PeakSeg: Peak detection via constrained optimal Segmentation

by Guillem Rigaill (Université d'Evry, France), Toby Dylan Hocking and Guillaume Bourque (McGill University, Canada).

Introduction:

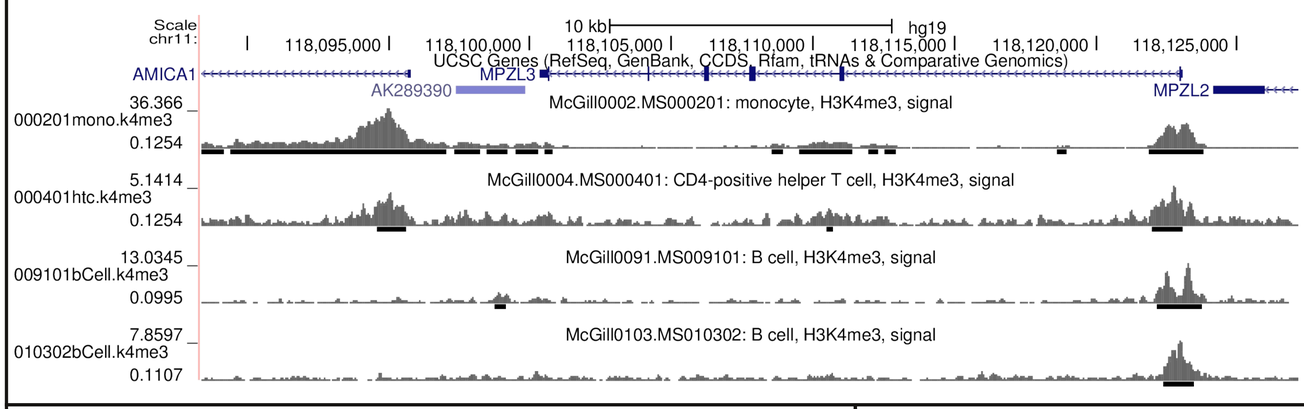
ChIP-seq data characterize the genomic binding positions for proteins such as histones and transcription factors.

The goal of a peak detection algorithm is to filter out background noise, and define the precise genomic positions of "peaks" with many aligned sequence reads.

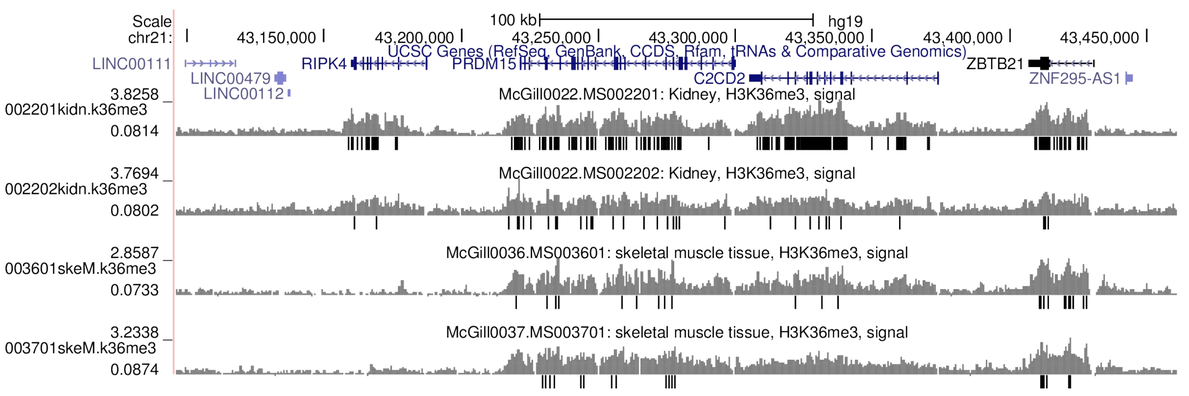
Problem:

unsupervised macs2 peak detection algorithm (Zhang et al. Model-based analysis of ChIP-Seq, Genome Biology 2008) does not match visually obvious peaks.

Data from http://epigenomesportal.ca H3K4me3 sharp peaks,



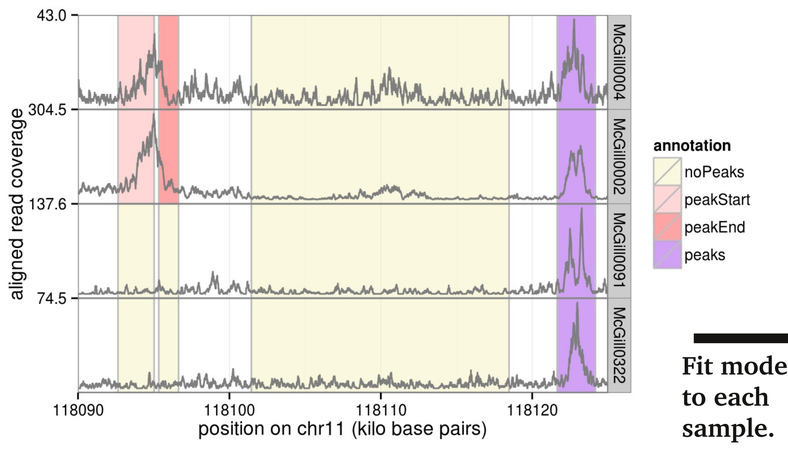
H3K36me3 broadly enriched domains.



Benchmark data set:

manually annotated regions containing peaks or no peaks.
http://cbio.ensmp.fr/~thocking/chip-seq-chunk-db/

- Seven data sets containing 12,826 manually annotated regions across 37 samples, 8 cell types, from 4 annotators.
- peakStart/peakEnd regions should contain exactly 1 peak start/end.
- peaks regions should contain at least 1 overlapping peak.
- Below: 16 regions across 4 samples created by annotator TDH, 12 possible false positives, 8 possible false negatives.



Fit models to each sample.

Proposed Method: constrained maximum likelihood segmentation.

The Supervised Peak Detection Problem

- A ChIP-seq profile on a single chromosome with d base pairs is a vector $\mathbf{y} = \begin{bmatrix} y_1 & \cdots & y_d \end{bmatrix} \in \mathbb{Z}_+^d$ of counts of aligned sequence reads.
- A peak caller is a function $c : \mathbb{Z}_+^d \to \{0,1\}^d$ which returns 0 for background noise and 1 for a peak.
- We are given $i \in \{1, \ldots, n\}$ profiles $\mathbf{y}_i$ and annotated regions R_i.
- The goal is to find a peak caller with minimal error on some test profiles:

$$\underset{c}{\text{minimize}} \sum_{i \in \text{test}} E[c(\mathbf{y}_i), R_i],$$

where E is the number of false positive and false negative regions R_i.

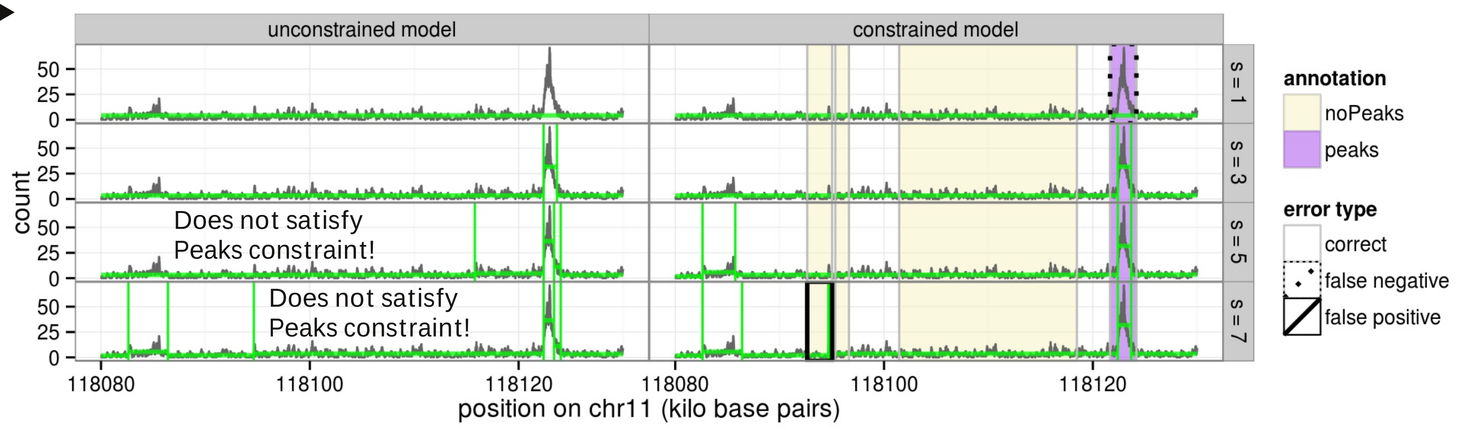
Unsupervised PeakSeg: constrained segmentation

$$\tilde{\mathbf{m}}^s(\mathbf{y}) = \underset{\mathbf{m} \in \mathbb{R}^d}{\arg\min} \quad \rho(\mathbf{m}, \mathbf{y})$$

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

Peaks constraint: $P_j(\mathbf{m}) \in \{0,1\}$ for all $j \in \{1, \ldots, d\}$.

- The Poisson loss function is $\rho(\mathbf{m}, \mathbf{y}) = \sum_{j=1}^d m_j - y_j \log m_j$.
- $\text{Segments}(\mathbf{m}) \in \{1, \ldots, d\}$ is the number of piecewise constant segments of the mean vector $\mathbf{m}$.
- The indicator for a peak at base j is $P_j(\mathbf{m}) = \sum_{k=1}^j \text{sign}(m_k - m_{k-1})$.
- Geometric interpretation of **Peaks constraint**: segment mean must change up, down, up, down, ...
- For example for $s = 5$ segments, mean values should satisfy $\mu_1 < \mu_2 > \mu_3 < \mu_4 > \mu_5$.



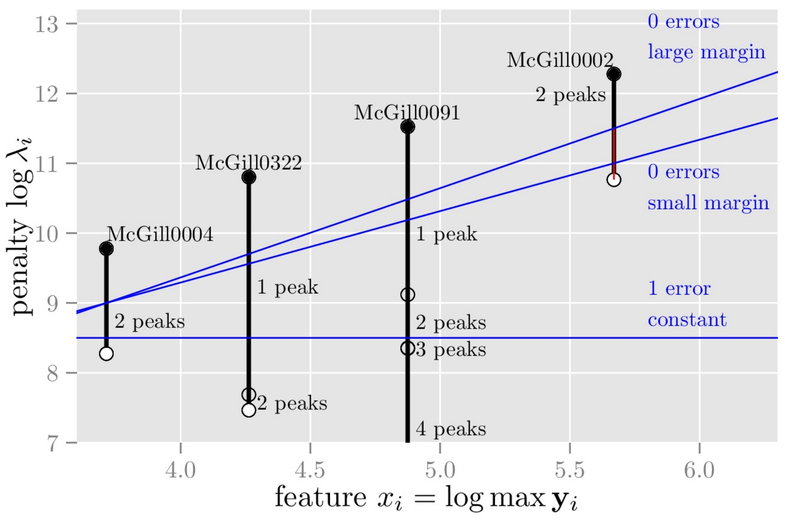
Supervised PeakSeg: learning a penalty function

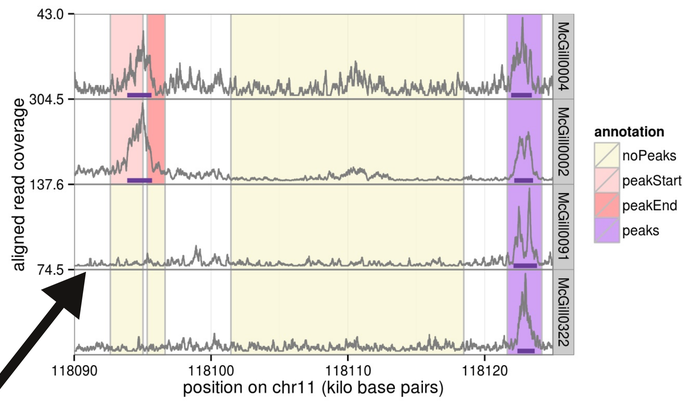
Reference: Hocking, Rigaill, et al. Learning Sparse Penalties for Change-point Detection using Max Margin Interval Regression. ICML 2013.

- Given a positive penalty $\lambda \in \mathbb{R}_+$, the optimal number of segments is

$$s^*(\lambda, \mathbf{y}) = \underset{s \in \{1,3,\ldots,s_{\max}\}}{\arg\min} \rho\left[\tilde{\mathbf{m}}^s(\mathbf{y}), \mathbf{y}\right] + \lambda s.$$

- Sample-specific penalty values $\log \lambda_i = f(\mathbf{x}_i) = \beta + \mathbf{w}^\intercal \mathbf{x}_i$.
- An $m = 2$-dimensional feature vector $\mathbf{x}_i = \begin{bmatrix} \log \max \mathbf{y}_i & \log d_i \end{bmatrix}$, where d_i is the number of base pairs for sample i.
- **Geometric interpretation: interval regression.** the minimum error penalty function f intersects the target interval of penalty values for each sample.
- For separable data, find the penalty function f with largest margin:





Apply learned penalty function to select optimal number of peaks for each sample.

For real data: minimize a smooth convex squared hinge loss $\ell_i : \mathbb{R} \to \mathbb{R}_+$ which depends on the annotated region data R_i:

$$\hat{f} = \underset{f}{\arg\min} \sum_{i=1}^n \ell_i \left[f(\mathbf{x}_i)\right].$$

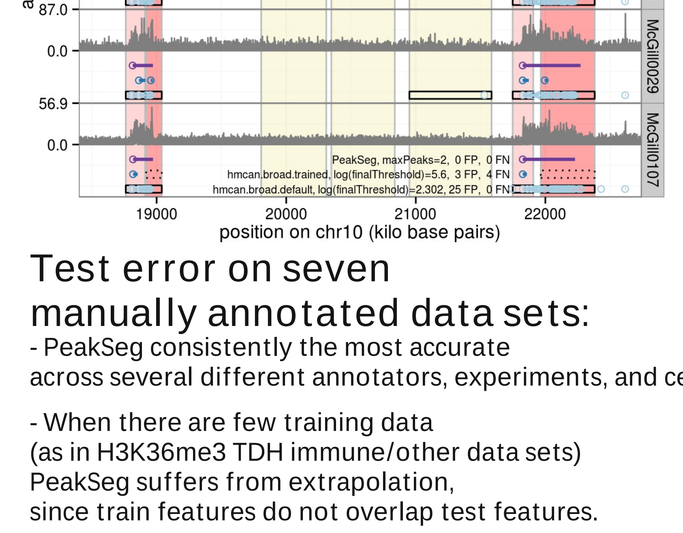
To make a prediction on a test sample with profile $\mathbf{y}$ and features $\mathbf{x}$,

- Compute the predicted penalty $\hat{\lambda} = \exp \hat{f}(\mathbf{x})$,
- the predicted number of segments $\hat{s} = s^*(\hat{\lambda}, \mathbf{y})$,
- and finally the predicted peaks $\mathbf{P}\left[\tilde{\mathbf{m}}^{\hat{s}}(\mathbf{y})\right]$, where $\mathbf{P}[\mathbf{m}] = \begin{bmatrix} P_1(\mathbf{m}) & \cdots & P_d(\mathbf{m}) \end{bmatrix} \in \{0,1\}^d$.

Results:

State-of-the-art peak detection for both sharp and broad peaks.

Some examples of detected peaks:



Test error on seven manually annotated data sets:

- PeakSeg consistently the most accurate across several different annotators, experiments, and cell types.

- When there are few training data (as in H3K36me3 TDH immune/other data sets) PeakSeg suffers from extrapolation, since train features do not overlap test features.

See extended results (test ROC curves, peak calls) on interactive figure:
http://cbio.ensmp.fr/~thocking/figure-dp-peaks-interactive/

Code available!

Previous work: unconstrained model computed using Segmentor3IsBack R package, Cleynen et al. (2014). Errors computed using PeakError R package.
https://github.com/tdhock/PeakError

This work: constrained model computed using dynamic programming algorithm, PeakSegDP R package.
https://github.com/tdhock/PeakSegDP

Conclusions/future work:

- First supervised peak detector that learns from manually annotated regions with and without peaks.
- State-of-the-art peak detection on both sharp H3K4me3 and broad H3K36me3 profiles.
- Apply to other histone mark types and transcription factor ChIP data.
- Current dynamic programming algorithm has time complexity quadratic in number of data to segment (base pairs), but Pruned Dynamic Programming (Rigaill arXiv:1004.0887) is a linear time algorithm that could be used.
- Other features for penalty learning problem?
- Detecting peaks in the same genomic positions across several samples?