Change Point Detection in Genomic Data

Alan Chau   Ana Ignatieva
James Thornton   Lorenzo Pacchiardi

November 1, 2018

Long Chain DNA Sequences

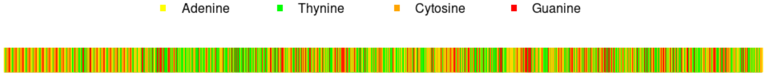


Figure 1: Plot of a DNA with 5000 base pair

- Seemingly long and messy spectrum of A, T, C and G. Can you observe any structure?

Long Chain DNA Sequences

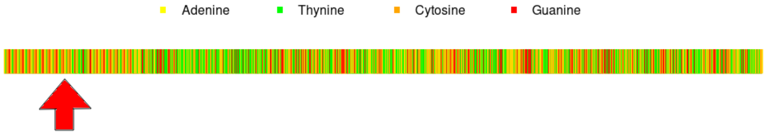


Figure 2: Plot of a DNA with 5000 base pair

- I found one after staring at the data for 2 days.

Long Chain DNA Sequences

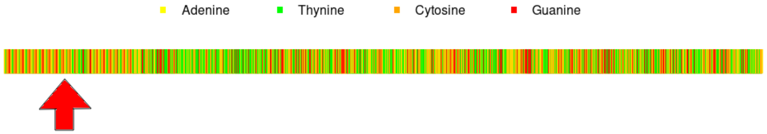


Figure 2: Plot of a DNA with 5000 base pair

- I found one after staring at the data for 2 days.
- Is there a systematic way of doing this structure discovery?

Long Chain DNA Sequences

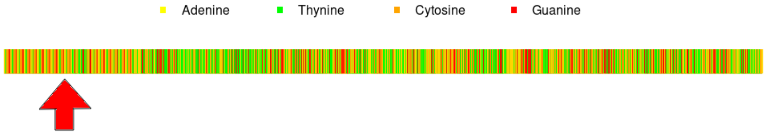


Figure 2: Plot of a DNA with 5000 base pair

- I found one after staring at the data for 2 days.
- Is there a systematic way of doing this structure discovery?
- In other words, can we detect change points systematically along this sequence?

Talk Overview

- Alan Chau - Exact algorithms for detections
- Lorenzo Pacchiardi - Maximum Likelihood and approximate search methods
- James Thornton - Bottom up approach and Bayesian methods
- Ana Ignatieva - Hidden Markov Model

Problem Statement

- Consider a sequence of data $\mathbf{y} = \{y_t\}_{t=1}^{T}$ where y_t can be continuous, discrete or categorical.
- Configure the set of possible change points as $\mathcal{T}_T := \{\tau : 0 = \tau_0 < \tau_1 < ... < \tau_K < \tau_{K+1} = T\}$
- We aim to minimise the following:

$$V(\tau, \mathbf{y}) = \min_{\tau \in \mathcal{T}_T} \sum_{k=0}^{K} \mathcal{C}(y_{\tau_k+1:\tau_{k+1}}) + \text{Pen}\{\tau, \beta\} \tag{1}$$

where $\mathcal{C}$ is some cost function measuring heterogeneity of the segment.

Optimal Partitioning

- One obvious choice is to pick the L_0 norm, βK .
- Set $F(T)$ as $V(\tau, \mathbf{y})$ restricted to the domain $\mathcal{T}_T$, we then realise:

$$
\begin{aligned}
F(T) &= \min_{\tau \in \mathcal{T}_T} \left\{ \sum_{k=0}^{K} \left[\mathcal{C}(y_{\tau_k+1:\tau_{k+1}}) + \beta \right] \right\} && (2) \\
&= \min_{t} \left\{ \sum_{k=0}^{K-1} \left[\mathcal{C}(y_{\tau_k+1:\tau_{k+1}}) + \beta \right] + \mathcal{C}(y_{t+1:T}) + \beta \right\} && (3) \\
&= \min_{t} \{ F(t) + \mathcal{C}(y_{t+1:T}) + \beta \} && (4)
\end{aligned}
$$

- We can then use a dynamic program and start from $F(1)$ and obtain an exact solution in $\mathcal{O}(T^2)$

Pruned Exact Linear Time (PELT)

- We get exactness but it is slow.

Pruned Exact Linear Time (PELT)

- We get exactness but it is slow.
- We can use pruning to reduce complexity and achieve approximate linear run time.

Pruned Exact Linear Time (PELT)

- We get exactness but it is slow.
- We can use pruning to reduce complexity and achieve approximate linear run time.

Theorem

We assume there exits a constant K such that for all $t < s < T$,

$$\mathcal{C}(y_{t+1:s}) + \mathcal{C}(y_{s+1:T}) + K \leq \mathcal{C}(y_{t+1:T})$$

Then if

$$F(t) + \mathcal{C}(y_{t+1:s}) + K \geq F(s)$$

holds, at a future time $T > s$, t can never be the optimal last change point prior to T.

Pruned Exact Linear Time (PELT)

Theorem

We assume there exits a constant K such that for all $t < s < T$,

$$\mathcal{C}(y_{t+1:s}) + \mathcal{C}(y_{s+1:T}) + K \leq \mathcal{C}(y_{t+1:T})$$

Then if

$$F(t) + \mathcal{C}(y_{t+1:s}) + K \geq F(s)$$

holds, at a future time $T > s$, t can never be the optimal last change point prior to T.

- Intuitively speaking, if the condition is met, then s is always a better changepoint than t, thus we don't have to consider that anymore in future steps.

Pruned Exact Linear Time

```
[H] Initialise: Let $T$ be the number of data, set $F(0) = -\beta$,
$cp(0) = NULL$ and $R_1 = \{0\}$
Iterate: For $\tau^* = 1, ..., T$
  1  Compute $F(\tau^*) = \min_{\tau \in R_{\tau^*}} [F(\tau) + \mathcal{C}(y_{\tau+1} : \tau^*) + \beta]$
  2  Let $\tau' = \arg\min_{\tau \in R_{\tau^*}} [F(\tau) + \mathcal{C}(y_{\tau+1:\tau^*}) + \beta]$
  3  set $cp(\tau^*) = [cp(\tau'), \tau']$
  4  set $R_{\tau^*+1} = \{\tau \in R_{\tau^*} \cap \tau^* : F(\tau) + \mathcal{C}(y_{\tau+1:\tau^*}) + K \leq F(\tau^*)\}$
Output:  A vector of change points in $cp(T)$
```

Application

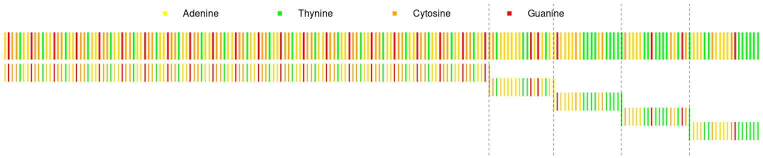
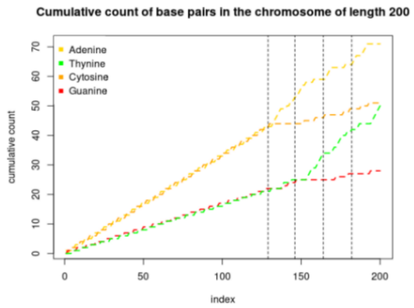


Figure 3: PELT applied to a sequence of DNA with length 200

Application



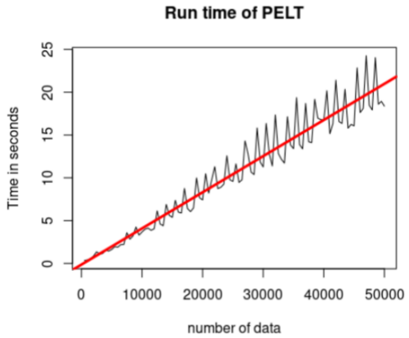


Figure 4: (left) PELT applied to the first 200 DNA sequence. (right) Run time analysis on PELT

Problem with large sequences

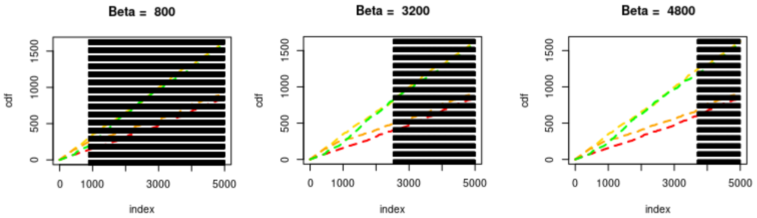


Figure 5: Limitation on the PELT method with large sequence

Pruned Exact Linear Time with Modified L_0 penalty

To tackle this concentration of change points, we introduce a modified penalty for the PELT algorithm to control the spread,

$$\text{Pen}_{mL_0}(\tau, \beta) = 3K\beta \log(T) + \beta \log(\beta) \sum_{k=0}^{K} \log(\frac{\tau_{k+1} - \tau_k}{T}) \qquad (5)$$

This can be easily incorporate to PELT by setting:

$$\mathcal{C}'(y_{a+1:b}) \leftarrow \mathcal{C}(y_{a+1:b}) + \beta \log(\beta) \log\left(\frac{b-a}{T}\right) \quad \beta' \leftarrow 3\beta \log(T) \qquad (6)$$

Results

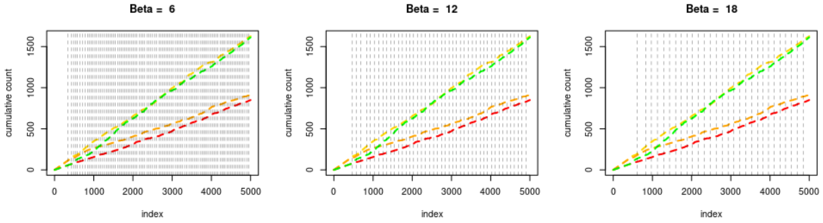


Figure 6: More evenly spread change points

MLE of transition matrix

Consider a standard Markov Chain. MLE for the transition probability:

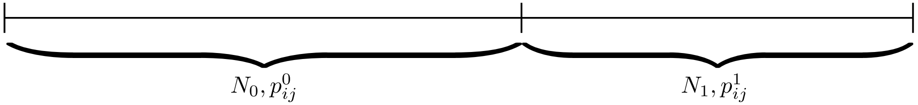
$$\hat{p}_{ij} = \frac{n_{ij}}{\sum_{j=1}^{m} n_{ij}} = \frac{n_{ij}}{n_i} \longrightarrow p_{ij}$$

n_{ij}: number of observed transitions $i \rightarrow j$

MLE of transition matrix

If the inferred segment is actually the composition of two MCs with different transition probabilities, we get:

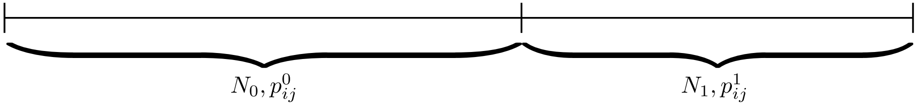
$$\hat{p}_{ij} \longrightarrow \frac{N_0 p_i^0 p_{ij}^0 + N_1 p_i^1 p_{ij}^1}{N_0 p_i^0 + N_1 p_i^1}, \quad N_0,\ N_1 \to \infty$$



MLE of transition matrix

If the inferred segment is actually the composition of two MCs with different transition probabilities, we get:

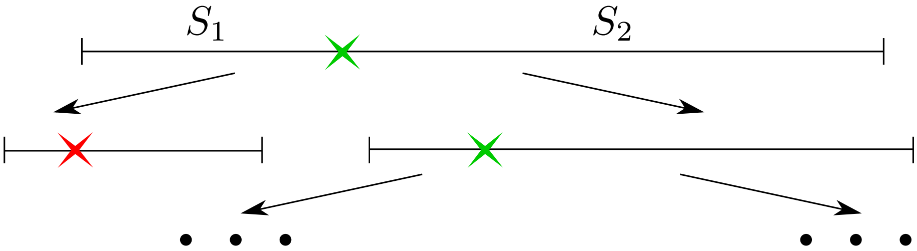
$$\hat{p}_{ij} \longrightarrow \frac{N_0 p_i^0 p_{ij}^0 + N_1 p_i^1 p_{ij}^1}{N_0 p_i^0 + N_1 p_i^1}, \quad N_0,\ N_1 \to \infty$$



$\hat{p}_{ij} \to p_{ij}^0 \iff N_1/N_0 \to 0$. This is usually not guaranteed in change point detection algorithms.

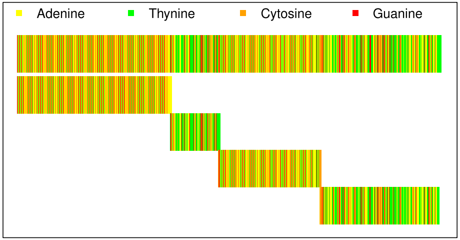
Markov model of genome segmentation (?)

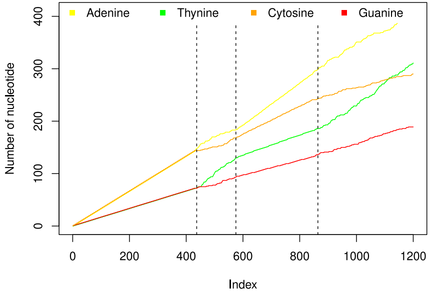
- Recursive binary segmentation procedure.
- Maximizes Jensen Shannon divergence between the two subsequences $S_1,\ S_2$: $D_{JS}(S_1, S_2) = H(S) - \pi_1 H(S_1) - \pi_2 H(S_2)$
- Each segmentation is accepted if it satisfies the BIC criterion $\Delta\mathcal{C}_{BIC} < 0 \iff 2ND_{JS}(S_1, S_2) > 16 \ln N$



Experiment on the genomic data

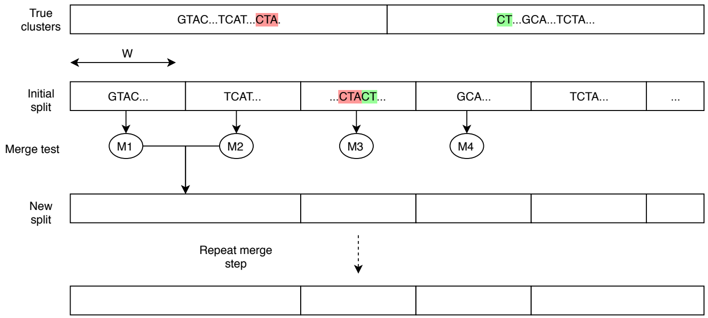
- Time complexity $\mathcal{O}(N \log_2 K)$, K: number of change points.
- Algorithm was run on the first 5 million nucleotides. Segments for the first 1500 are reported in picture.





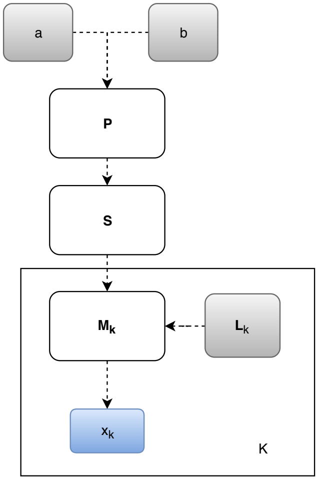
Grid-Merge Heuristic

Split chain index $\rightarrow$ Merge on distance between MLE transition matrices $\rightarrow \ldots$



- Applicable for large data-sets
- Intuitive interpretation
- Approximation error

Bayesian Change Point Detection



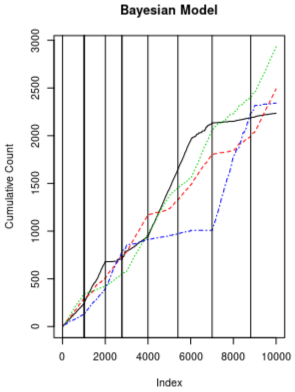
Generative Model:

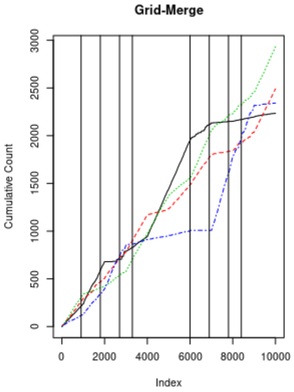
- $P \sim BetaDiag(a, b)$
- $\mathcal{S} \sim MarkovChain(P)$ where $s_1 = 1$
- $M_k | \mathcal{S} \sim DirMat(\boldsymbol{\lambda})\ \forall k \in [1 : K]$
- $x^k_{1:n_k} | \mathcal{S}, M_k \sim MarkovChain(M_k)$

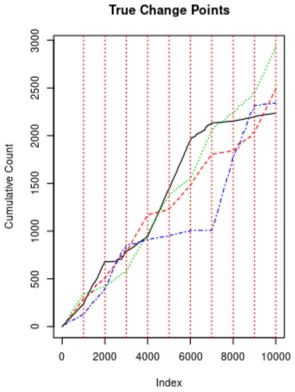
Implementation:

- Conjugate
- MCMC
  - Blocked Gibbs Sampling
  - Simulated annealing

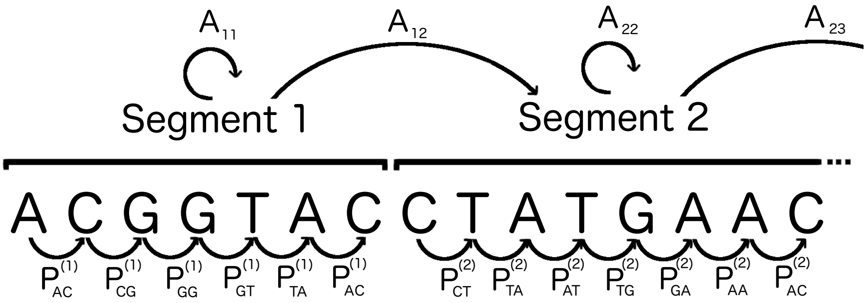
Results on Simulated Data







HMM



- K hidden states, transition matrix A
- Emissions $y_1, \ldots, y_T$, transitions $P_{ij}^{(s)}$
- Changepoints = hidden state transitions
- Used for genome data (**??**)

Hidden state transitions

For changepoints want structure of A:

$$A = \begin{pmatrix} 1-\lambda_1 & \lambda_1 & 0 & \dots & 0 \\ 0 & 1-\lambda_2 & \lambda_2 & \dots & 0 \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1-\lambda_{K-1} & \lambda_{K-1} \\ 0 & \dots & 0 & 0 & 1 \end{pmatrix}$$

- Small probability of transitioning to next state
- Can't revisit previous states
- Can't jump ahead

Parameter fitting

Baum-Welch (**?**):

- Version of E-M
- Need to incorporate Markov dependency of emissions
- Iteratively compute $\mathbb{P}(S_t|y_1, \ldots, y_T)$
- Expected number of times (ij) appears in state s

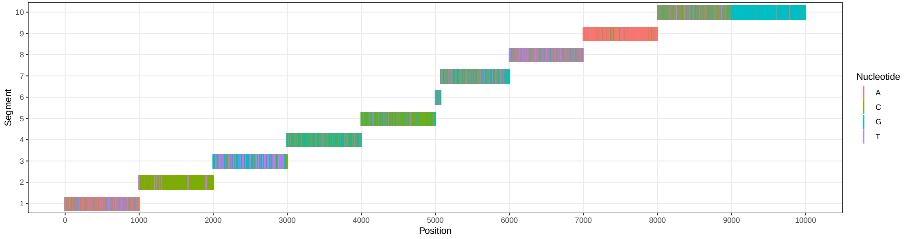
  $\implies$ update $P_{ij}^{(s)}$
- Expected number of state transitions $i \to j$

  $\implies$ update A_{ij}

Decode the HMM:

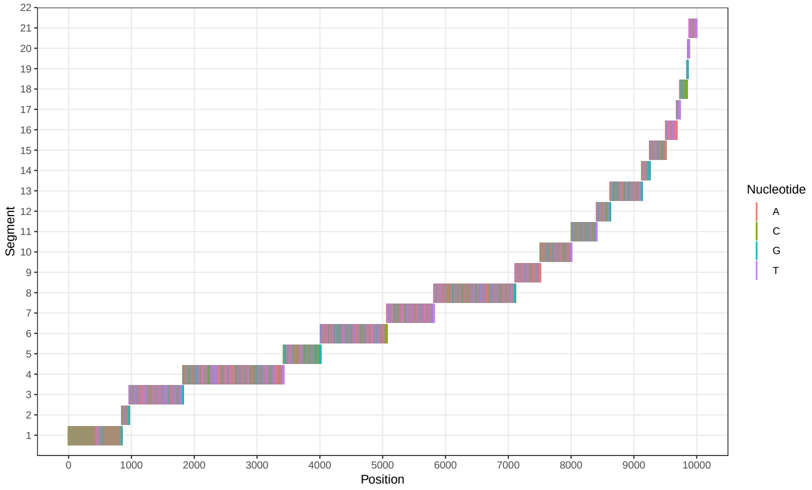
- Viterbi: get most likely path through states
- Posterior $\mathbb{P}(S_t|y_1, \ldots, y_T)$

Simulated data



- Actual changepoints at every 1,000
- Almost!

Real data



- Initial guess for max number of changepoints = 50
- HMM finds 22

Conclusions

- Flexible:
    - number of changepoints
    - incorporate information into structure of A
- General class of models
- Relatively slow