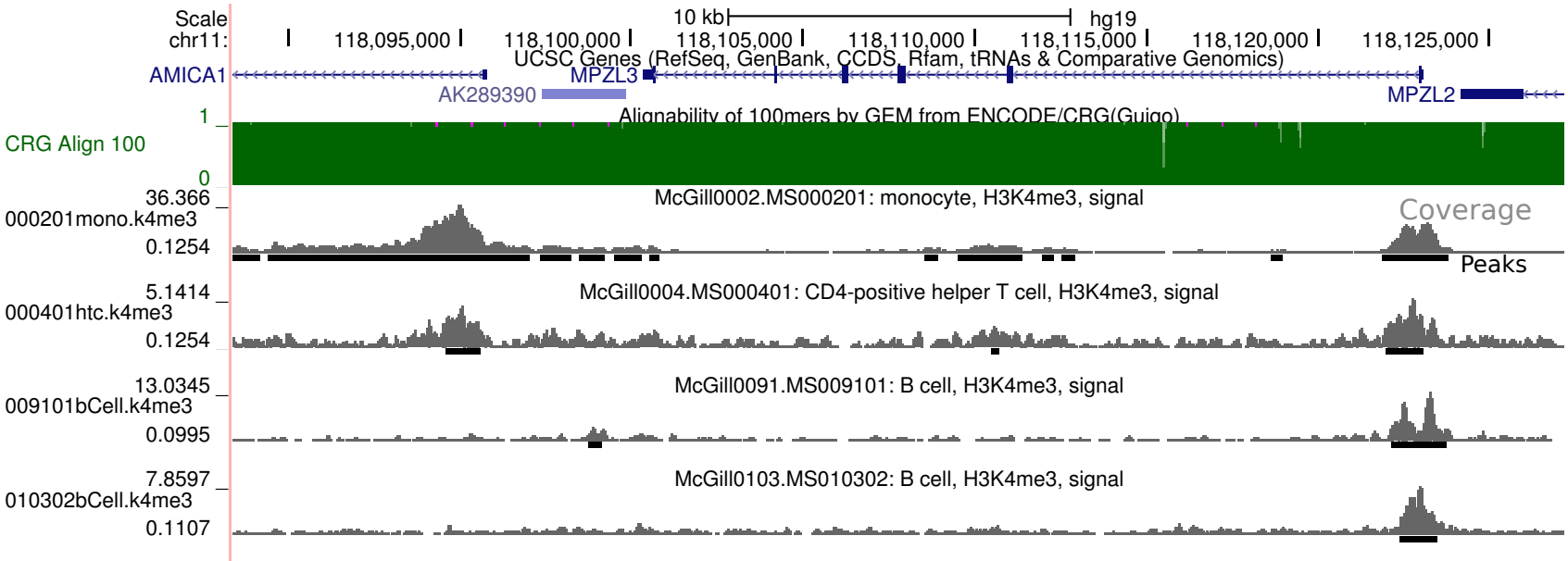
A Generalized Functional Pruning Optimal Partitioning (GFPOP) Algorithm for Peak Detection in Large Genomic Data

PeakSegDisk R package / arXiv:1810.00117

Toby Dylan Hocking, toby.hocking@nau.edu, joint work with Guillem Rigaill, Guillaume Bourque, Paul Fearnhead

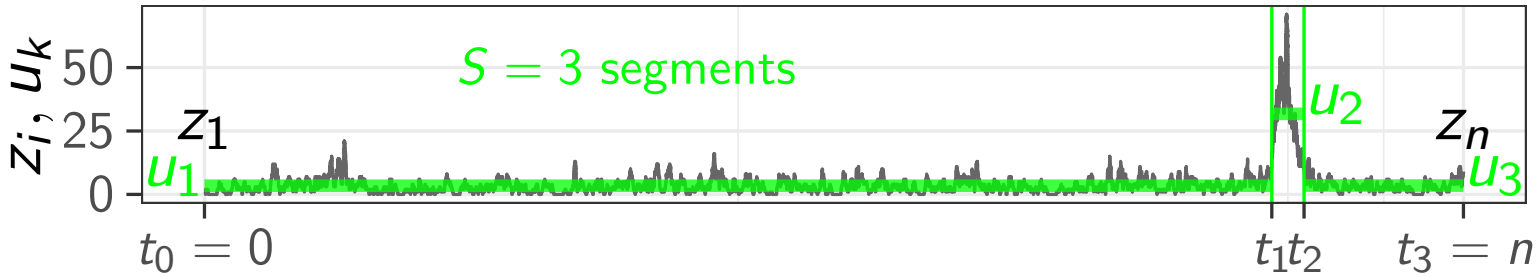
10 July 2019

Problem: find peaks in each of several samples



- Grey profiles are noisy aligned read count signals – peaks are genomic locations with protein binding sites.
- Black bars are peaks called by MACS2 (Zhang et al, 2008) – many false positives! (black bars where there is only noise)
- From a machine learning perspective, this is binary classification (positive=peaks, negative=noise).

Maximum likelihood changepoint detection with up-down constraints on adjacent segment means

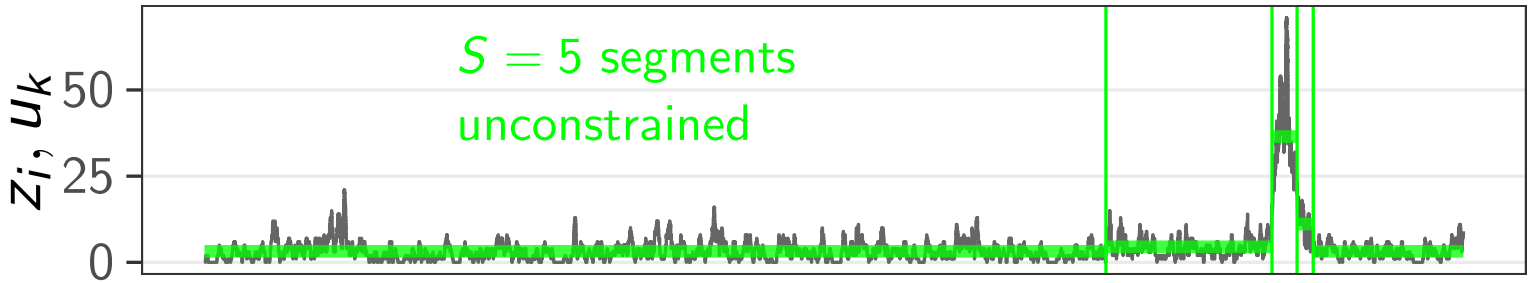


$$\underset{\substack{\mathbf{u}\in\mathbb{R}^S \\ 0=t_0<t_1<\cdots<t_{S-1}<t_S=n}}{\text{minimize}} \quad \sum_{k=1}^{S} \sum_{i=t_{s-1}+1}^{t_s} \ell(u_k, z_i)$$

$$\text{subject to} \quad u_{k-1} \leq u_k \ \forall k \in \{2, 4, \dots\},$$
$$u_{k-1} \geq u_k \ \forall k \in \{3, 5, \dots\}.$$

- One tuning parameter = number of segments $S \in \{1, 3, \dots\}$.
- Hard optimization problem, naively $O(n^S)$ time.
- Previous unconstrained model: not always up-down changes.
- Interpretable: $P = (S-1)/2$ peaks (segments 2, 4, ...).
- H *et al.*, *ICML* 2015: $O(Sn^2)$ time approximate algorithm.
- H *et al.*, arXiv 2017: $O(Sn \log n)$ time optimal algorithm.

Maximum likelihood changepoint detection with up-down constraints on adjacent segment means

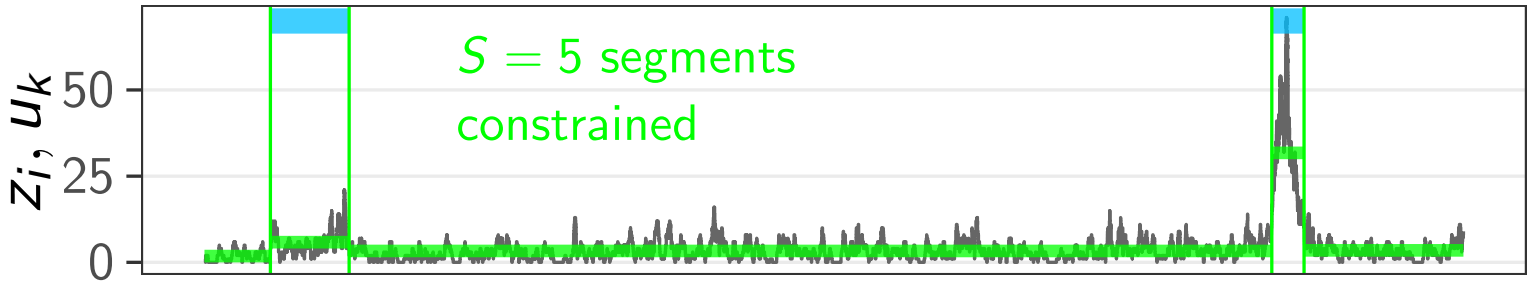


$$\underset{\substack{\mathbf{u}\in\mathbb{R}^S \\ 0=t_0<t_1<\cdots<t_{S-1}<t_S=n}}{\text{minimize}} \sum_{k=1}^{S} \sum_{i=t_{s-1}+1}^{t_s} \ell(u_k, z_i)$$

$$\text{subject to} \quad u_{k-1} \leq u_k \ \forall k \in \{2, 4, \dots\},$$
$$u_{k-1} \geq u_k \ \forall k \in \{3, 5, \dots\}.$$

- One tuning parameter = number of segments $S \in \{1, 3, \dots\}$.
- Hard optimization problem, naively $O(n^S)$ time.
- Previous unconstrained model: not always up-down changes.
- Interpretable: $P = (S-1)/2$ peaks (segments 2, 4, ...).
- H *et al.*, *ICML* 2015: $O(Sn^2)$ time approximate algorithm.
- H *et al.*, arXiv 2017: $O(Sn \log n)$ time optimal algorithm.

Maximum likelihood changepoint detection with up-down constraints on adjacent segment means



$$\underset{\substack{\mathbf{u}\in\mathbb{R}^S \\ 0=t_0<t_1<\cdots<t_{S-1}<t_S=n}}{\text{minimize}} \quad \sum_{k=1}^{S} \sum_{i=t_{s-1}+1}^{t_s} \ell(u_k, z_i)$$

$$\text{subject to} \quad u_{k-1} \leq u_k \ \forall k \in \{2, 4, \dots\},$$
$$u_{k-1} \geq u_k \ \forall k \in \{3, 5, \dots\}.$$

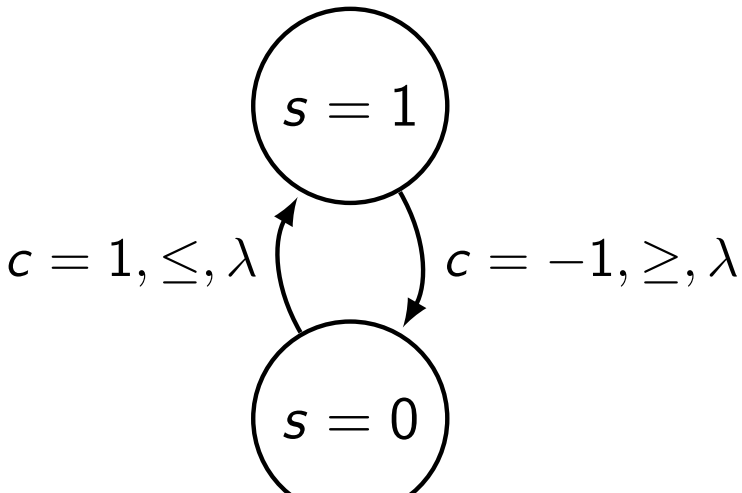
- One tuning parameter = number of segments $S \in \{1, 3, \dots\}$.
- Hard optimization problem, naively $O(n^S)$ time.
- Previous unconstrained model: not always up-down changes.
- Interpretable: $P = (S-1)/2$ peaks (segments 2, 4, ...).
- H *et al.*, *ICML* 2015: $O(Sn^2)$ time approximate algorithm.
- H *et al.*, arXiv 2017: $O(Sn \log n)$ time optimal algorithm.

Penalized changepoint problem with up-down constraints on adjacent segment means

H *et al.*, *arXiv* 2018. Generalized Functional Pruning Optimal Partitioning Algorithm (GFPOP) solves for one penalty $\lambda \in \mathbb{R}_+$ in $O(n \log n)$ time.

$$\underset{\substack{\mathbf{m}\in\mathbb{R}^n,\ \mathbf{s}\in\{0,1\}^n \\ \mathbf{c}\in\{-1,0,1\}^{n-1}}}{\text{minimize}} \quad \sum_{i=1}^{n} \ell(m_i, z_i) + \lambda \sum_{i=1}^{n-1} I(c_i \neq 0)$$

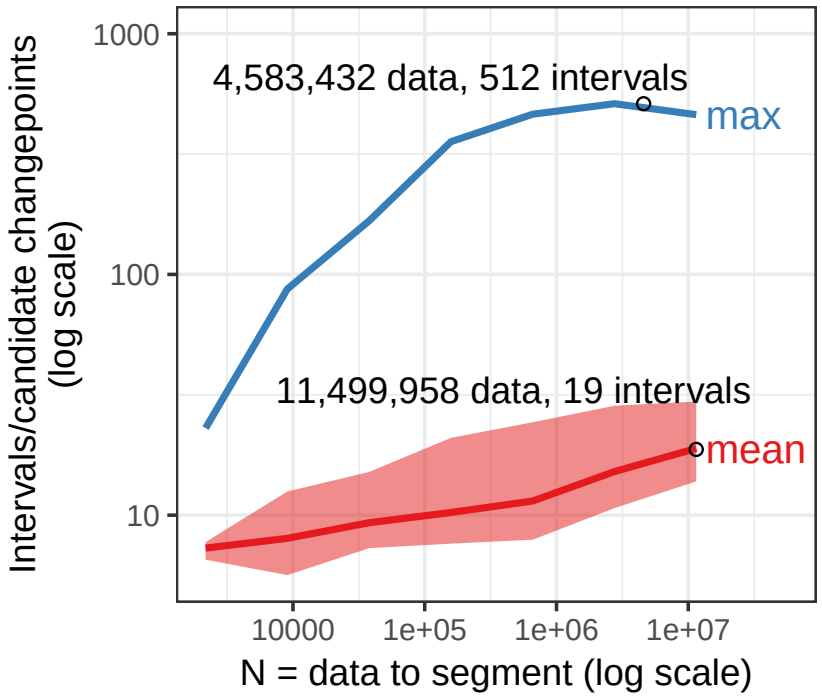
subject to
- no change: $c_t = 0 \Rightarrow m_t = m_{t+1}$ and $s_t = s_{t+1}$
- go up: $c_t = 1 \Rightarrow m_t \leq m_{t+1}$ and $(s_t, s_{t+1}) = (0, 1)$,
- go down: $c_t = -1 \Rightarrow m_t \geq m_{t+1}$ and $(s_t, s_{t+1}) = (1, 0)$.



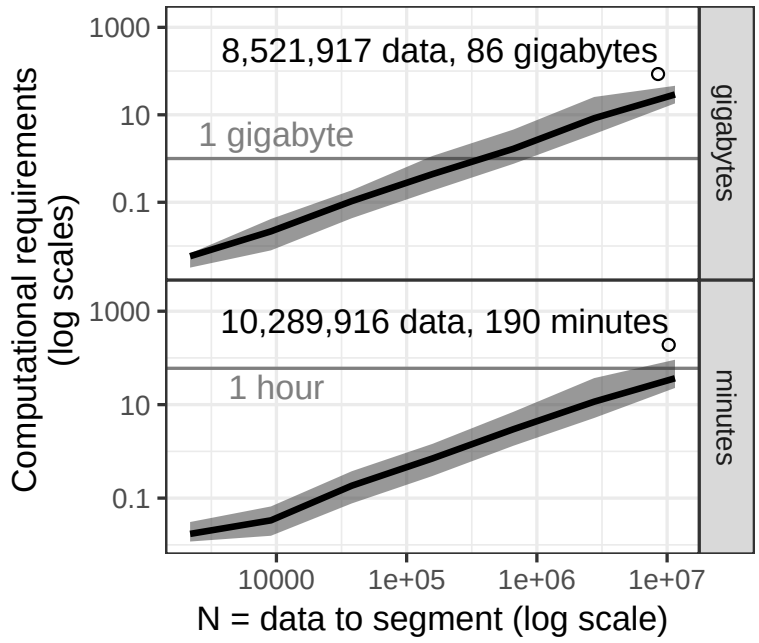
Nodes=states s,
Edges=changes c (constraint, penalty).

Proposed algorithm is fast, empirically $O(n \log n)$

$O(\log n)$ candidate changepoints.



Overall $O(n \log n)$ time/space.



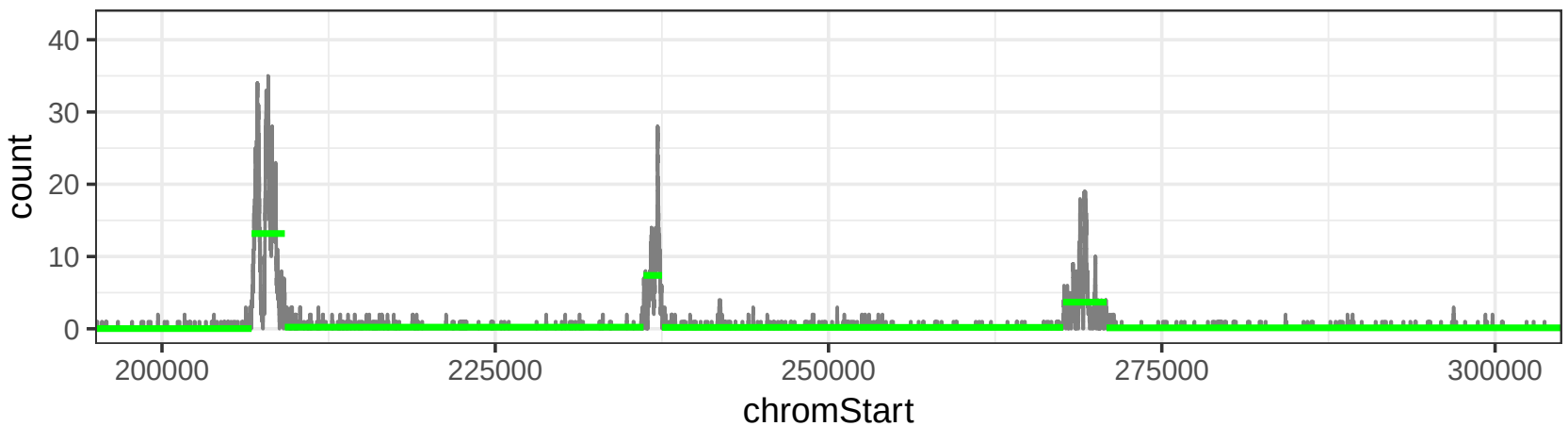
Example: $n = 10^7$ data, 1000 peaks. Previous algo would require 150 TB of storage space and 12 weeks of computation time! With the right penalty λ the proposed algo takes 1 hour, 80 GB.

First save data as coverage.bedGraph file with n lines, so algorithm only needs $O(\log n)$ memory / $O(n \log n)$ disk

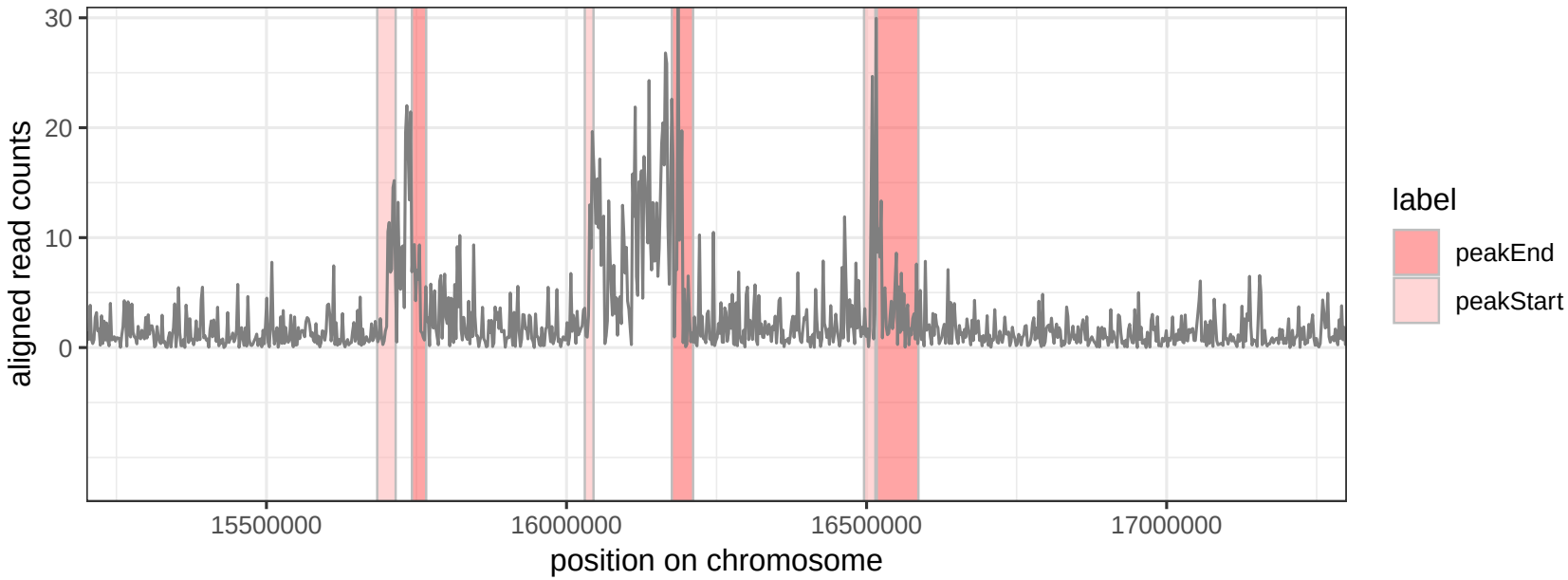
```
write.table(Mono27ac$coverage,
  "Mono27ac/chr11-60000-580000/coverage.bedGraph",
  col.names=FALSE, row.names=FALSE, quote=FALSE,
  sep="\t")

fit <- PeakSegDisk::problem.PeakSegFPOP(
  "Mono27ac/chr11-60000-580000", "10000")

fit$loss

##    penalty segments peaks  bases bedGraph.lines
## 1:   10000       15     7 520000           6921
##    mean.pen.cost total.loss equality.constraints
## 1:     0.2189332   43845.26                     0
##    mean.intervals max.intervals megabytes seconds
## 1:       14.42581            41  10.27206   0.824
```

Visualization of count data and segmentation/peaks

```
library("ggplot2")
ggplot() + theme_bw() +
  coord_cartesian(xlim = c(2e5, 3e5)) +
  geom_step(aes(
    chromStart, count),
    color = "grey50", data = Mono27ac$coverage) +
  geom_segment(aes(
    chromStart, mean, xend = chromEnd, yend = mean),
    color = "green", size = 1, data = fit$segments)
```

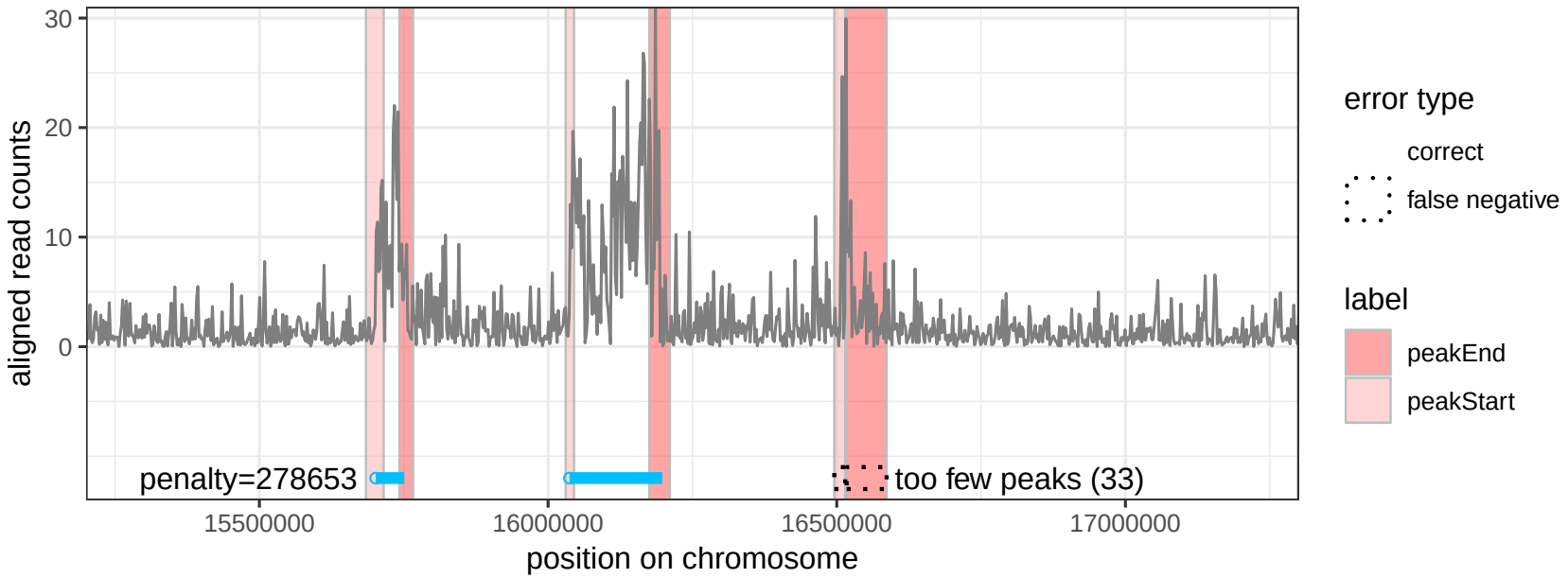


Labels used to determine optimal number of peaks



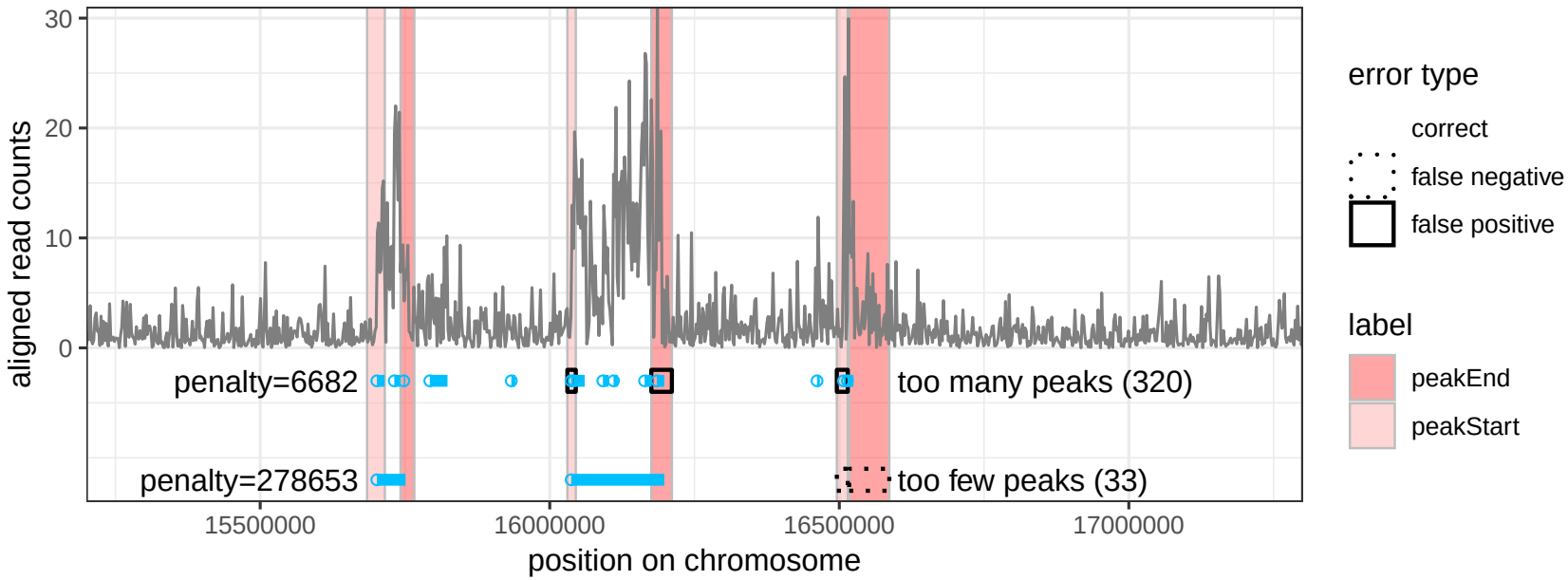
Visually labeled regions (H *et al.*, *Bioinformatics* 2017).

Labels used to determine optimal number of peaks



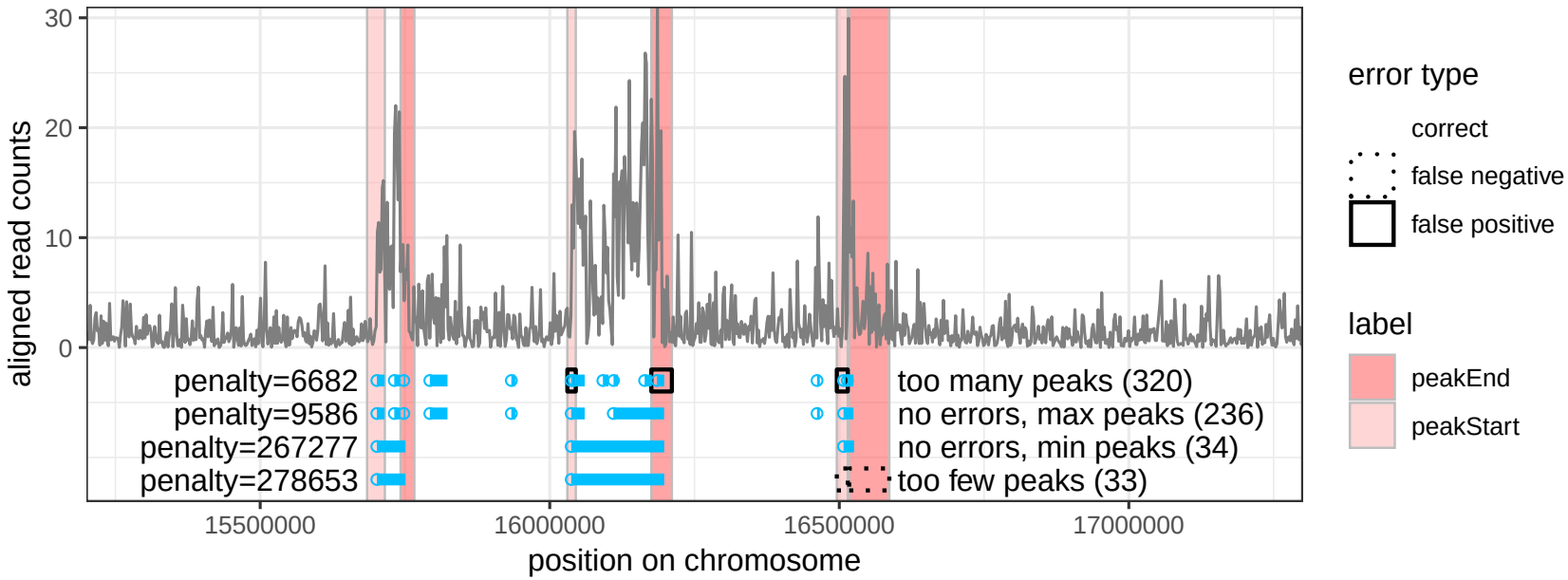
Penalty too large, too few peaks, 2 false negative labels.

Labels used to determine optimal number of peaks



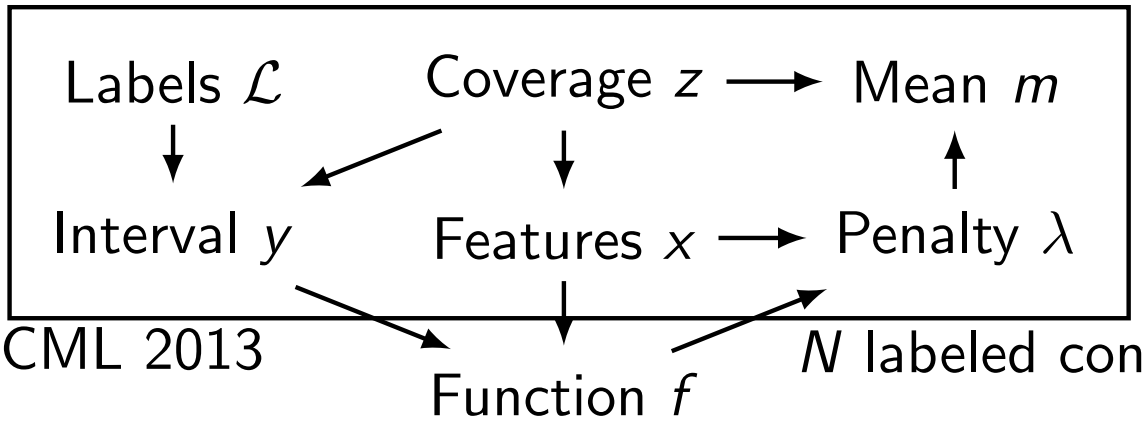
Penalty too small, too many peaks, 3 false positive labels.

Labels used to determine optimal number of peaks



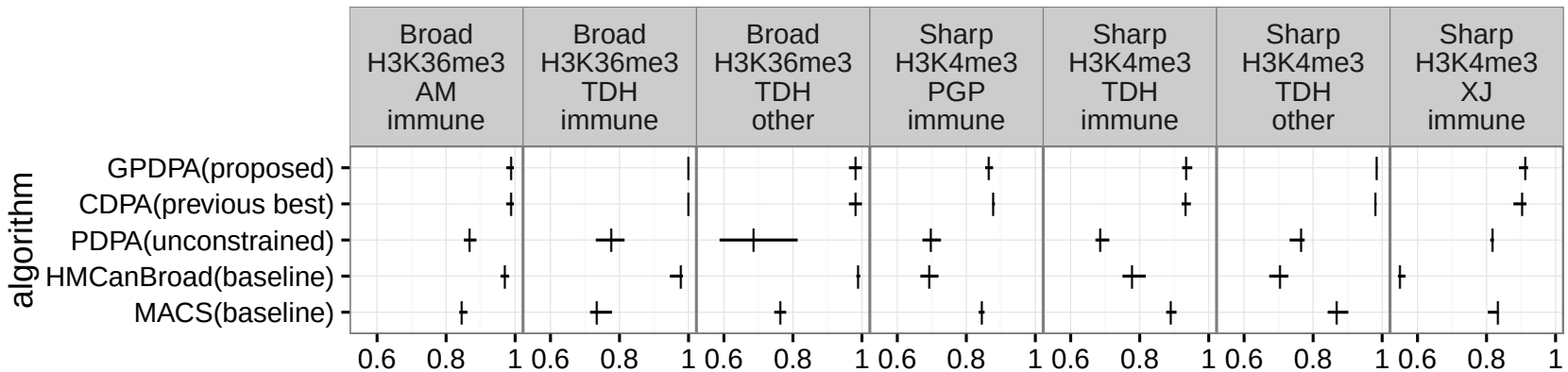
Models with 34–236 peaks have no label errors (midpoint=135).

Supervised changepoint penalty learning is highly accurate for both broad and sharp data/pattern types



H *et al.*, ICML 2013

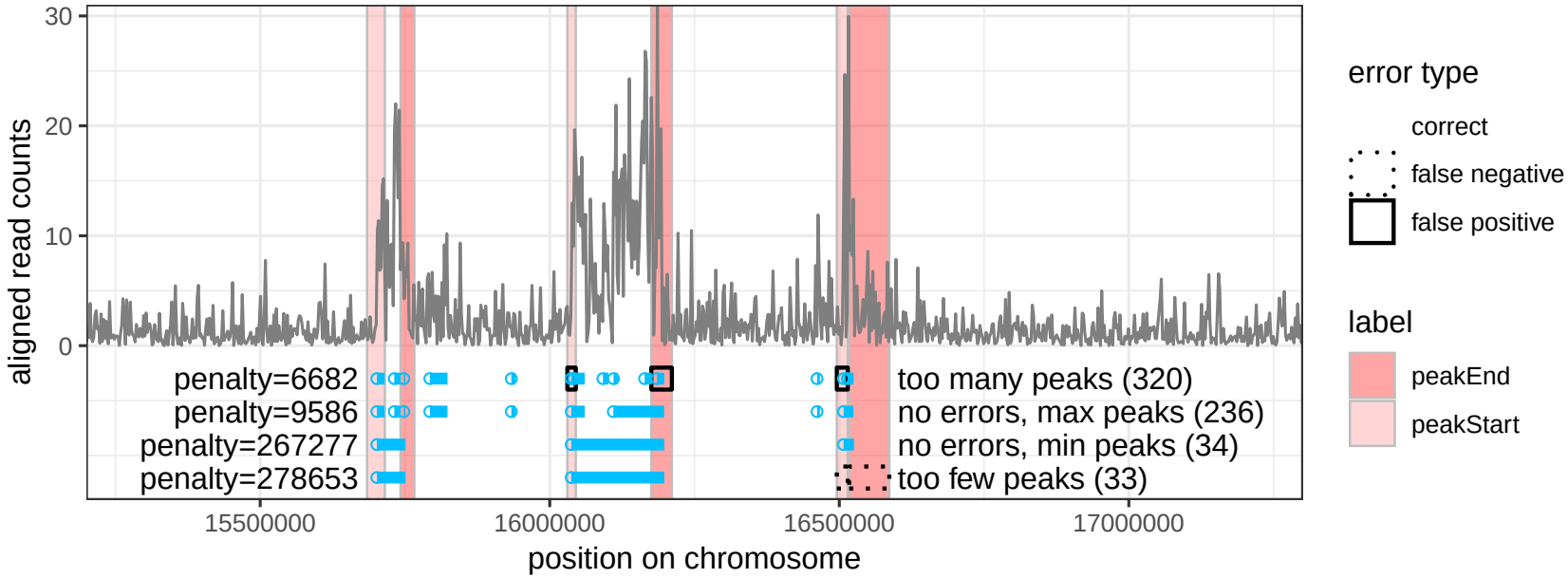
N labeled contigs/samples



H *et al.*, *arXiv* 2017.

- 4-fold cross-validation: train on 3/4 of labels, test on 1/4.
- Proposed up-down constrained algorithms (CDPA, GPDPA) state-of-the-art.

Conclusions / thanks!



- Proposed GFPOP algorithm computes global minimum of constrained changepoint problem in $O(n \log n)$ time, $O(n \log n)$ disk, $O(\log n)$ memory.
- Peak predictions are highly accurate in labeled genomic data with two patterns (H3K4me3 sharp, H3K36me3 broad).
- C++ code with R interface: `https://CRAN.R-project.org/package=PeakSegDisk`
- Contact me: toby.hocking@nau.edu