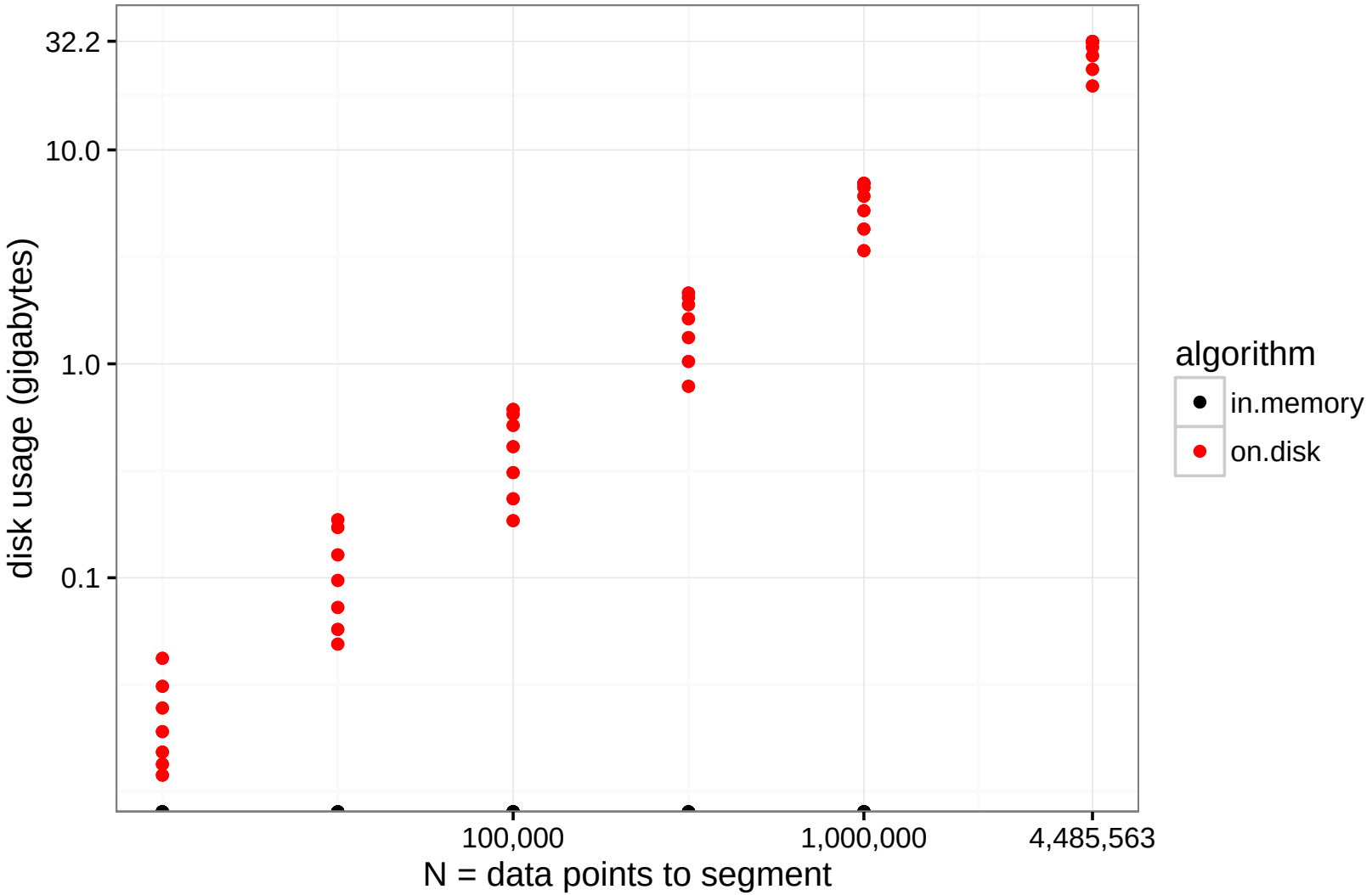

- R package in memory: $O(N \log N)$ time.
- Command line program on disk: $O(N \log N)$ time.

# Memory requirements reasonable for on-disk version



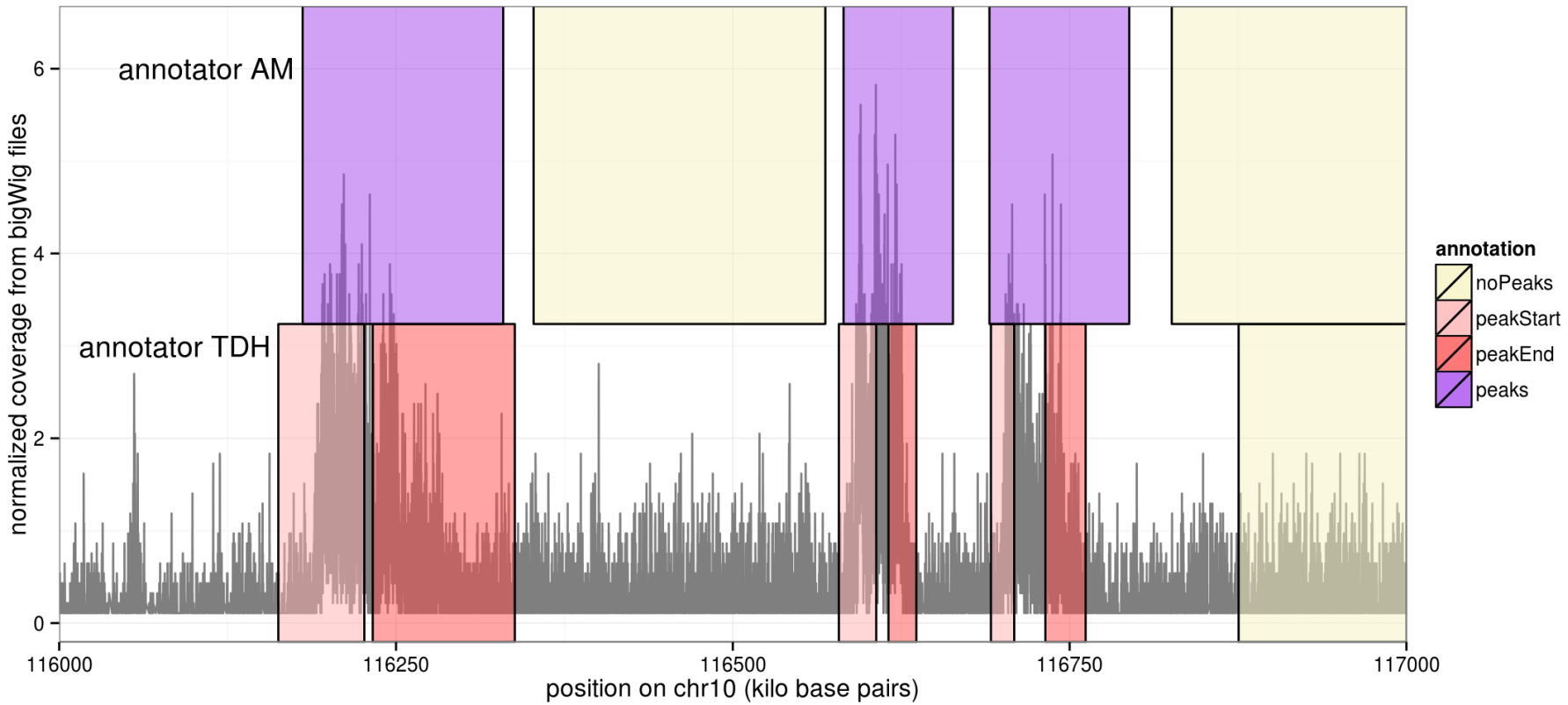

- R package in memory: $O(N \log N)$ memory.
- Command line program on disk: $O(\log N)$ memory.

# Disk usage reasonable



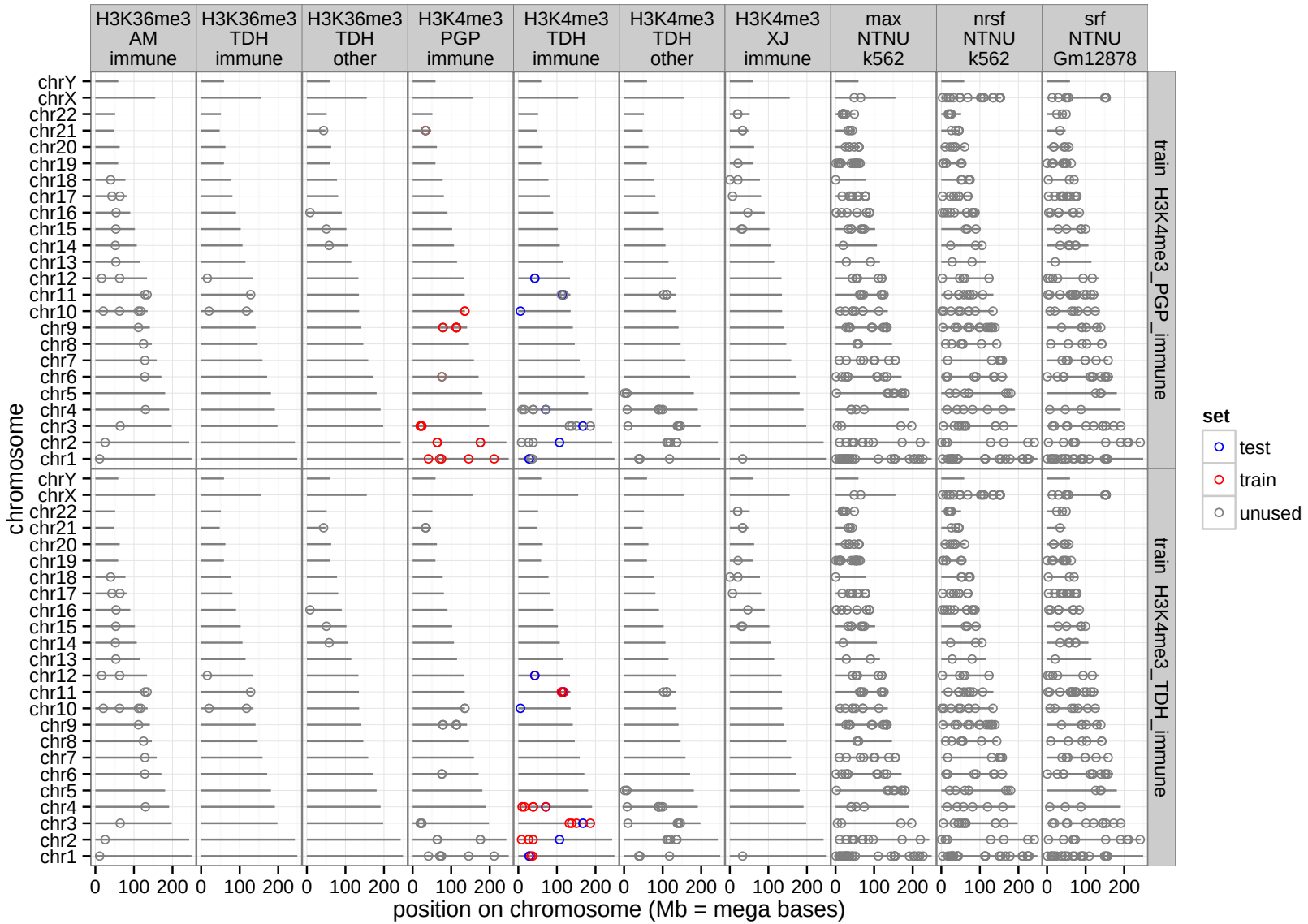

- R package in memory: no disk usage.
- Command line program: $O(N \log N)$ disk space (temporary).

# Two annotators provide consistent labels, but different precision



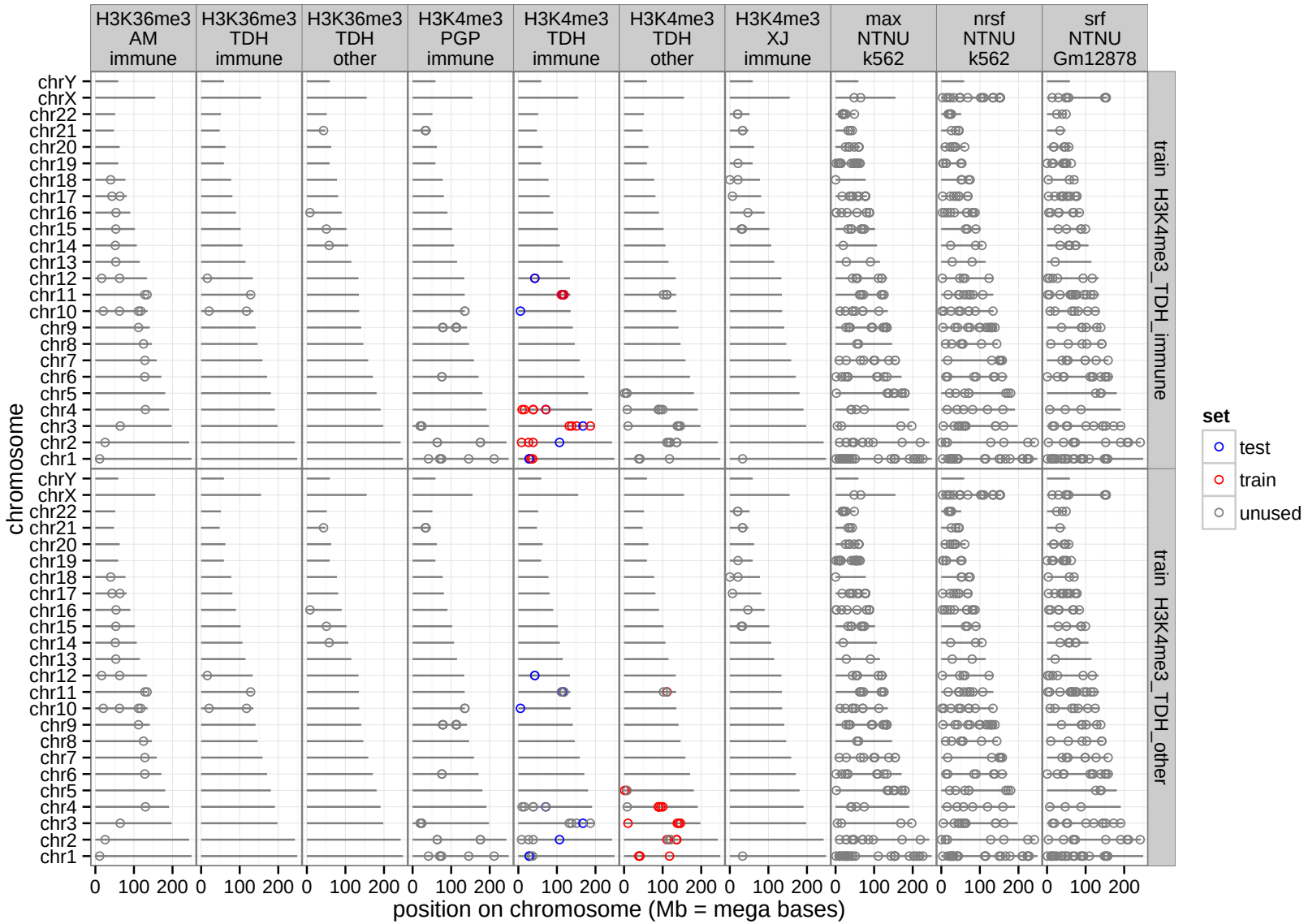

- TDH peakStart/peakEnd more precise than AM peaks.
- AM noPeaks more precise than TDH no label.

# Train on one person, test on another (same histone mark and samples)

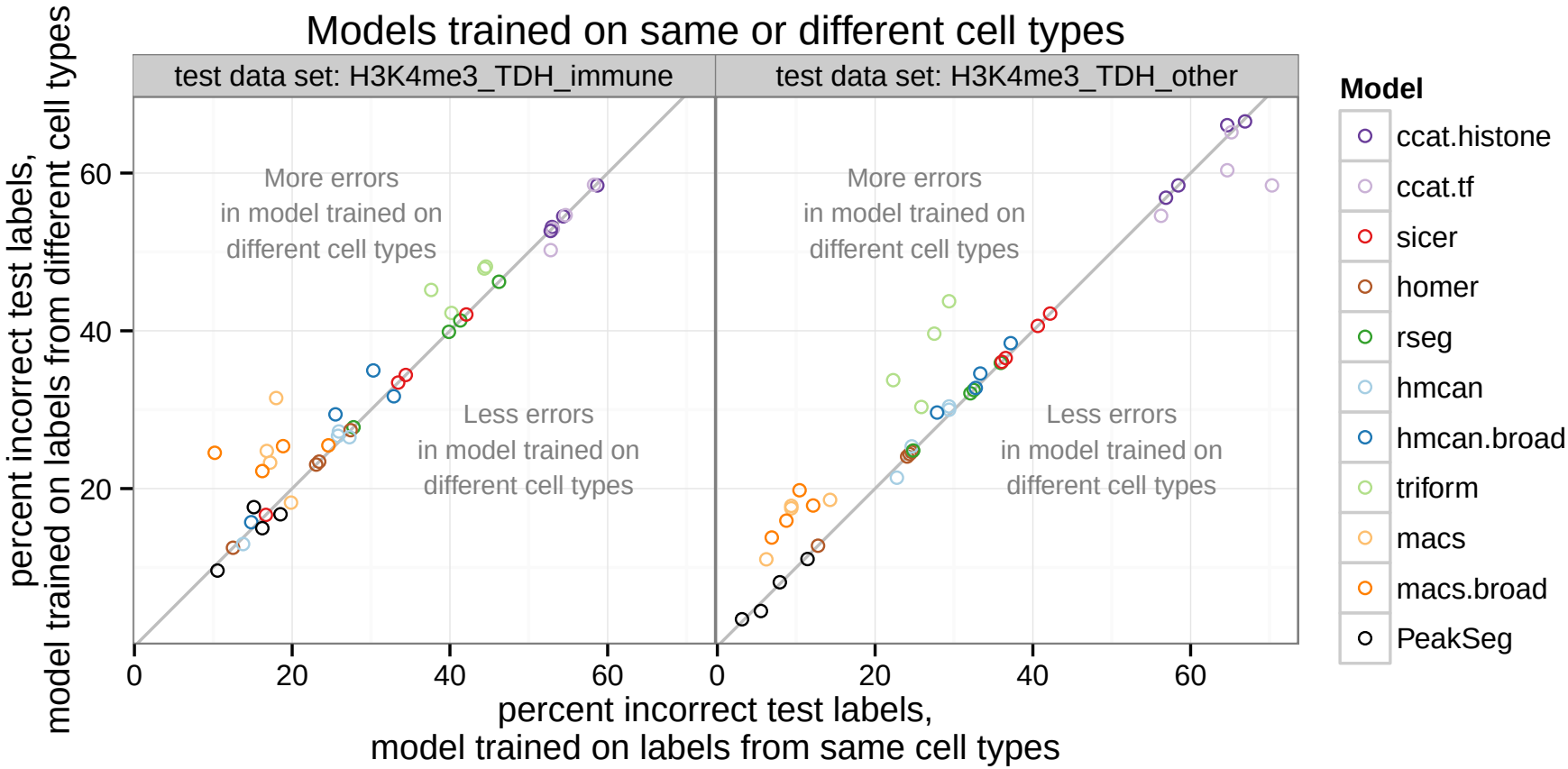


# Train on one person, test on another (same histone mark and samples)

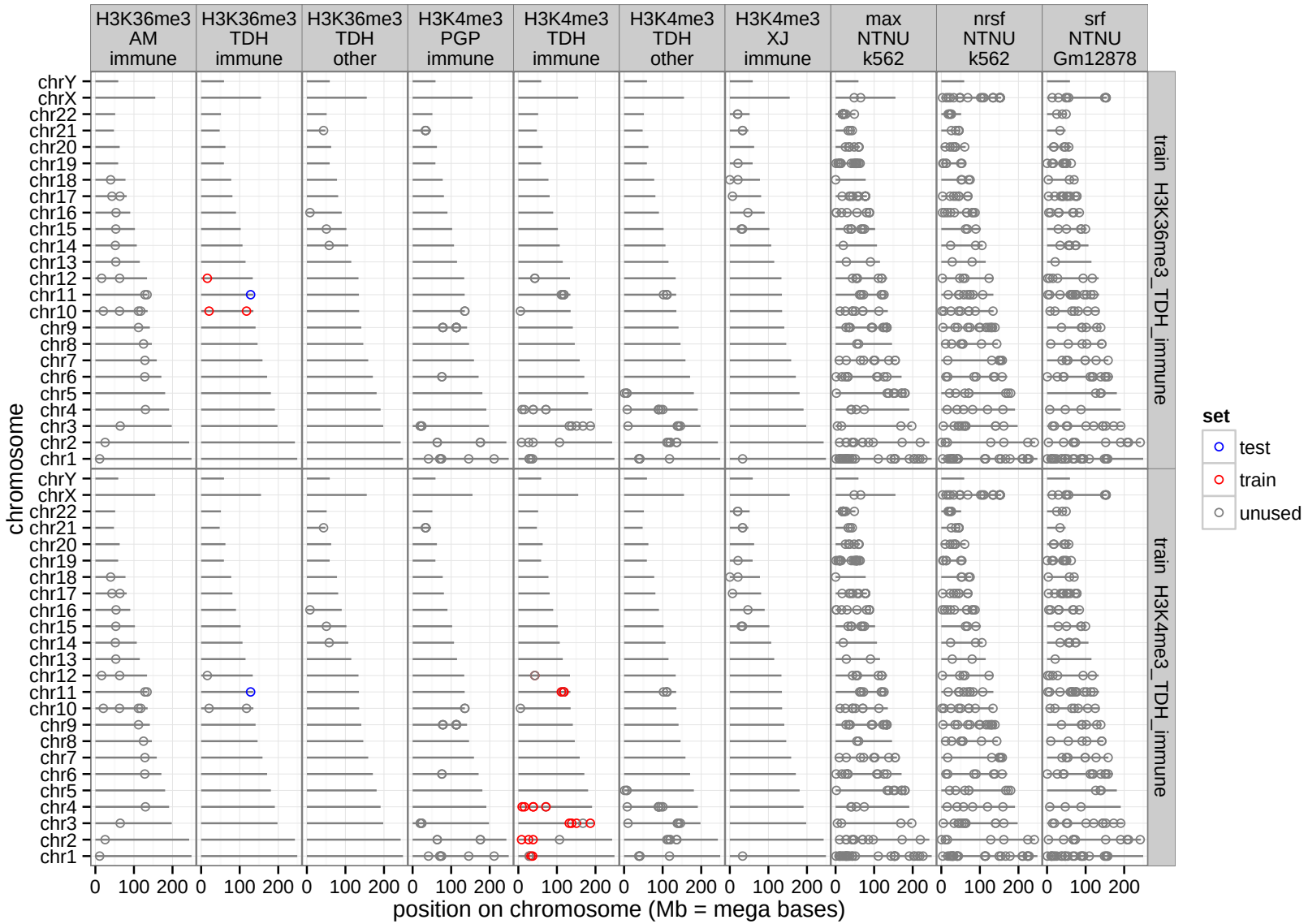
# Train on some samples, test on others (same histone mark and person)



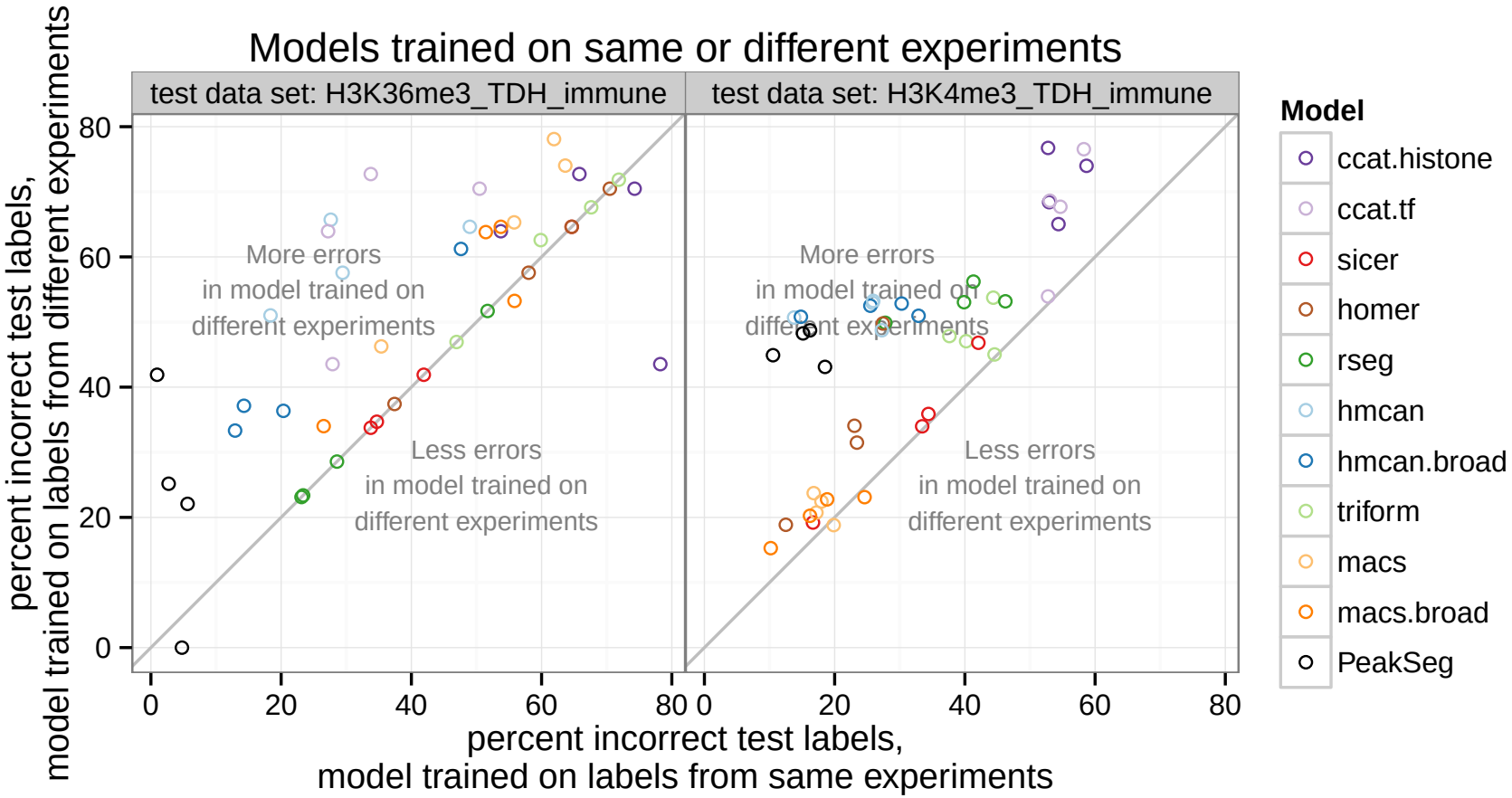
# Train on some samples, test on others (same histone mark and person)



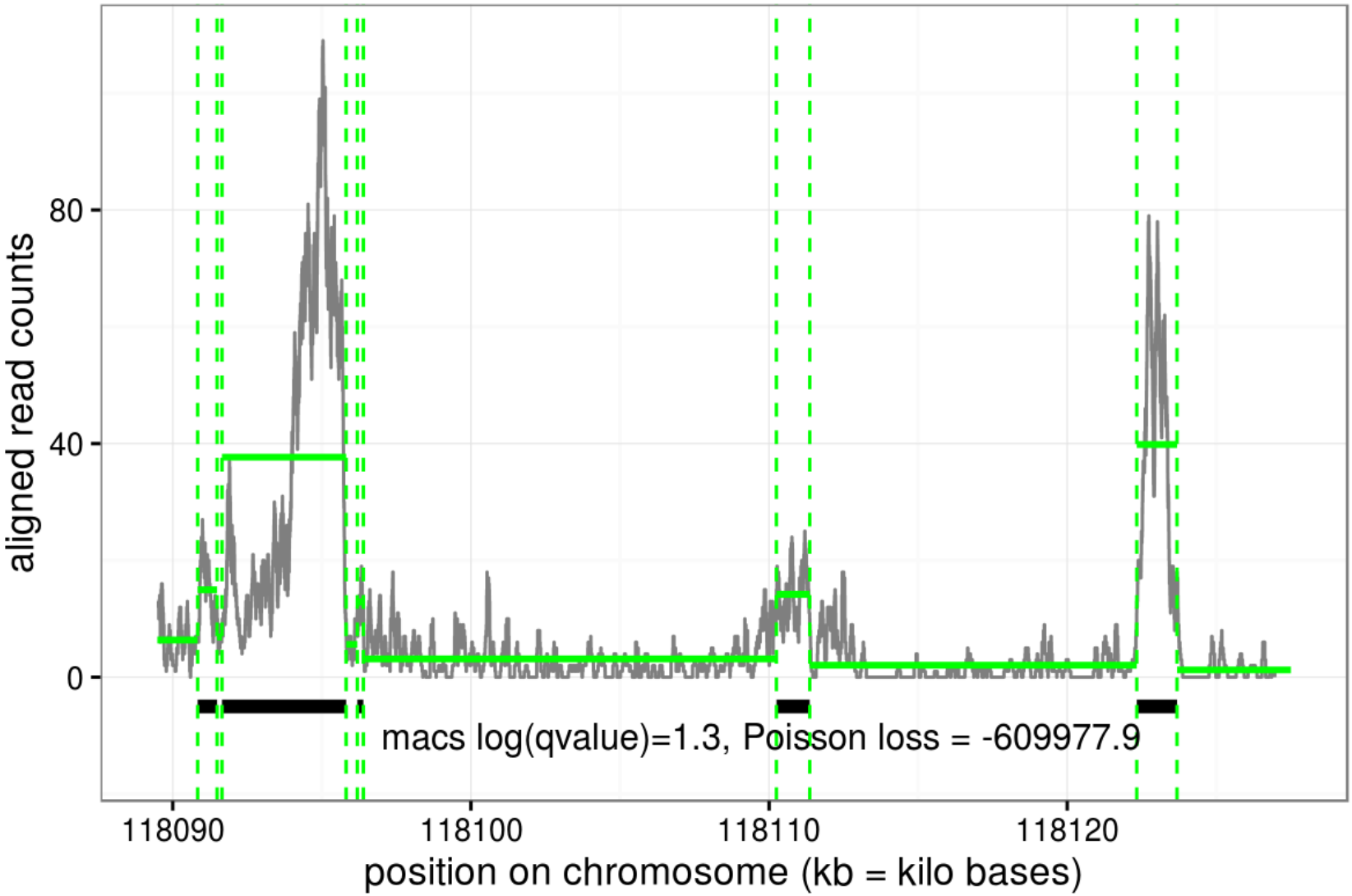
# Train on one histone mark, test on another (same person and samples)



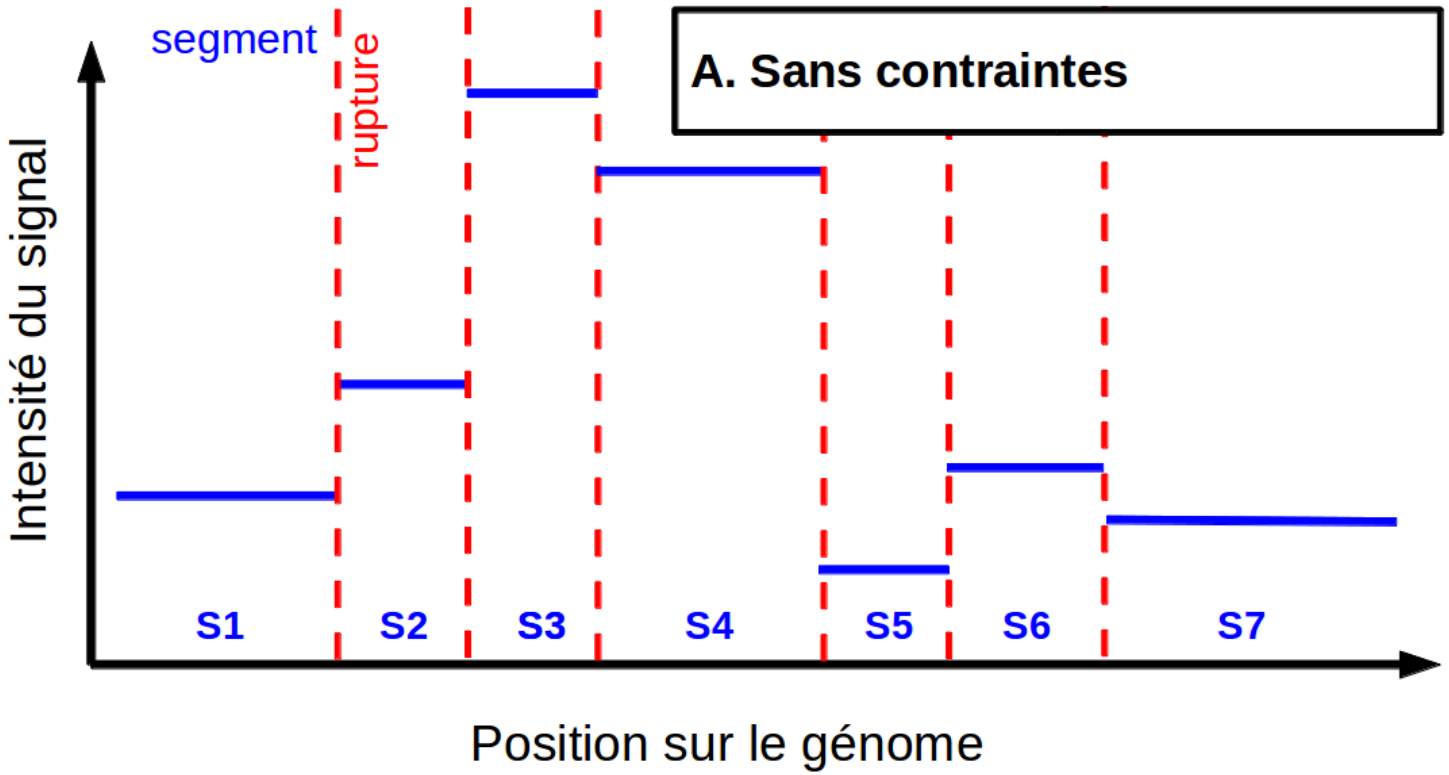
# Train on one histone mark, test on another (same person and samples)



# PeakSeg with 2 peaks more likely than default macs

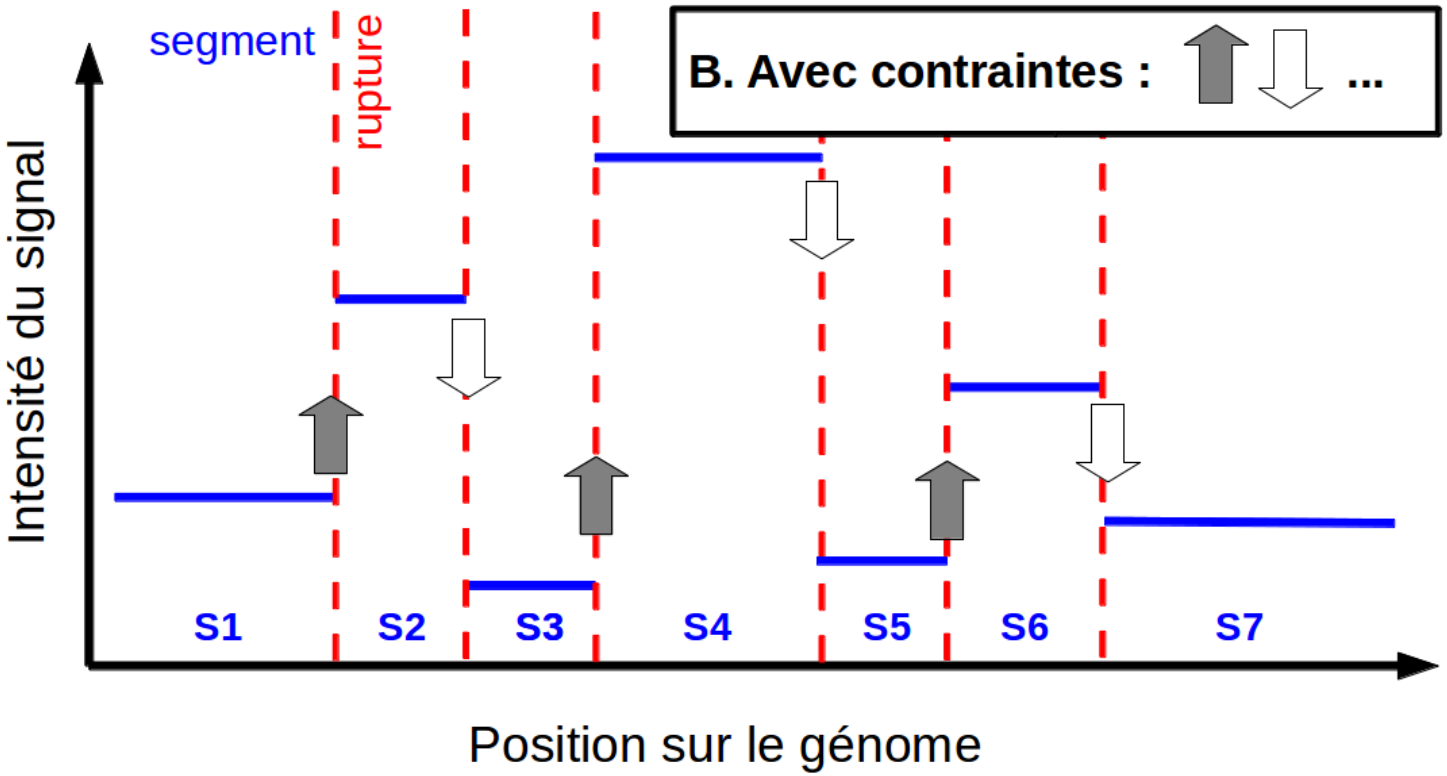


# Unconstrained model can have any changes




- The model above does NOT verify the PeakSeg up-down constraint (second change is up, should be down).
- Not directly interpretable as background and peaks.

# PeakSeg constraint forces up then down changes




- Novelty of PeakSeg model with respect to previous optimal segmentation models: the up-down constraint.
- Interpretable as background (S1, S3, ...) and peaks (S2, S4, ...).

# What does our solution say about the PeakSeg solution?

- Our algo exactly solves a problem with **non-strict inequality** constraints.
- For example, $N = 3$ data points and $S = 3$ segments,

$$\min_{m_1 \leq m_2 \geq m_3} \sum_{t=1}^{N} m_t - z_t \log m_t.$$

- But the PeakSeg problem has **strict inequality** constraints:

$$\min_{m_1 < m_2 > m_3} \sum_{t=1}^{N} m_t - z_t \log m_t.$$

When our algo returns equal values for adjacent segment means,

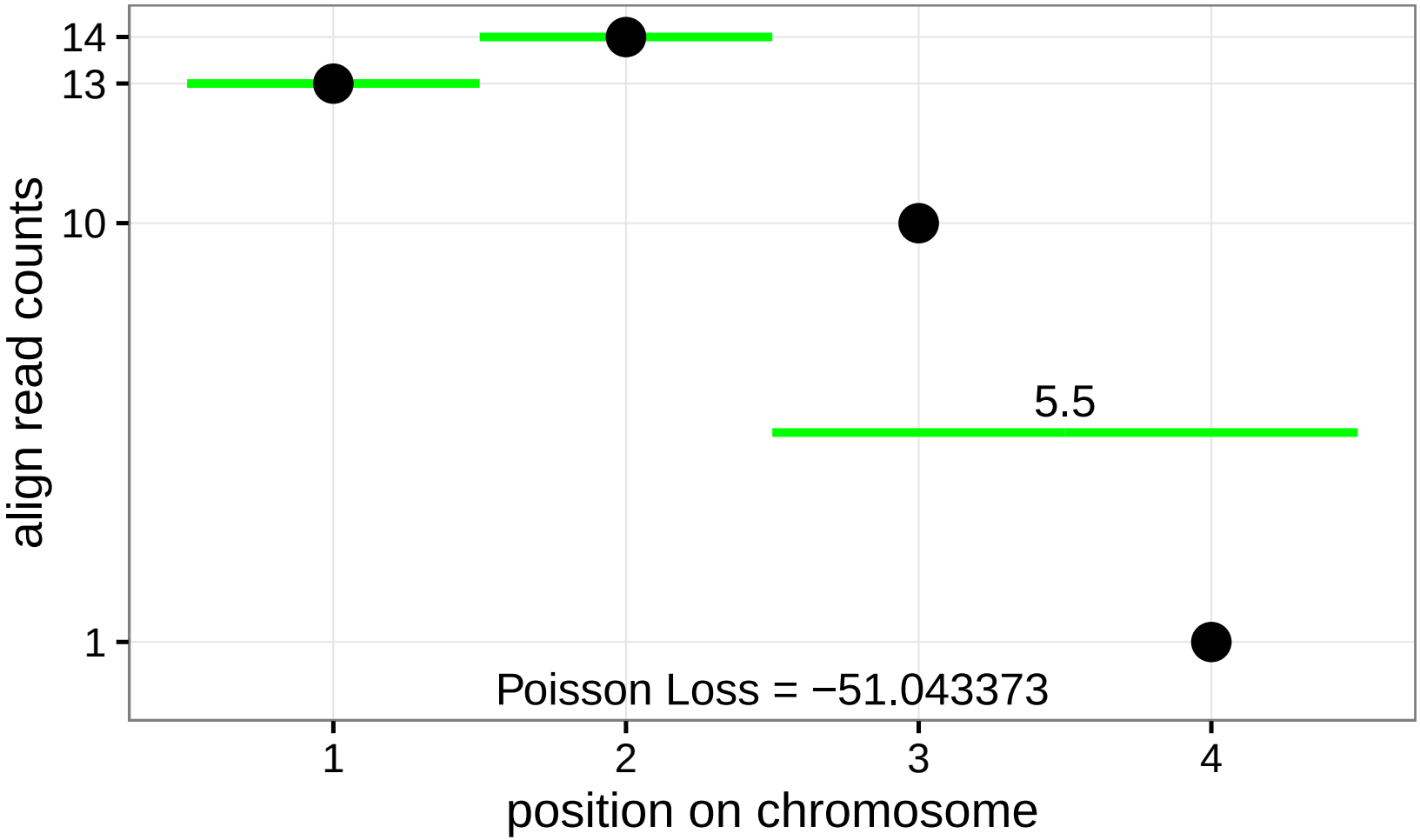
- Our solution is not feasible for the PeakSeg problem, and
- The PeakSeg solution is undefined.

# Example: 13, 14, 10, 1

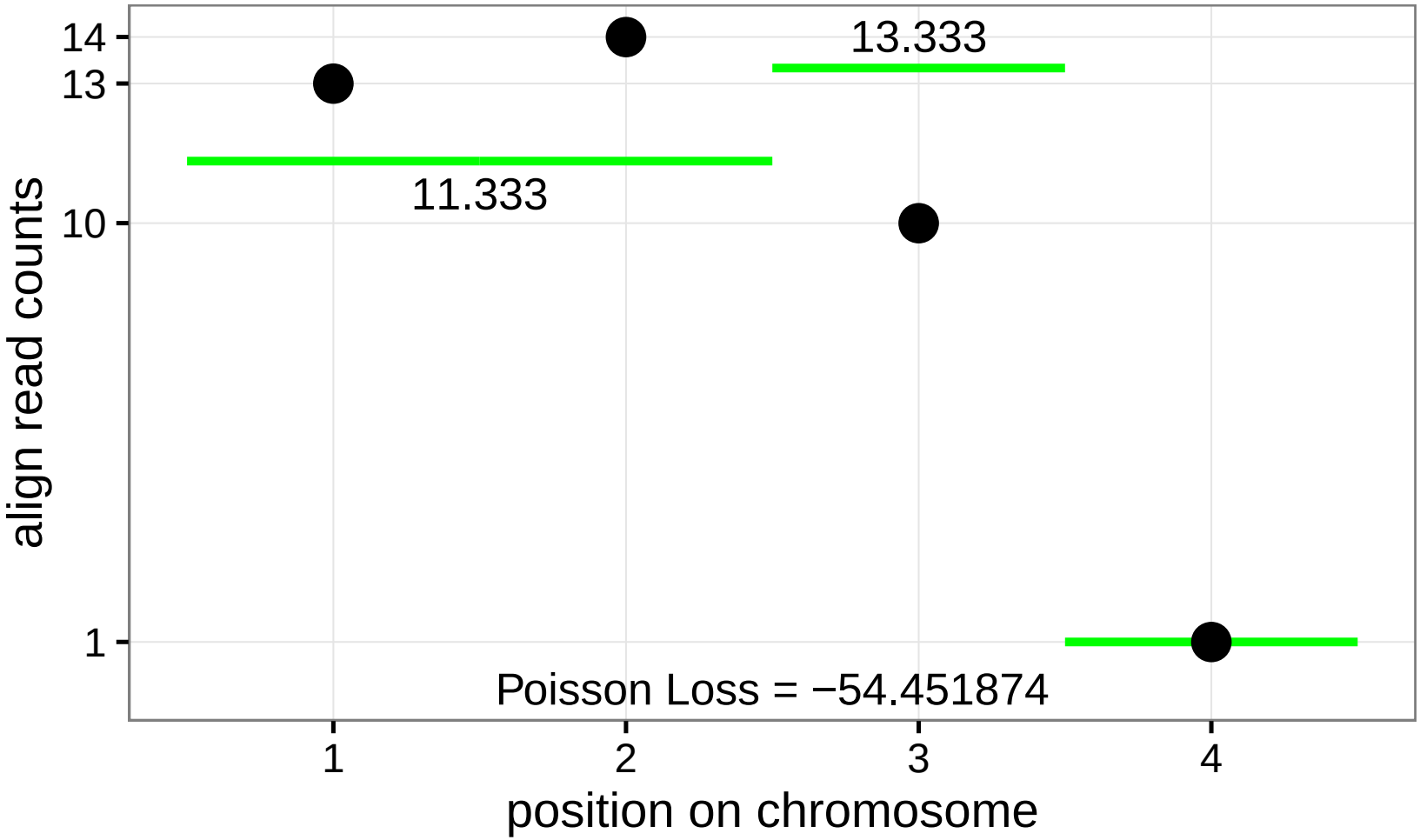
What do you think is the best model with 1 peak?
(3 segments, 1 change up, 1 change down)

Guessing game:
`https://github.com/tdhock/PeakSegFPOP-paper/blob/master/figure-min-undefined.R`

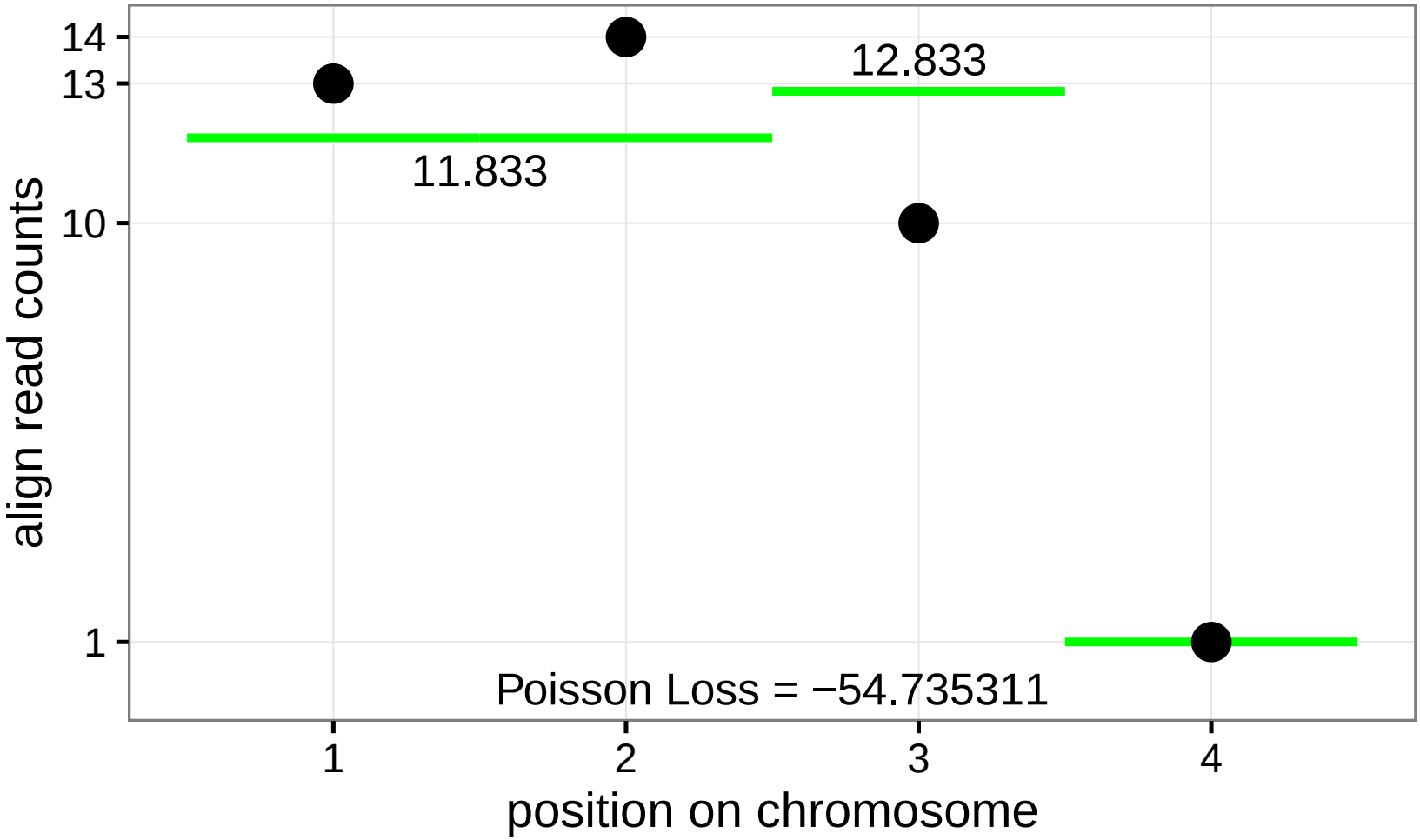
# PeakSegDP returns this highly suboptimal model



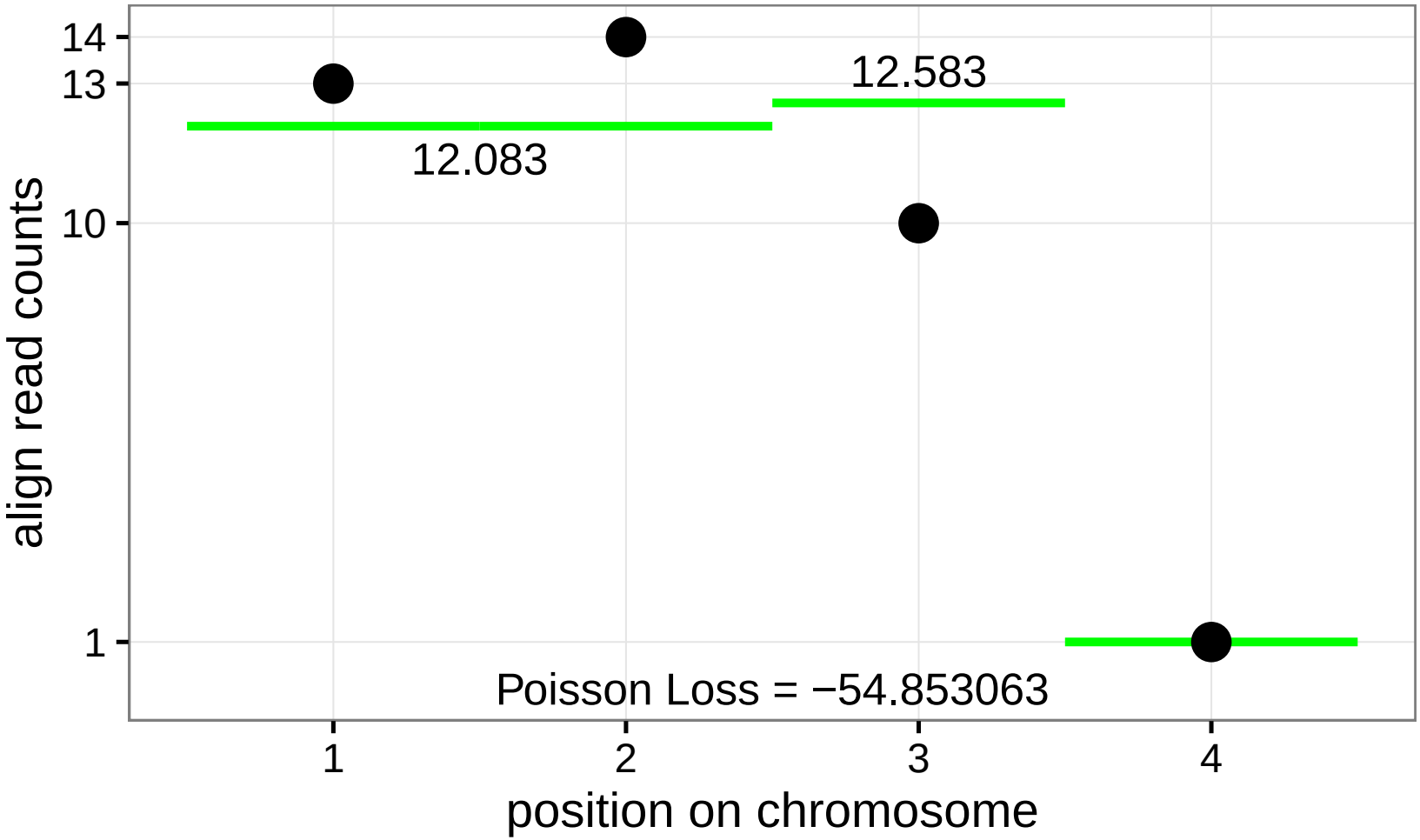
# A better model



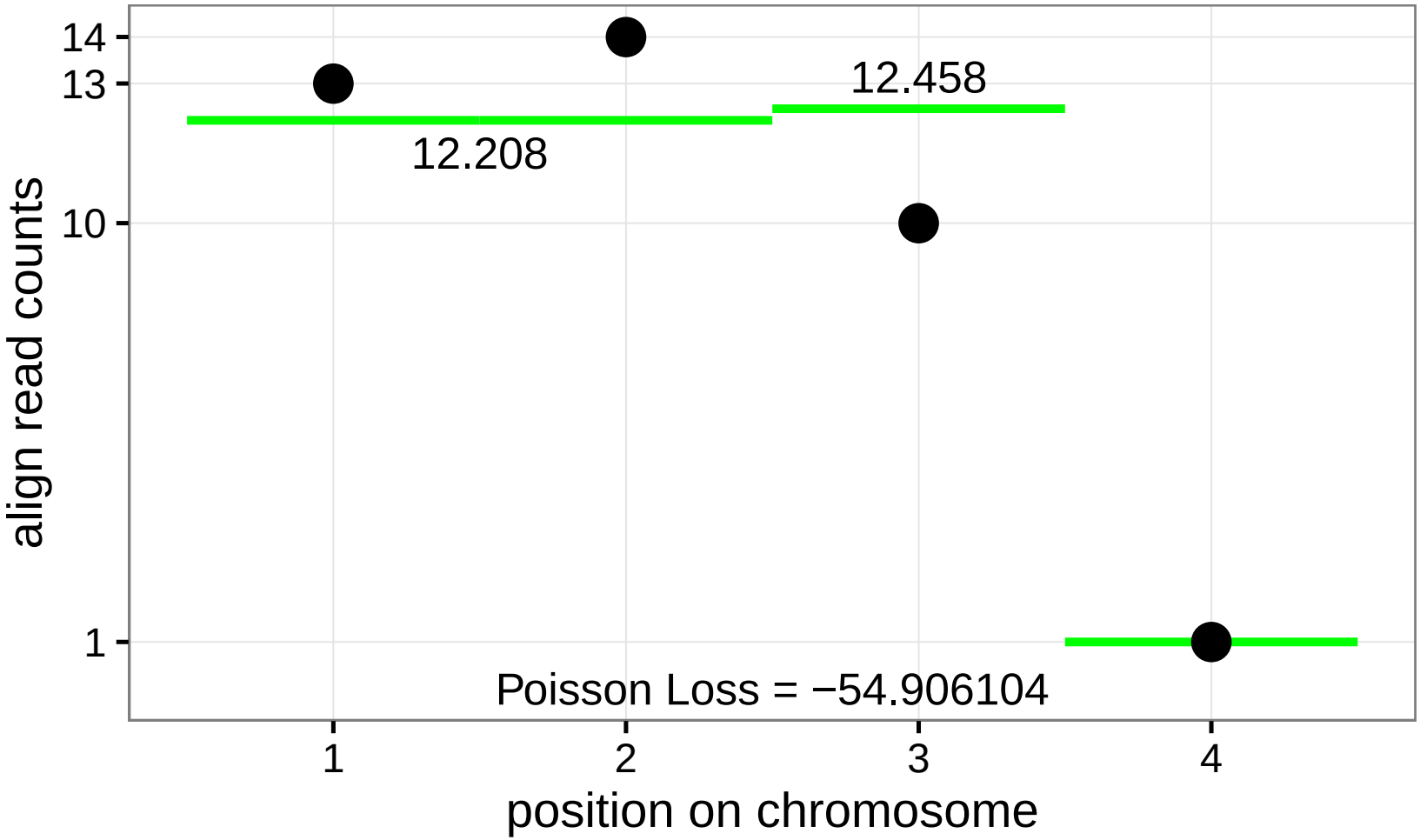
# Even better


14
13
10
1
align read counts
12.833
11.833
Poisson Loss = −54.735311
1
2
3
4
position on chromosome

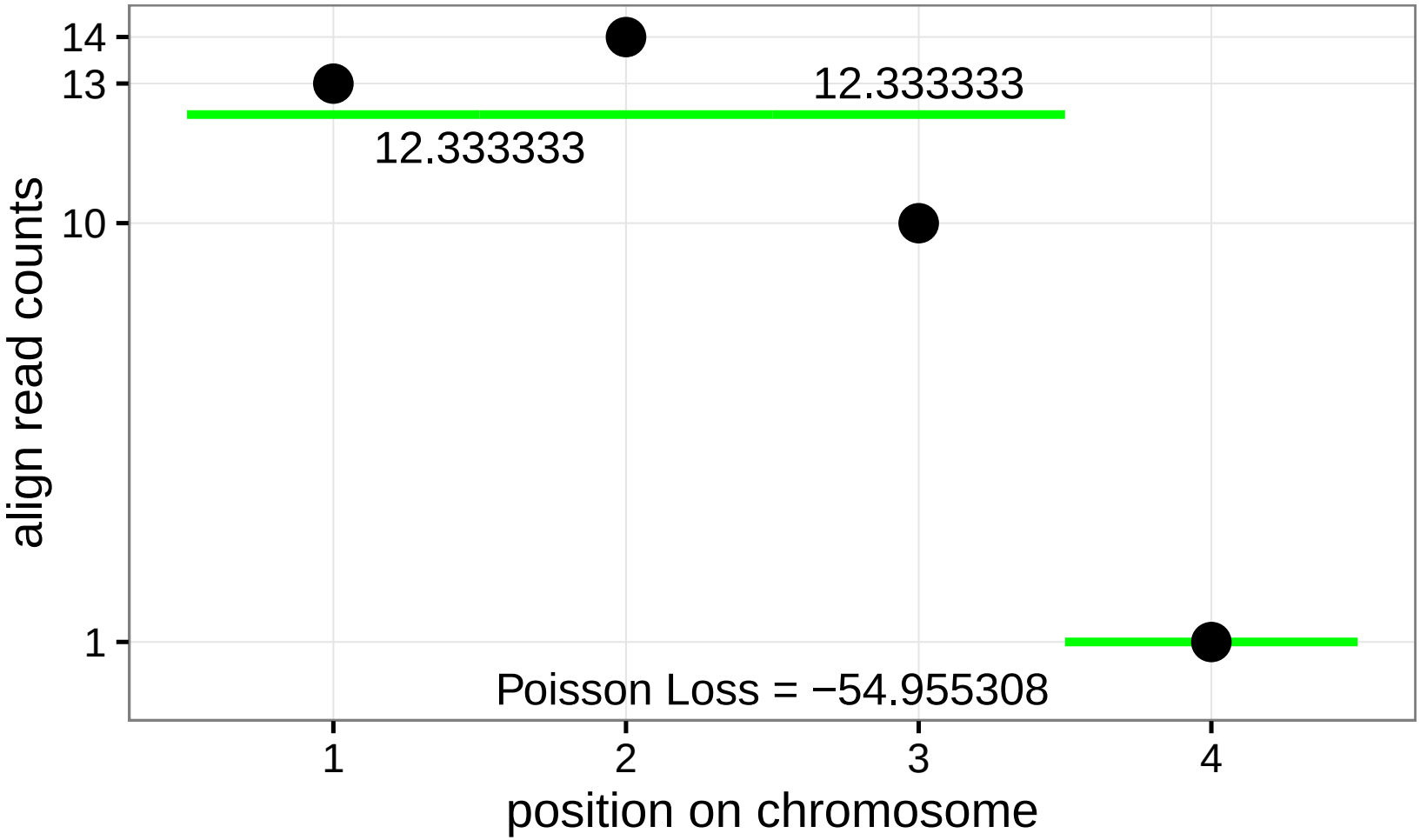
# Still better



# We can keep going forever



# Best model is not feasible



# Unconstrained maximum likelihood segmentation

- We have a sequence of $n$ count data $\mathbf{y} \in \mathbb{Z}_+^n$ to segment.
- Choose the number of segments $S \in \{1, 2, \ldots, n\}$.

$$\begin{aligned} \underset{\substack{\mathbf{m}\in\mathbb{R}^n \\ \mathbf{c}\in\{-1,0,1\}^{n-1}}}{\text{minimize}} \quad & \sum_{t=1}^{n} \ell(m_t, z_t) \\ \text{subject to} \quad & 1 + \sum_{t=1}^{n-1} I(c_t \neq 0) = S, \\ & c_t = -1 \Rightarrow m_t > m_{t+1} \text{ (change down)} \\ & c_t = 0 \Rightarrow m_t = m_{t+1} \text{ (no change)} \\ & c_t = 1 \Rightarrow m_t < m_{t+1} \text{ (change up)} \end{aligned}$$

- The Poisson loss function is $\ell(m, y) = m - y \log m$.
- Every $t$ such that $c_t \neq 0$ is a change-point.

# The PeakSeg constrained maximum likelihood model

$$
\begin{aligned}
\underset{\substack{\mathbf{m}\in\mathbb{R}^n \\ \mathbf{c}\in\{-1,0,1\}^{n-1}}}{\text{minimize}} \quad & \sum_{t=1}^{n} \ell(m_t, z_t) \\
\text{subject to} \quad & 1 + \sum_{t=1}^{n-1} I(c_t \neq 0) = S, \\
& c_t = -1 \Rightarrow m_t > m_{t+1} \text{ (change down)} \\
& c_t = 0 \Rightarrow m_t = m_{t+1} \text{ (no change)} \\
& c_t = 1 \Rightarrow m_t < m_{t+1} \text{ (change up)} \\
& \forall t \in \{1, \ldots, n-1\},\ P_t(\mathbf{c}) \in \{0, 1\}.
\end{aligned}
$$

The only difference with the unconstrained problem is that we have added the constraint $P_t(\mathbf{c}) = \sum_{i=1}^{t} c_i \in \{0, 1\}$.

# The problem solved by the PeakSegPDPA

$$\underset{\substack{\mathbf{m}\in\mathbb{R}^n \\ \mathbf{c}\in\{-1,0,1\}^{n-1}}}{\text{minimize}} \quad \sum_{t=1}^{n} \ell(m_t, z_t)$$

$$\text{subject to} \quad 1 + \sum_{t=1}^{n-1} I(c_t \neq 0) = S,$$

$$c_t = -1 \Rightarrow m_t \geq m_{t+1} \text{ (change down or no change)}$$

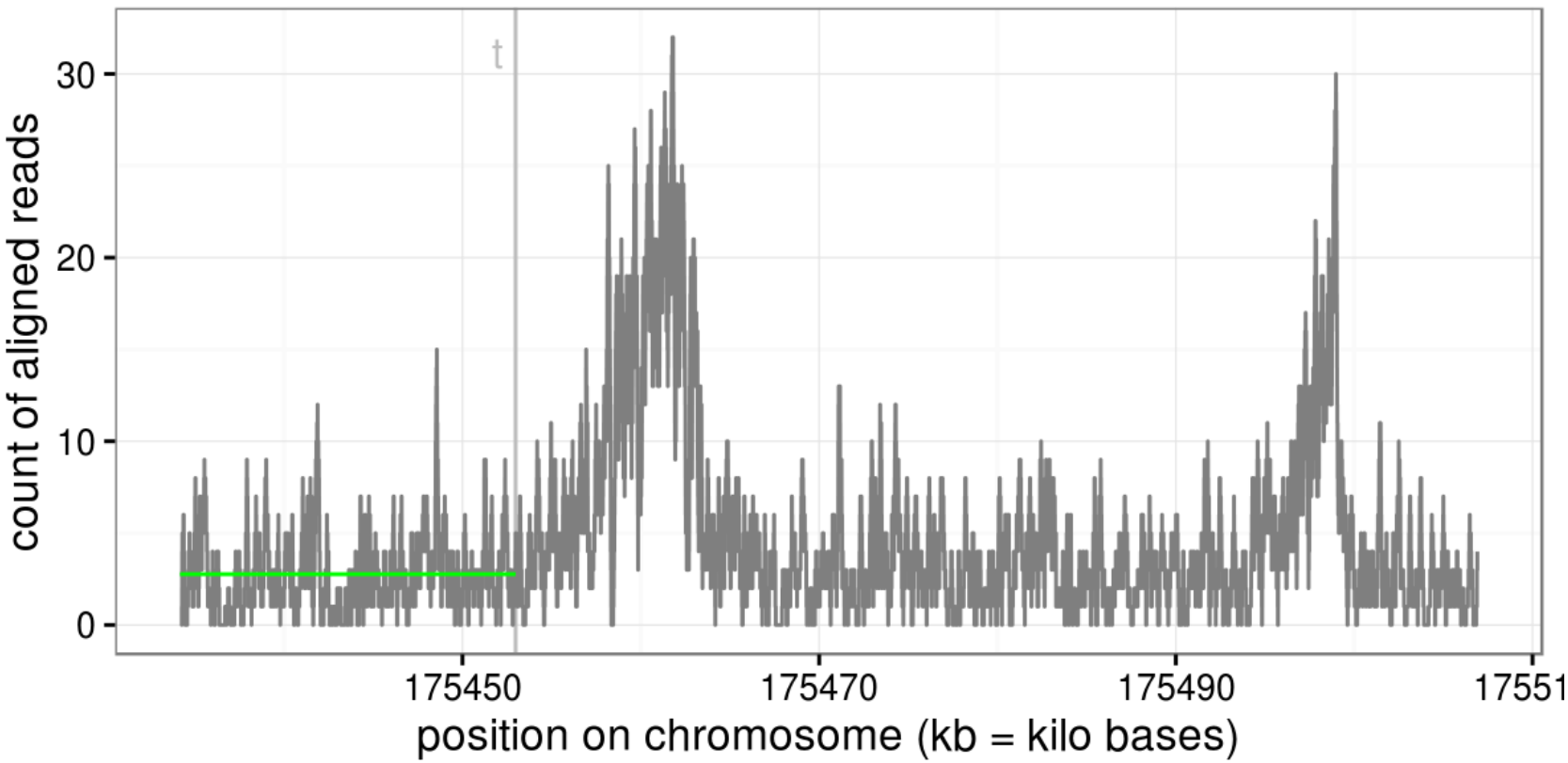
$$c_t = 0 \Rightarrow m_t = m_{t+1} \text{ (no change)}$$

$$c_t = 1 \Rightarrow m_t \leq m_{t+1} \text{ (change up or no change)}$$

$$\forall t \in \{1, \ldots, n-1\},\ P_t(\mathbf{c}) \in \{0, 1\}.$$

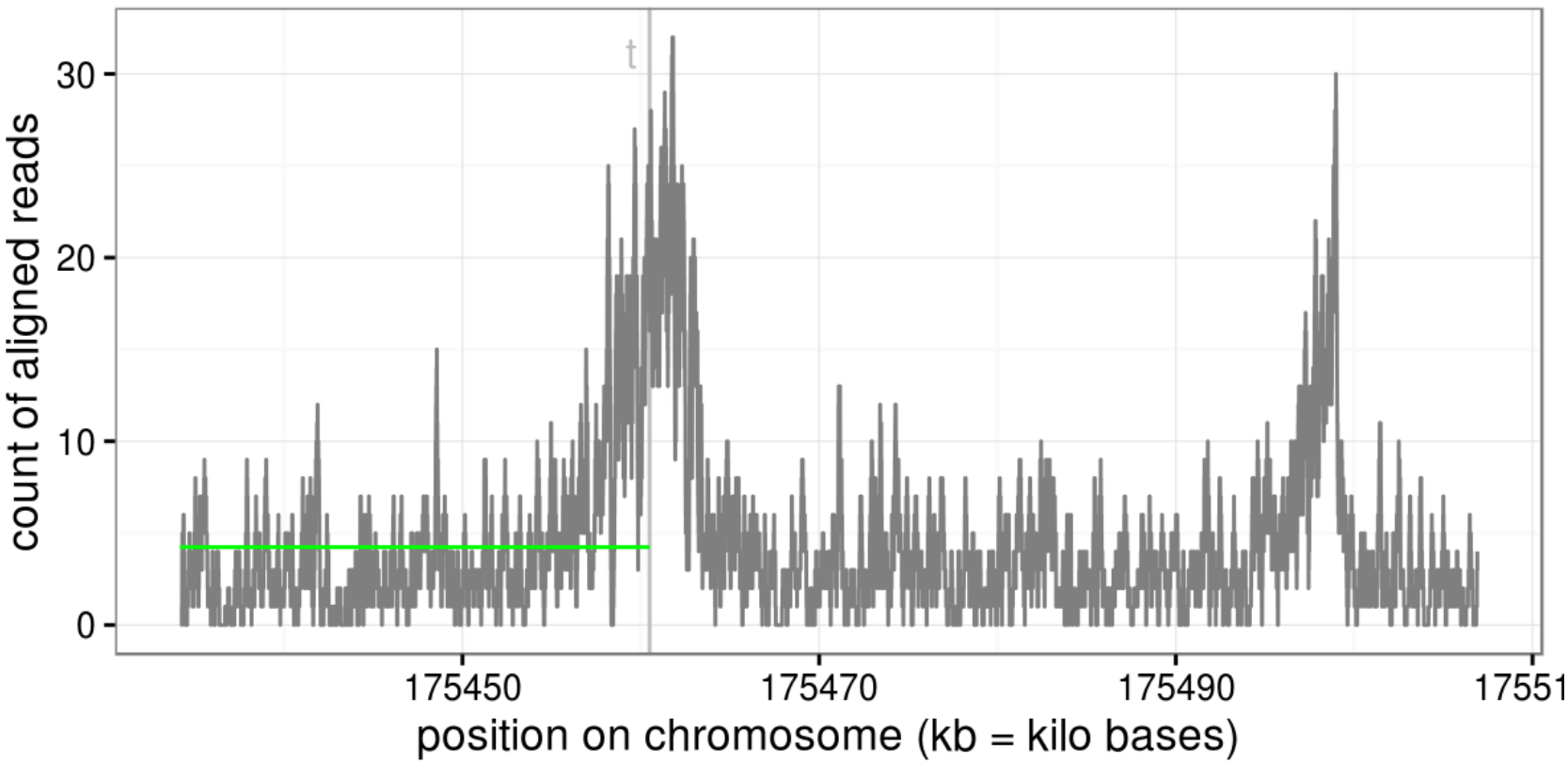
- The only difference with the **PeakSeg** problem is that we have changed the strict inequality constraints to non-strict inequality constraints.
- This model has **at most** $S$ distinct segment means (some may be equal due to the non-strict equality constraints).

# Computation of optimal loss $\mathcal{L}_{s,t}$ for $s = 1$ segments up to data point $t$



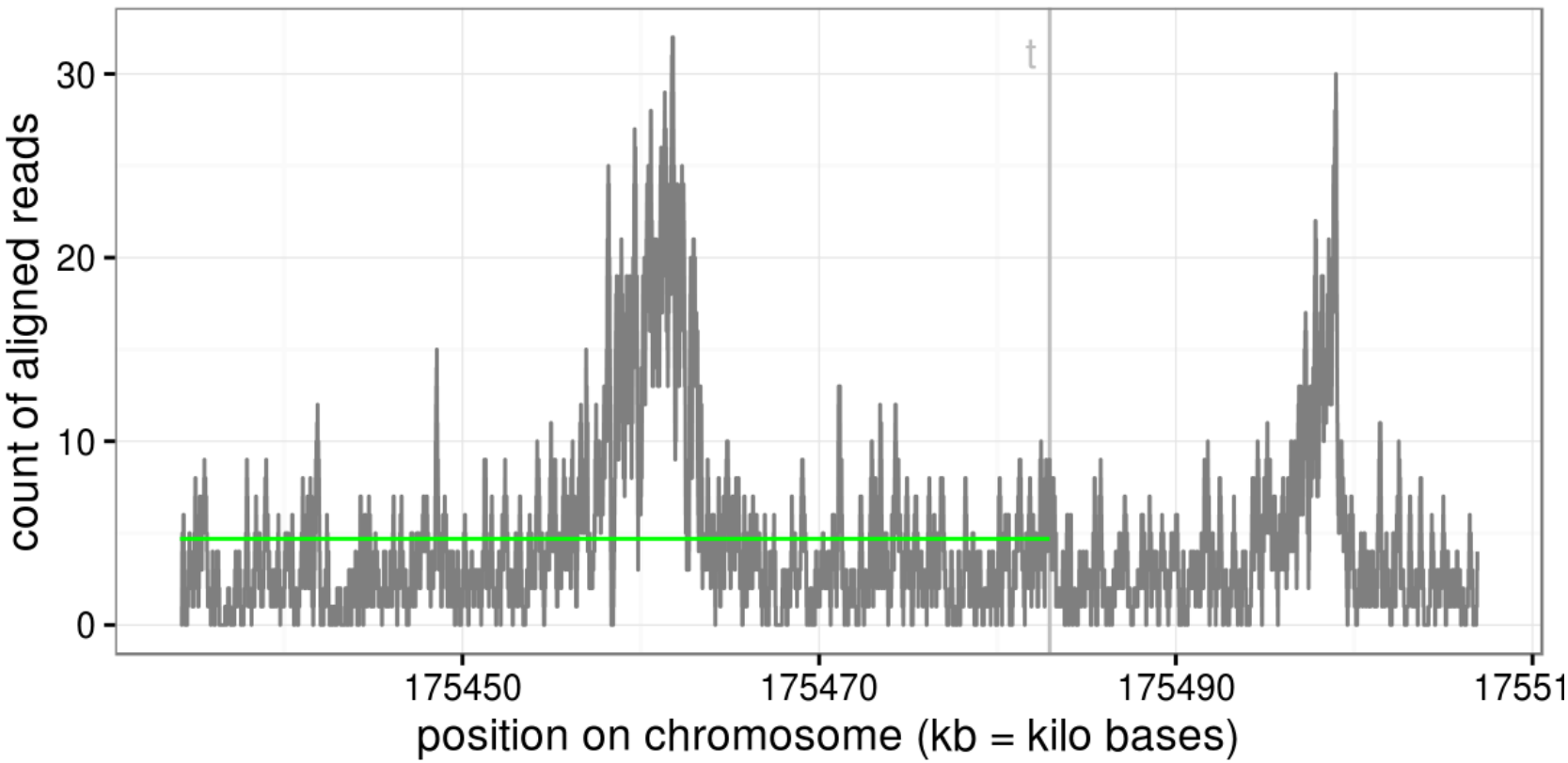

$$\mathcal{L}_{1,t} = \underbrace{c_{(0,t]}}_{\text{optimal loss of 1st segment } (0,t]}$$

# Computation of optimal loss $\mathcal{L}_{s,t}$ for $s = 1$ segments up to data point $t$



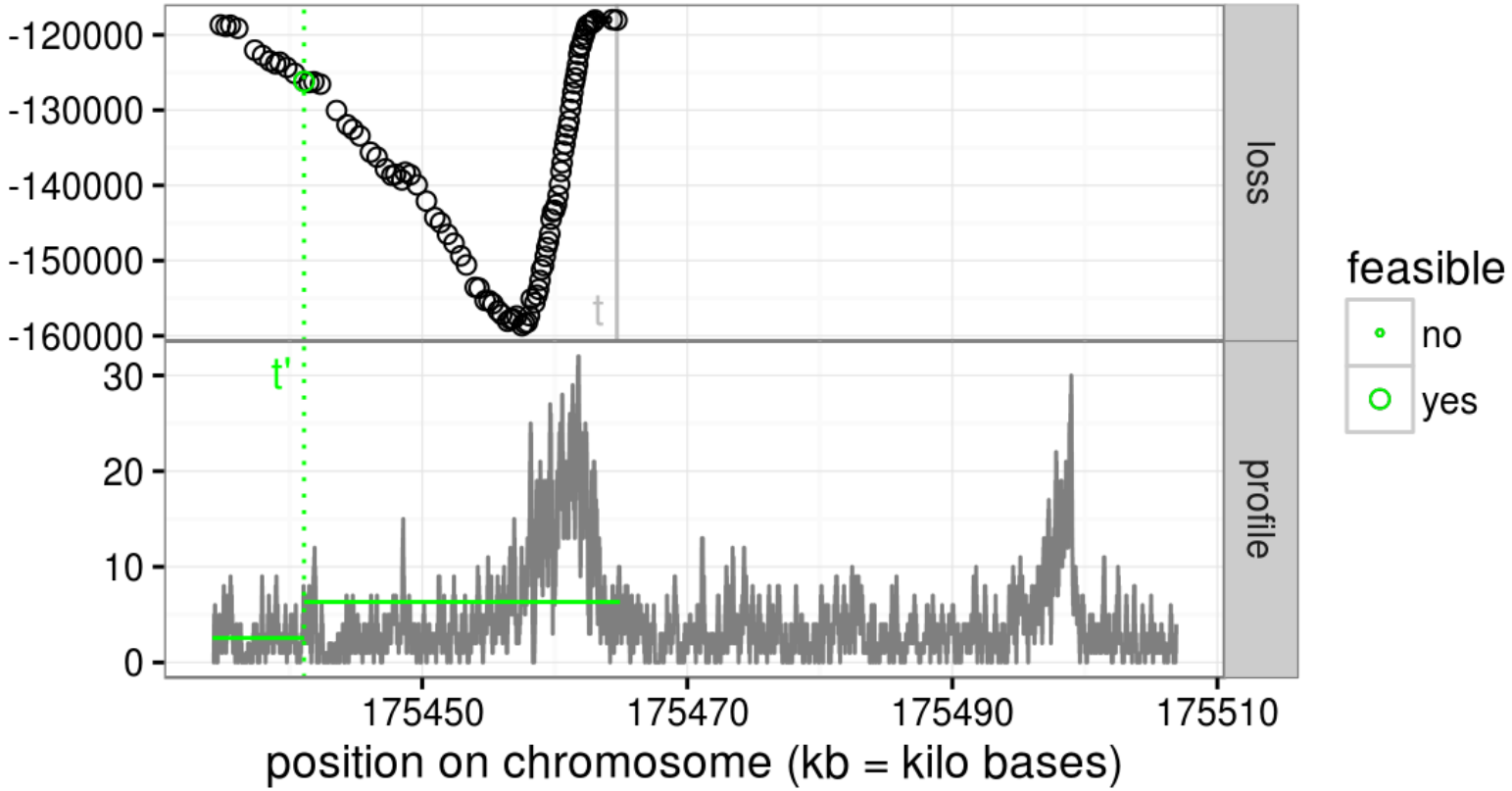

$$\mathcal{L}_{1,t} = \underbrace{c_{(0,t]}}_{\text{optimal loss of 1st segment } (0,t]}$$

# Computation of optimal loss $\mathcal{L}_{s,t}$ for $s = 1$ segments up to data point $t$



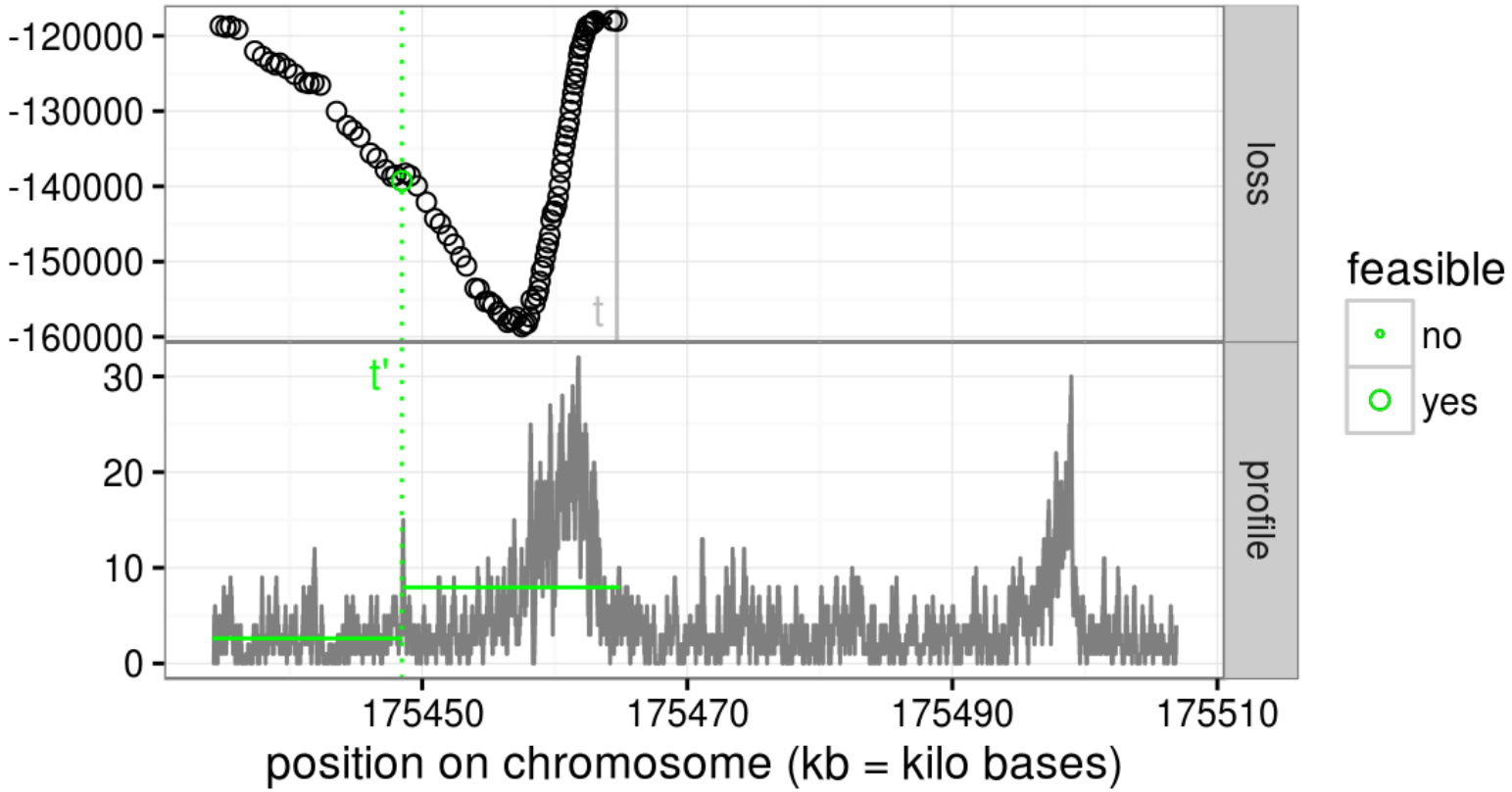

$$\mathcal{L}_{1,t} = \underbrace{c_{(0,t]}}_{\text{optimal loss of 1st segment } (0,t]}$$

# Computation of optimal loss $\mathcal{L}_{s,t}$ for $s = 2$ segments up to data point $t < d$



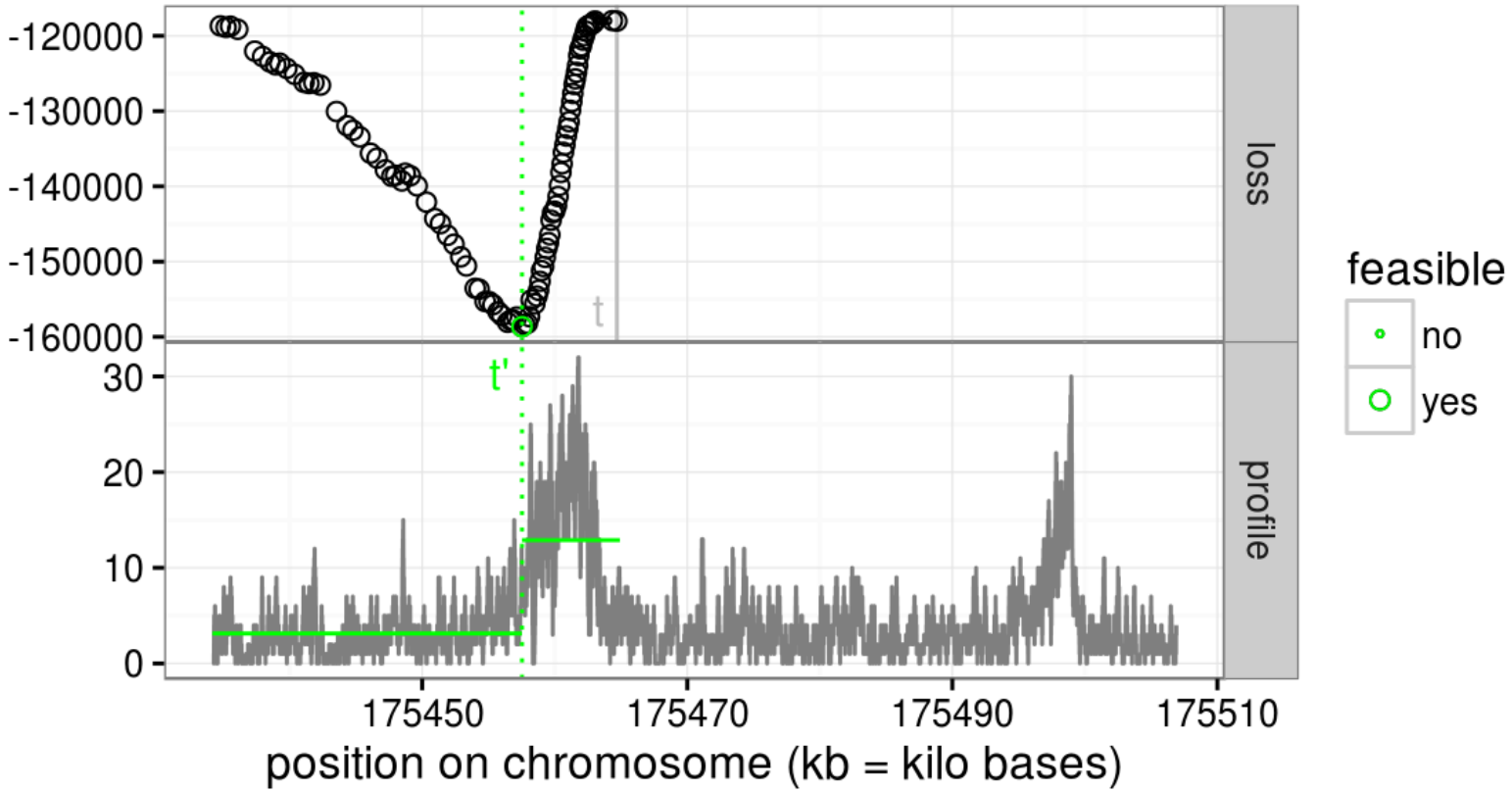

$$\mathcal{L}_{2,t} = \min_{t'<t} \underbrace{\mathcal{L}_{1,t'}}_{\text{optimal loss in 1 segment up to } t'} + \underbrace{c_{(t',t]}}_{\text{optimal loss of 2nd segment } (t',t]}$$

# Computation of optimal loss $\mathcal{L}_{s,t}$ for $s = 2$ segments up to data point $t < d$



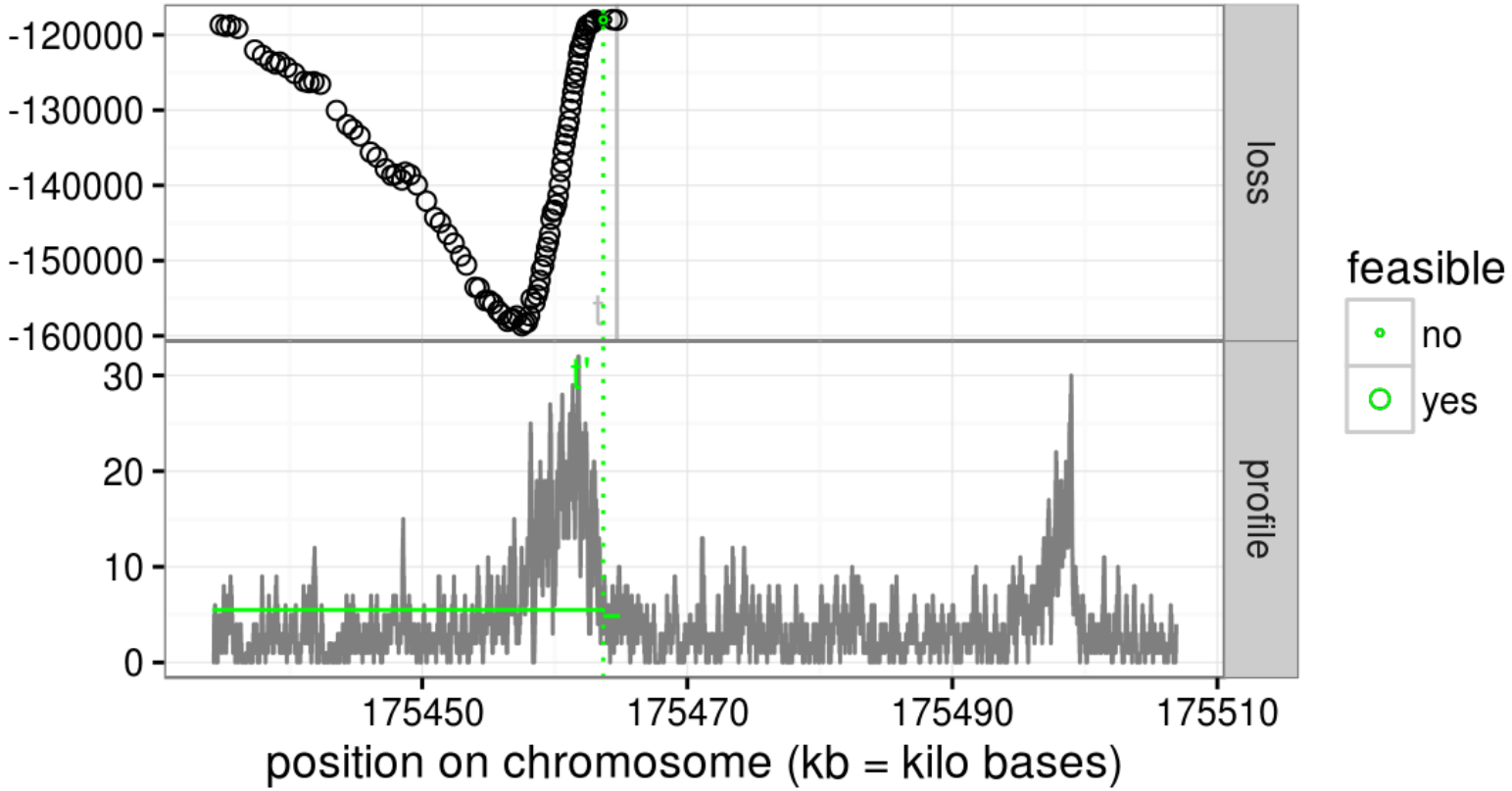

$$\mathcal{L}_{2,t} = \min_{t'<t} \underbrace{\mathcal{L}_{1,t'}}_{\text{optimal loss in 1 segment up to } t'} + \underbrace{c_{(t',t]}}_{\text{optimal loss of 2nd segment } (t',t]}$$

# Computation of optimal loss $\mathcal{L}_{s,t}$ for $s = 2$ segments up to data point $t < d$



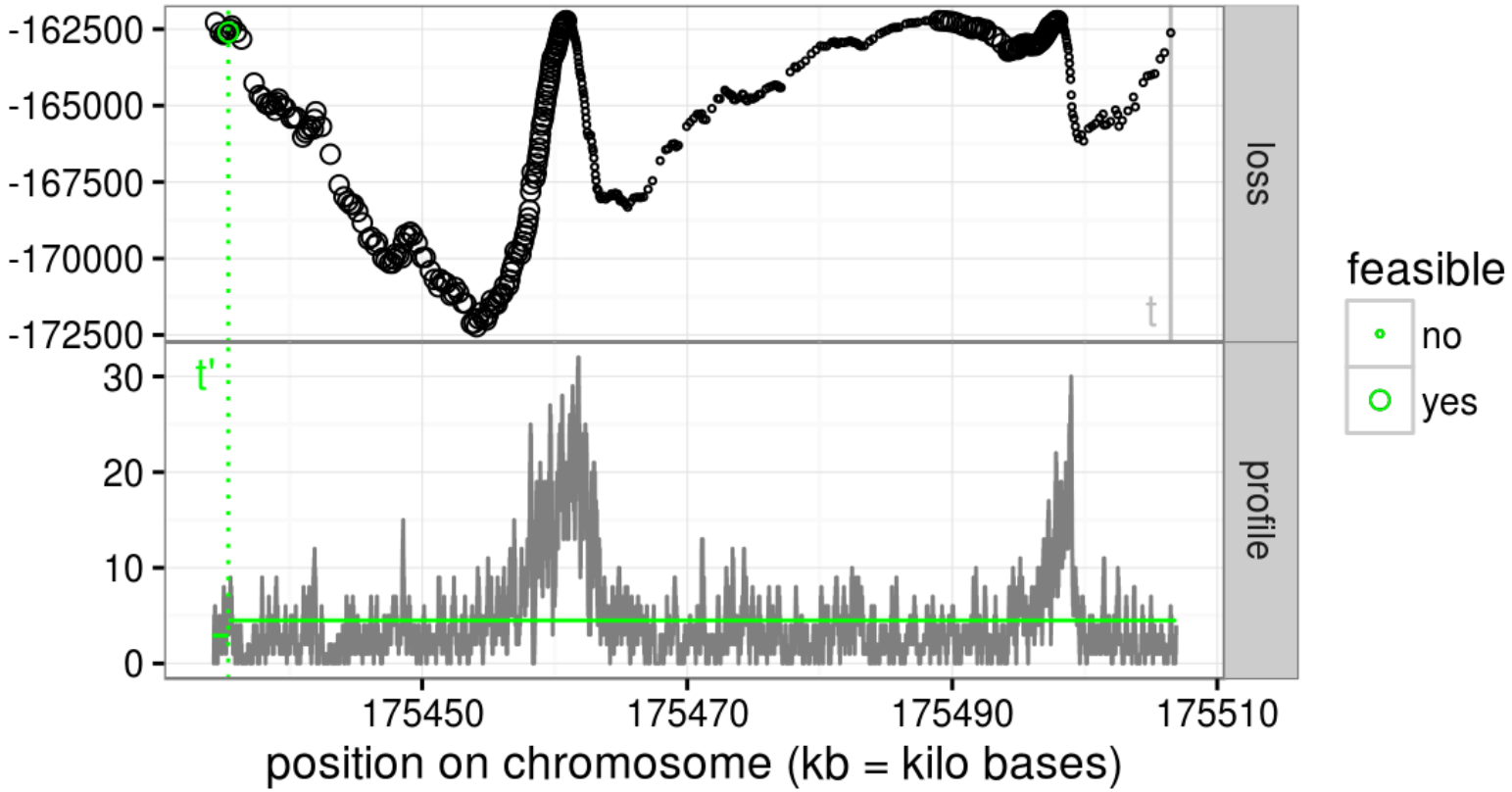

$$\mathcal{L}_{2,t} = \min_{t'<t} \underbrace{\mathcal{L}_{1,t'}}_{\text{optimal loss in 1 segment up to } t'} + \underbrace{c_{(t',t]}}_{\text{optimal loss of 2nd segment } (t',t]}$$

# Computation of optimal loss $\mathcal{L}_{s,t}$ for $s = 2$ segments up to data point $t < d$



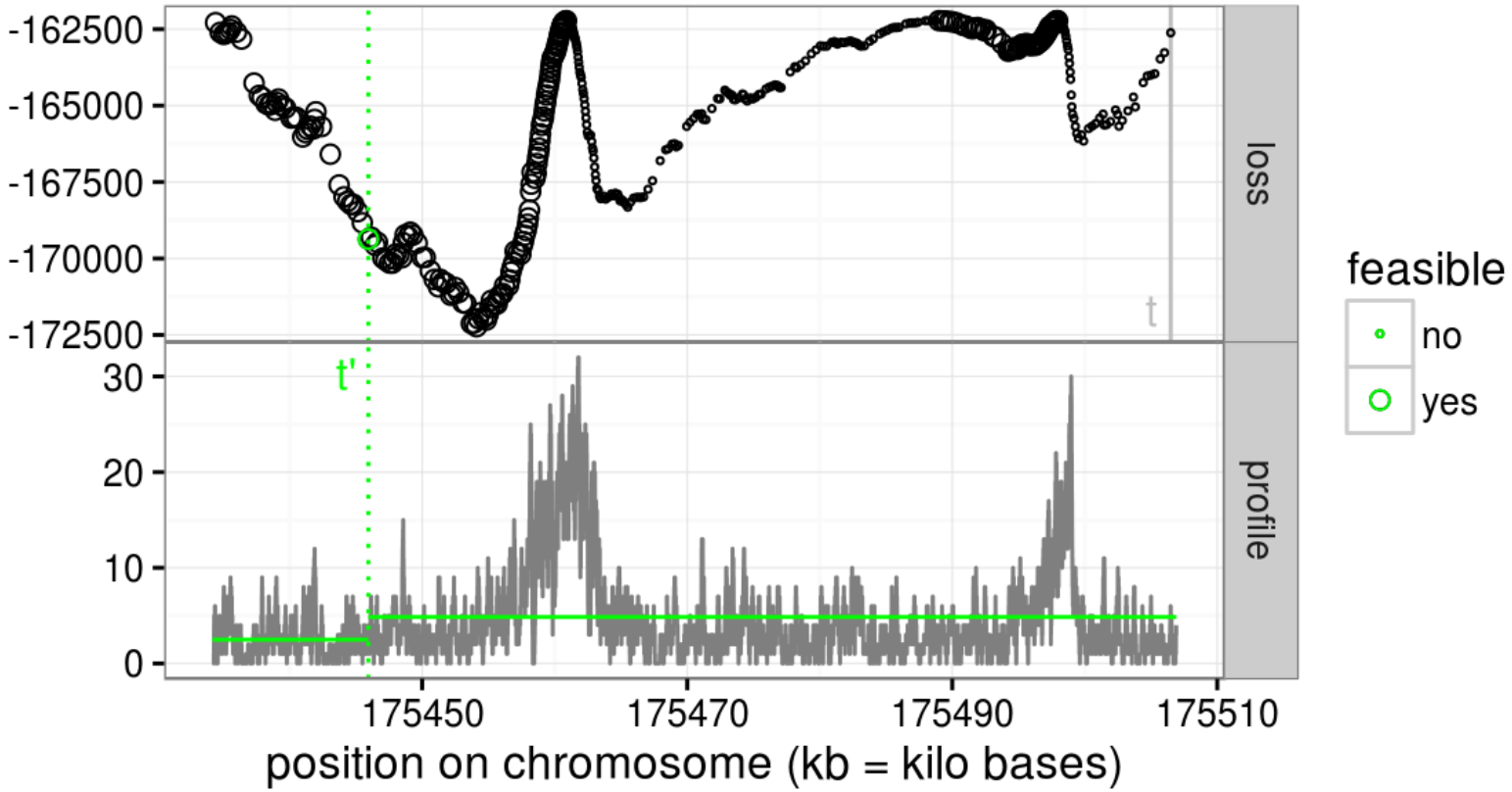

$$\mathcal{L}_{2,t} = \min_{t'<t} \underbrace{\mathcal{L}_{1,t'}}_{\text{optimal loss in 1 segment up to } t'} + \underbrace{c_{(t',t]}}_{\text{optimal loss of 2nd segment } (t',t]}$$

# Computation of optimal loss $\mathcal{L}_{s,t}$ for $s = 2$ segments up to last data point $t = d$



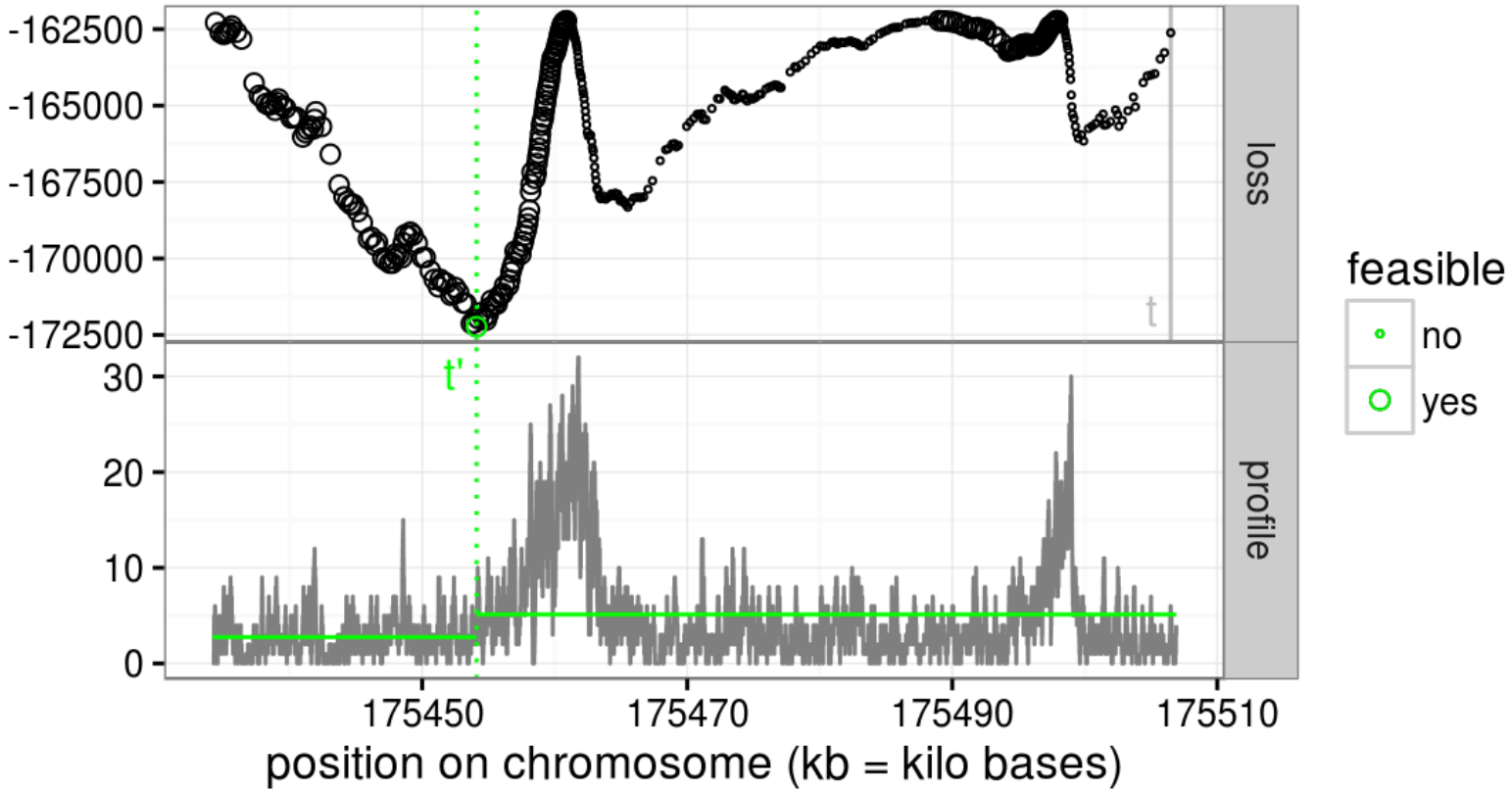

$$\mathcal{L}_{2,t} = \min_{t'<t} \underbrace{\mathcal{L}_{1,t'}}_{\text{optimal loss in 1 segment up to } t'} + \underbrace{c_{(t',t]}}_{\text{optimal loss of 2nd segment } (t',t]}$$

# Computation of optimal loss $\mathcal{L}_{s,t}$ for $s = 2$ segments up to last data point $t = d$



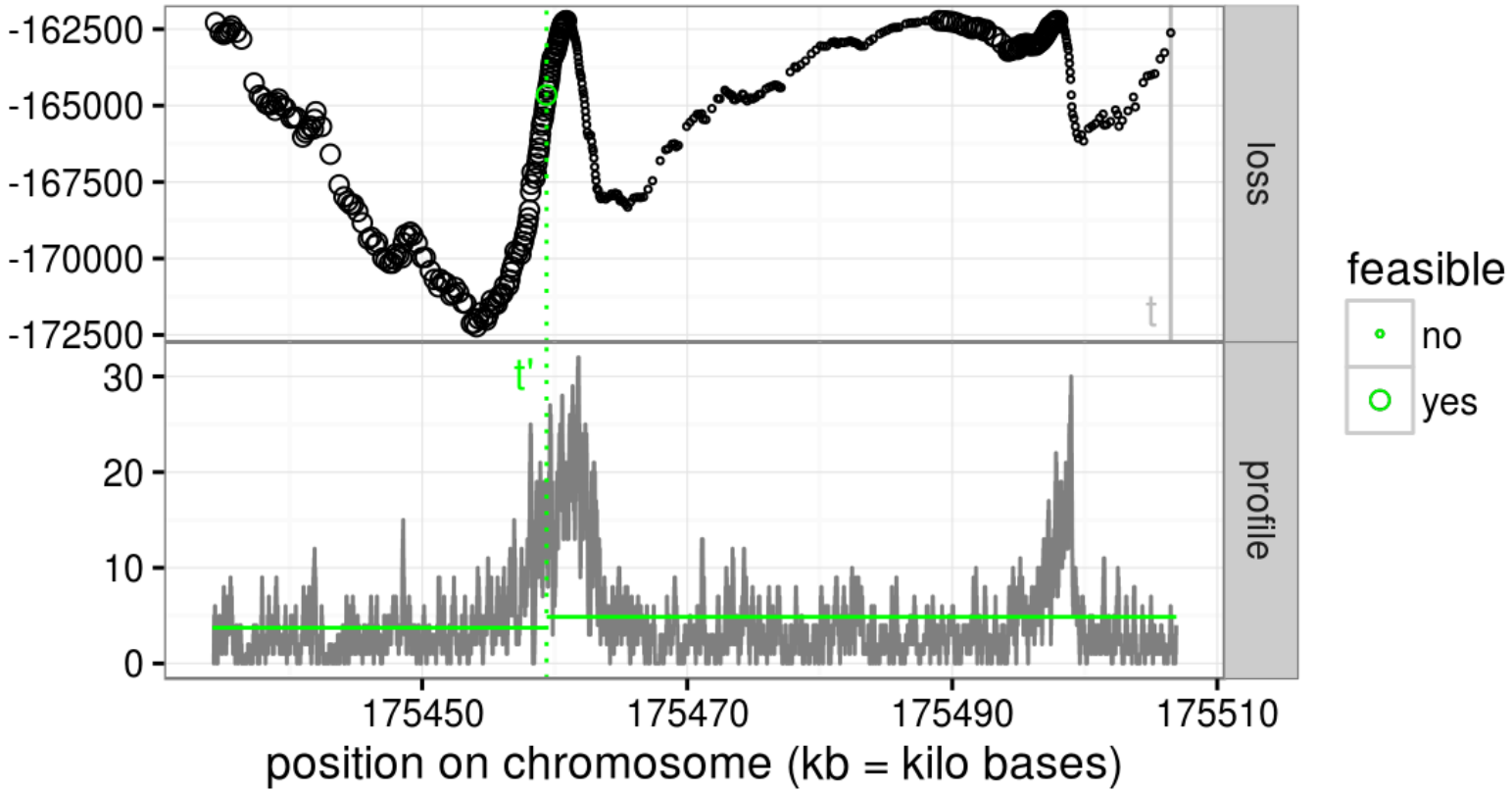

$$\mathcal{L}_{2,t} = \min_{t'<t} \underbrace{\mathcal{L}_{1,t'}}_{\text{optimal loss in 1 segment up to } t'} + \underbrace{c_{(t',t]}}_{\text{optimal loss of 2nd segment } (t',t]}$$

# Computation of optimal loss $\mathcal{L}_{s,t}$ for $s = 2$ segments up to last data point $t = d$



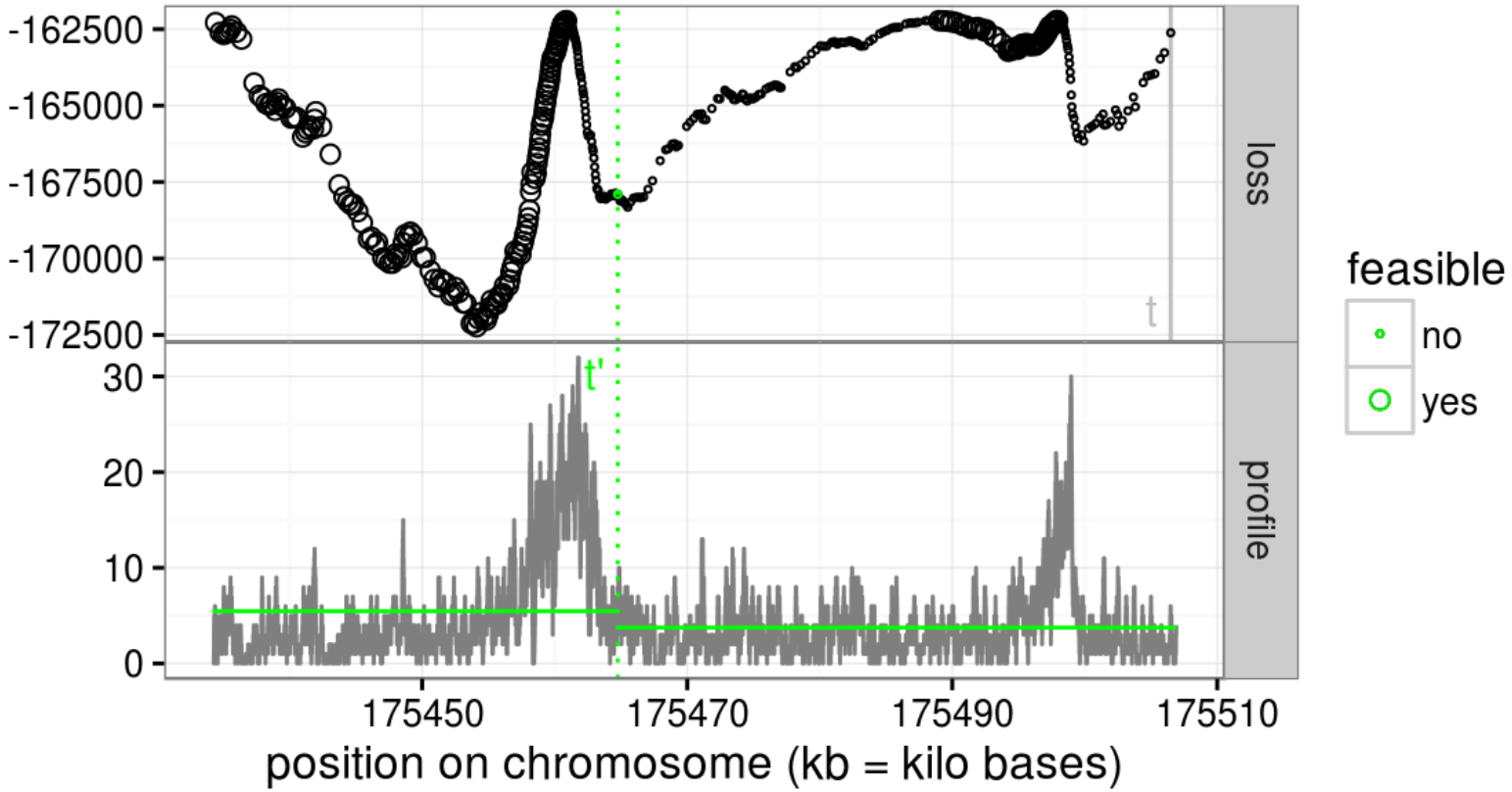

$$\mathcal{L}_{2,t} = \min_{t'<t} \underbrace{\mathcal{L}_{1,t'}}_{\text{optimal loss in 1 segment up to } t'} + \underbrace{c_{(t',t]}}_{\text{optimal loss of 2nd segment } (t',t]}$$

# Computation of optimal loss $\mathcal{L}_{s,t}$ for $s = 2$ segments up to last data point $t = d$



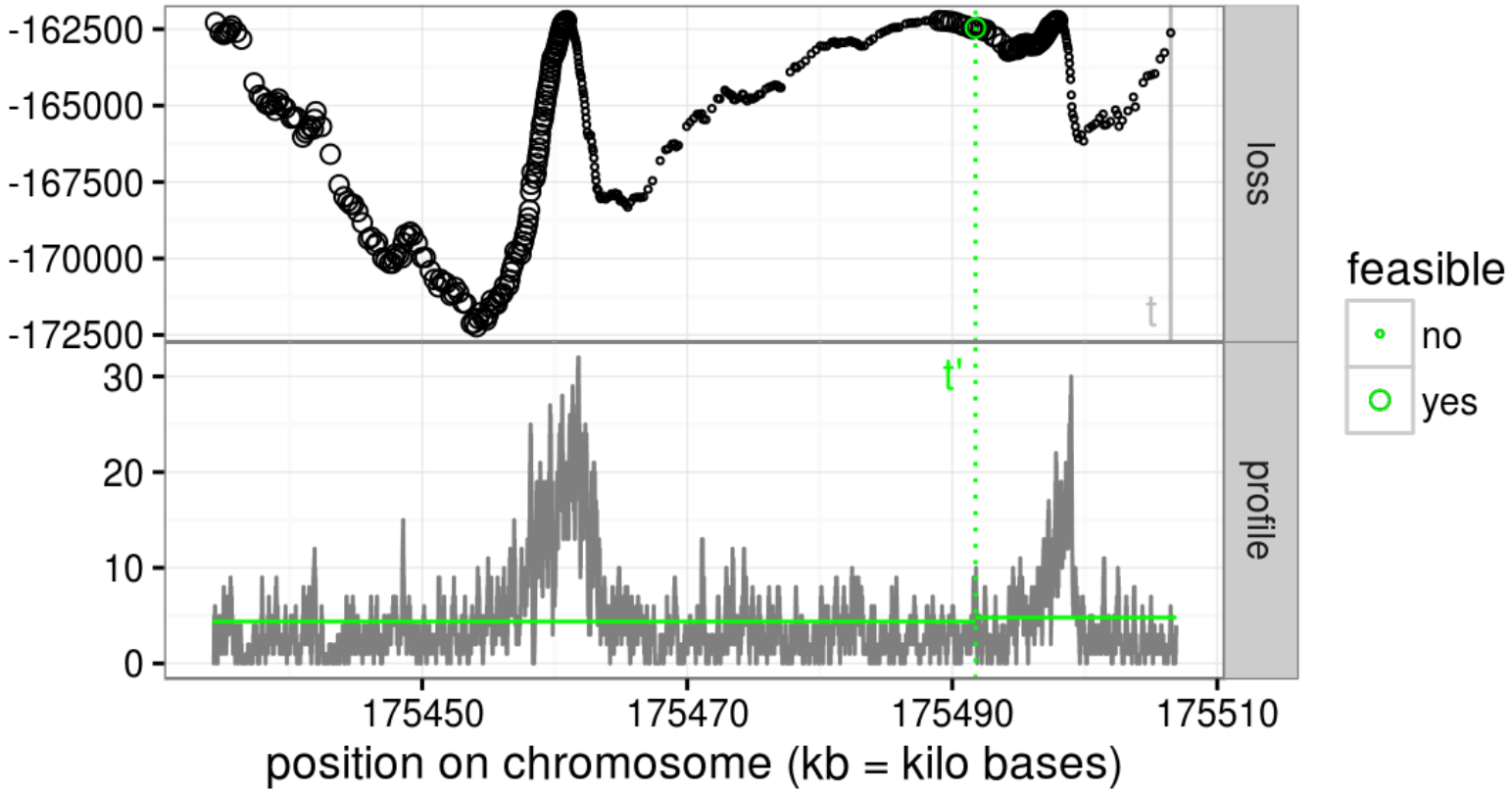

$$\mathcal{L}_{2,t} = \min_{t'<t} \underbrace{\mathcal{L}_{1,t'}}_{\text{optimal loss in 1 segment up to } t'} + \underbrace{c_{(t',t]}}_{\text{optimal loss of 2nd segment } (t',t]}$$

# Computation of optimal loss $\mathcal{L}_{s,t}$ for $s = 2$ segments up to last data point $t = d$




$$\mathcal{L}_{2,t} = \min_{t'<t} \underbrace{\mathcal{L}_{1,t'}}_{\text{optimal loss in 1 segment up to } t'} + \underbrace{c_{(t',t]}}_{\text{optimal loss of 2nd segment } (t',t]}$$

# Computation of optimal loss $\mathcal{L}_{s,t}$ for $s = 2$ segments up to last data point $t = d$




$$\mathcal{L}_{2,t} = \min_{t'<t} \underbrace{\mathcal{L}_{1,t'}}_{\text{optimal loss in 1 segment up to } t'} + \underbrace{c_{(t',t]}}_{\text{optimal loss of 2nd segment } (t',t]}$$

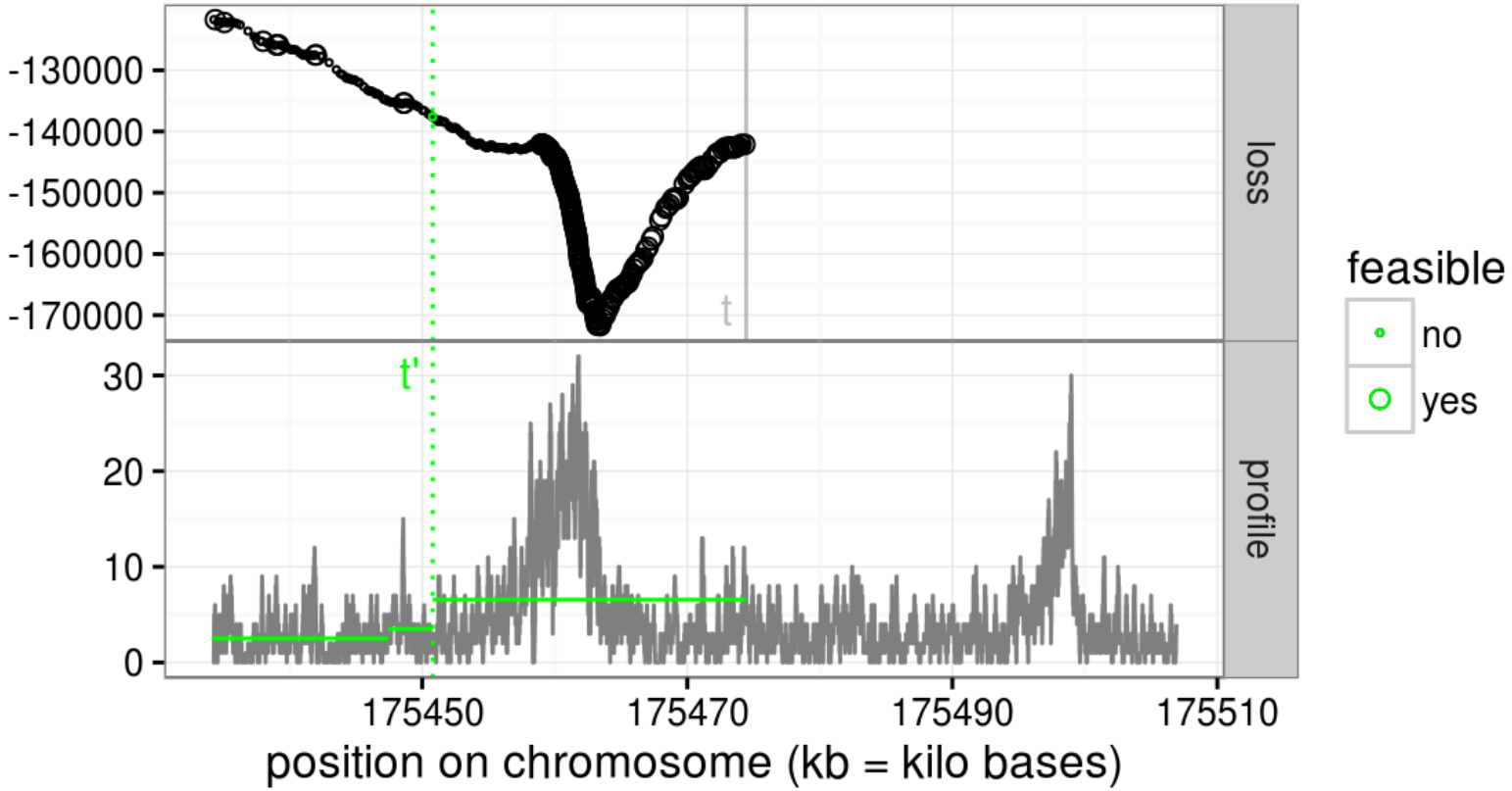
# Dynamic programming is faster than grid search for $s > 2$ segments

Computation time in number of data points $N$:

| segments $s$ | grid search | dynamic programming |
|---|---|---|
| 1 | $O(N)$ | $O(N)$ |
| 2 | $O(N^2)$ | $O(N^2)$ |
| 3 | $O(N^3)$ | $O(N^2)$ |
| 4 | $O(N^4)$ | $O(N^2)$ |
| $\vdots$ | $\vdots$ | $\vdots$ |

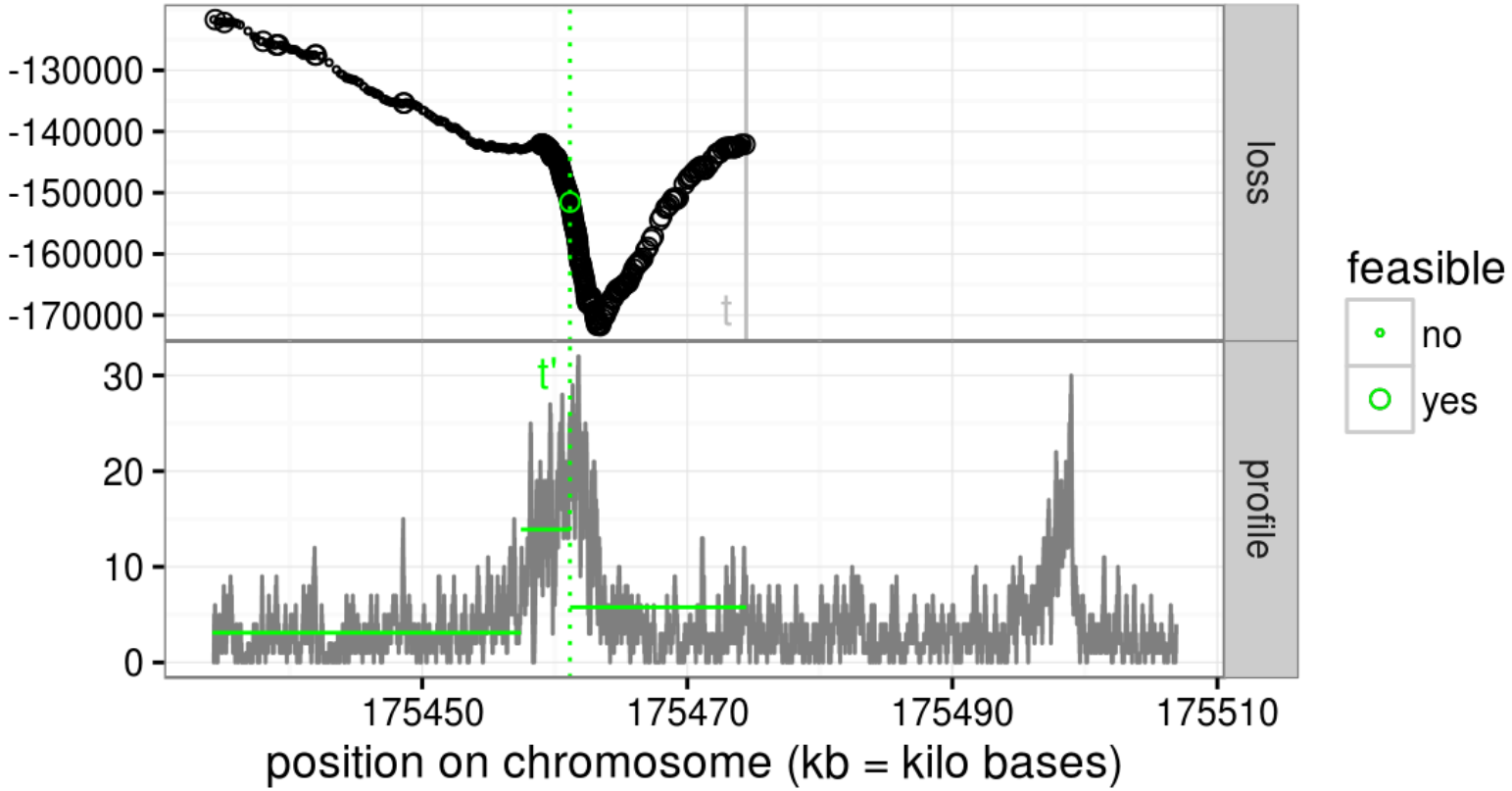
For example $N = 5735$ data points to segment.
$N^2 = 32890225$
$N^3 = 188625440375$
$\vdots$

# Computation of optimal loss $\mathcal{L}_{s,t}$ for $s = 3$ segments up to data point $t$



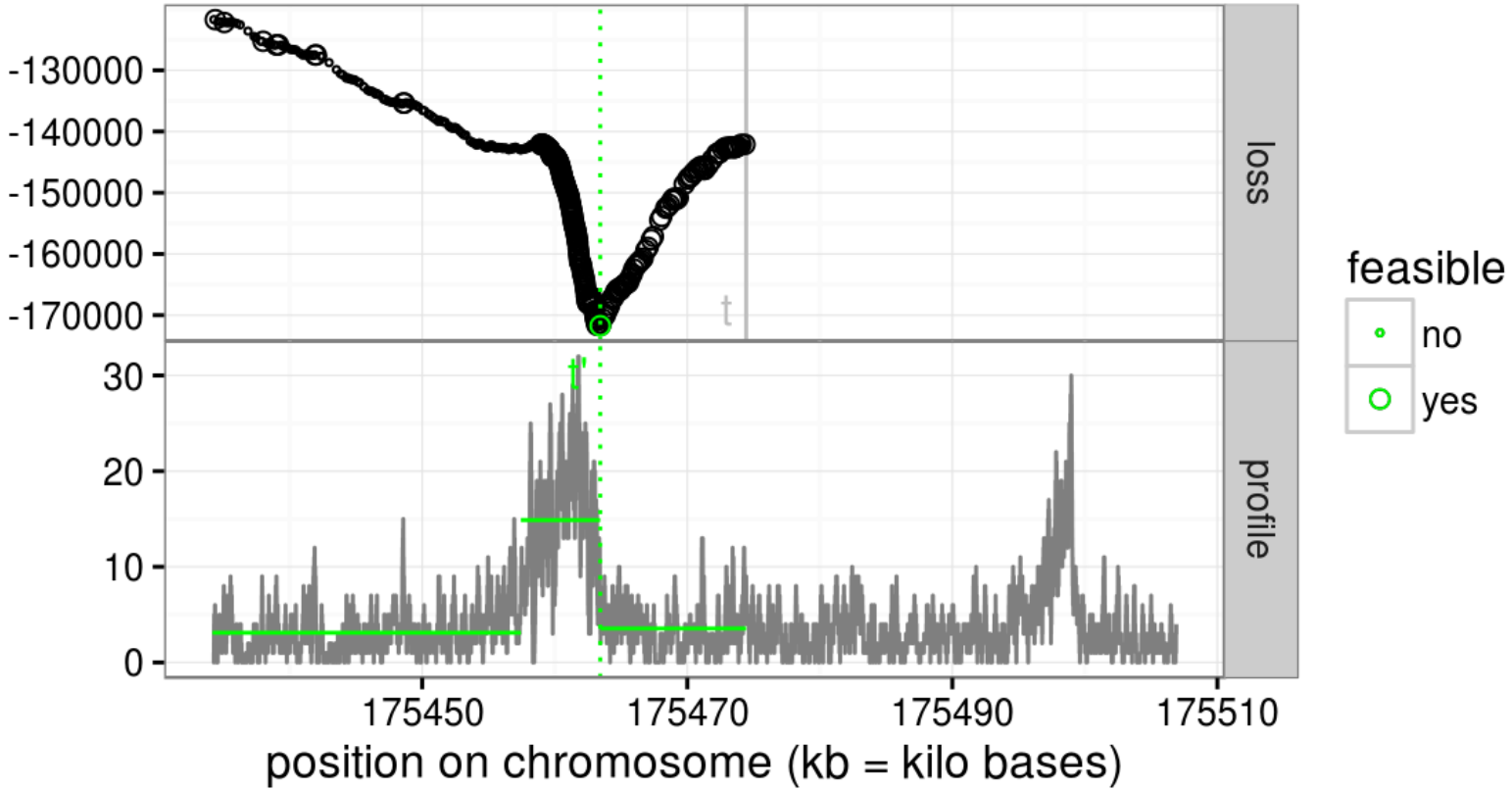

$$\mathcal{L}_{3,t} = \min_{t'<t} \underbrace{\mathcal{L}_{2,t'}}_{\text{optimal loss in 2 segments up to } t'} + \underbrace{c_{(t',t]}}_{\text{optimal loss of 3rd segment } (t',t]}$$

# Computation of optimal loss $\mathcal{L}_{s,t}$ for $s = 3$ segments up to data point $t$



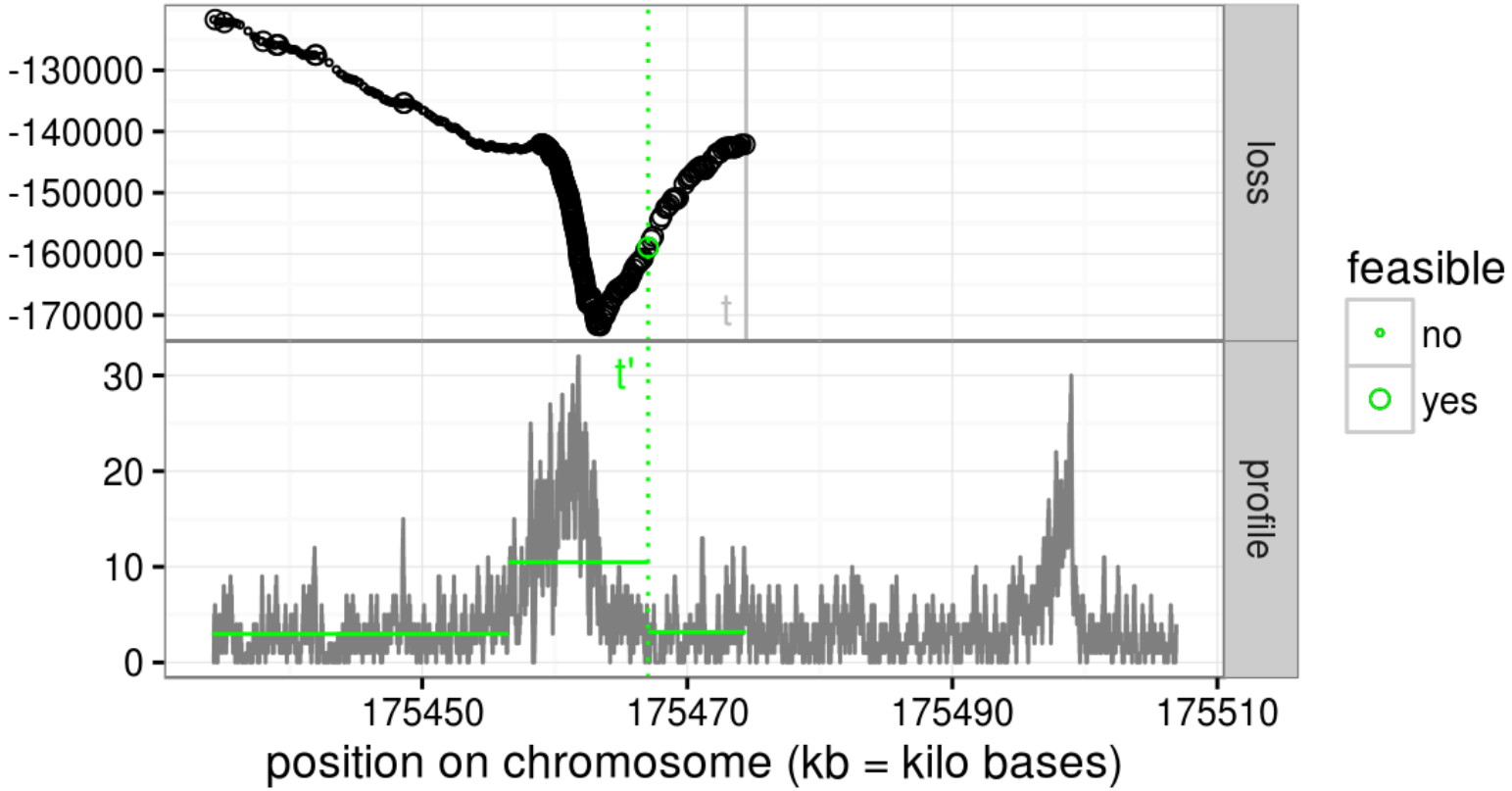

$$\mathcal{L}_{3,t} = \min_{t'<t} \underbrace{\mathcal{L}_{2,t'}}_{\text{optimal loss in 2 segments up to } t'} + \underbrace{c_{(t',t]}}_{\text{optimal loss of 3rd segment } (t',t]}$$

# Computation of optimal loss $\mathcal{L}_{s,t}$ for $s = 3$ segments up to data point $t$



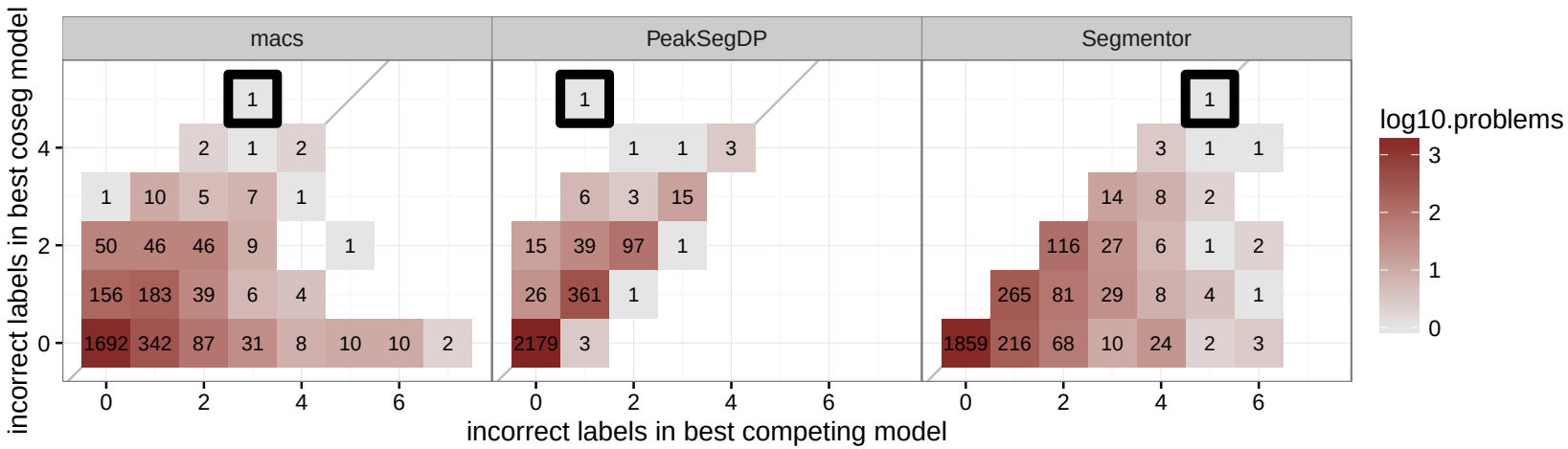

$$\mathcal{L}_{3,t} = \min_{t'<t} \underbrace{\mathcal{L}_{2,t'}}_{\text{optimal loss in 2 segments up to } t'} + \underbrace{c_{(t',t]}}_{\text{optimal loss of 3rd segment } (t',t]}$$

# Computation of optimal loss $\mathcal{L}_{s,t}$ for $s = 3$ segments up to data point $t$

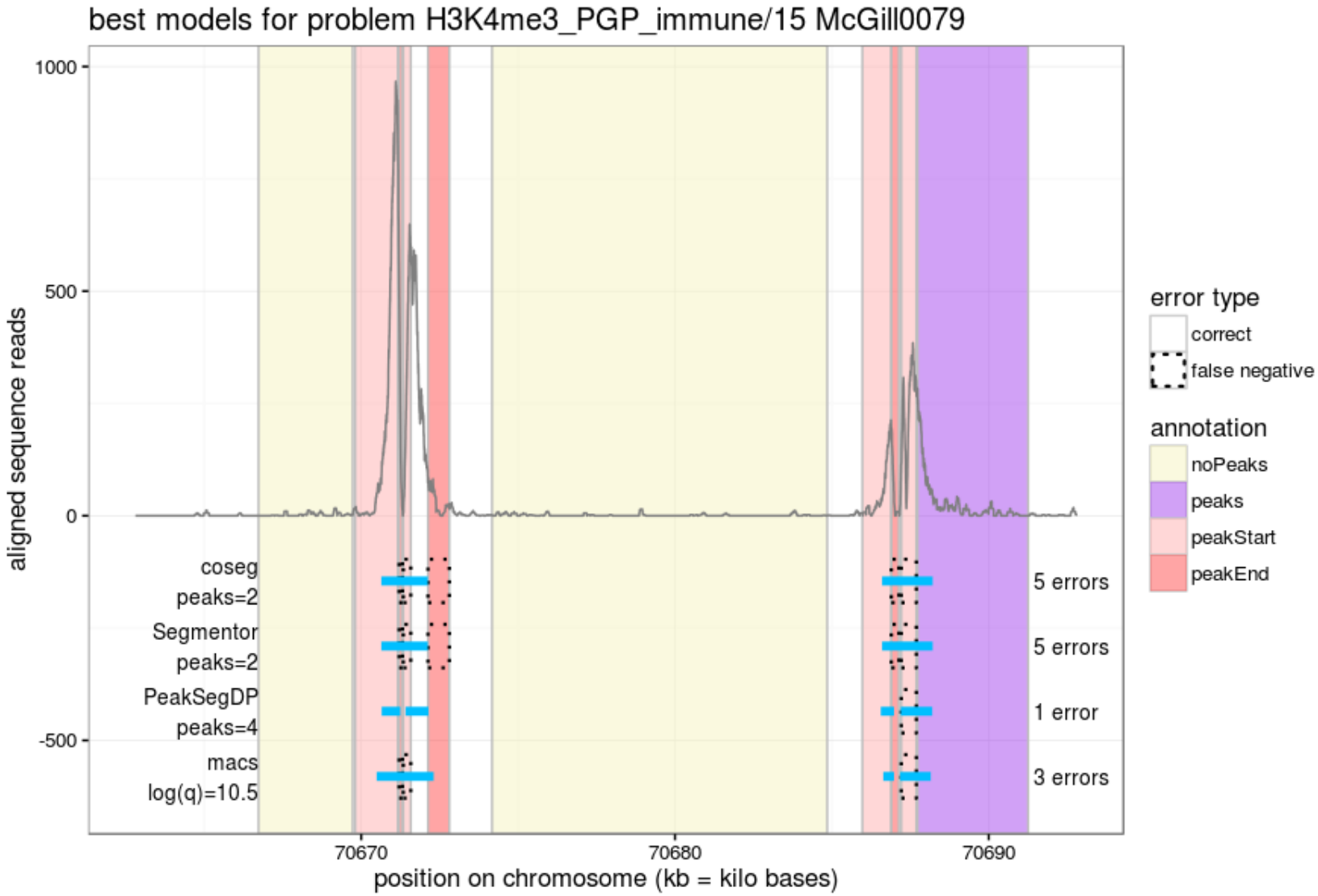



$$\mathcal{L}_{3,t} = \min_{t'<t} \underbrace{\mathcal{L}_{2,t'}}_{\text{optimal loss in 2 segments up to } t'} + \underbrace{c_{(t',t]}}_{\text{optimal loss of 3rd segment } (t',t]}$$

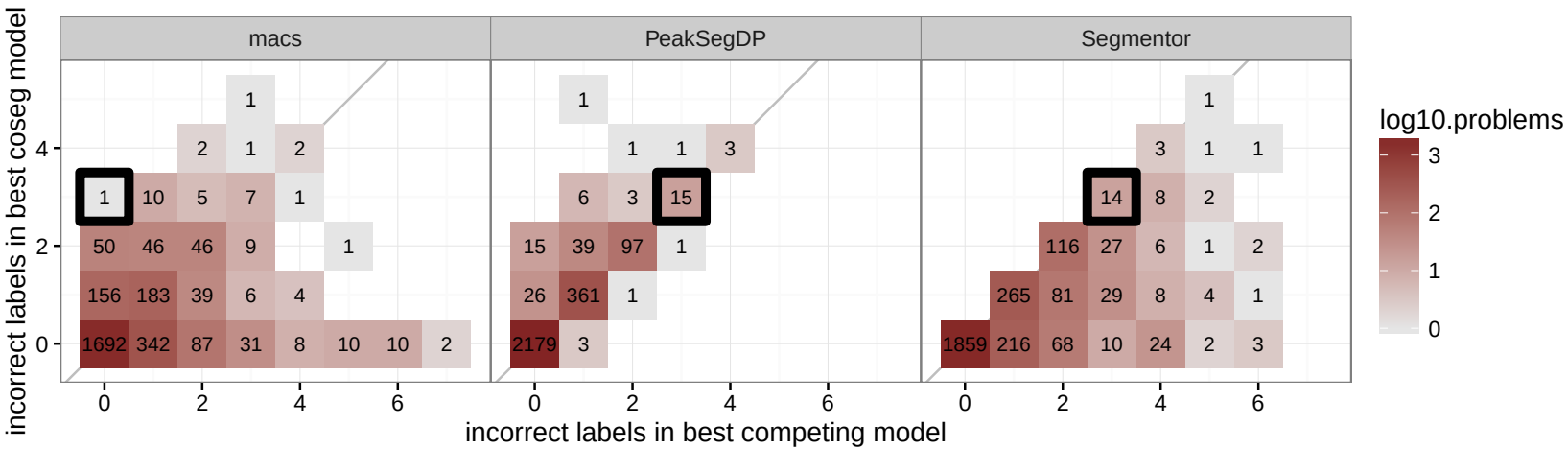
# 5 errors for coseg/Segmentor, only 1 error for PeakSegDP



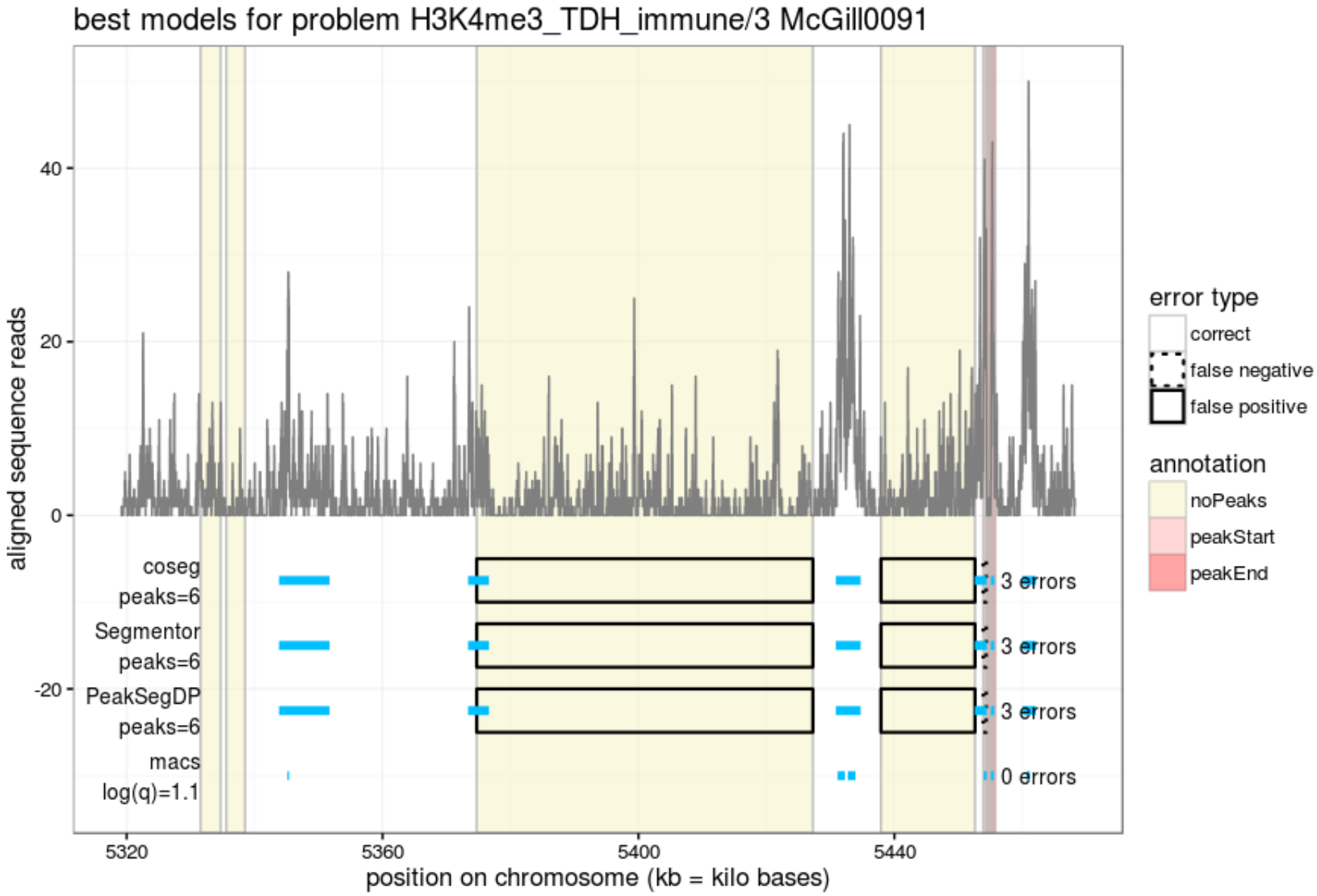
# 5 errors for coseg/Segmentor, only 1 error for PeakSegDP



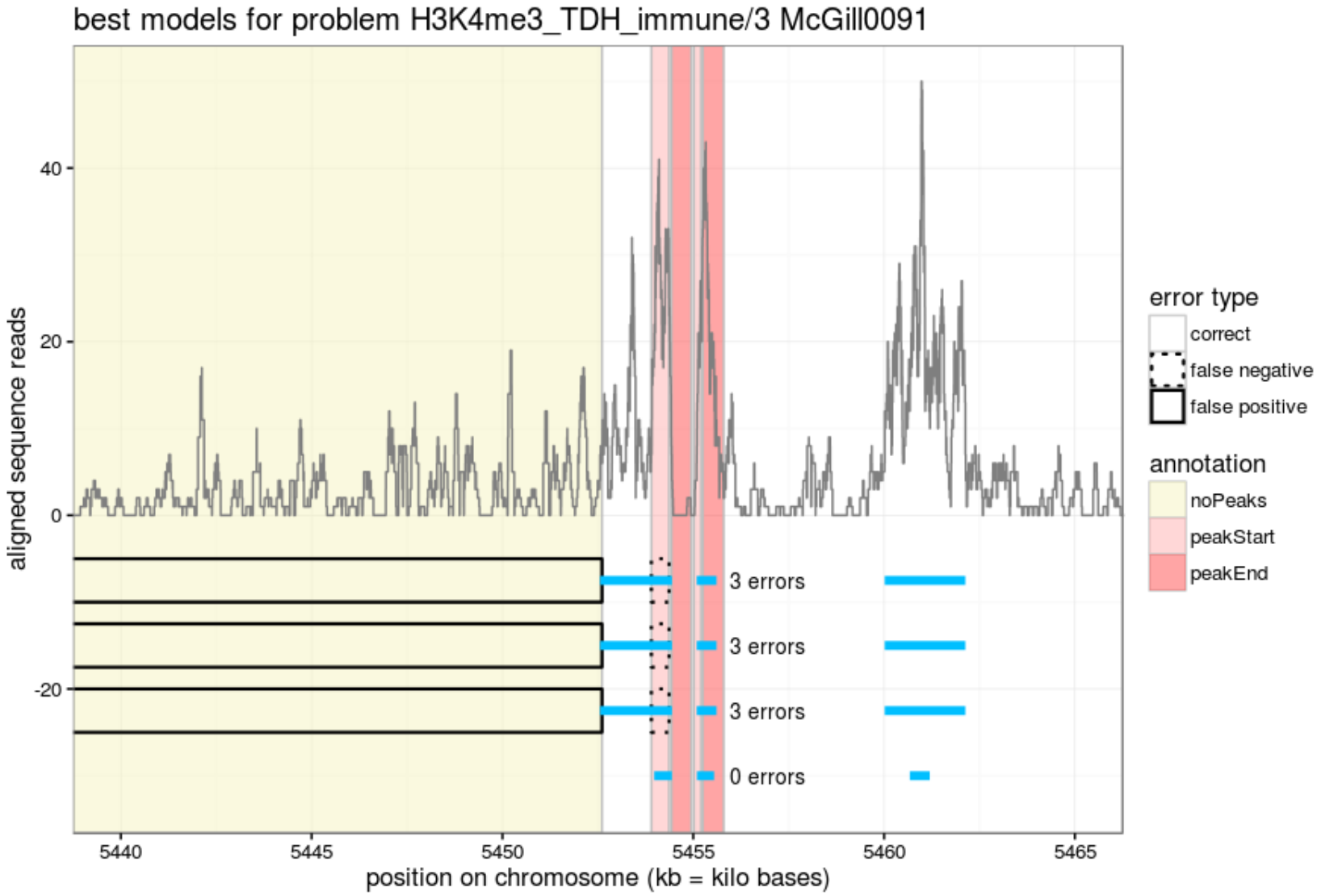
# Constrained optimization worse than macs



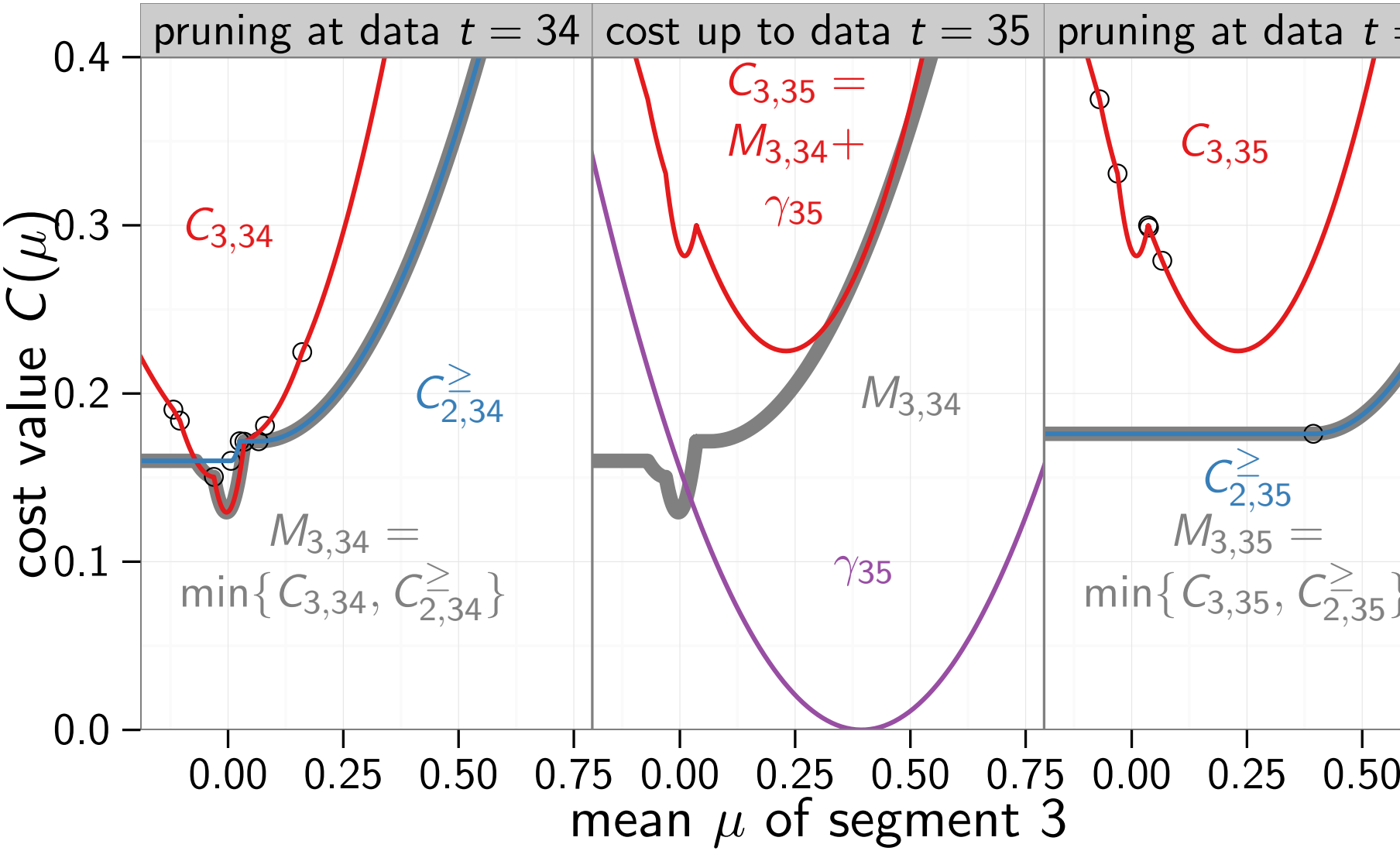
# Constrained optimization worse than macs



# Constrained optimization worse than macs

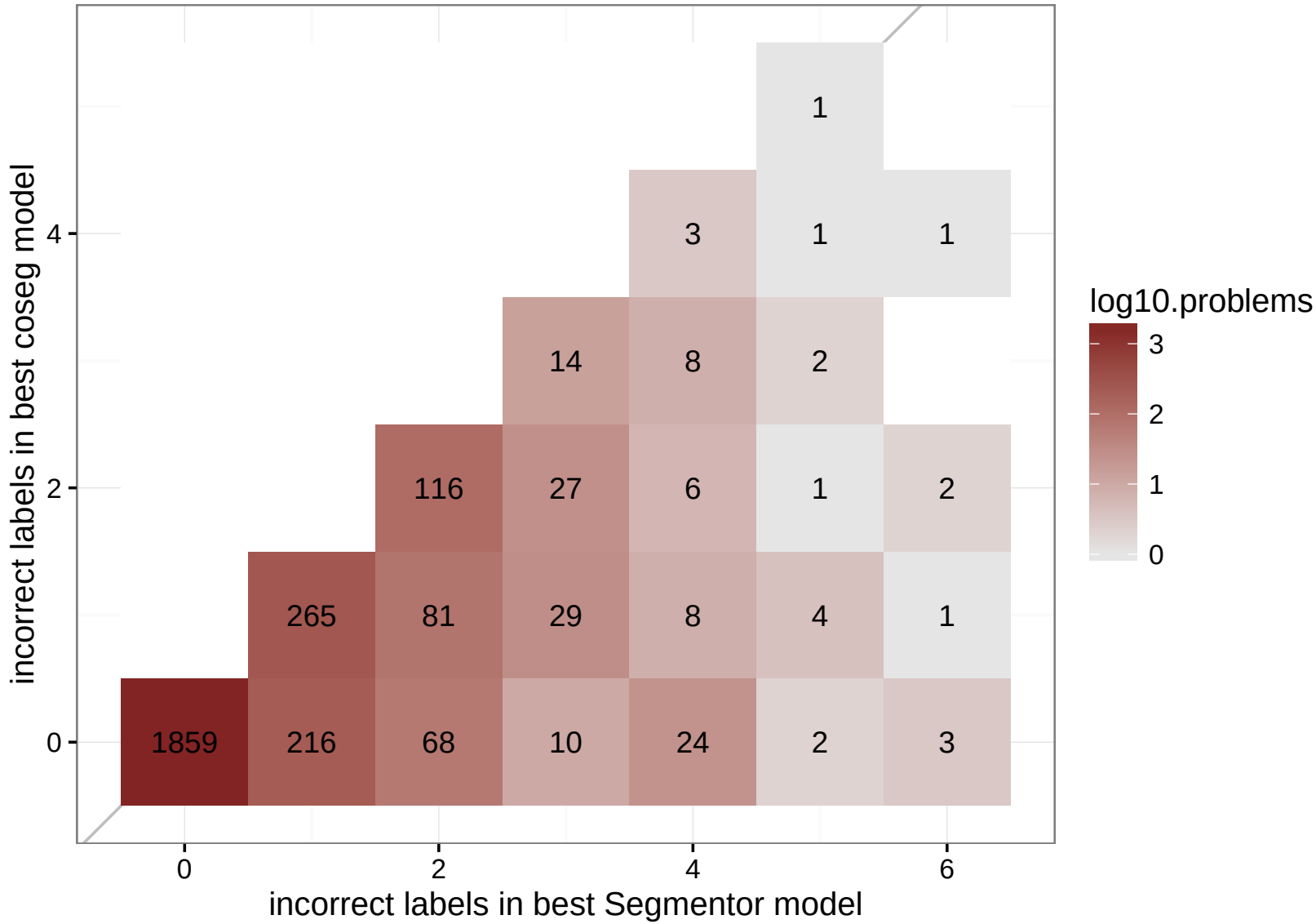


# Functional pruning complete example

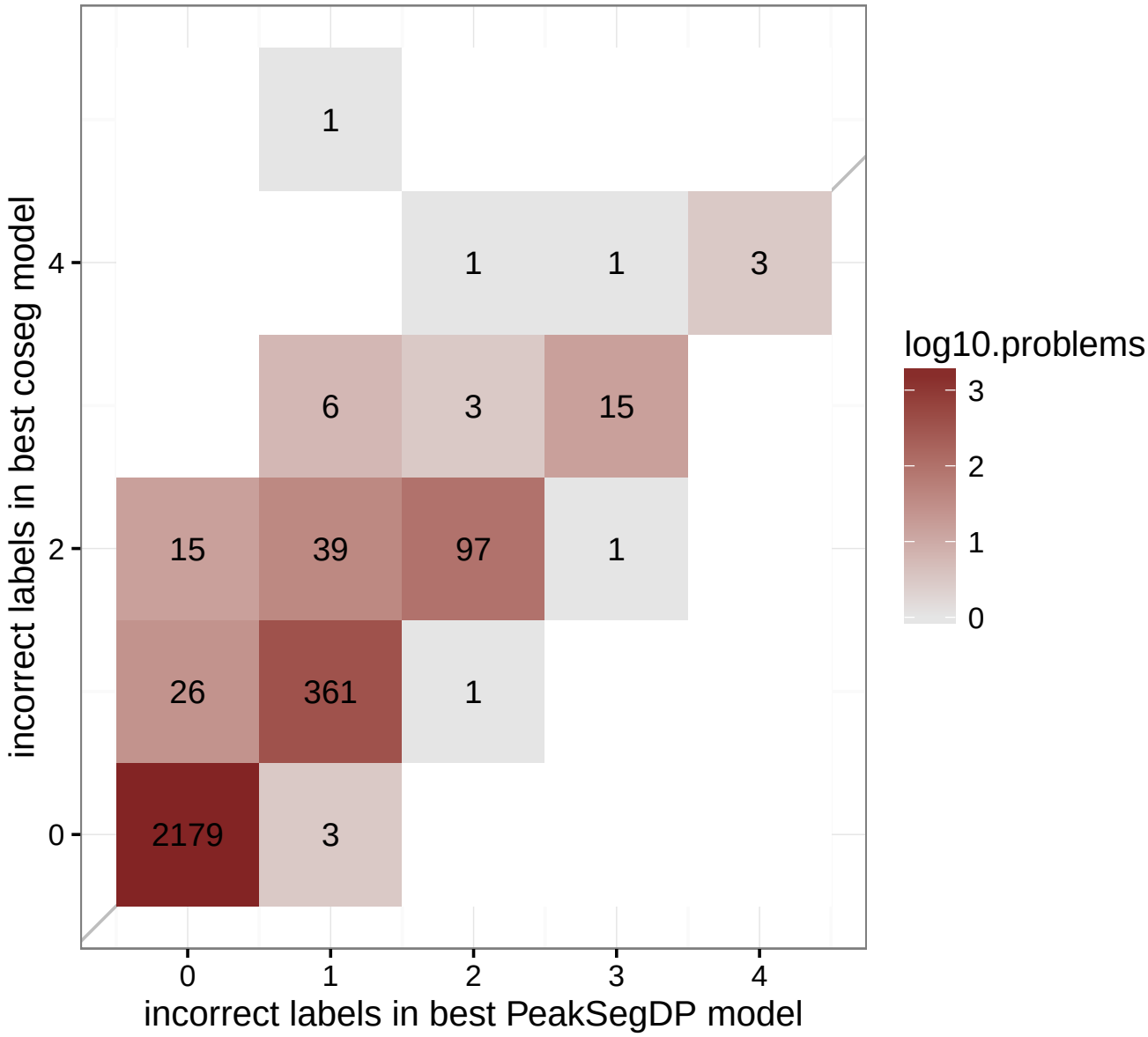



https://github.com/tdhock/PeakSegFPOP-paper

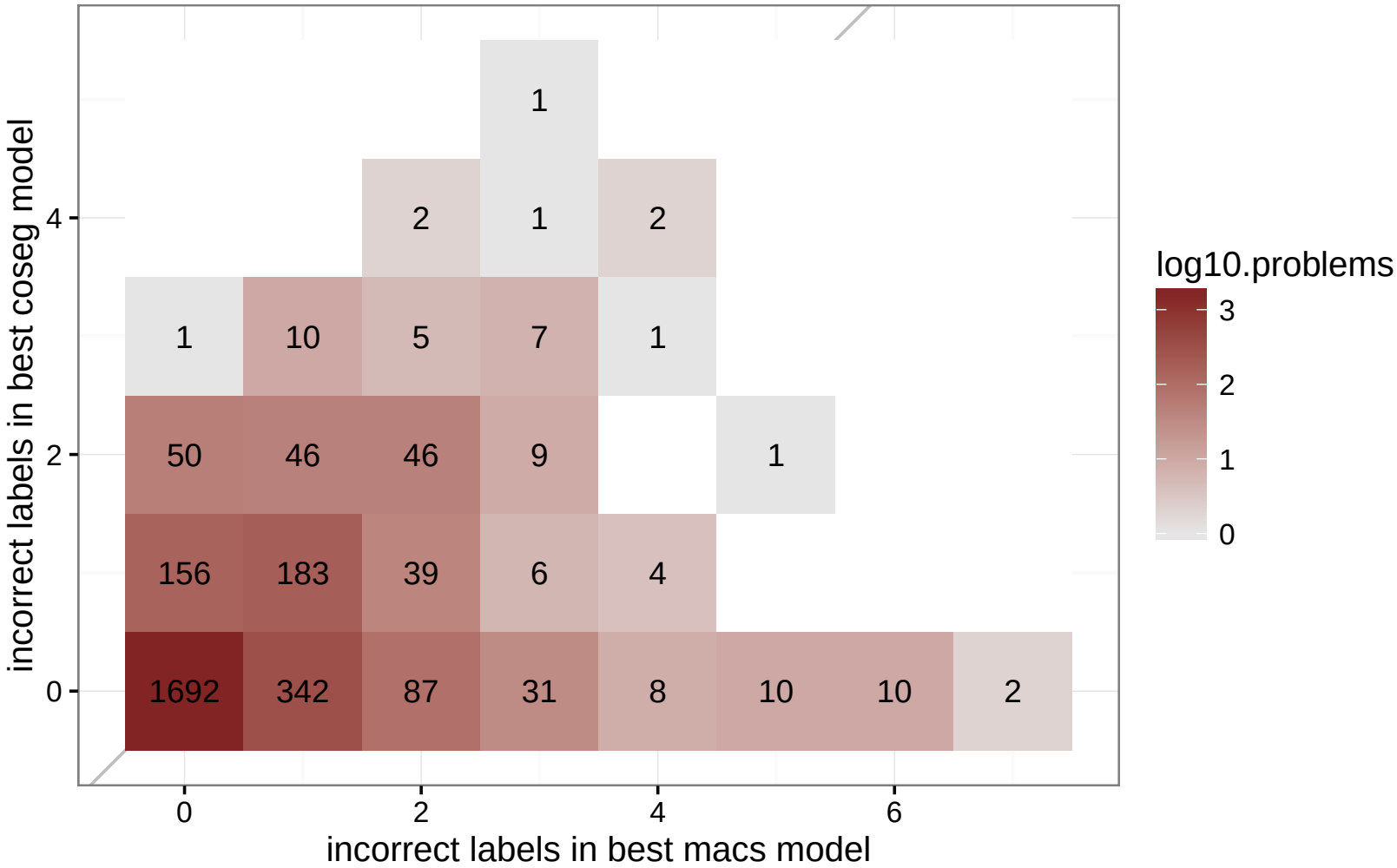
# New coseg algorithm more accurate than unconstrained maximum likelihood Poisson model (Segmentor)



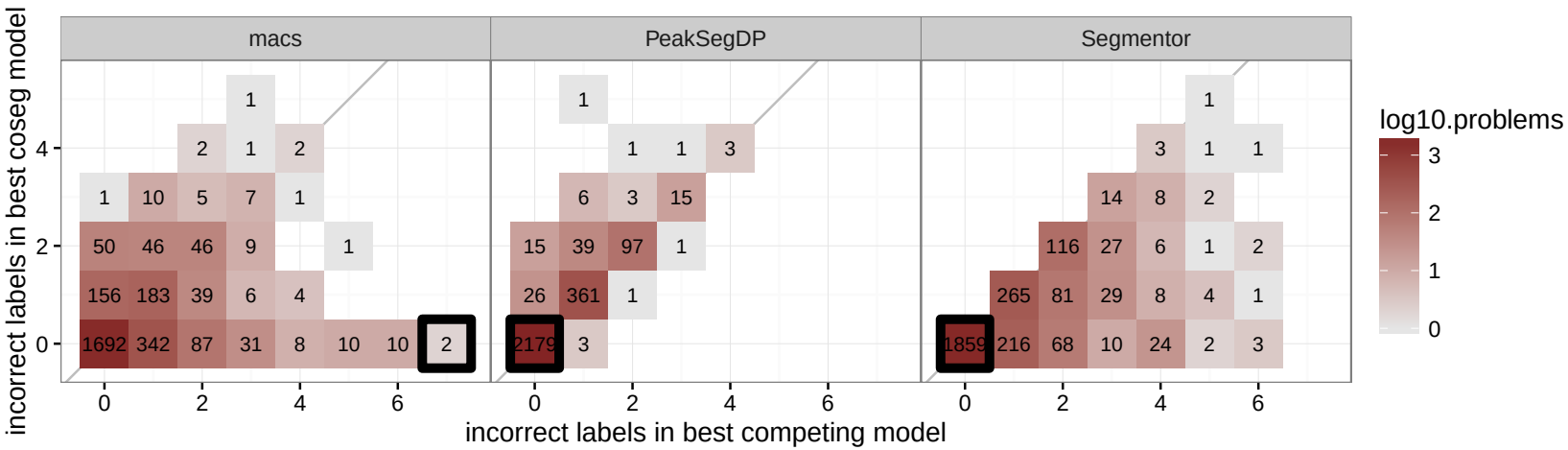
# New coseg algorithm mostly agrees with slower inexact DP



# New coseg algorithm sometimes disagrees with macs



# Constrained optimization better than macs



# 0 errors for coseg/PeakSegDP, 6 errors for Segmentor