

PeakSeg: constrained optimal **Segmentation** and supervised penalty learning for **Peak** detection in count data

Toby Dylan Hocking

toby.hocking@mail.mcgill.ca

joint work with Guillem Rigaill and Guillaume Bourque

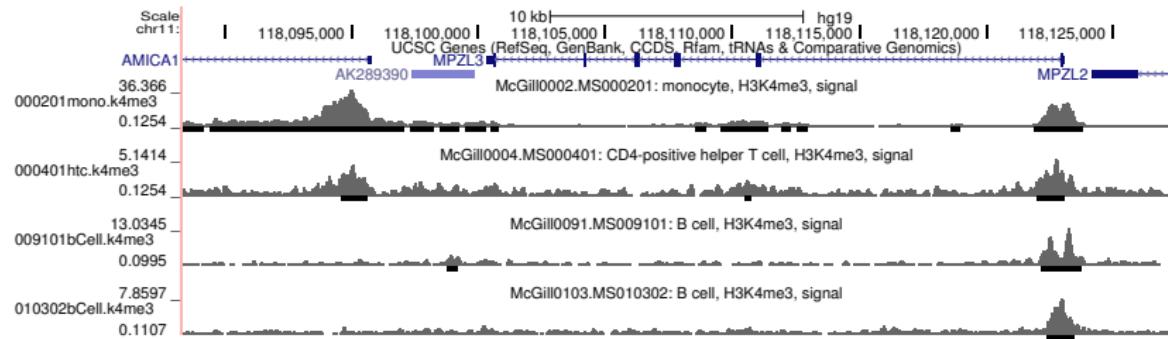
7 July 2015

ChIP-seq data and previous work on peak detection

New PeakSeg model: constrained optimal segmentation

Train and test error results, conclusions

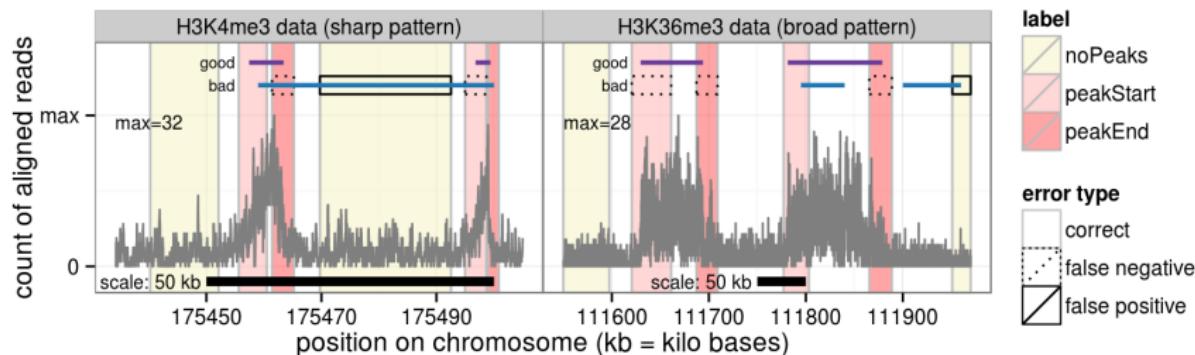
Problem: many false positives in unsupervised peak detectors



- ▶ Grey signal is noisy count data (\approx protein binding).
- ▶ Binary classification for every sample and position:
negative class = background noise,
positive class = “peaks.”
- ▶ Black bands show peak predictions of “Model-based analysis of ChIP-Seq” (MACS), Zhang et al, 2008 (default parameters).

Peak detector accuracy can be quantified using manually annotated region labels

Supervised learning framework (arXiv:1409.6209).



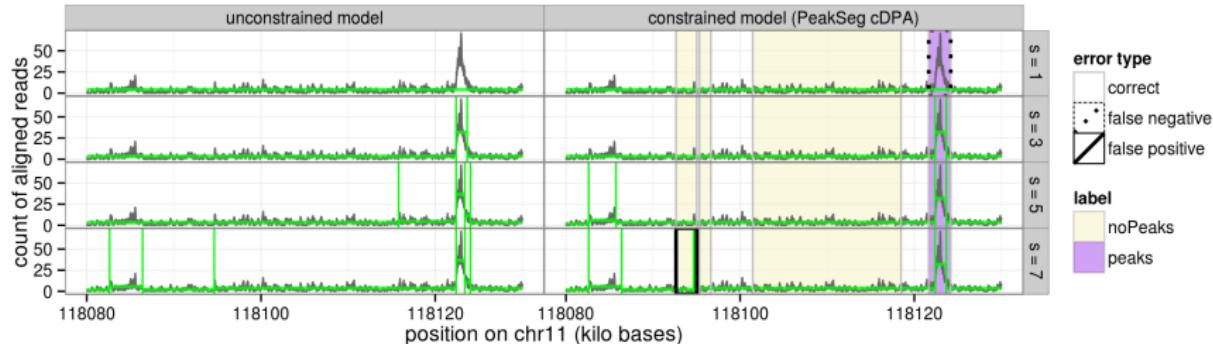
- ▶ Good peaks have 0 incorrect regions.
- ▶ Bad peaks have 7 incorrect regions.
- ▶ Goal: minimize number of incorrect test labels.

ChIP-seq data and previous work on peak detection

New PeakSeg model: constrained optimal segmentation

Train and test error results, conclusions

Maximum likelihood Poisson segmentation models



- ▶ Previous work: unconstrained maximum likelihood mean for s segments ($s - 1$ changes).
- ▶ This paper: constraint enforces up, down, up, down (and not up, up, down).
- ▶ Odd-numbered segments are background noise, even-numbered segments are peaks.

PeakSeg: constrained maximum likelihood segmentation

For each number of segments $s \in \{1, \dots, s_{\max}\}$,
the PeakSeg model for the mean vector \mathbf{m} is defined as:

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

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 $j > 1$,

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

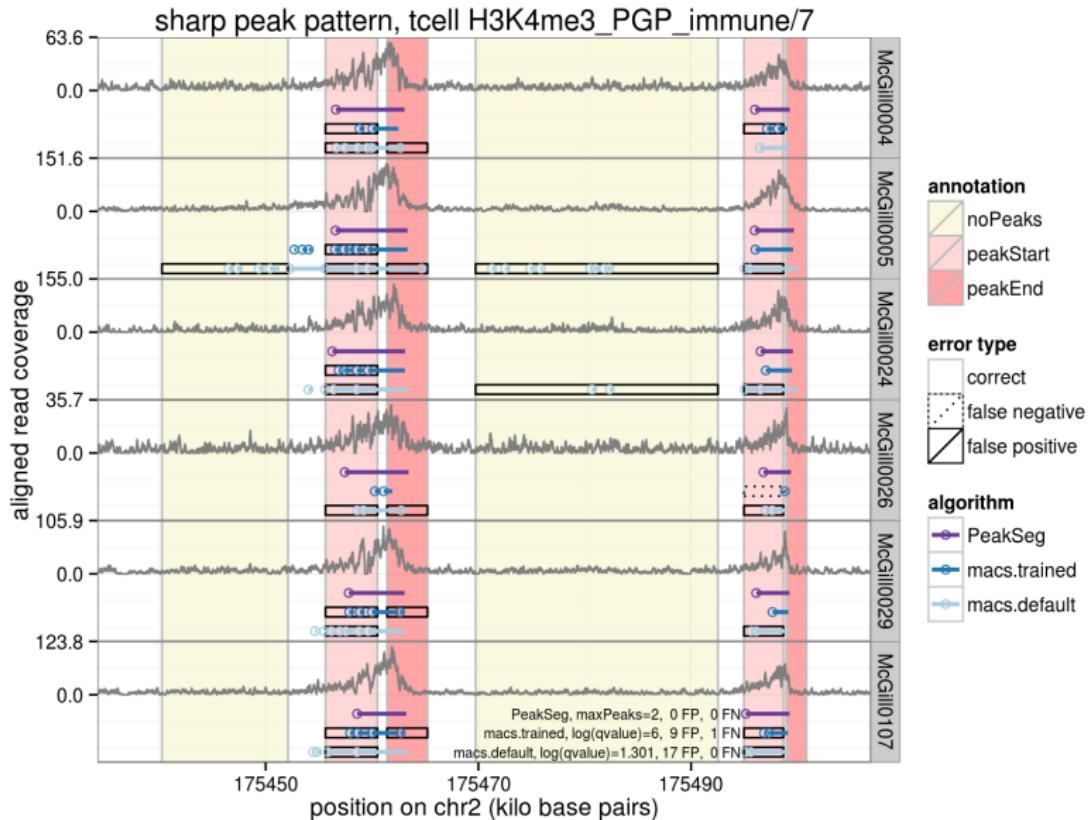
We propose cDPA = a constrained dynamic programming algorithm, which computes s_{\max} models in $O(s_{\max} d^2)$ time.

ChIP-seq data and previous work on peak detection

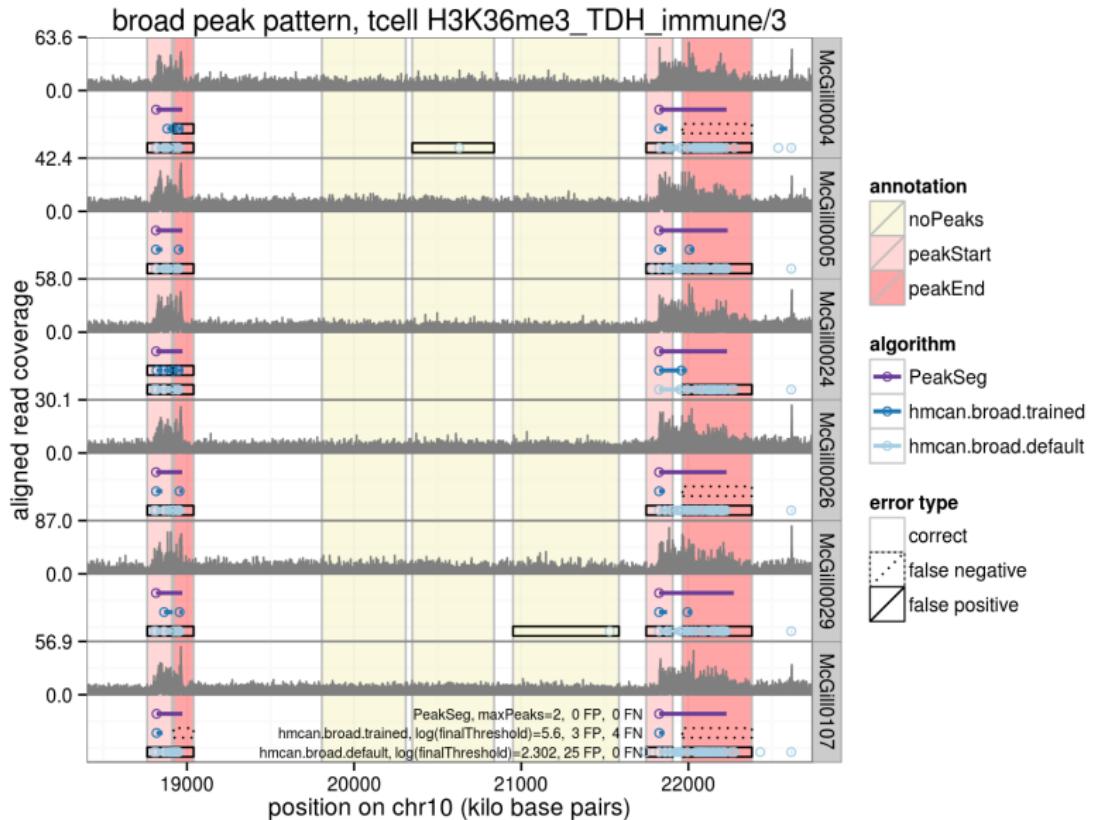
New PeakSeg model: constrained optimal segmentation

Train and test error results, conclusions

Train error on H3K4me3 data



Train error on H3K36me3 data



AIC/BIC and oracle penalty function learning

Supervised learning method of Hocking, Rigaill, et al (ICML 2013).

Oracle model complexity of Cleynen and Lebarbier (2014).

Predicted number of segments for each profile i :

$$\hat{s}_i = \arg \min_s \text{PoissonLoss} [\tilde{\mathbf{m}}^s(\mathbf{y}_i), \mathbf{y}_i] + \underbrace{h(p, d_i)}_{\text{given}} \underbrace{\lambda_i}_{\text{learned}},$$

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 | learned λ_i | parameters | learning algorithm |
|------|---|--|-----------------------|
| *.0 | AIC=2, BIC= $\log d_i$ | none | unsupervised |
| *.1 | β | $\beta \in \mathbb{R}_+$ | 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 | $\exp(\beta + \mathbf{w}^\top \mathbf{x}_i)$ | $\beta \in \mathbb{R}, \mathbf{w} \in \mathbb{R}^{40}$ | regularized int. reg. |

AIC/BIC and oracle penalty function learning

Supervised learning method of Hocking, Rigaill, et al (ICML 2013).

Oracle model complexity of Cleynen and Lebarbier (2014).

Predicted number of segments for each profile i :

$$\hat{s}_i = \arg \min_s \text{PoissonLoss} [\tilde{\mathbf{m}}^s(\mathbf{y}_i), \mathbf{y}_i] + \underbrace{h(p, d_i)}_{\substack{\text{given} \\ \text{learned}}} \lambda_i,$$

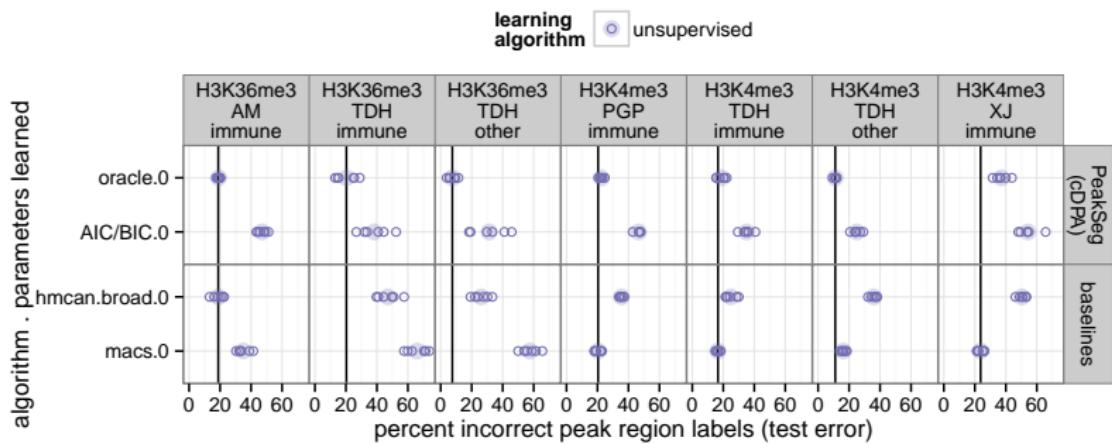
Names: (model complexity).(number of parameters learned):

| 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 | learned λ_i | parameters | learning algorithm |
|-------------|---|--|---------------------------|
| *.0 | $\text{AIC}=2$, $\text{BIC}=\log d_i$ | none | unsupervised |
| *.1 | β | $\beta \in \mathbb{R}_+$ | 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 | $\exp(\beta + \mathbf{w}^\top \mathbf{x}_i)$ | $\beta \in \mathbb{R}, \mathbf{w} \in \mathbb{R}^{40}$ | regularized int. reg. |

Unsupervised constrained optimization algorithm works for both H3K36me3 and H3K4me3 data types

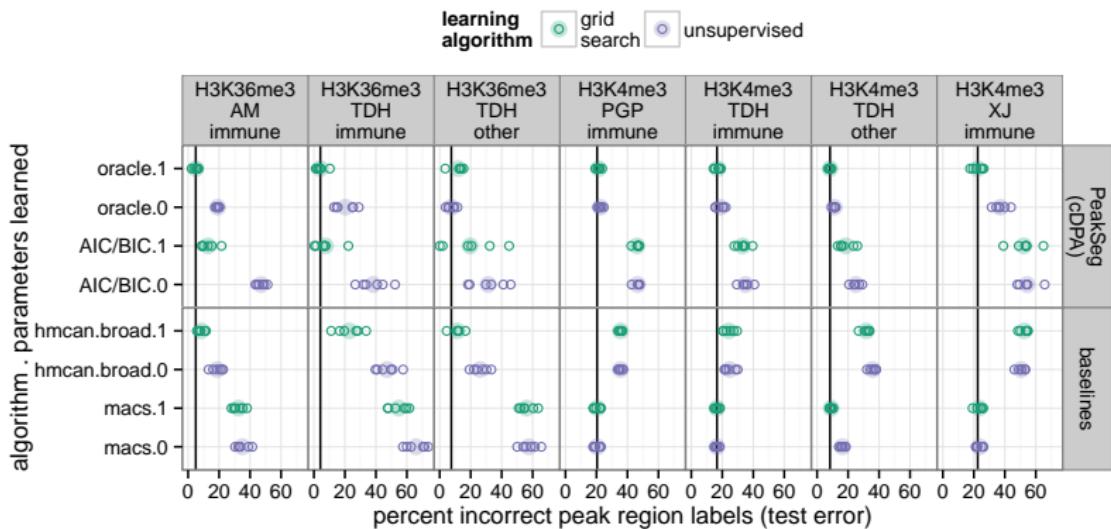
...except in the H3K4me3_XJ_immune data set.



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

Training 1 parameter with grid search reduces test error

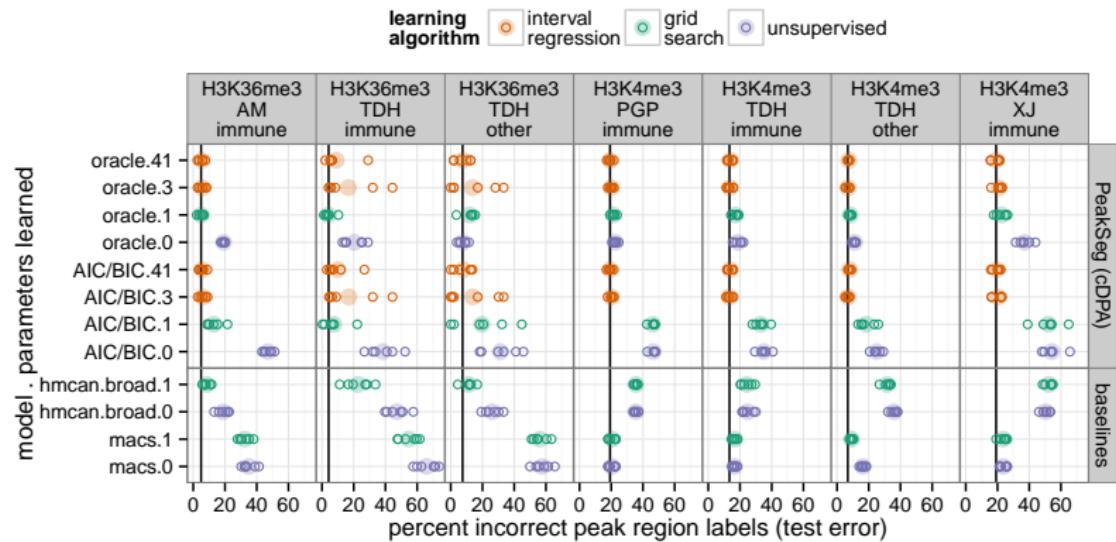
...except for macs, good defaults for 3/4 H3K4me3 data sets.



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

Training several parameters with interval regression further reduces test error

...except when there are few train data (H3K36me3_TDH).



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

Conclusions and future work

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

- ▶ New segmentation model with up, down, up, down constraint.
- ▶ First supervised peak detection algorithm.
- ▶ State-of-the-art peak detection for both H3K4me3 and H3K36me3 profiles.
- ▶ Oracle model complexity more accurate than AIC/BIC.

Future work:

- ▶ Constrained version of Pruned Dynamic Programming (Rigaill arXiv:1004.0887) to compute in $O(d \log d)$ time.
- ▶ Efficient algorithm which provably computes PeakSeg model?
- ▶ Theoretically optimal features for the penalty learning problem?
- ▶ Feature learning based on profile count data.
- ▶ Overlapping peaks at the same positions across samples (arXiv:1506.01286).

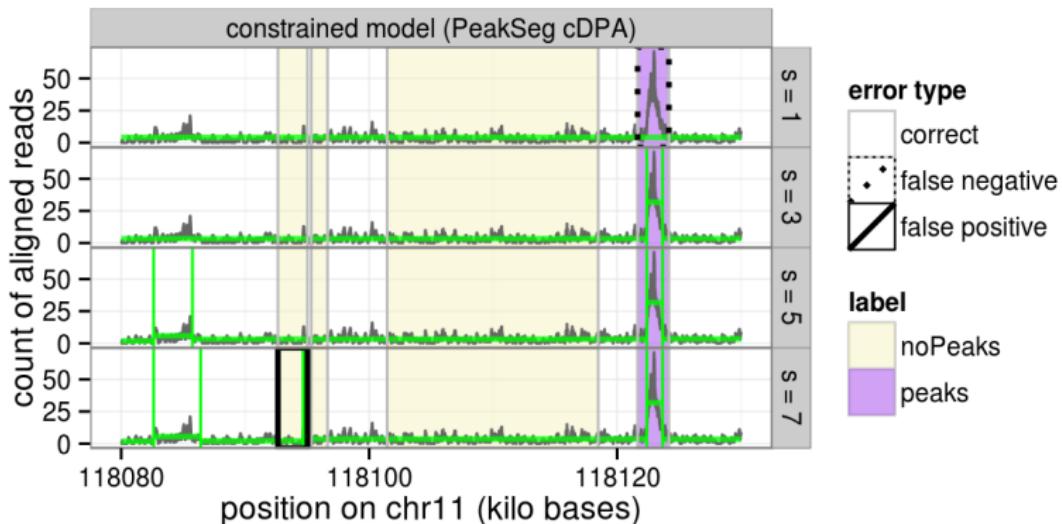
Thanks for your attention!

Write me at 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.

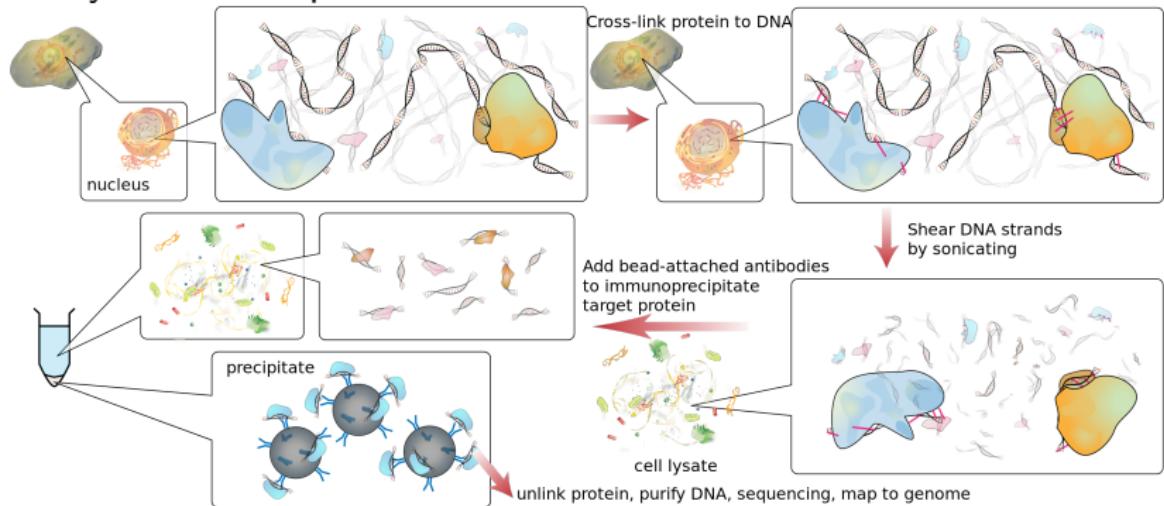
PeakSeg accuracy can be quantified using labels



- ▶ 1, 3, 5, 7 segments = 0, 1, 2, 3 peaks ($2p + 1 = s$).
- ▶ Models with $s \in \{1, 7\}$ segments have 1 incorrect region.
- ▶ Models with $s \in \{3, 5\}$ segments are perfect.
- ▶ Goal for $i \in \{1, \dots, n\}$ profiles:
predict profile-specific segments \hat{s}_i with minimum errors.

Chromatin immunoprecipitation sequencing (ChIP-seq)

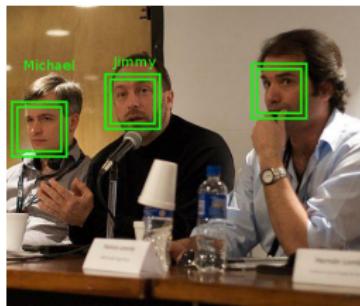
Analysis of DNA-protein interactions.



Source: “ChIP-sequencing,” Wikipedia.

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

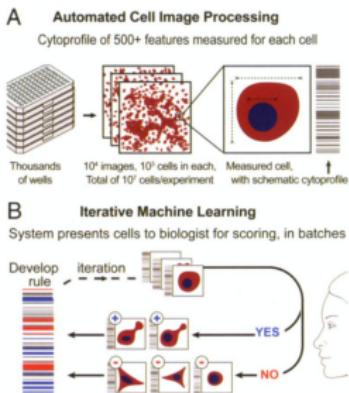
Photos



Labels: names

CVPR 2013
246 papers

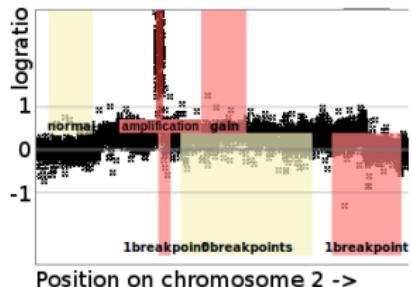
Cell images



phenotypes

CellProfiler
873 citations

Copy number profiles

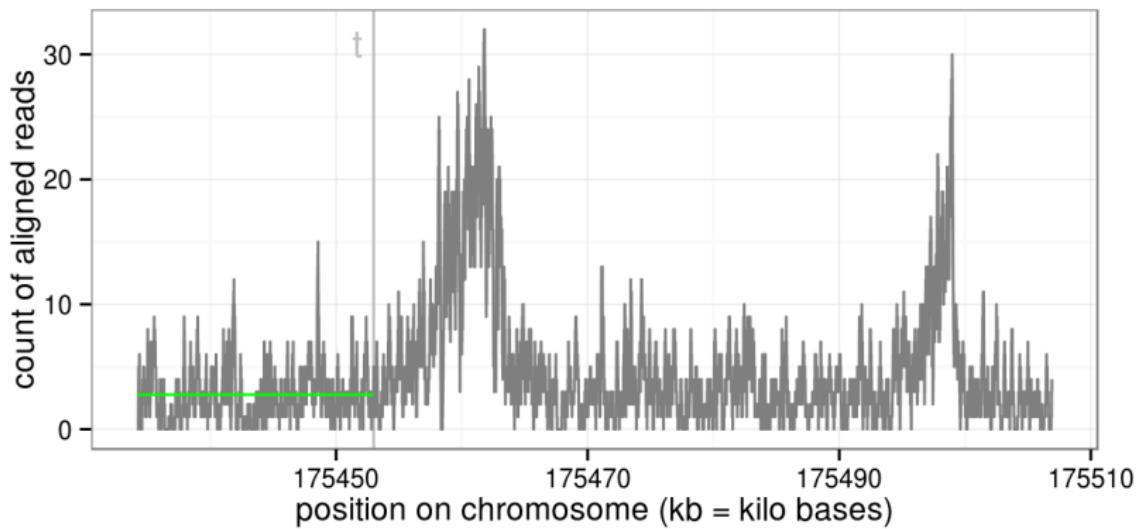


alterations

SegAnnDB
H, et. al. 2014.

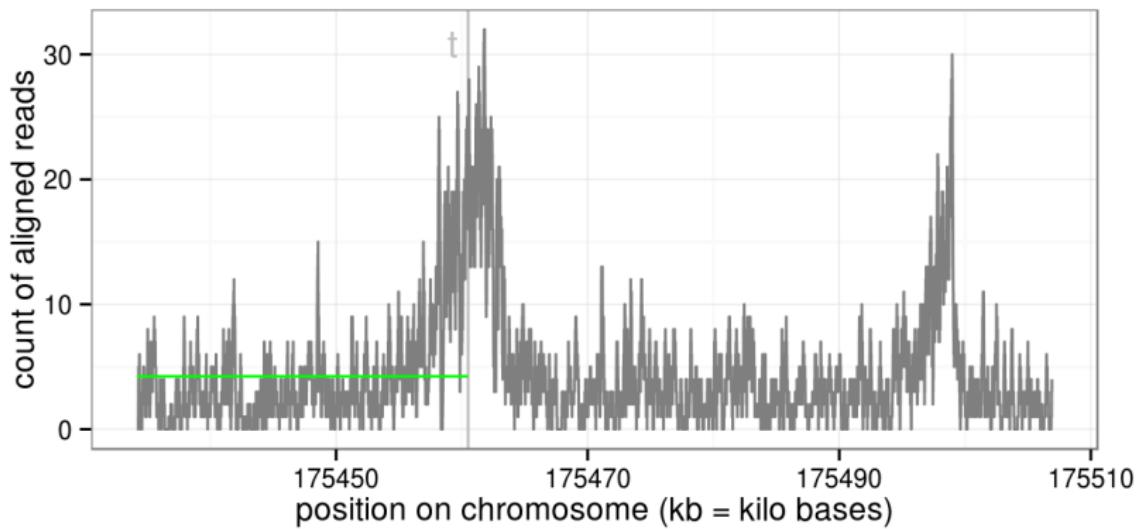
Sources: http://en.wikipedia.org/wiki/Face_detection
Jones et al PNAS 2009. Scoring diverse cellular morphologies in image-based screens with iterative feedback and machine learning.

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



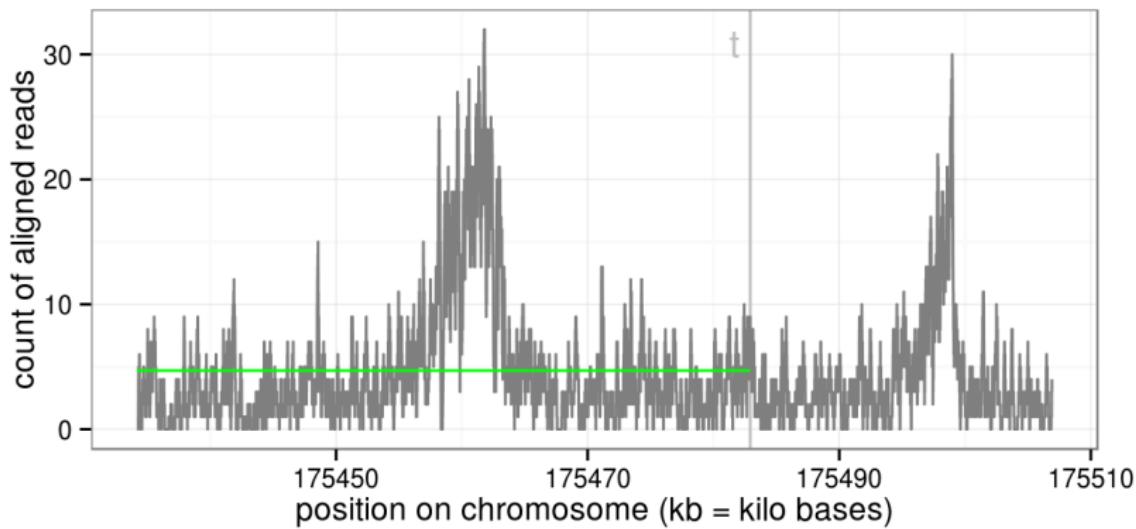
$$\mathcal{L}_{1,t} = \underbrace{c_{(0,t]}}_{\text{optimal loss of 1st segment } (0,t]}$$

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



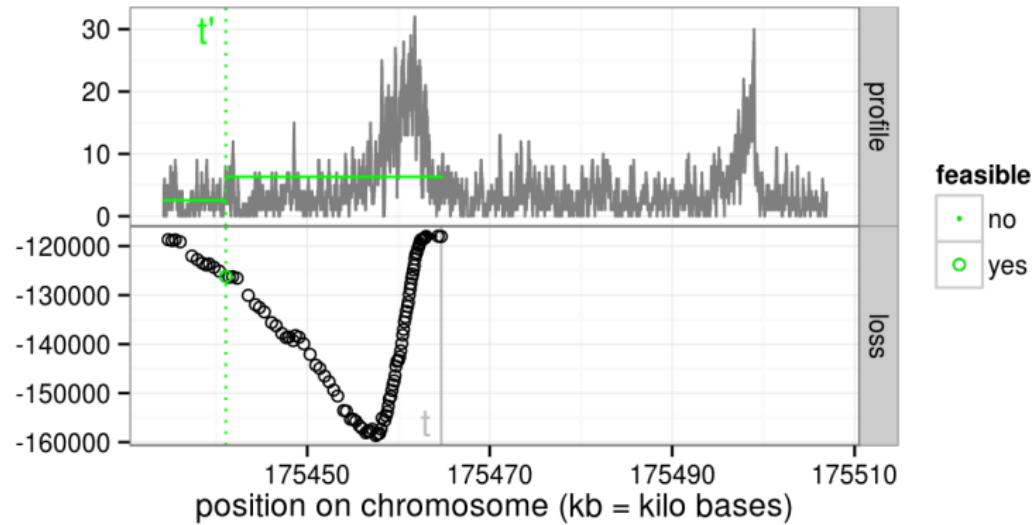
$$\mathcal{L}_{1,t} = \underbrace{c_{(0,t]}}_{\text{optimal loss of 1st segment } (0, t]}$$

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



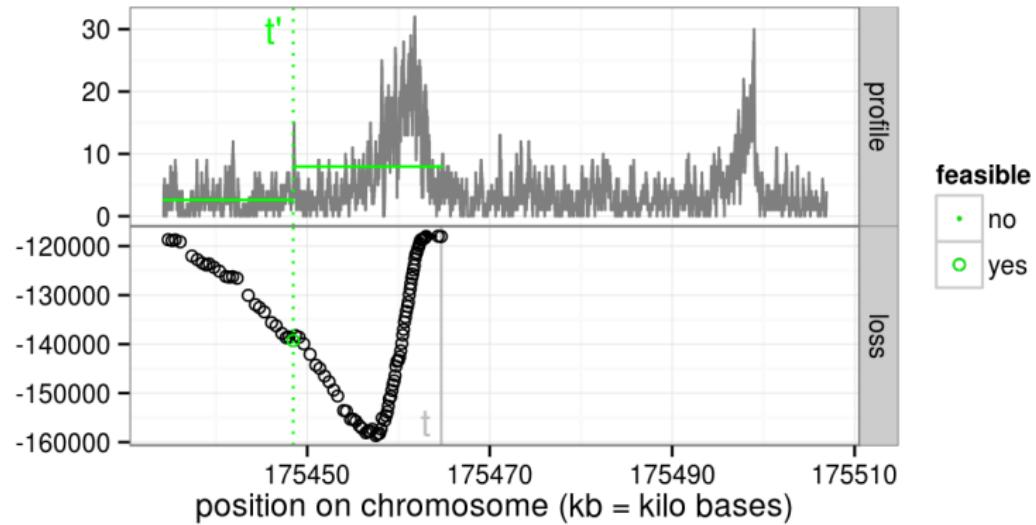
$$\mathcal{L}_{1,t} = \underbrace{c_{(0,t]}}_{\text{optimal loss of 1st segment } (0,t]}$$

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



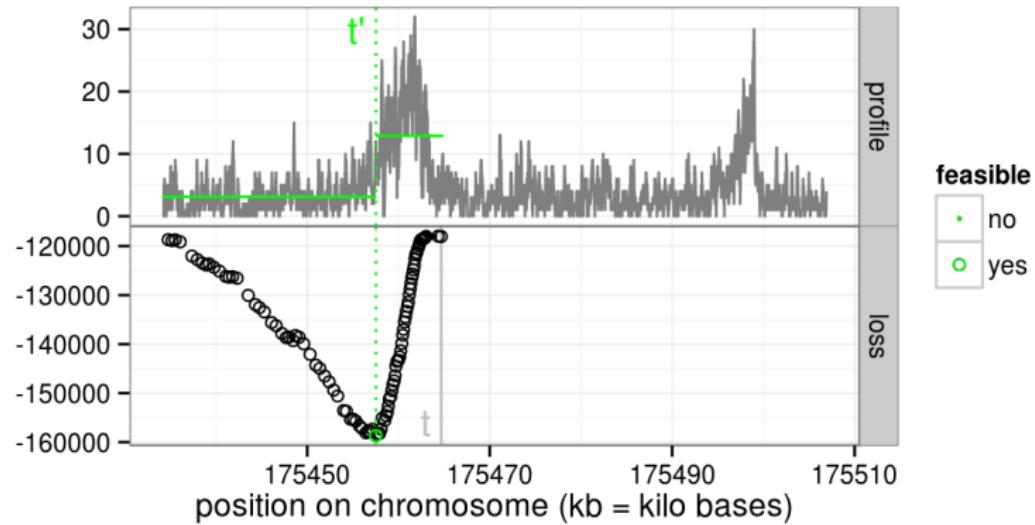
$$\mathcal{L}_{2,t} = \min_{t' < t} \underbrace{\mathcal{L}_{1,t'}}_{\text{optimal loss in 1 segment up to } t'} + \underbrace{c(t',t]}_{\text{optimal loss of 2nd segment } (t', t]}$$

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



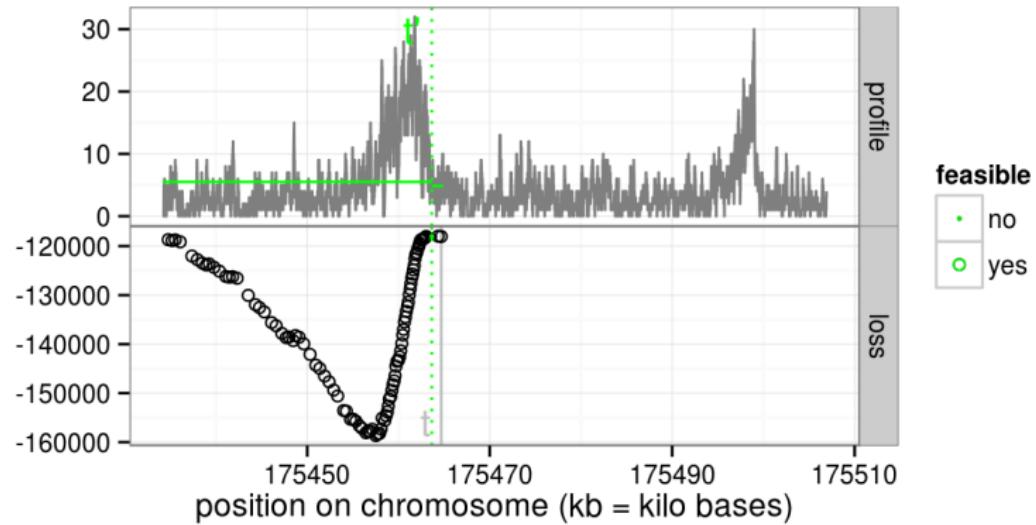
$$\mathcal{L}_{2,t} = \min_{t' < t} \underbrace{\mathcal{L}_{1,t'}}_{\text{optimal loss in 1 segment up to } t'} + \underbrace{c(t',t]}_{\text{optimal loss of 2nd segment } (t', t]}$$

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



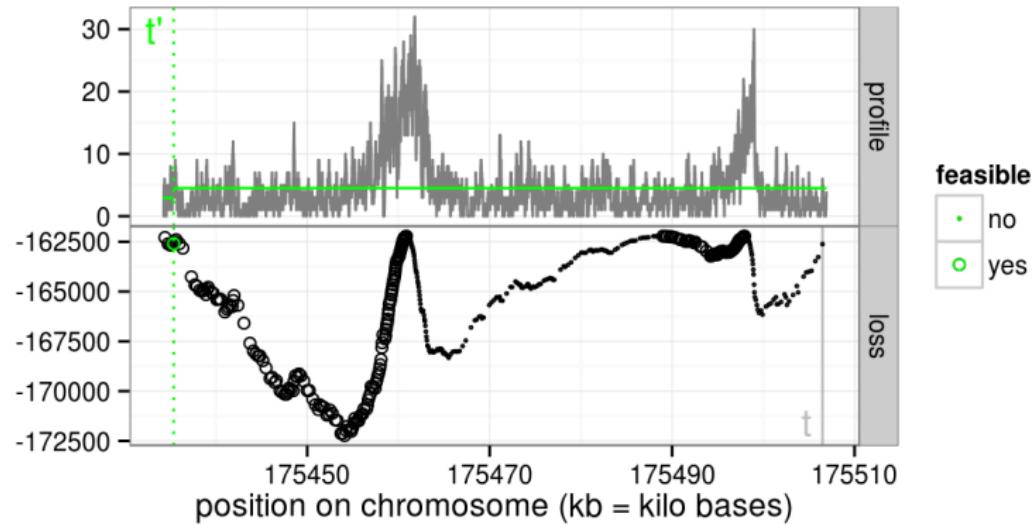
$$\mathcal{L}_{2,t} = \min_{t' < t} \underbrace{\mathcal{L}_{1,t'}}_{\text{optimal loss in 1 segment up to } t'} + \underbrace{c(t', t]}_{\text{optimal loss of 2nd segment } (t', t]}$$

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



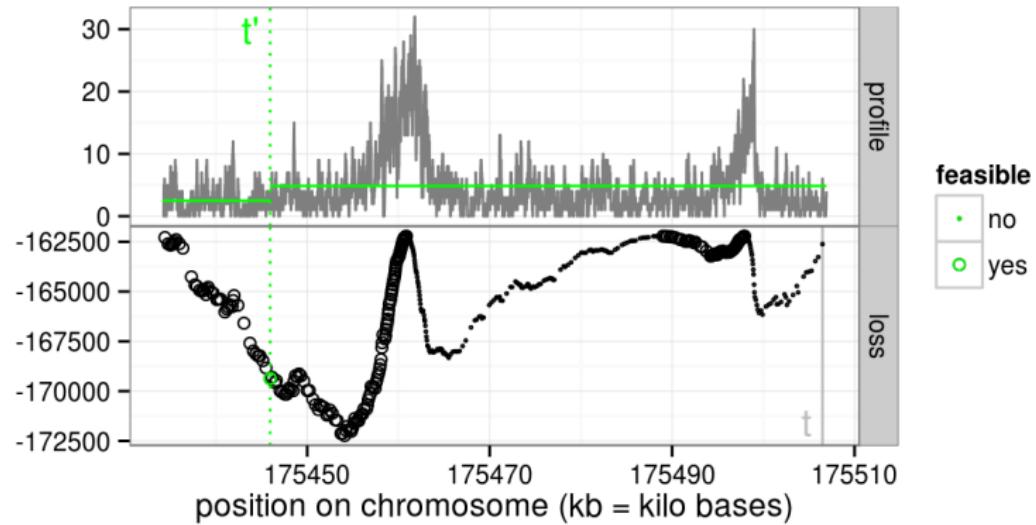
$$\mathcal{L}_{2,t} = \min_{t' < t} \underbrace{\mathcal{L}_{1,t'}}_{\text{optimal loss in 1 segment up to } t'} + \underbrace{c(t',t]}_{\text{optimal loss of 2nd segment } (t', t]}$$

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



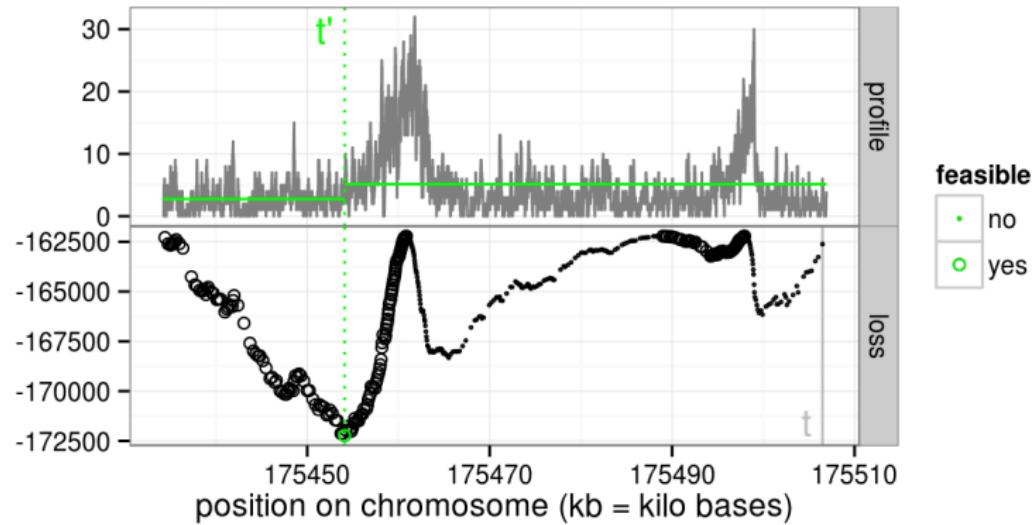
$$\mathcal{L}_{2,t} = \min_{t' < t} \underbrace{\mathcal{L}_{1,t'}}_{\text{optimal loss in 1 segment up to } t'} + \underbrace{c(t',t]}_{\text{optimal loss of 2nd segment } (t', t]}$$

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



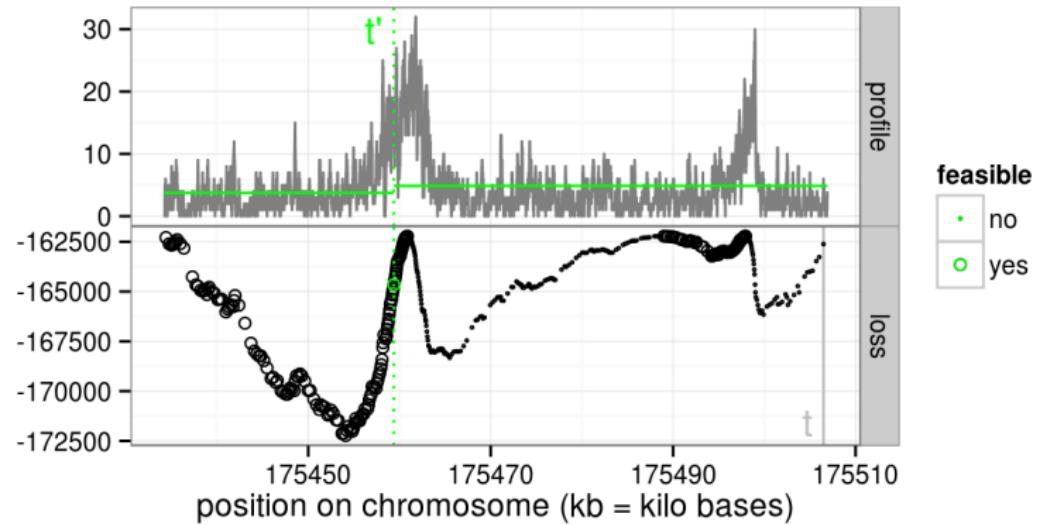
$$\mathcal{L}_{2,t} = \min_{t' < t} \underbrace{\mathcal{L}_{1,t'}}_{\text{optimal loss in 1 segment up to } t'} + \underbrace{c(t',t]}_{\text{optimal loss of 2nd segment } (t', t]}$$

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



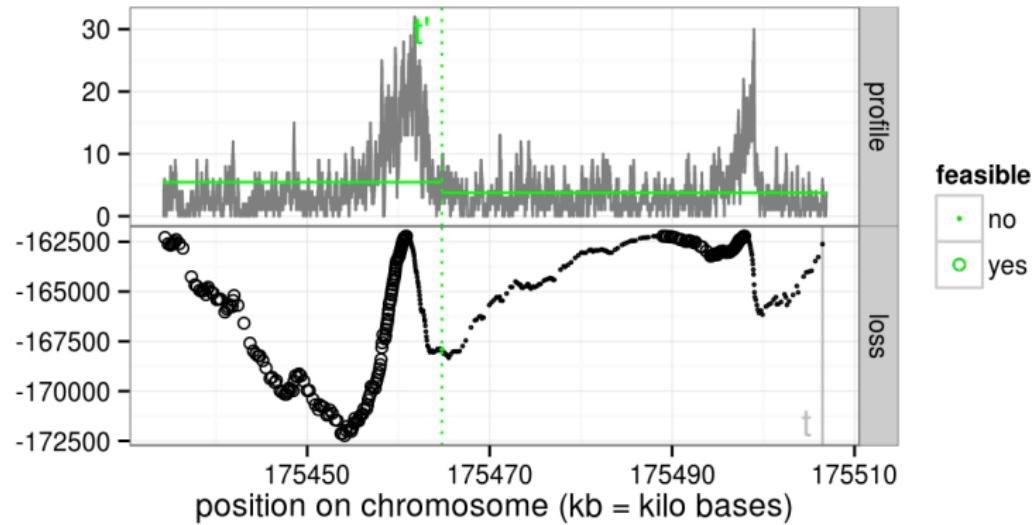
$$\mathcal{L}_{2,t} = \min_{t' < t} \underbrace{\mathcal{L}_{1,t'}}_{\text{optimal loss in 1 segment up to } t'} + \underbrace{c(t',t]}_{\text{optimal loss of 2nd segment } (t', t]}$$

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



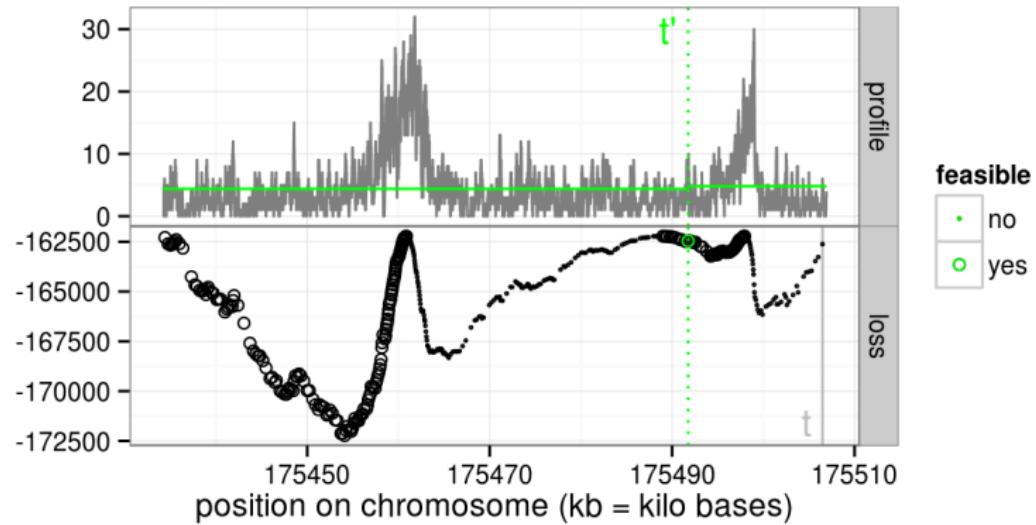
$$\mathcal{L}_{2,t} = \min_{t' < t} \underbrace{\mathcal{L}_{1,t'}}_{\text{optimal loss in 1 segment up to } t'} + \underbrace{c(t',t]}_{\text{optimal loss of 2nd segment } (t', t]}$$

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 | $O(d^4)$ | $O(d^2)$ |
| ⋮ | ⋮ | ⋮ |

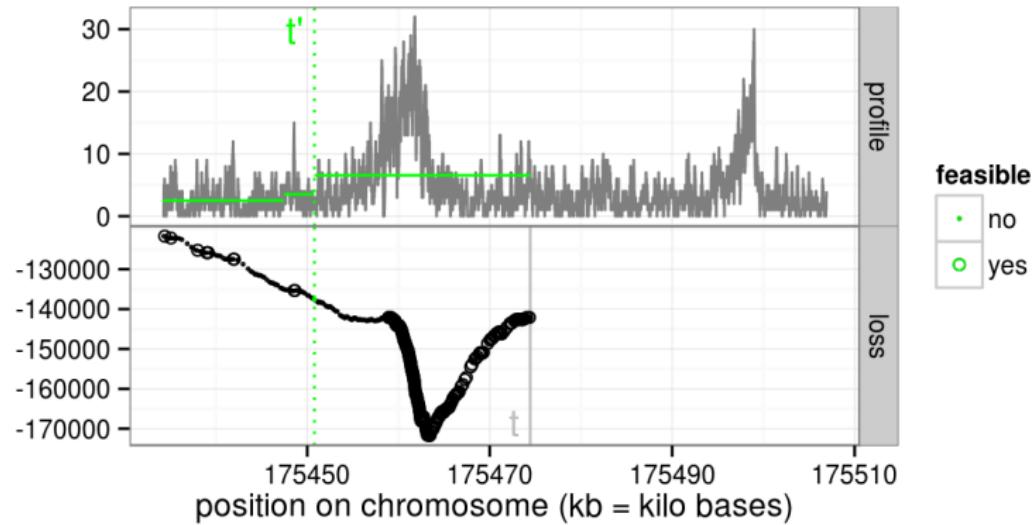
For example $d = 5735$ data points to segment.

$$d^2 = 32890225$$

$$d^3 = 188625440375$$

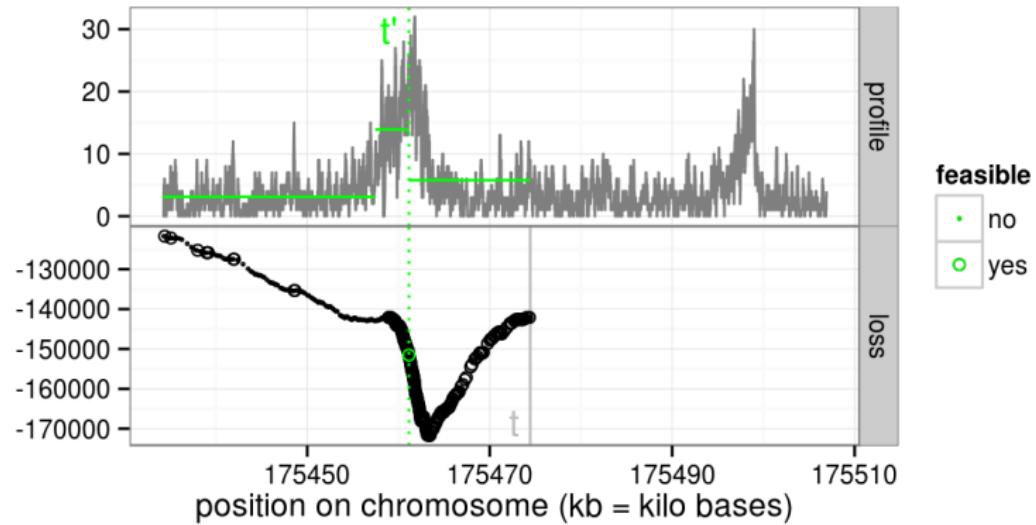
⋮

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



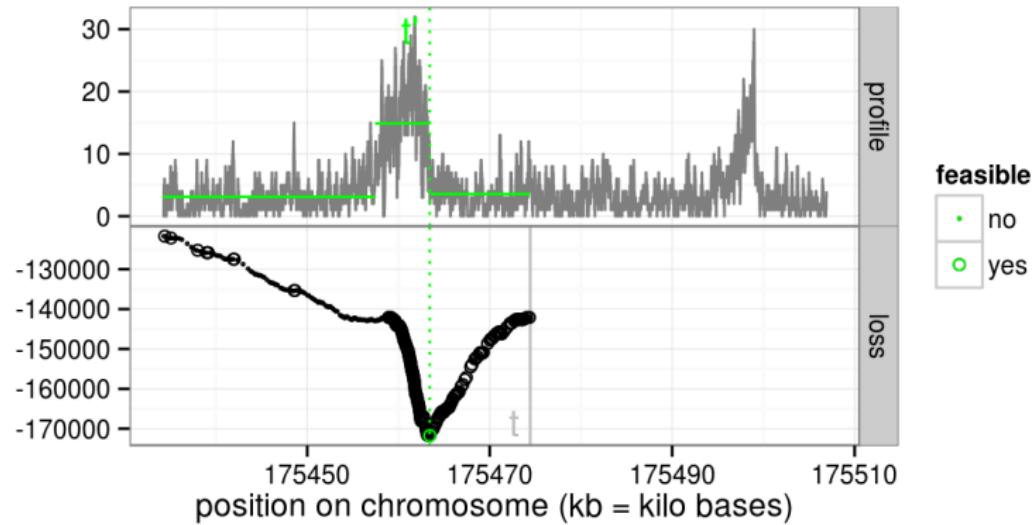
$$\mathcal{L}_{3,t} = \min_{t' < t} \underbrace{\mathcal{L}_{2,t'}}_{\text{optimal loss in 2 segments up to } t'} + \underbrace{c(t',t]}_{\text{optimal loss of 3rd segment } (t',t]}$$

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



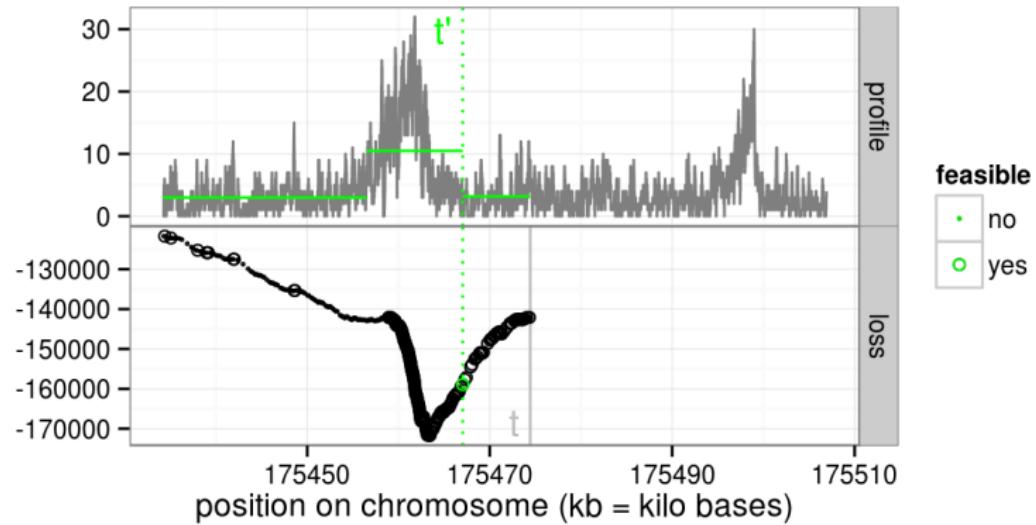
$$\mathcal{L}_{3,t} = \min_{t' < t} \underbrace{\mathcal{L}_{2,t'}}_{\text{optimal loss in 2 segments up to } t'} + \underbrace{c(t',t]}_{\text{optimal loss of 3rd segment } (t',t]}$$

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 α_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 $\tilde{\mathbf{m}}^{p^*[\exp f(\mathbf{x})]}(\mathbf{y})$.

Benchmark: 7 annotated region data sets

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

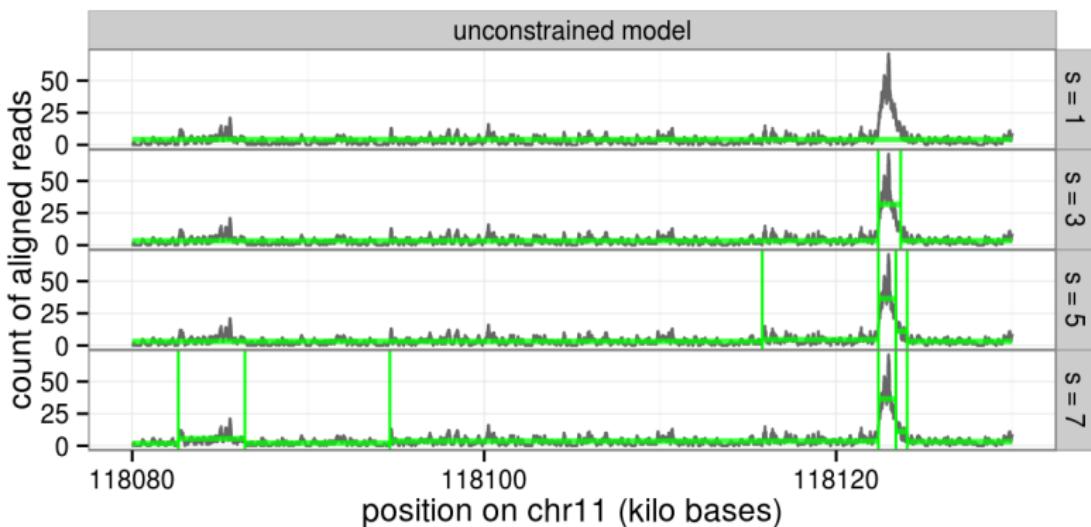
- ▶ 4 annotators (AM, TDH, PGP, XJ).
- ▶ 8 cell types.
- ▶ 37 annotated H3K4me3 profiles (sharp peaks).
- ▶ 29 annotated H3K36me3 profiles (broadly enriched domains).
- ▶ 12,826 annotated regions in total.
- ▶ 2752 separate segmentation problems.

Used the cDPA on the annotated data.

- ▶ cDPA computed models with 0, ..., 9 peaks (for 99.5% of problems).
- ▶ For the biggest problem, cDPA took 3 hours. ($d = 88,509$ data points, 3.5 megabases)
- ▶ macs takes about 90 minutes for one whole genome.

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

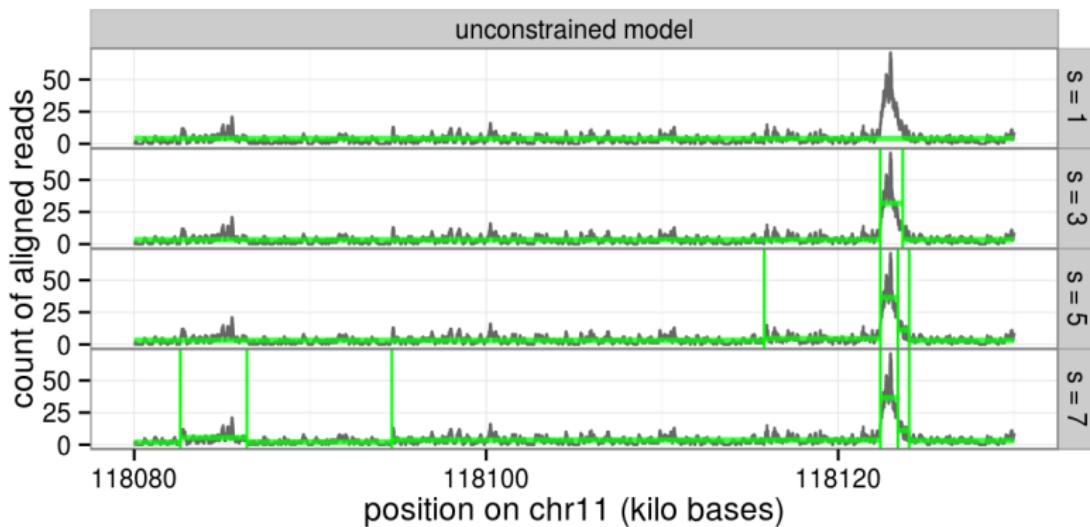
- ▶ Let $\mathbf{y} = [y_1 \ \cdots \ y_d] \in \mathbb{Z}_+^d$ be the aligned read counts for one sample and one genomic region.
- ▶ Fix $s_{\max} = 19$, the maximum number of segments.
- ▶ For each number of segments $s \in \{1, \dots, s_{\max}\}$, we want:

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

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

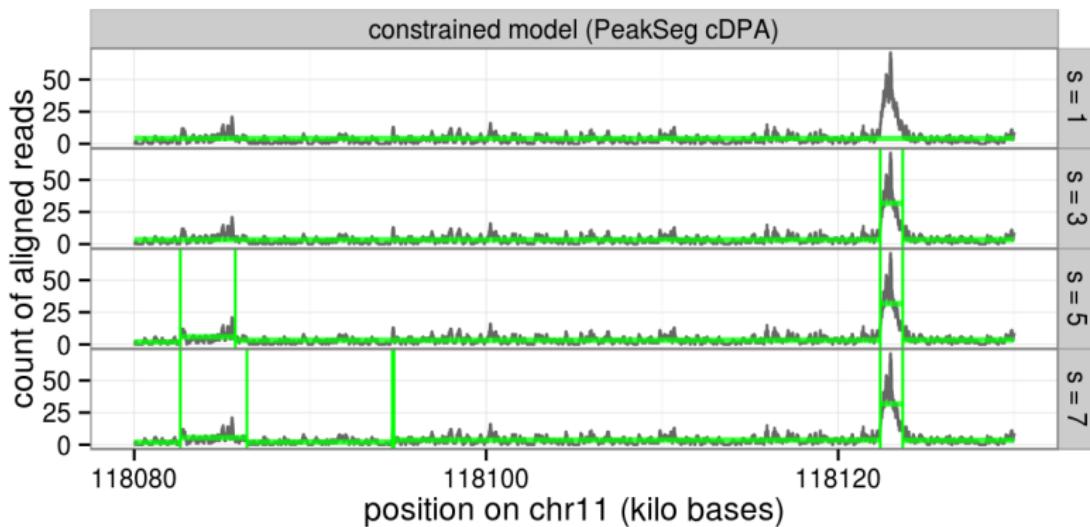
- ▶ Pruned Dynamic Programming (Rigaill arXiv:1004.0887) returns $\hat{\mathbf{m}}^1(\mathbf{y}), \dots, \hat{\mathbf{m}}^{s_{\max}}(\mathbf{y})$ in $O(s_{\max} d \log d)$ time.

Maximum likelihood segmentations



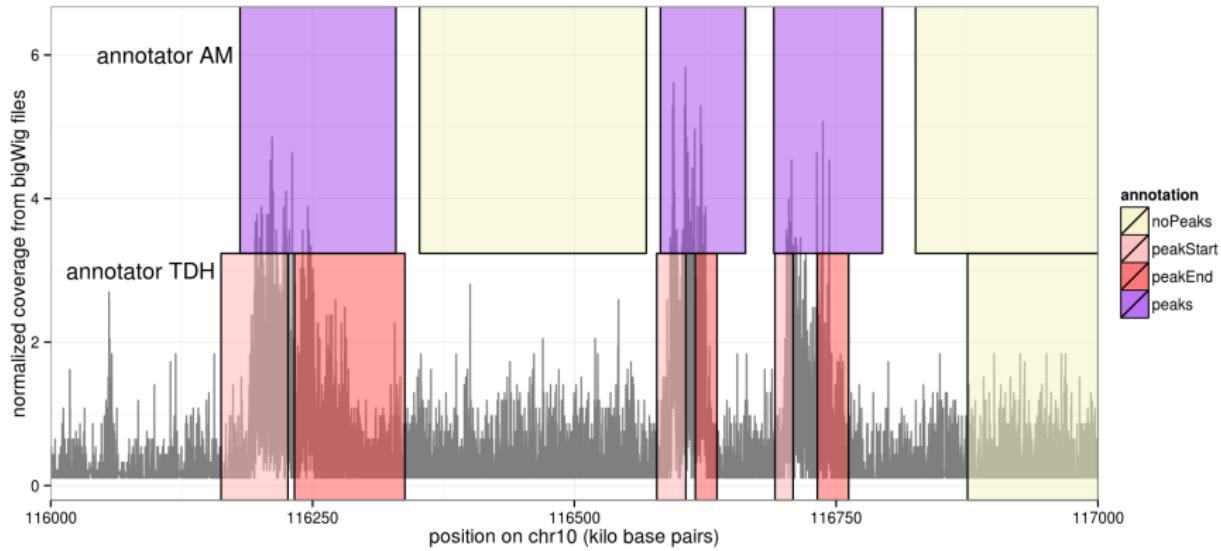
Model with $s = 5$ segments changes up, up, down, down.
How to define peaks? Introduce a threshold parameter?

Constrained maximum likelihood segmentations



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

Two annotators provide consistent labels, but different precision



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

Comparison on annotated McGill benchmark data set

Hocking et al, 2014, arXiv:1409.6209.

- ▶ Manually annotate regions with or without peaks.
<http://cbio.ensmp.fr/~thocking/chip-seq-chunk-db/>
- ▶ Tune 1 parameter that affects the number of peaks.
- ▶ Choose the parameter that minimizes the annotation error.

Results:

- ▶ MACS best for H3K4me3 (sharp peak pattern),
- ▶ HMCan.broad best for H3K36me3 (broad peak pattern).
- ▶ Consistent across 4 annotators (PhD students, postdocs).
- ▶ 10–20% test error rates.

Can we do better than unsupervised peak detectors? Yes!

We propose **PeakSeg**, a new model with efficient algorithms for supervised peak detection.

- ▶ Input: several ChIP-seq profiles, manually annotated regions.
- ▶ New methods for peak detection:
 - ▶ Constrained optimal segmentation.
 - ▶ Efficient supervised learning using manually annotated regions.
- ▶ Output: predicted peaks for each profile.

State-of-the-art peak detection accuracy (on both sharp H3K4me3 and broad H3K36me3 profiles).

Can we do better than unsupervised peak detectors? Yes!

We propose **PeakSeg**, a new model with efficient algorithms for supervised peak detection.

- ▶ Input: several ChIP-seq profiles, **manually annotated regions**.
- ▶ New methods for peak detection:
 - ▶ Constrained optimal segmentation.
 - ▶ Efficient **supervised learning using manually annotated regions**.
- ▶ Output: predicted peaks for each profile.

State-of-the-art peak detection accuracy (on both sharp H3K4me3 and broad H3K36me3 profiles).

Existing 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?

How to choose model parameters?

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

```
[-g GSIZEx  
[-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
```