PeakSeg: constrained optimal segmentation and supervised penalty learning for peak detection in count data

by Toby Dylan Hocking toby.hocking@mail.mcgill.ca, Guillem Rigaill, and Guillaume Bourque.

Introduction:

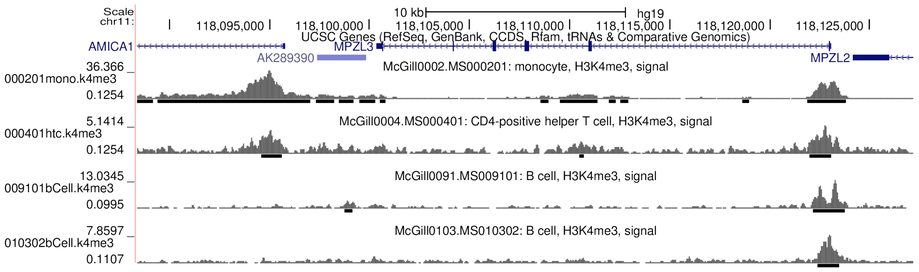
ChIP-seq data characterize the genomic binding positions for proteins such as histones and transcription factors.

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.

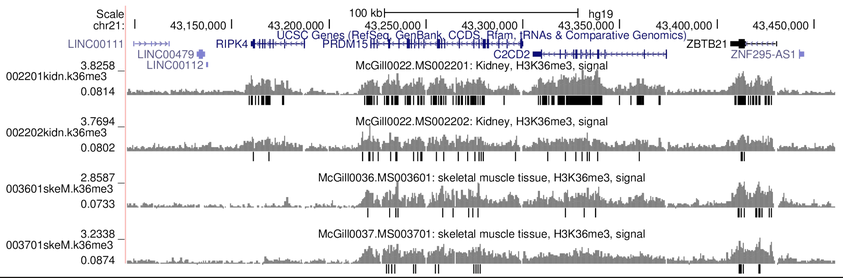
Problem:

unsupervised macs2 peak detection algorithm with default parameters (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.



Proposed Method: constrained maximum likelihood segmentation + supervised penalty learning.

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 region labels L_i.
- The goal is to find a peak caller with minimal error on some test profiles:

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

where E is the number of false positive and false negative labels L_i.

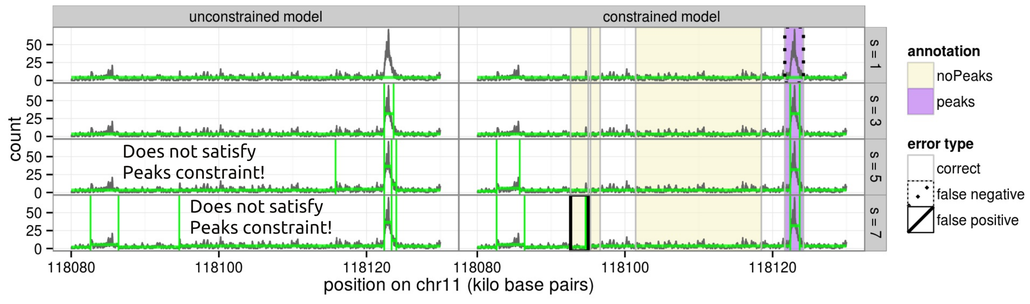
PeakSeg: constrained Poisson segmentation

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

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

$$\textbf{Peaks constraint:} \quad P_j(\mathbf{m}) \in \{0,1\} \text{ for all } j \in \{2, \ldots, d\}.$$

- $\text{PoissonLoss}(\mathbf{m}, \mathbf{y}) = \sum_{j=1}^d m_j - y_j \log m_j$.
- $\text{Segments}(\mathbf{m}) = \sum_{j=2}^d I(m_j \neq m_{j-1}) \in \{1, \ldots, d\}$ 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 $\mu_1 < \mu_2 > \mu_3 < \mu_4 > \mu_5$.



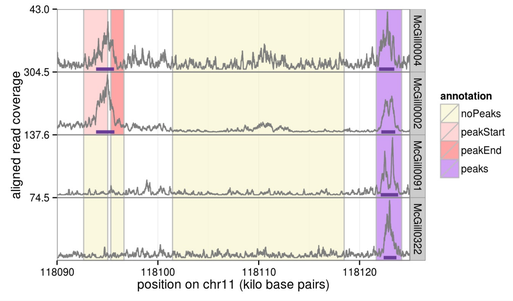
Supervised penalty learning

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

- Train on count data $\mathbf{y}_i \in \mathbb{Z}_+^{d_i}$ and labels L_i for i profiles.
- For a positive penalty $\lambda \in \mathbb{R}_+$, the optimal number of segments is

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

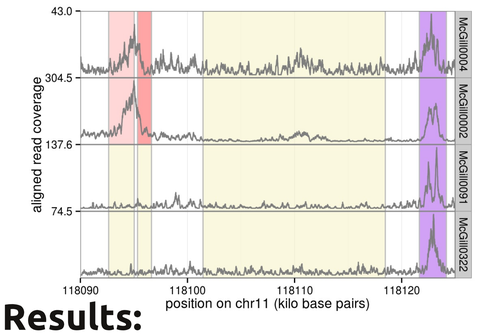
- Model complexity $h(s, d_i)$ is either AIC/BIC or oracle (see table below).
- Profile-specific penalty values $\log \lambda_i = f(\mathbf{x}_i) = \beta + \mathbf{w}^\intercal \mathbf{x}_i$.
- Learn the penalty f with minimal incorrect regions L_i on a train data set.
- At test time compute the predicted penalty $\hat{\lambda} = \exp \hat{f}(\mathbf{x})$ and the predicted number of segments $\hat{s} = s^*(\hat{\lambda}, \mathbf{y})$.



Benchmark data set:

manually annotated regions containing peaks or no peaks (Hocking et al. arXiv:1409.6209). http://cbio.ensmp.fr/~thocking/chip-seq-chunk-db/

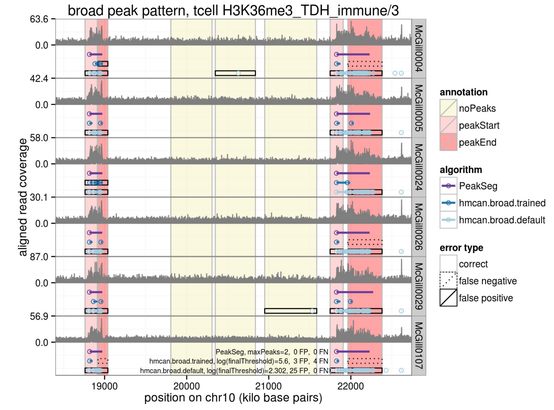
- Seven data sets containing 12,826 manually annotated regions across 37 samples, 8 cell types, from 4 annotators.
- peakStart/peakEnd regions should contain exactly 1 peak start/end.
- peaks regions should contain at least 1 overlapping peak.
- Below: 16 regions across 4 samples created by annotator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



Results:

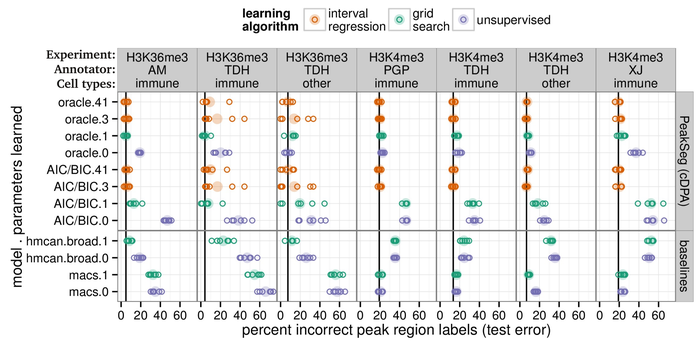
State-of-the-art peak detection for both sharp and broad peaks.

Some examples of detected peaks:



Results, continued: state-of-the-art test error on seven benchmark data sets

- Repeat six times: randomly split data set into half train, half test.
- Train baseline macs, hmcan.broad algorithms by choosing the best scalar significance threshold.
- PeakSeg consistently the most accurate across several different annotators, experiments, and cell types.
- When there are very few training data (as in H3K36me3 TDH immune/other data sets) PeakSeg suffers from extrapolation, since train features do not overlap test features.
- Oracle model complexity of Cleynen and Lebarbier (arXiv:1301.2534) more accurate than simpler AIC/BIC-type penalty.



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	$\exp(\beta + \mathbf{w}^\intercal \mathbf{x}_i)$	$\beta \in \mathbb{R}, \mathbf{w} \in \mathbb{R}^{40}$	regularized int. reg.

Conclusions:

PeakSeg showed state-of-the-art peak detection on both sharp H3K4me3 and broad H3K36me3 profiles.
- Heuristic/approximate constrained dynamic programming algorithm (cDPA) searches for most likely peak locations.
- First supervised peak detector that learns from manually annotated region labels with and without peaks.
- Sometimes not as accurate as other algorithms when there are very few manually annotated regions in train set.

Future work:

- Current algorithm (cDPA) has time complexity quadratic in number of data to segment (base pairs), but pruned dynamic programming (Rigaill arXiv:1004.0887) is a log-linear time algorithm that could be used.
- An algorithm that provably recovers the PeakSeg model? (see paper for example where cDPA does not find PeakSeg)
- Other features for penalty learning problem, based on Poisson segmentation model selection theory?
- Instead of taking features for granted, learn them based on profile count data.
- Overlapping peaks at the same positions across samples (PeakSegJoint model, arXiv:1506.01286.

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: PeakSeg computed using constrained dynamic programming algorithm (cDPA), PeakSegDP R package. https://github.com/tdhock/PeakSegDP