

# A qsub pipeline for supervised ChIP-seq peak detection

Toby Dylan Hocking

[toby.hocking@mail.mcgill.ca](mailto:toby.hocking@mail.mcgill.ca)

joint work with Guillem Rigaill and Guillaume Bourque

2 April 2015

## ChIP-seq data and previous work on unsupervised peak detection

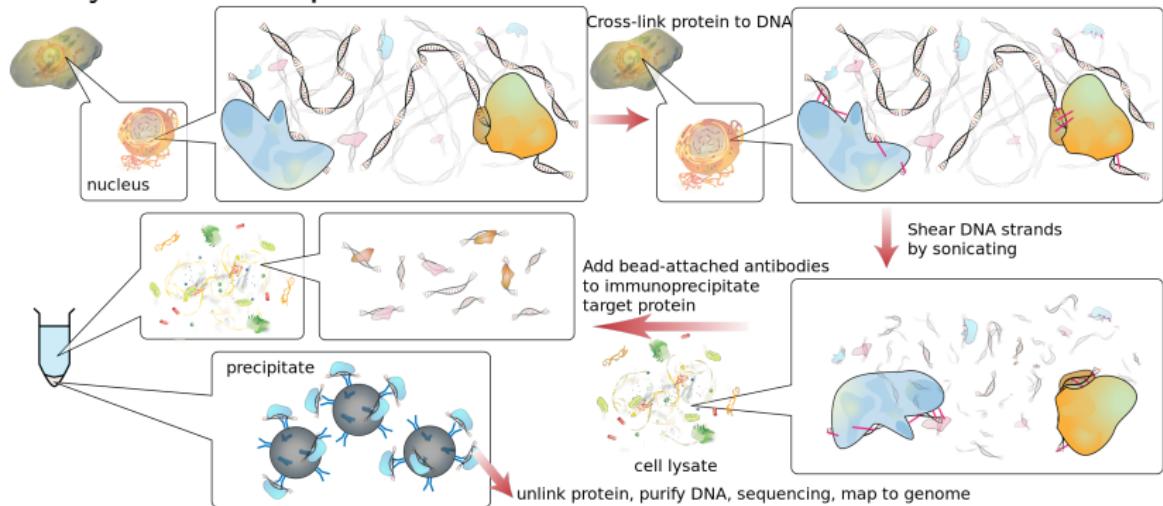
Supervised peak detection using annotated region labels

A qsub pipeline for supervised peak detection

Results on the McGill benchmark data set, conclusions

# Chromatin immunoprecipitation sequencing (ChIP-seq)

## Analysis of DNA-protein interactions.

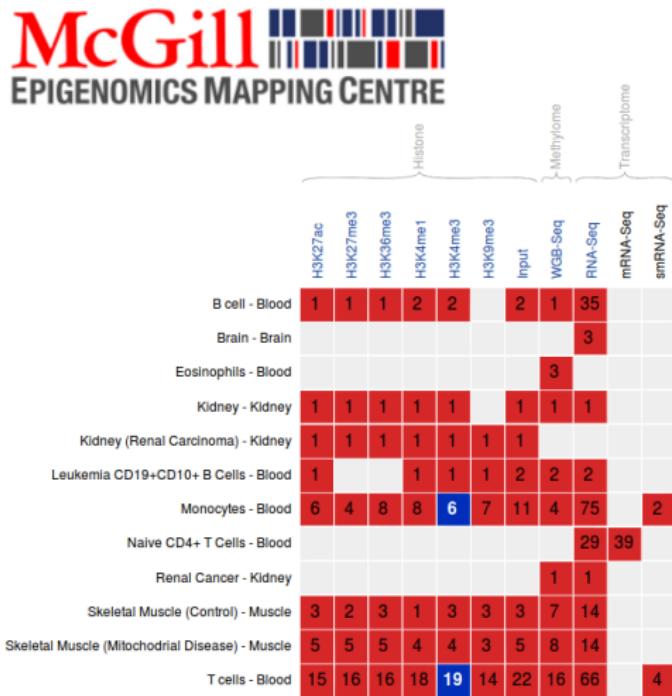


Source: “ChIP-sequencing,” Wikipedia.

# Data downloaded from Epigenomes Portal

McGill EMC: Datasets - Mozilla Firefox  
McGill EMC: Datasets   
[epigenomesportal.ca/edcc/](http://epigenomesportal.ca/edcc/)

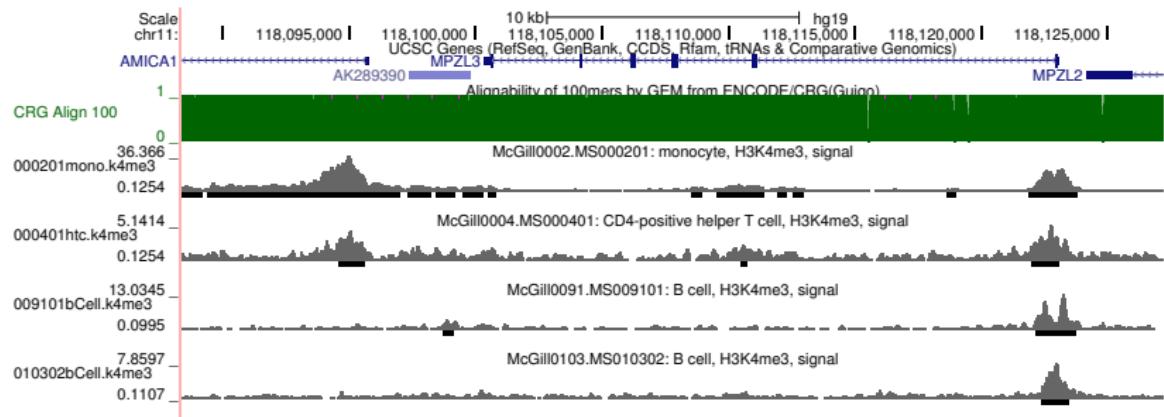
## McGill EPIGENOMICS MAPPING CENTRE



	H3K27ac	H3K27me3	H3K36me3	H3K4me1	H3K4me3	H3K9me3	Input	WGB-Seq	RNA-Seq	mRNA-Seq	smRNA-Seq
B cell - Blood	1	1	1	2	2		2	1	35		
Brain - Brain									3		
Eosinophils - Blood								3			
Kidney - Kidney	1	1	1	1	1		1	1	1		
Kidney (Renal Carcinoma) - Kidney	1	1	1	1	1	1	1				
Leukemia CD19+CD10+ B Cells - Blood	1			1	1	1	2	2	2		
Monocytes - Blood	6	4	8	8	6	7	11	4	75	2	
Naive CD4+ T Cells - Blood								29	39		
Renal Cancer - Kidney							1	1			
Skeletal Muscle (Control) - Muscle	3	2	3	1	3	3	3	7	14		
Skeletal Muscle (Mitochondrial Disease) - Muscle	5	5	5	4	4	3	5	8	14		
T cells - Blood	15	16	16	18	19	14	22	16	66	4	

[Visualize in Genome Browser](#)[Get track hub link](#)[Download tracks](#)

# Goal: find peaks in each of several samples



## Existing unsupervised peak detection algorithms

- ▶ Model-based analysis of ChIP-Seq (MACS), Zhang et al, 2008.
- ▶ SICER, Zang et al, 2009.
- ▶ HOMER findPeaks, Heinz et al, 2010.
- ▶ RSEG, Song and Smith, 2011.
- ▶ Histone modifications in cancer (HMCan), Ashoor et al, 2013.
- ▶ ... dozens of others.

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

- ▶ ...algorithm?
- ▶ ...parameters?

# Problem: how to choose model parameters?

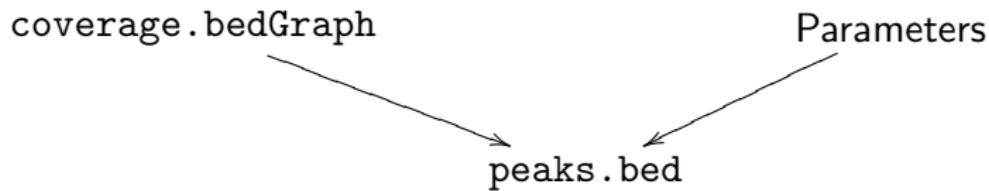
19 parameters for Model-based analysis of ChIP-Seq (MACS), Zhang et al, 2008.

```
[-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
```

# Summary of unsupervised peak detectors



ChIP-seq data and previous work on unsupervised peak detection

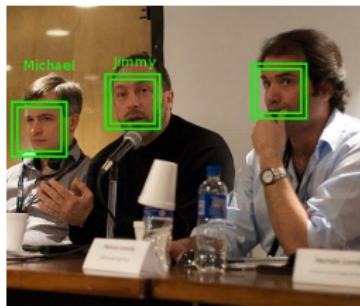
Supervised peak detection using annotated region labels

A qsub pipeline for supervised peak detection

Results on the McGill benchmark data set, conclusions

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

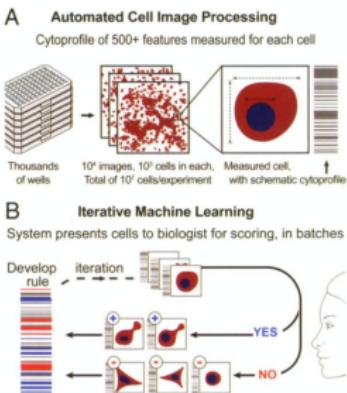
Photos



Labels: names

CVPR 2013  
246 papers

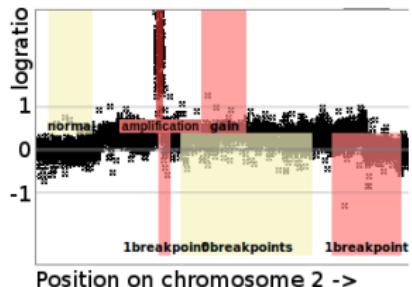
Cell images



phenotypes

CellProfiler  
873 citations

Copy number profiles

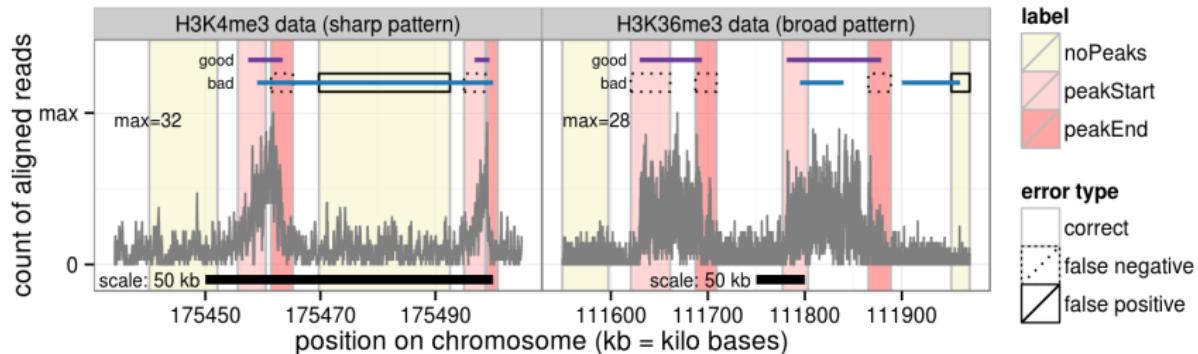


alterations

SegAnnDB  
H, et. al. 2014.

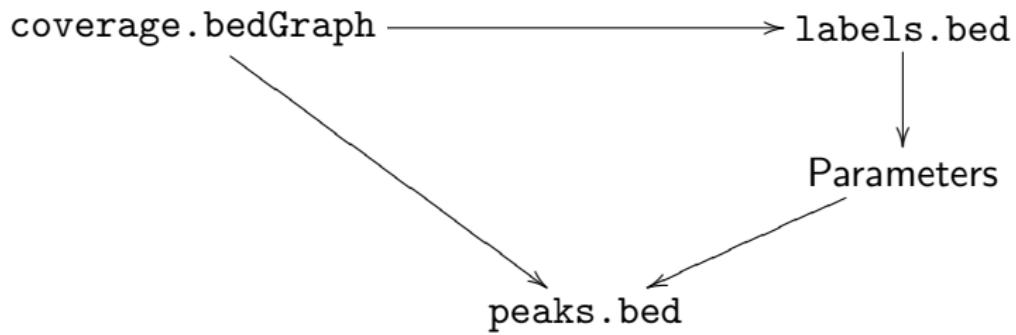
Sources: [http://en.wikipedia.org/wiki/Face\\_detection](http://en.wikipedia.org/wiki/Face_detection)  
Jones et al PNAS 2009. Scoring diverse cellular morphologies in image-based screens with iterative feedback and machine learning.

# Peak detector accuracy can be quantified using manually annotated region labels

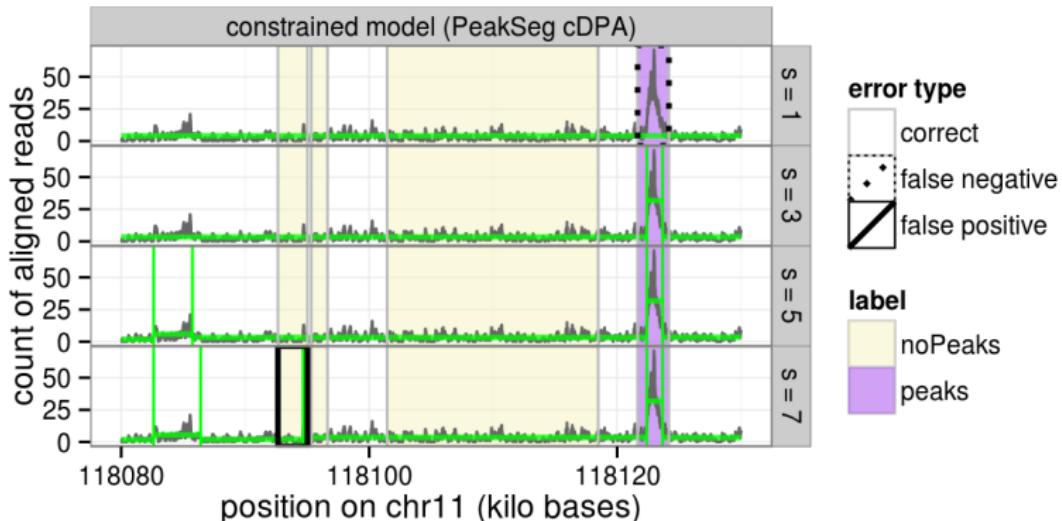


- ▶ Good peaks have 0 incorrect regions.
- ▶ Bad peaks have 7 incorrect regions.
- ▶ Goal: minimize number of incorrect regions.

## Summary of supervised peak detection



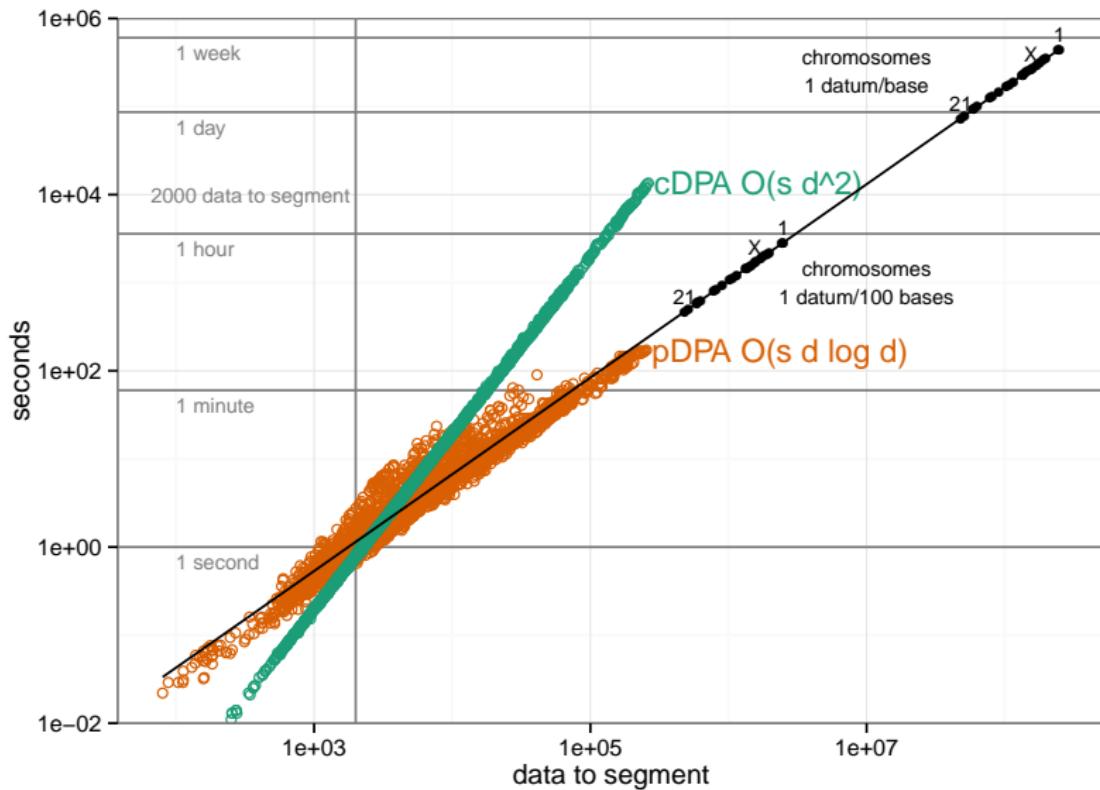
# PeakSeg constrained maximum likelihood model



State-of-the-art peak detection (Hocking et al, ICML2015), but constrained Dynamic Programming Algorithm = cDPA has  $O(s_{\max} d^2)$  time complexity:

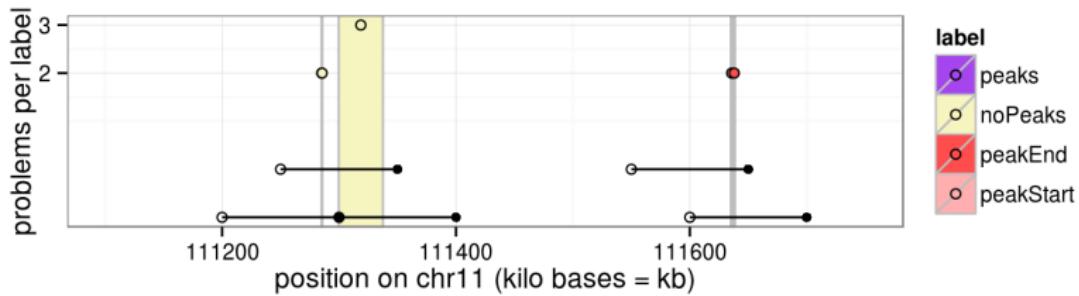
- ▶  $s_{\max}$  = maximum number of segments/peaks,
- ▶  $d$  = data points to segment.

# Timings on benchmark data sets



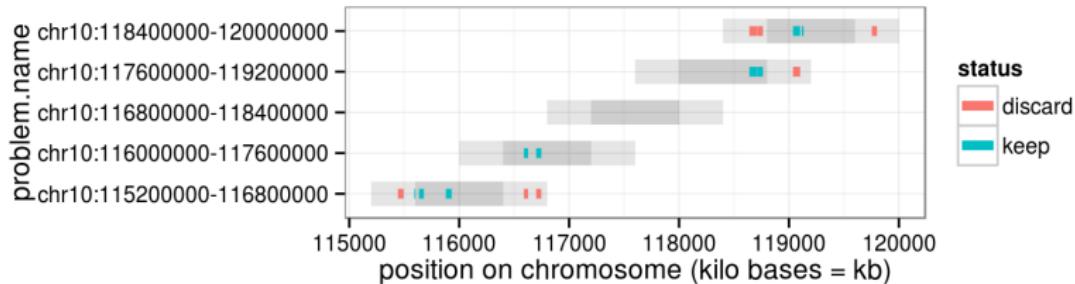
How to use cDPA on whole genome?

## Break the genome into separate segmentation problems



Segmentation problems (black) with labeled regions (colors).

## Only keep peaks in the center of each problem



Segmentation problems (light grey)  
with regions where peaks are kept (dark grey).

## Summary of proposed PeakSeg model

- ▶ Fix  $d = 2000$  data points (bins) per segmentation problem.
- ▶ Split the genome into separate segmentation problems of size  $R$ .
- ▶ Try several different resolutions  $R$ ,  
select the resolution  $R$  with minimal train error.
- ▶ Learn the PeakSeg model using the labeled problems.
- ▶ Use the model to predict peaks on all problems.

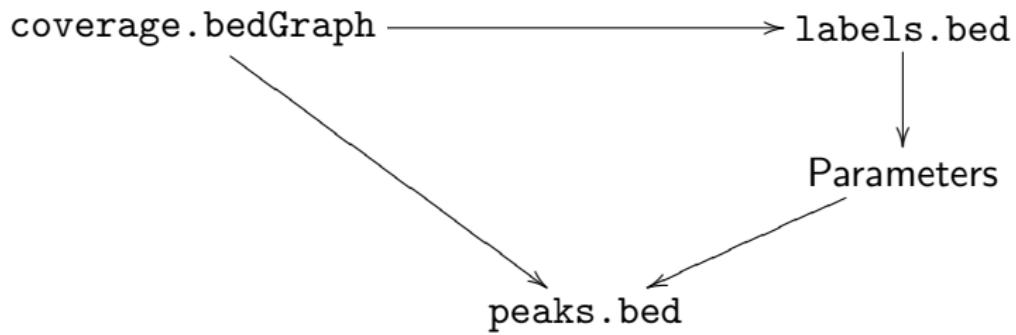
ChIP-seq data and previous work on unsupervised peak detection

Supervised peak detection using annotated region labels

A qsub pipeline for supervised peak detection

Results on the McGill benchmark data set, conclusions

## Summary of supervised peak detection



ChIP-seq data and previous work on unsupervised peak detection

Supervised peak detection using annotated region labels

A qsub pipeline for supervised peak detection

Results on the McGill benchmark data set, conclusions

## Benchmark labeled data sets

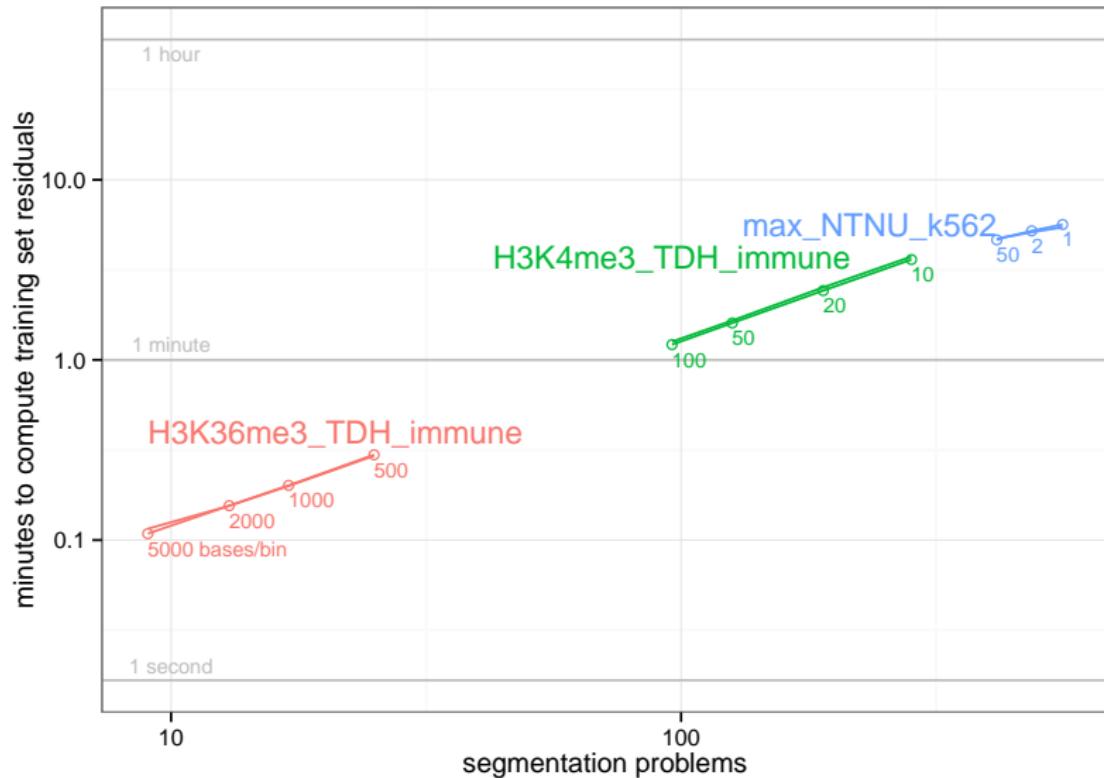
<http://cbio.ensmp.fr/~thocking/chip-seq-chunk-db/>

- ▶ 4 annotators (AM, TDH, PGP, XJ).
- ▶ 8 cell types.
- ▶ 37 annotated H3K4me3 profiles (sharp peaks).
- ▶ 29 annotated H3K36me3 profiles (broadly enriched domains).
- ▶ 12,826 annotated regions in total.

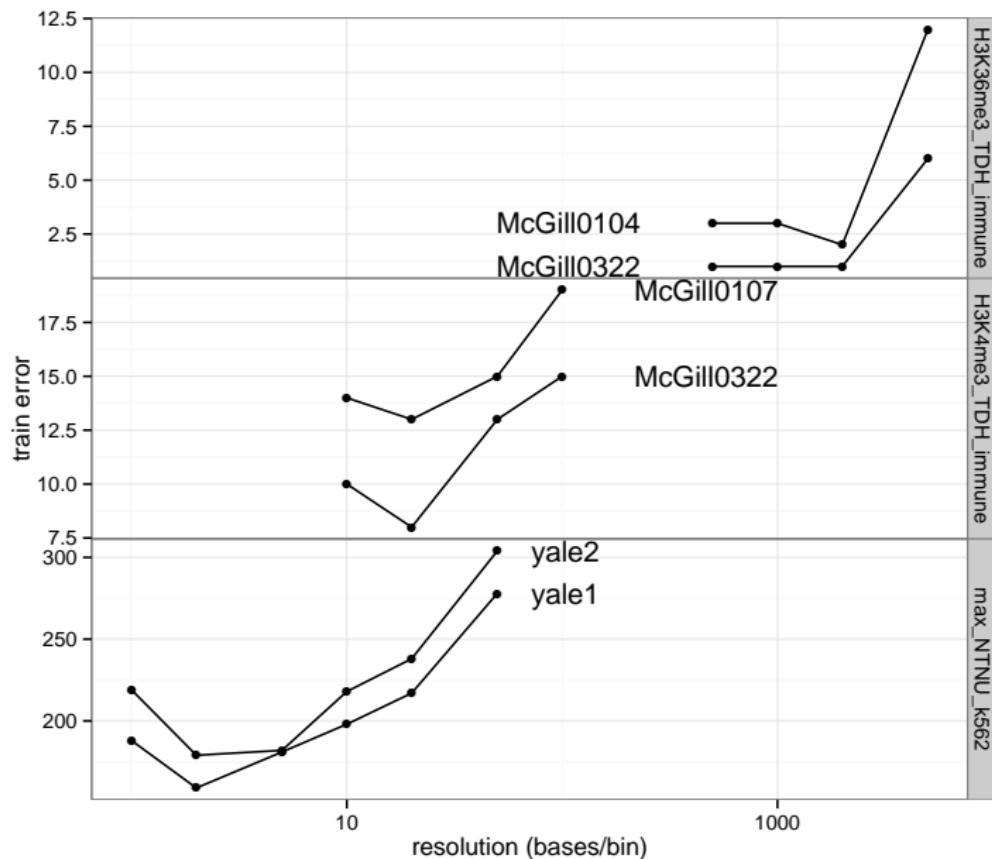
<http://cbio.ensmp.fr/~thocking/table-TF-benchmark/>

- ▶ 1 annotator.
- ▶ 3 TF types: max, nrSF, srf.
- ▶ For each TF: 1 cell type, 2 replicates.

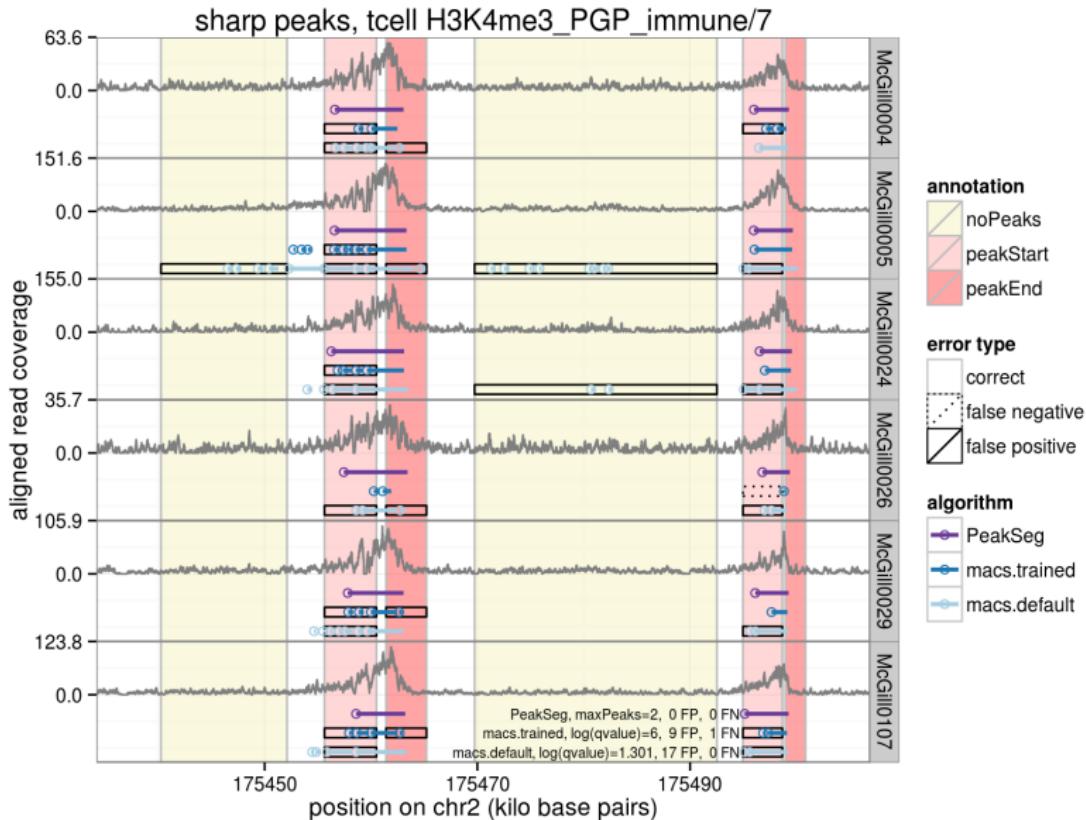
# Training time proportional to number of segmentation problems



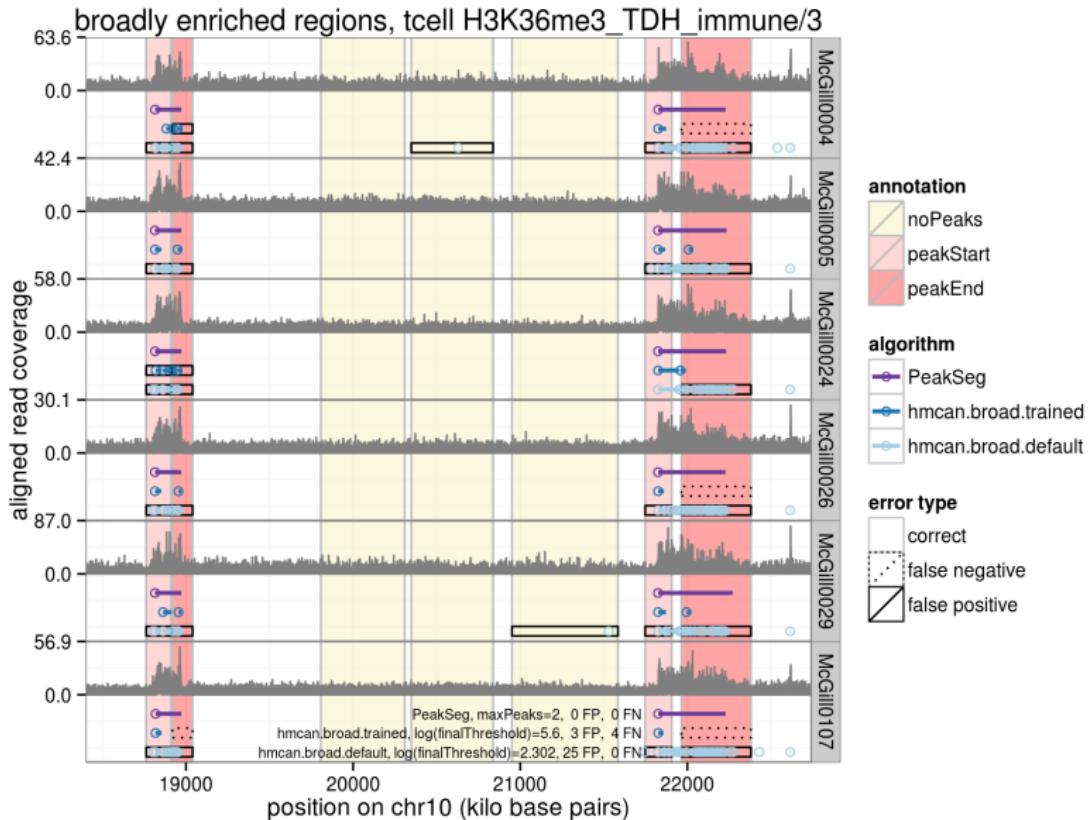
## Best resolution depends on data type



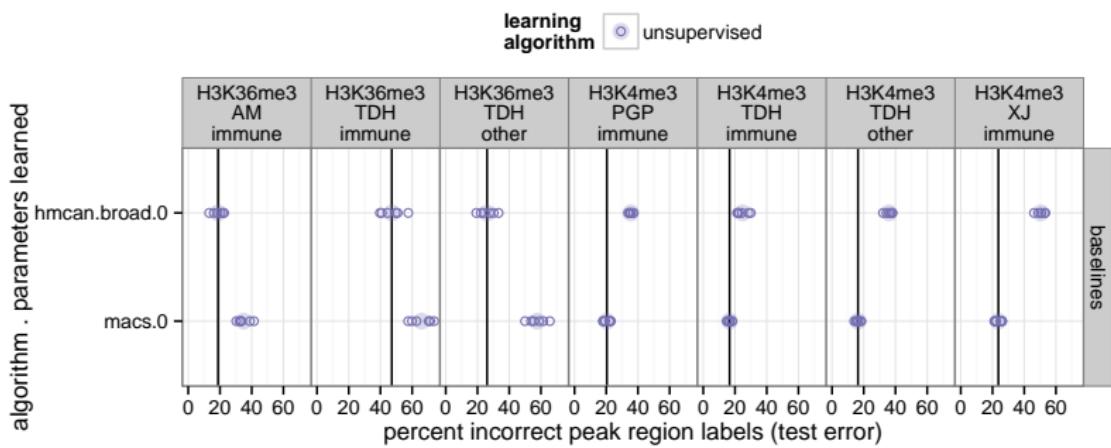
# Best results on H3K4me3 data



# Best results on H3K36me3 data



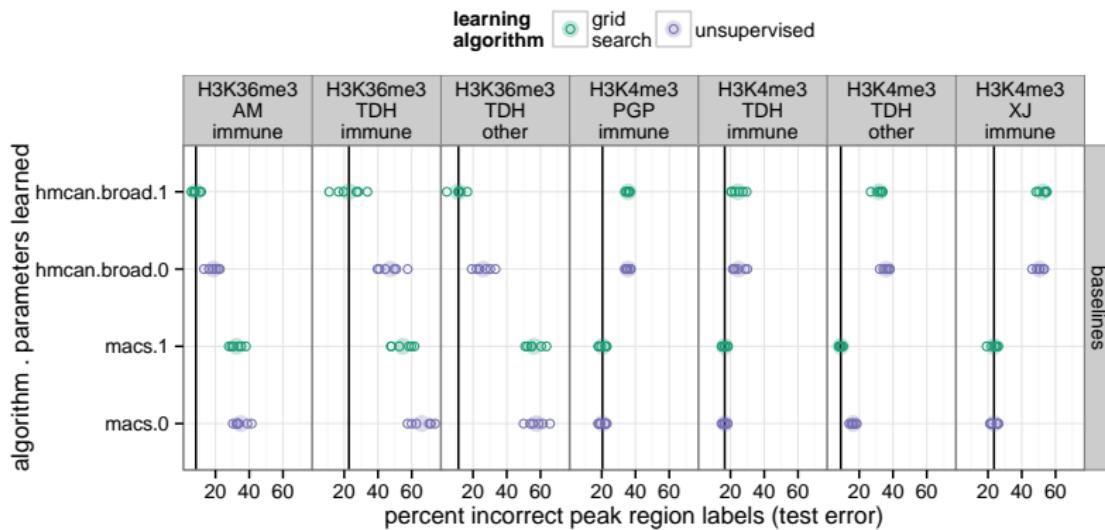
# hmcan.broad better for H3K36me3, macs better for H3K4me3



Six train/test splits (open circles) and mean (shaded circle).

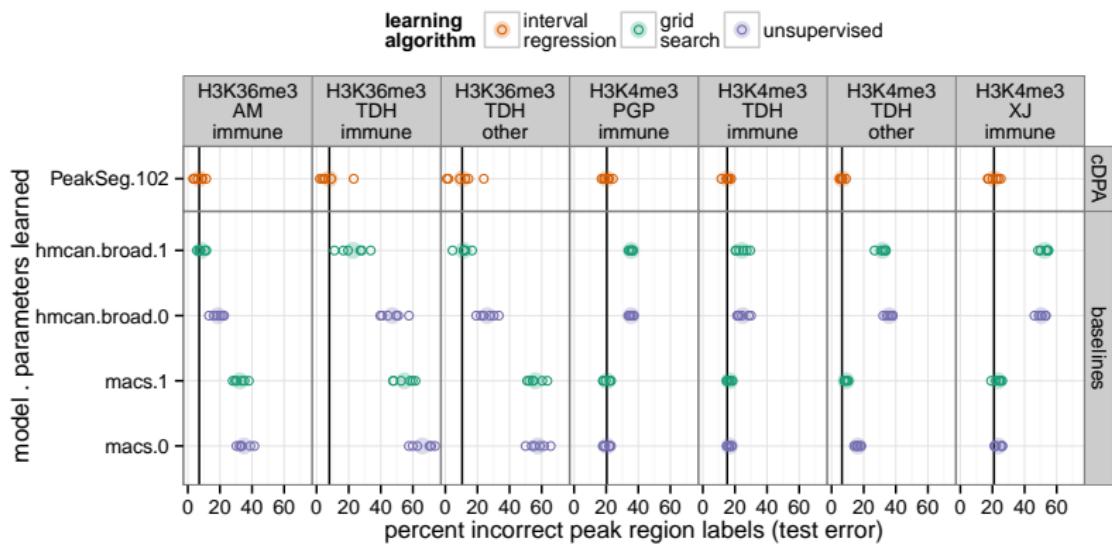
# Training 1 parameter with grid search reduces test error

...except for macs, good defaults for 3/4 H3K4me3 data sets.



Six train/test splits (open circles) and mean (shaded circle).

# Supervised genome-wide PeakSeg method works for both data types



Six train/test splits (open circles) and mean (shaded circle).

## Conclusions and future work

PeakSeg: **Peak** detection via constrained optimal **Segmentation**.

- ▶ First peak detector with efficient multi-parameter supervised learning algorithm.
- ▶ State-of-the-art peak detection for both H3K4me3 and H3K36me3 profiles.

Future work:

- ▶ Detecting the same peaks across several profiles?
- ▶ Faster algorithm via constrained version of

Time	Models	Algorithm	Author
$O(s_{\max} d \log d)$	$s_{\max}$	pDPA	Rigaill
$O(d \log d)$	1	FPOP	Maistone et al.

$d$  = data points to segment,  $s_{\max}$  = max segments.

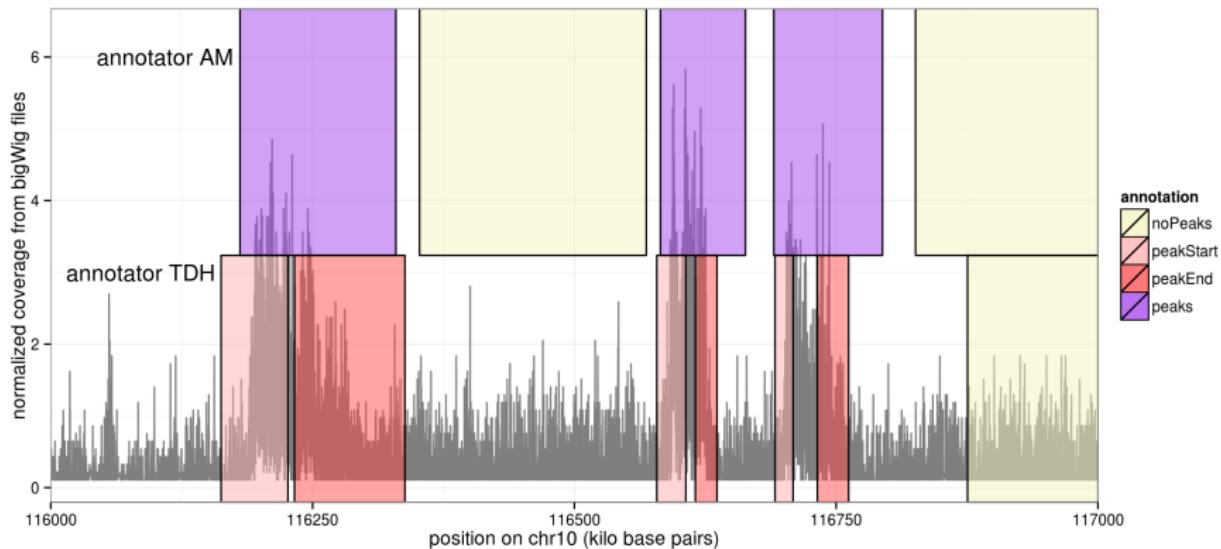
# Thanks for your attention!

Write me at [toby.hocking@mail.mcgill.ca](mailto:toby.hocking@mail.mcgill.ca) to collaborate!

Source code for slides, figures, paper online!  
<https://github.com/tdhock/PeakSeg-paper>

Supplementary slides appear after this one.

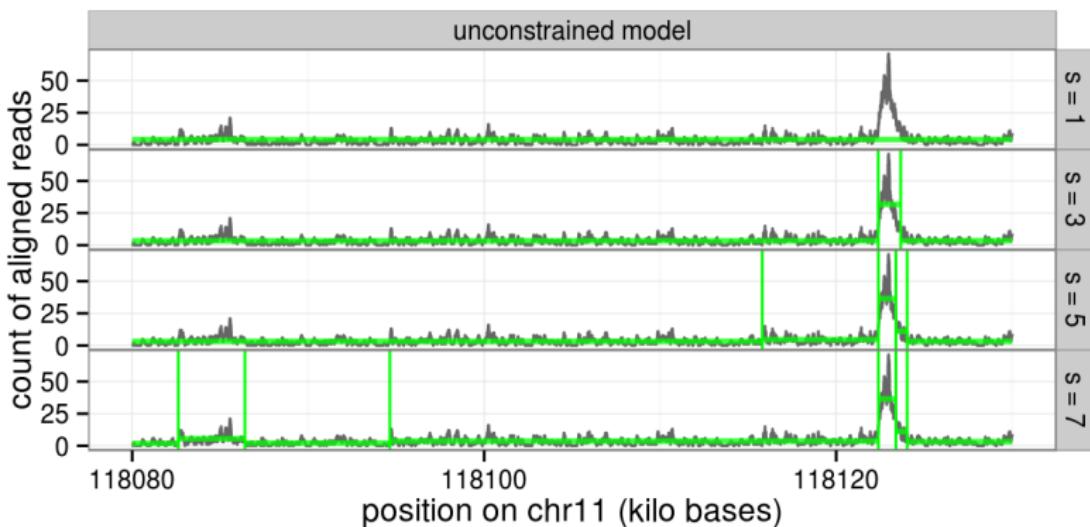
## Two annotators provide consistent labels, but different precision



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

## Maximum likelihood segmentations

For a coverage profile  $\mathbf{y} \in \mathbb{Z}_+^d$ , find the mean vector  $\hat{\mathbf{m}}^s(\mathbf{y}) \in \mathbb{R}^d$  with maximum Poisson likelihood, given  $s$  segments ( $s - 1$  change-points).



Computed via Segmentor3IsBack R package (Cleynen et al. 2014)

## Previous work: maximum likelihood segmentation

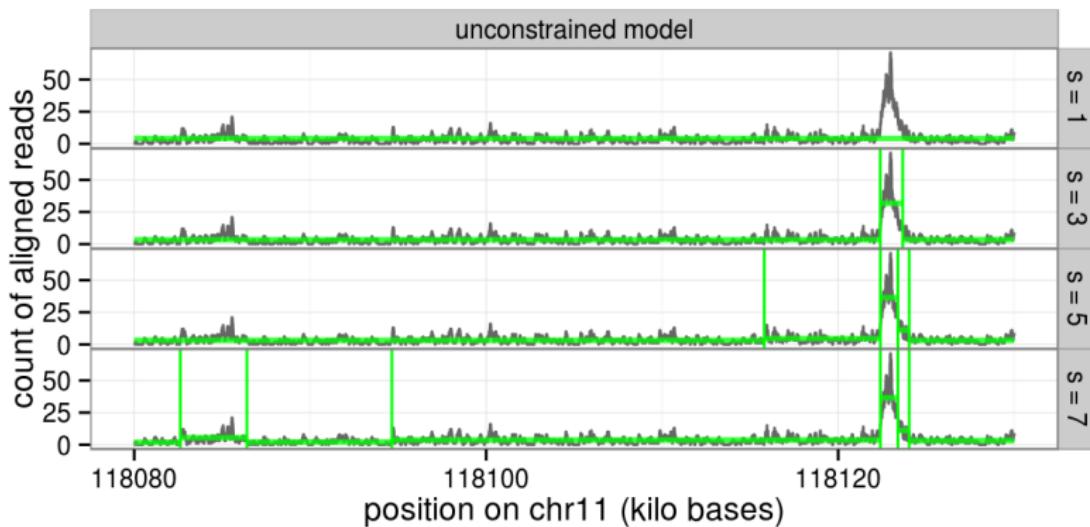
- ▶ Let  $\mathbf{y} = [ y_1 \ \cdots \ y_d ] \in \mathbb{Z}_+^d$  be the aligned read counts for one sample and one genomic region.
- ▶ Fix  $s_{\max} = 19$ , the maximum number of segments.
- ▶ For each number of segments  $s \in \{1, \dots, s_{\max}\}$ , we want:

$$\hat{\mathbf{m}}^s(\mathbf{y}) = \arg \min_{\mathbf{m} \in \mathbb{R}^d} \sum_{j=1}^d m_j - y_j \log m_j \text{ (Poisson loss)}$$

such that  $\text{Segments}(\mathbf{m}) = s$ .

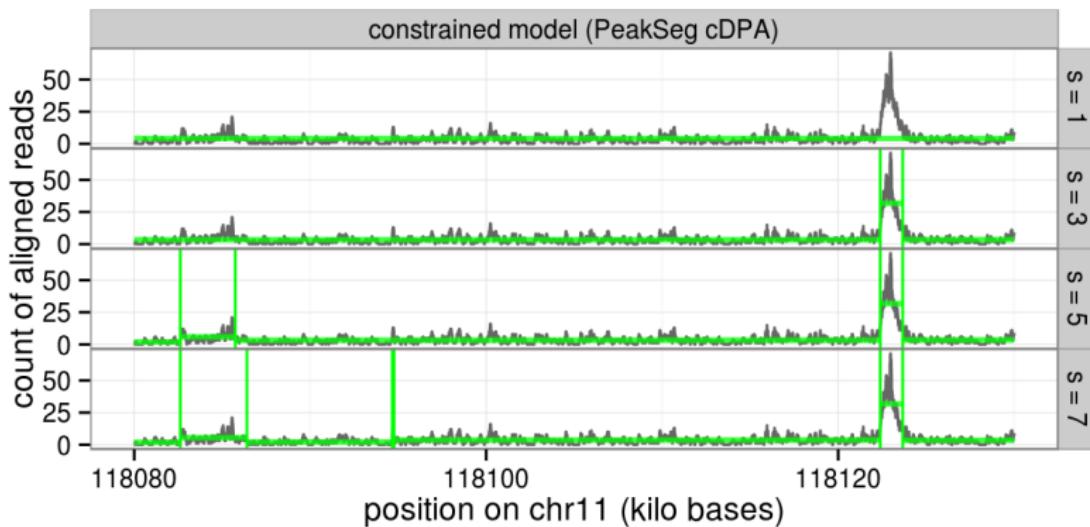
- ▶ Pruned Dynamic Programming Algorithm = pDPA (Rigaill arXiv:1004.0887) returns  $\hat{\mathbf{m}}^1(\mathbf{y}), \dots, \hat{\mathbf{m}}^{s_{\max}}(\mathbf{y})$  in  $O(s_{\max} d \log d)$  time.

# Maximum likelihood segmentations



Model with  $s = 5$  segments changes up, up, down, down.  
How to define peaks? Introduce a threshold parameter?

## Constrained maximum likelihood segmentations



Model with  $s = 5$  segments changes up, down, up, down.  
Peaks are even-numbered segments.

## PeakSeg: constrained maximum likelihood segmentation

For each number of segments  $s \in \{1, \dots, s_{\max}\}$ , we want:

$$\tilde{\mathbf{m}}^s(\mathbf{y}) = \arg \min_{\mathbf{m} \in \mathbb{R}^d} \sum_{j=1}^d m_j - y_j \log m_j$$

such that  $\text{Segments}(\mathbf{m}) = s$ ,

$$\forall j \in \{1, \dots, d\}, \quad P_j(\mathbf{m}) \in \{0, 1\}.$$

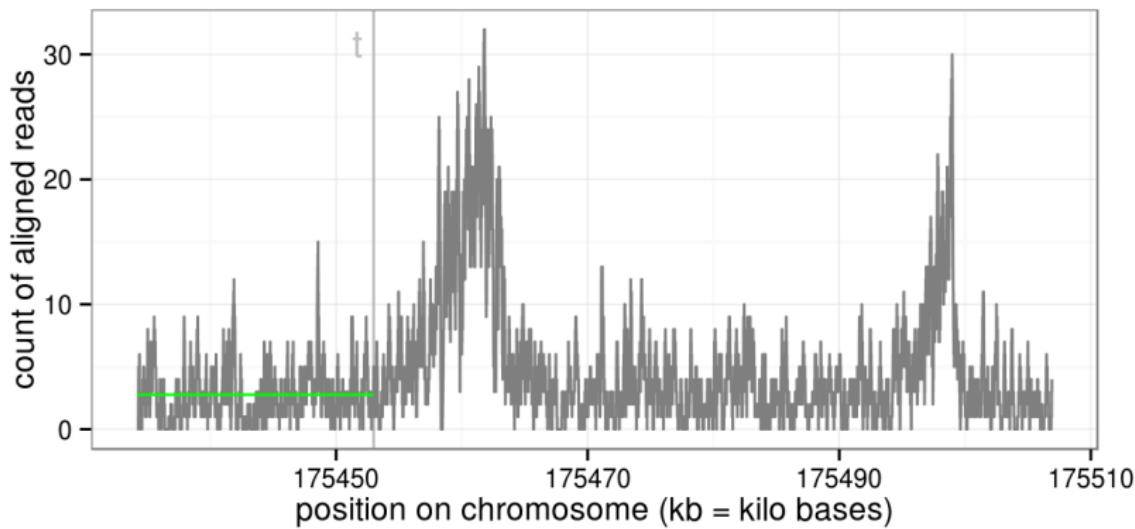
up, down, up, down constraint.

where the peak indicator  $P_1(\mathbf{m}) = 0$  and for  $j > 1$ ,

$$P_j(\mathbf{m}) = \sum_{k=2}^j \text{sign}(m_k - m_{k-1}).$$

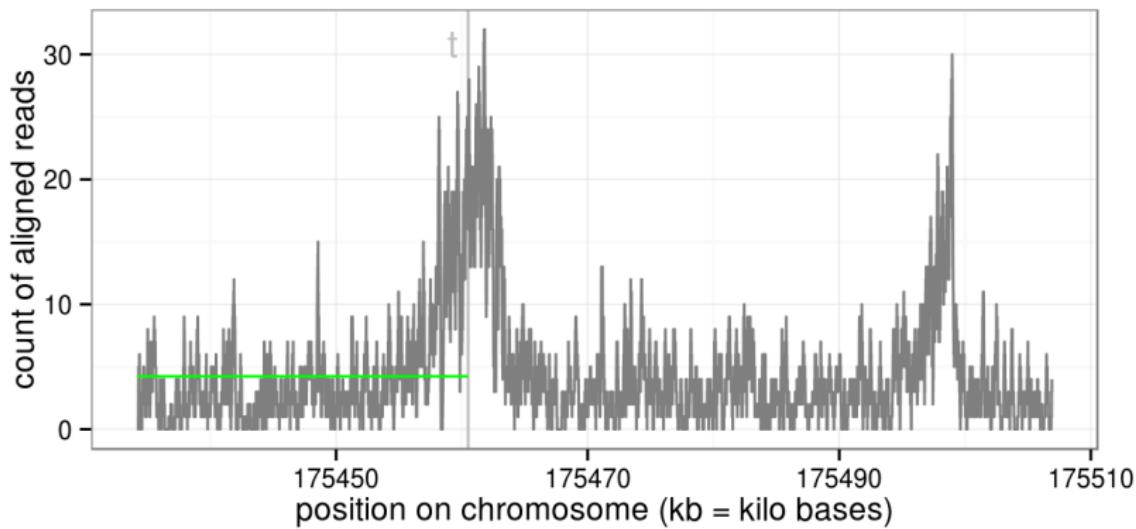
We propose cDPA = a constrained dynamic programming algorithm, which computes  $s_{\max}$  models in  $O(s_{\max}d^2)$  time.

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



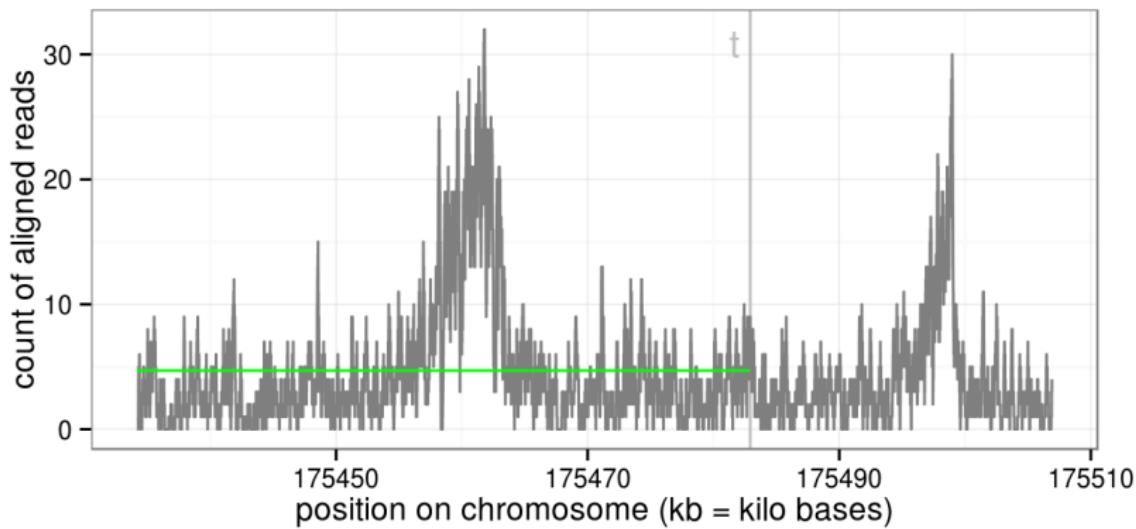
$$\mathcal{L}_{1,t} = \underbrace{c_{(0,t]}}_{\text{optimal loss of 1st segment } (0,t]}$$

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



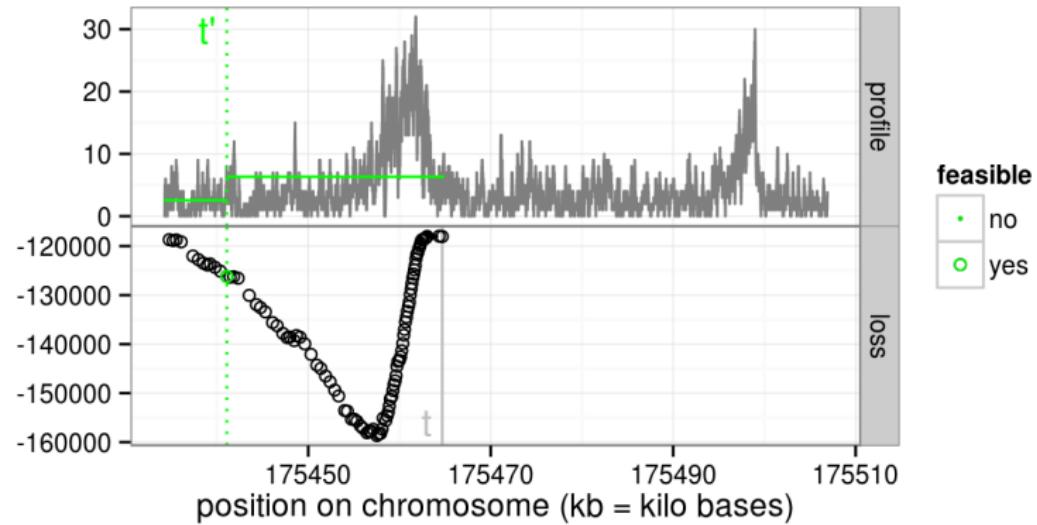
$$\mathcal{L}_{1,t} = \underbrace{c_{(0,t]}}_{\text{optimal loss of 1st segment } (0, t]}$$

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



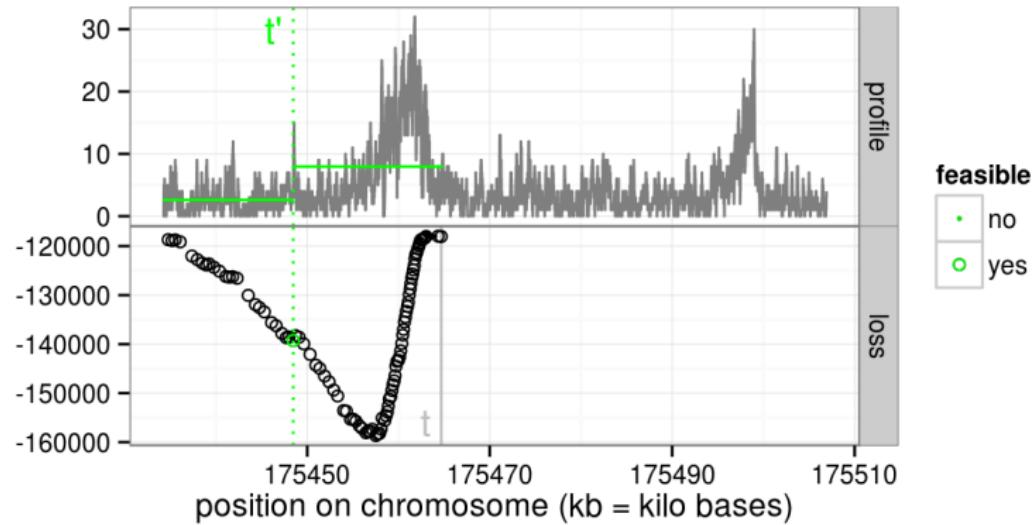
$$\mathcal{L}_{1,t} = \underbrace{c_{(0,t]}}_{\text{optimal loss of 1st segment } (0,t]}$$

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



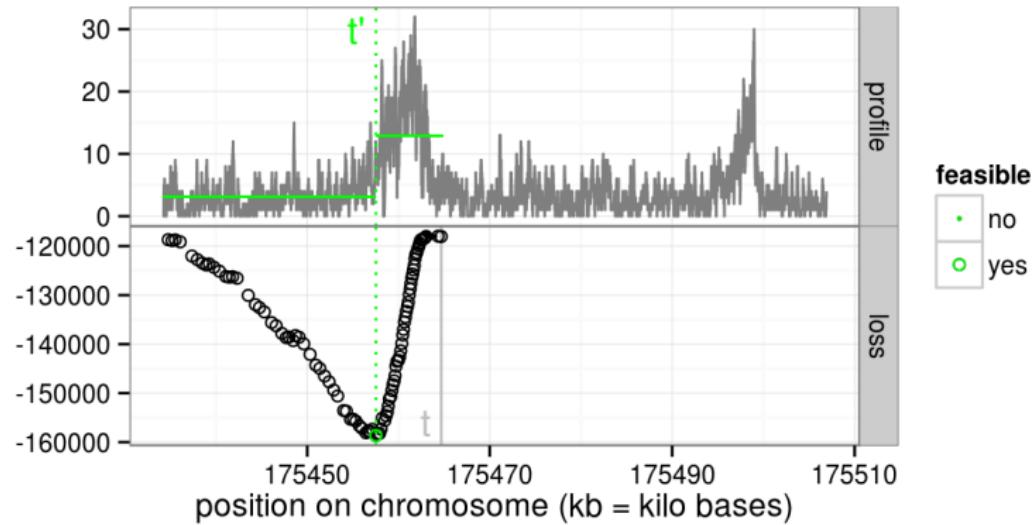
$$\mathcal{L}_{2,t} = \min_{t' < t} \underbrace{\mathcal{L}_{1,t'}}_{\text{optimal loss in 1 segment up to } t'} + \underbrace{c(t',t]}_{\text{optimal loss of 2nd segment } (t', t]}$$

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



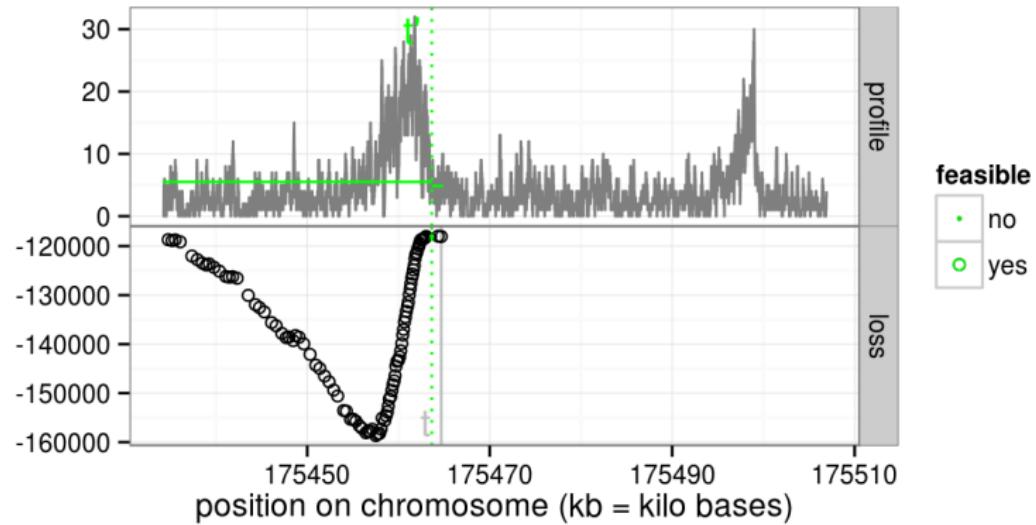
$$\mathcal{L}_{2,t} = \min_{t' < t} \underbrace{\mathcal{L}_{1,t'}}_{\text{optimal loss in 1 segment up to } t'} + \underbrace{c(t',t]}_{\text{optimal loss of 2nd segment } (t', t]}$$

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



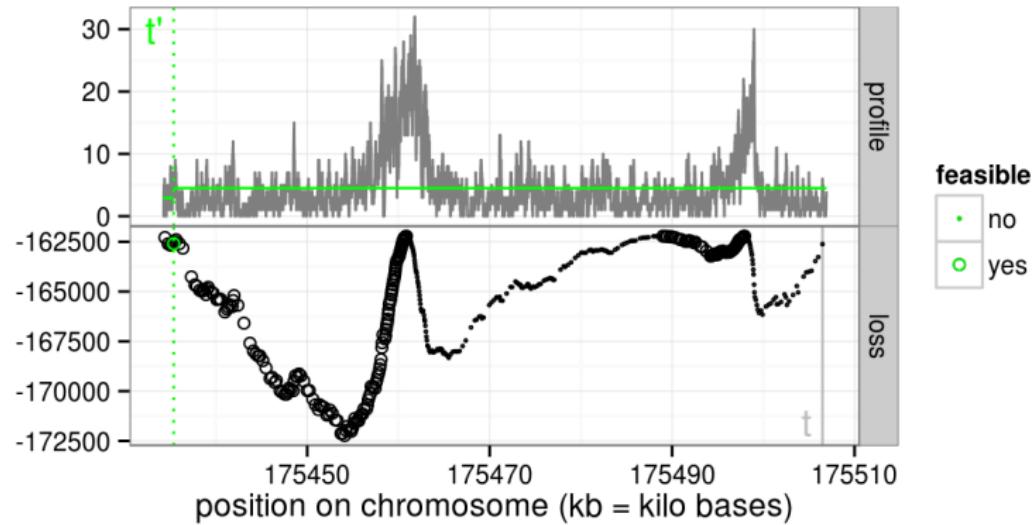
$$\mathcal{L}_{2,t} = \min_{t' < t} \underbrace{\mathcal{L}_{1,t'}}_{\text{optimal loss in 1 segment up to } t'} + \underbrace{c(t', t]}_{\text{optimal loss of 2nd segment } (t', t]}$$

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



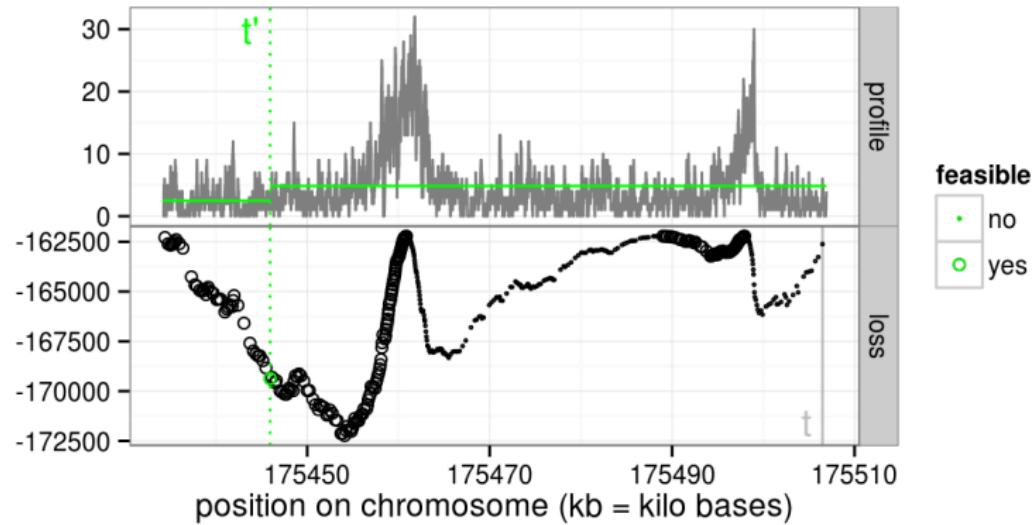
$$\mathcal{L}_{2,t} = \min_{t' < t} \underbrace{\mathcal{L}_{1,t'}}_{\text{optimal loss in 1 segment up to } t'} + \underbrace{c(t',t]}_{\text{optimal loss of 2nd segment } (t', t]}$$

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



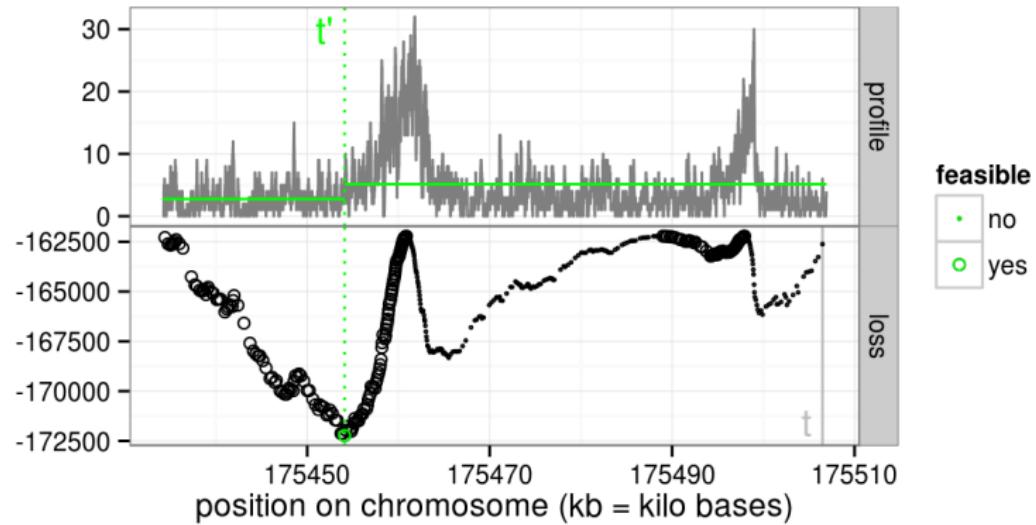
$$\mathcal{L}_{2,t} = \min_{t' < t} \underbrace{\mathcal{L}_{1,t'}}_{\text{optimal loss in 1 segment up to } t'} + \underbrace{c(t',t]}_{\text{optimal loss of 2nd segment } (t', t]}$$

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



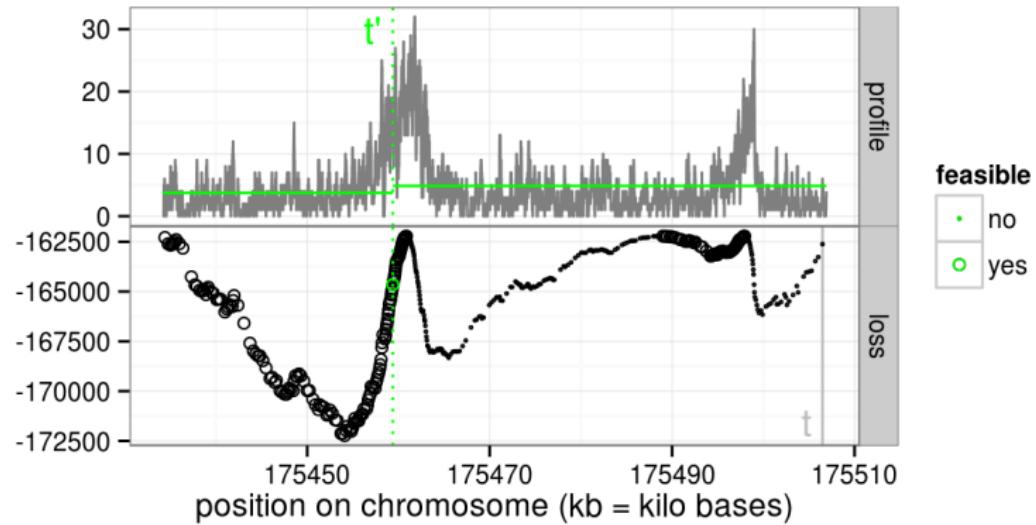
$$\mathcal{L}_{2,t} = \min_{t' < t} \underbrace{\mathcal{L}_{1,t'}}_{\text{optimal loss in 1 segment up to } t'} + \underbrace{c(t',t]}_{\text{optimal loss of 2nd segment } (t', t]}$$

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



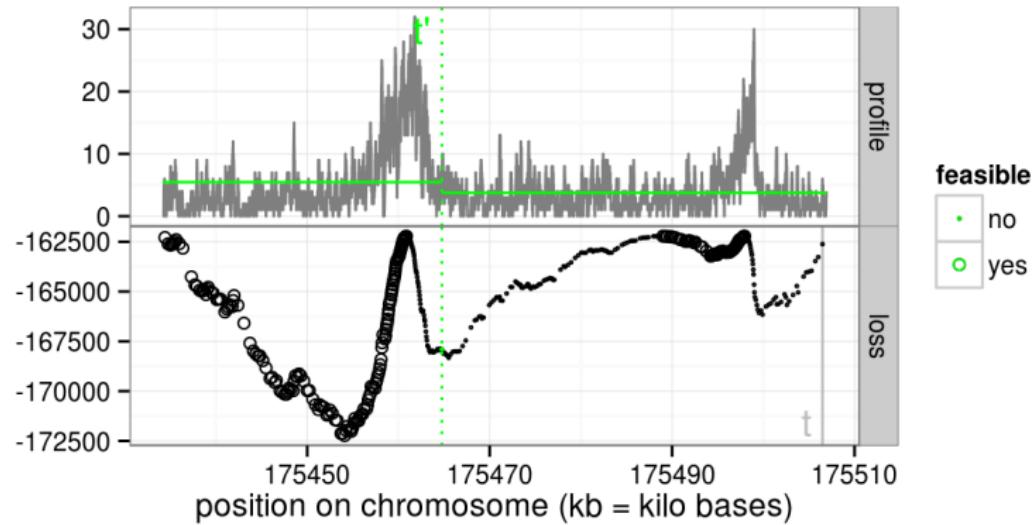
$$\mathcal{L}_{2,t} = \min_{t' < t} \underbrace{\mathcal{L}_{1,t'}}_{\text{optimal loss in 1 segment up to } t'} + \underbrace{c(t',t]}_{\text{optimal loss of 2nd segment } (t', t]}$$

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



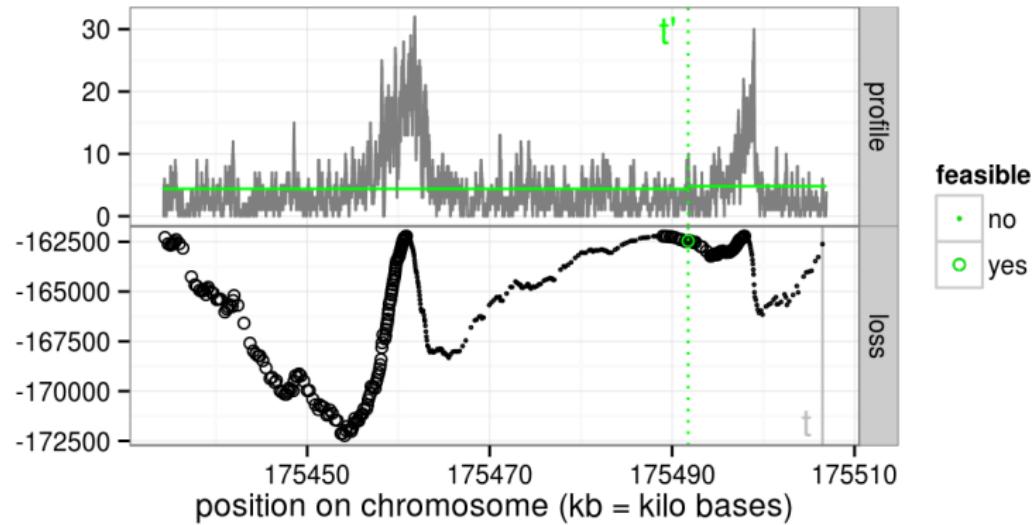
$$\mathcal{L}_{2,t} = \min_{t' < t} \underbrace{\mathcal{L}_{1,t'}}_{\text{optimal loss in 1 segment up to } t'} + \underbrace{c(t',t]}_{\text{optimal loss of 2nd segment } (t', t]}$$

# 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  $d$ :

segments $s$	grid search	dynamic programming
1	$O(d)$	$O(d)$
2	$O(d^2)$	$O(d^2)$
3	$O(d^3)$	$O(d^2)$
4	$O(d^4)$	$O(d^2)$
⋮	⋮	⋮

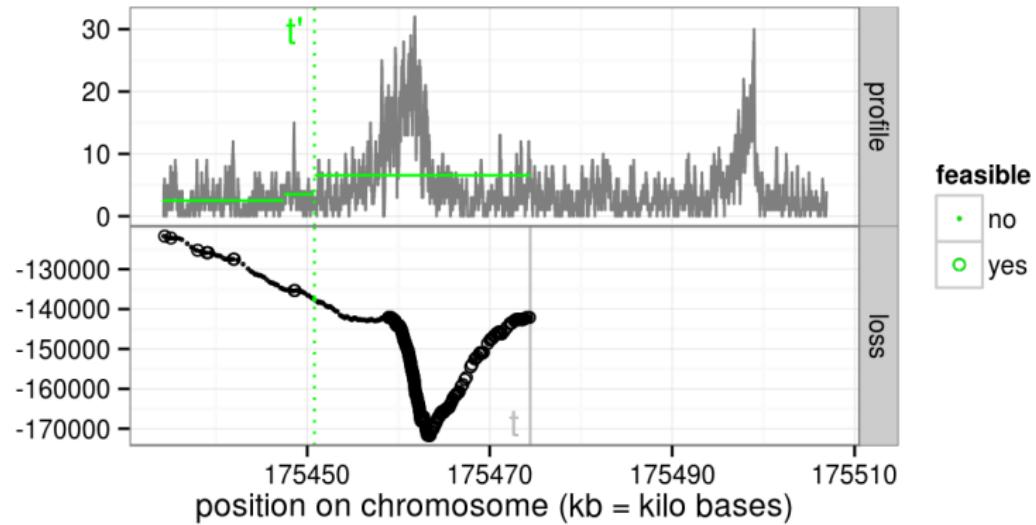
For example  $d = 5735$  data points to segment.

$$d^2 = 32890225$$

$$d^3 = 188625440375$$

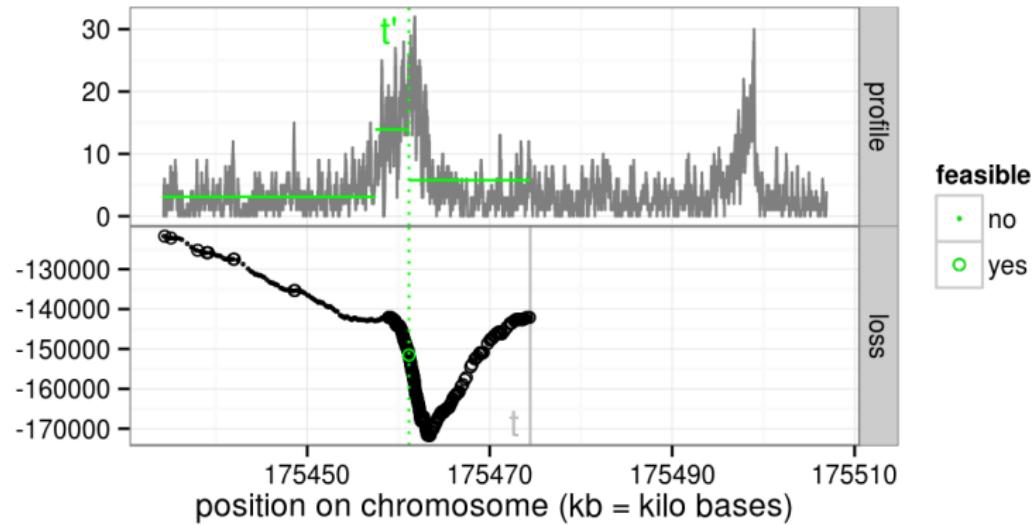
⋮

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



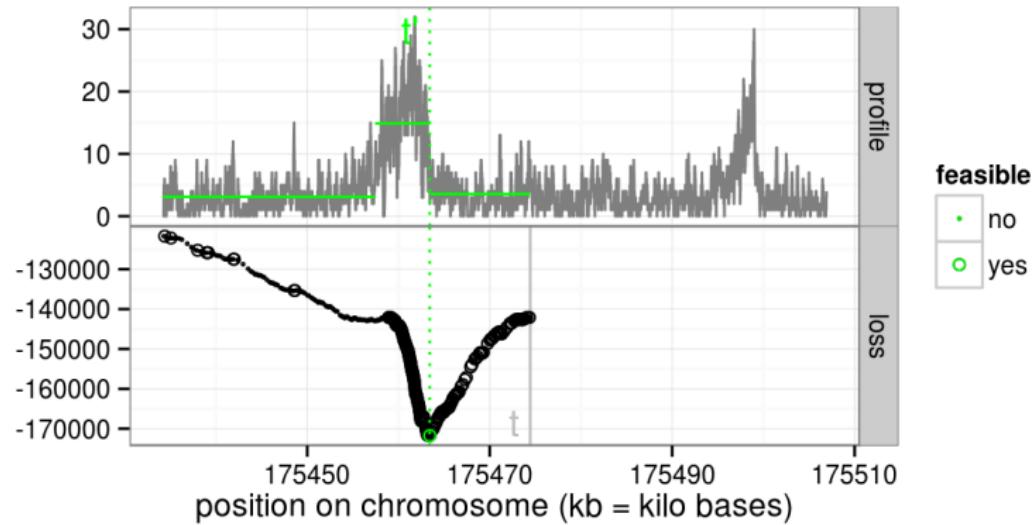
$$\mathcal{L}_{3,t} = \min_{t' < t} \underbrace{\mathcal{L}_{2,t'}}_{\text{optimal loss in 2 segments up to } t'} + \underbrace{c(t',t]}_{\text{optimal loss of 3rd segment } (t',t]}$$

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



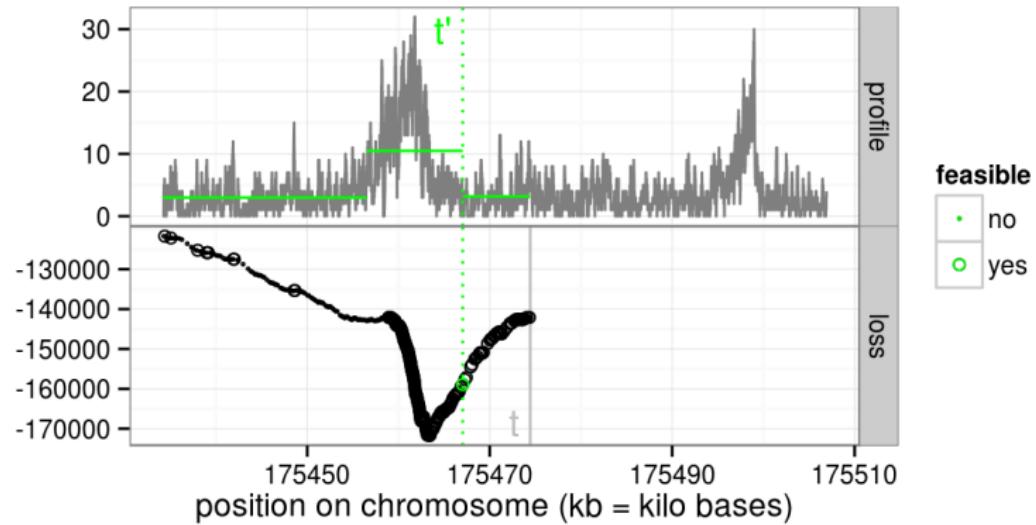
$$\mathcal{L}_{3,t} = \min_{t' < t} \underbrace{\mathcal{L}_{2,t'}}_{\text{optimal loss in 2 segments up to } t'} + \underbrace{c(t',t]}_{\text{optimal loss of 3rd segment } (t',t]}$$

# 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 $t$



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

## Step 1: compute annotation error functions

- ▶ Inputs: for  $i \in \{1, \dots, n\}$  samples, genomic profiles  $\mathbf{y}_i$ , annotated regions  $R_i$ .

	0 peaks	...	$p_{\max}$ peaks
segmentations	$\tilde{\mathbf{m}}^0(\mathbf{y}_i)$	...	$\tilde{\mathbf{m}}^{p_{\max}}(\mathbf{y}_i)$
annotation error	$e_i(0)$	...	$e_i(p_{\max})$

- ▶ R package <https://github.com/tdhock/PeakError/> computes the **annotation error**  $e_i : \{0, \dots, p_{\max}\} \rightarrow \mathbb{Z}_+$ .
- ▶ TD Hocking *et al.* Visual annotations and a supervised learning approach for evaluating and calibrating ChIP-seq peak detectors (arXiv:1409.6209).

## Step 2: compute model selection functions

For each sample/chromosome  $i \in \{1, \dots, n\}$ , for  $\lambda \in \mathbb{R}_+$ ,

- ▶ The **optimal number of peaks** function is

$$p_i^*(\lambda) = \arg \min_{p \in \{1, \dots, p_{\max}\}} \alpha_i^p + \lambda p,$$

where  $\alpha_i^p$  is the Poisson loss of the model with  $p$  peaks.

- ▶ The **penalized annotation error** function is

$$E_i(\lambda) = e_i [p_i^*(\lambda)],$$

where  $e_i(p)$  is the number of incorrect annotations for the model with  $p$  peaks.

**Peaks  $p_i^*$  and error  $E_i$  are non-convex, piecewise constant functions that can be computed exactly.**

## Step 3: learn a penalty function via interval regression

- ▶ Compute the target interval  $(\underline{L}_i, \bar{L}_i)$ .
- ▶  $\log \lambda_i \in (\underline{L}_i, \bar{L}_i) \Rightarrow$  optimal peak detection.
- ▶ Compute simple features  $\mathbf{x}_i \in \mathbb{R}^m$ , e.g. chromosome size, read counts, signal scale  $\log \max \mathbf{y}_i$ .
- ▶ Learn an optimal affine  $f(\mathbf{x}_i) = \beta + \mathbf{w}^\top \mathbf{x}_i = \log \lambda_i$ .
- ▶ Equivalent to learning a penalty  $\lambda_i = \exp f(\mathbf{x}_i)$ :

$$\begin{aligned} p_i^*[\exp f(\mathbf{x}_i)] &= \arg \min_p \alpha_i^p + p \exp f(\mathbf{x}_i) \\ &= \arg \min_p \alpha_i^p + p(\max \mathbf{y}_i)^w e^\beta. \end{aligned}$$

- ▶ Convex optimization problem, global optimum, variable selection (G Rigaill, TD Hocking, et al. ICML 2013).

## Summary of supervised PeakSeg algorithm

- ▶ Fix the maximum number of peaks  $p_{\max} = 10,000$ .
- ▶ For each sample/chromosome  $i \in \{1, \dots, n\}$ ,
  - ▶ **Unsupervised PeakSeg:** compute constrained maximum likelihood segmentations  $\tilde{\mathbf{m}}^0(\mathbf{y}_i), \dots, \tilde{\mathbf{m}}^{p_{\max}}(\mathbf{y}_i)$ .
  - ▶ Step 1: use annotated region labels to compute the annotation error  $e_i(0), \dots, e_i(p_{\max})$ .
  - ▶ Step 2: compute peaks  $p_i^*(\lambda)$ , error  $E_i(\lambda)$ , and target interval  $(\underline{L}_i, \bar{L}_i)$ .
- ▶ Step 3: learn a penalty  $\lambda_i = \exp f(\mathbf{x}_i)$  using features  $\mathbf{x}_i$  such as  $\log \max(\mathbf{y}_i)$ .
- ▶ Given an unlabeled chromosome  $(\mathbf{x}, \mathbf{y})$ , we predict  $\tilde{\mathbf{m}}^{p^*[\exp f(\mathbf{x})]}(\mathbf{y})$ .

## Penalty functions learned in this paper

Predicted number of segments for profile  $i \in \{1, \dots, n\}$ :

$$\hat{s}_i = \arg \min_{s \in \{1, 3, \dots, s_{\max} = 19\}} \overbrace{\rho[\tilde{\mathbf{m}}^s(\mathbf{y}_i), \mathbf{y}_i]}^{\text{Poisson loss}} + \overbrace{h(s, d_i)}^{\text{penalty}} \lambda_i,$$

Names: (model complexity).(number of parameters learned):

name	model complexity $h(s, d_i)$
AIC/BIC.*	$s$
oracle.*	$s \left(1 + 4\sqrt{1.1 + \log(d_i/s)}\right)^2$

name	smoothing $\lambda_i$	parameters	learning algorithm
*.0	AIC=2, BIC= $\log d_i$	none	unsupervised
*.1	$\beta$	$\beta \in \mathbb{R}_+$	grid search
*.3	$e^\beta d_i^{w_1} (\max \mathbf{y}_i)^{w_2}$	$\beta, w_1, w_2 \in \mathbb{R}$	interval regression
*.41	$\exp(\beta + \mathbf{w}^\top \mathbf{x}_i)$	$\beta \in \mathbb{R}, \mathbf{w} \in \mathbb{R}^{40}$	interval regression

( $d_i$  = number of data for profile  $i$ )

## Penalty functions learned in this paper

Predicted number of segments for profile  $i \in \{1, \dots, n\}$ :

$$\hat{s}_i = \arg \min_{s \in \{1, 3, \dots, s_{\max} = 19\}} \overbrace{\rho[\tilde{\mathbf{m}}^s(\mathbf{y}_i), \mathbf{y}_i]}^{\text{Poisson loss}} + \overbrace{h(s, d_i)}^{\text{penalty}} \lambda_i,$$

Names: (model complexity).(number of parameters learned):

name	model complexity $h(s, d_i)$
AIC/BIC.*	$s$
oracle.*	$s \left(1 + 4\sqrt{1.1 + \log(d_i/s)}\right)^2$

name	smoothing $\lambda_i$	parameters	learning algorithm
*.0	AIC=2, BIC=log $d_i$	none	unsupervised
*.1	$\beta$	$\beta \in \mathbb{R}_+$	grid search
*.3	$e^\beta d_i^{w_1} (\max \mathbf{y}_i)^{w_2}$	$\beta, w_1, w_2 \in \mathbb{R}$	interval regression
*.41	$\exp(\beta + \mathbf{w}^\top \mathbf{x}_i)$	$\beta \in \mathbb{R}, \mathbf{w} \in \mathbb{R}^{40}$	interval regression

( $d_i$  = number of data for profile  $i$ )