

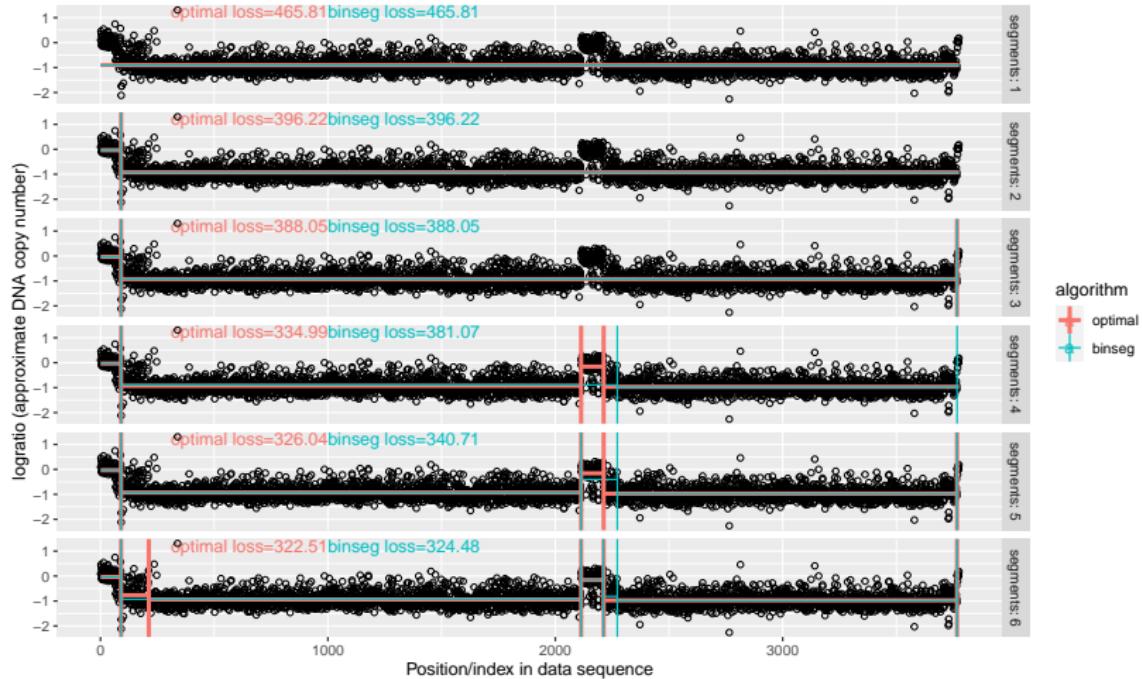
Optimal segmentation

Toby Dylan Hocking

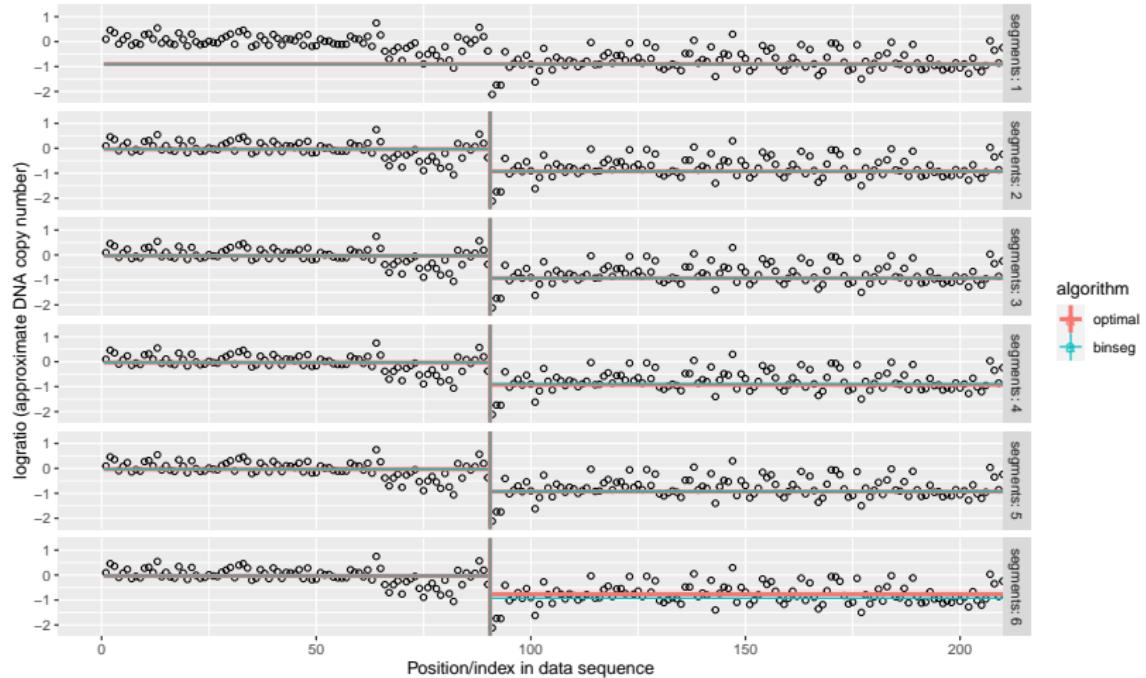
Motivation for optimal detection

- ▶ Binary segmentation is a greedy algorithm, so sometimes it chooses a changepoint which is sub-optimal for a larger model size.
- ▶ Is it possible to compute the changepoints and segments which are optimal for each model size? Yes, with dynamic programming.
- ▶ Is it desirable to compute the optimal changepoints and segments? Yes, see examples in next slides.

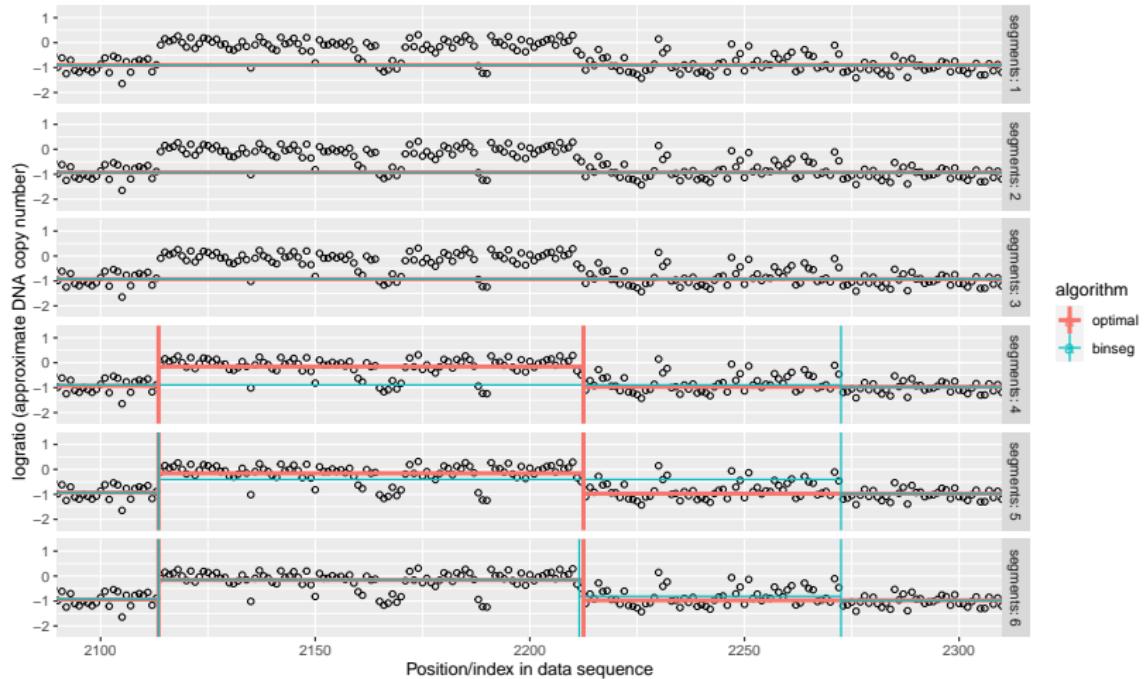
Example 1: stuck with sub-optimal change



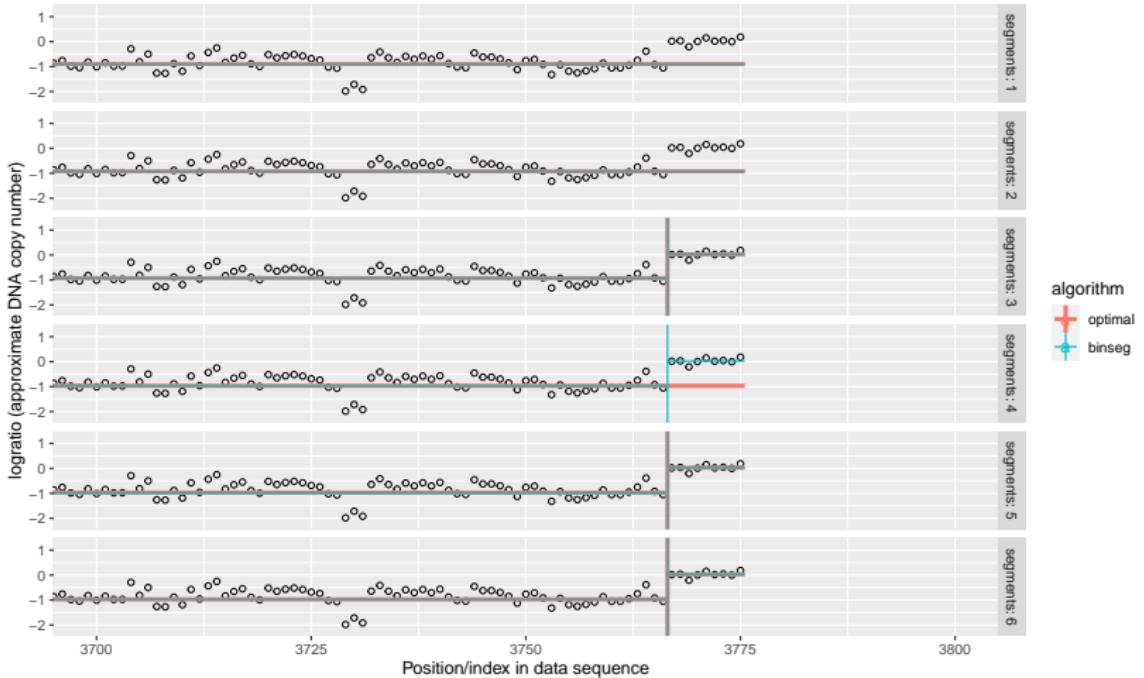
Zoom to start



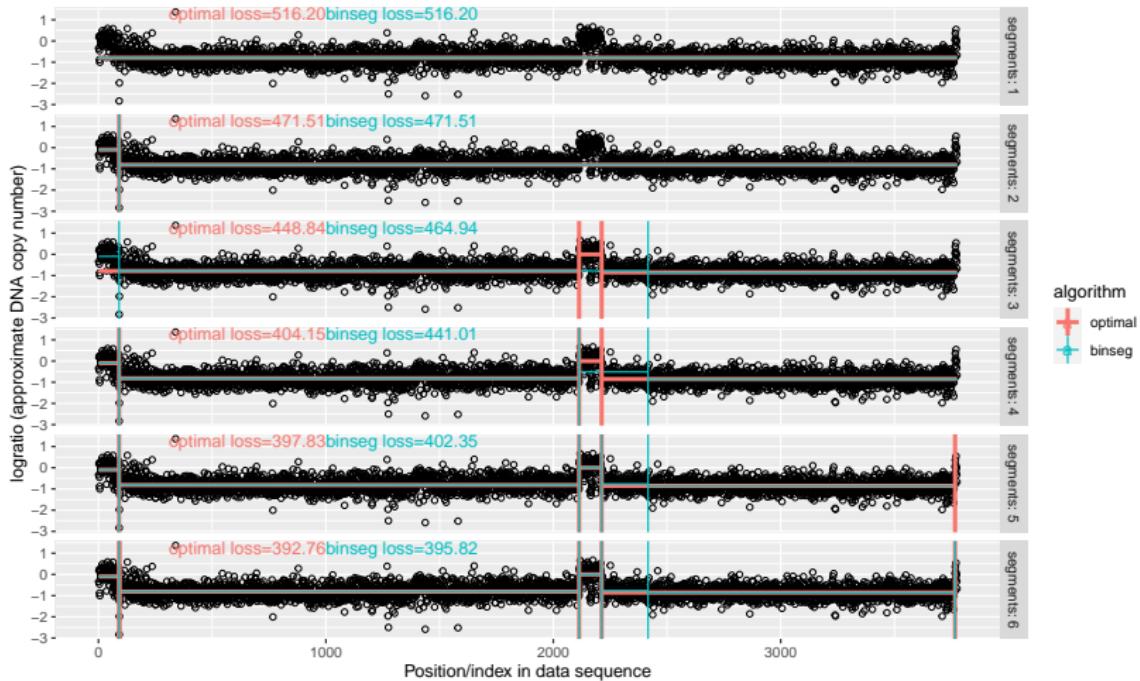
Zoom to center



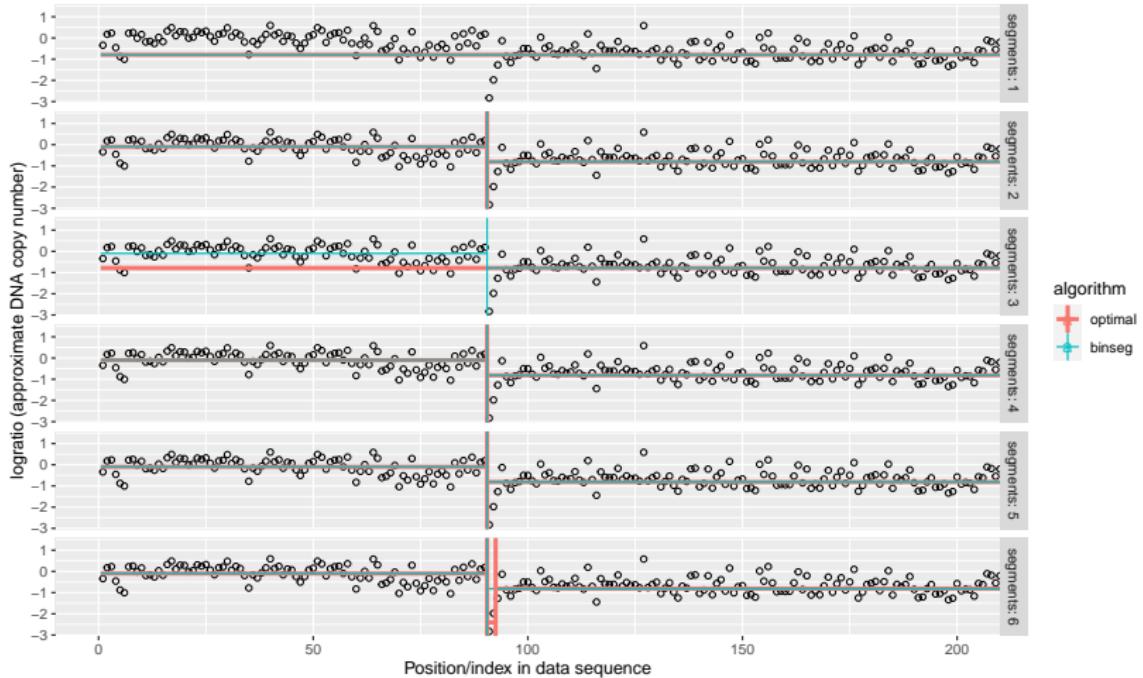
Zoom to end



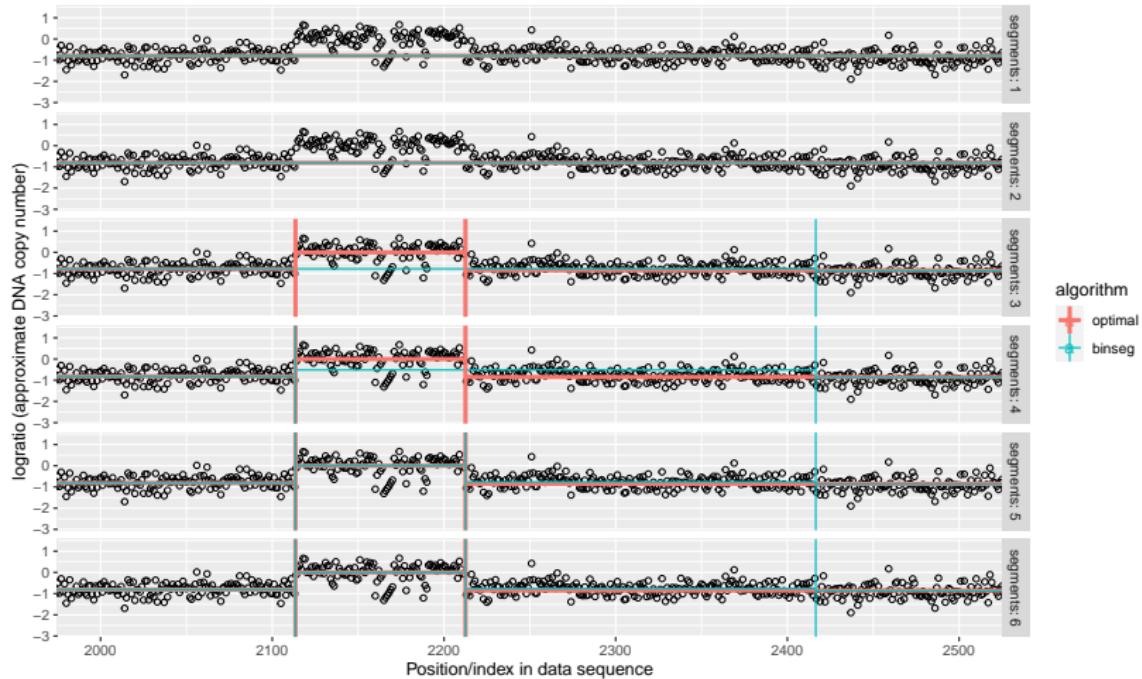
Example 2: missing a small change down



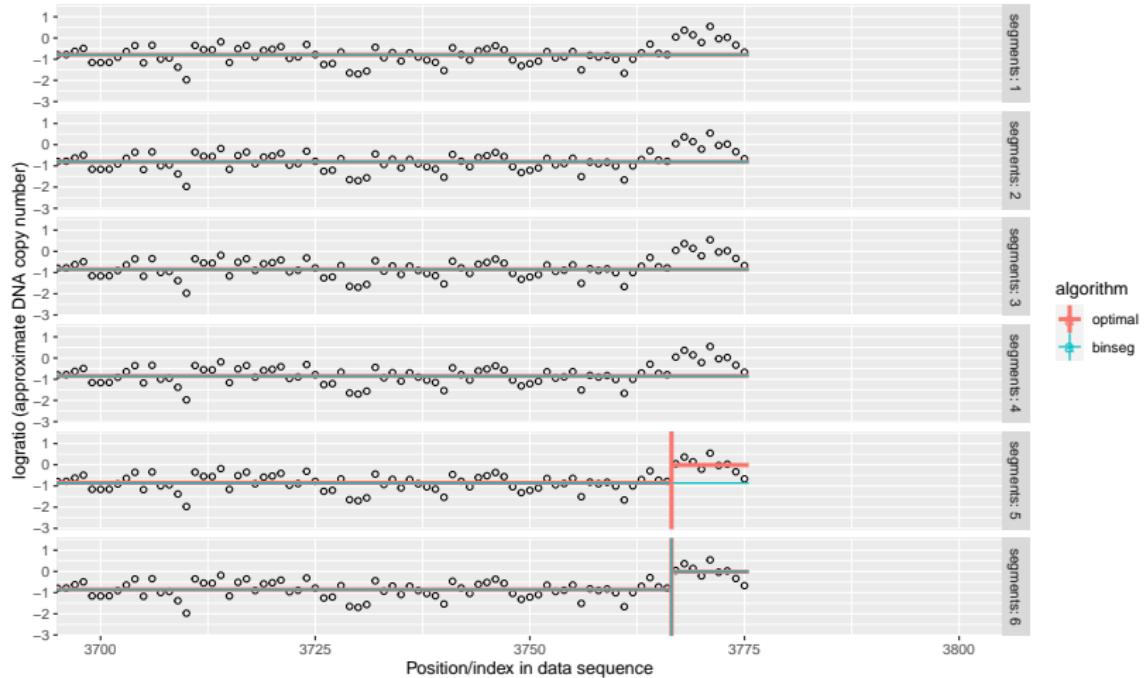
Zoom to start



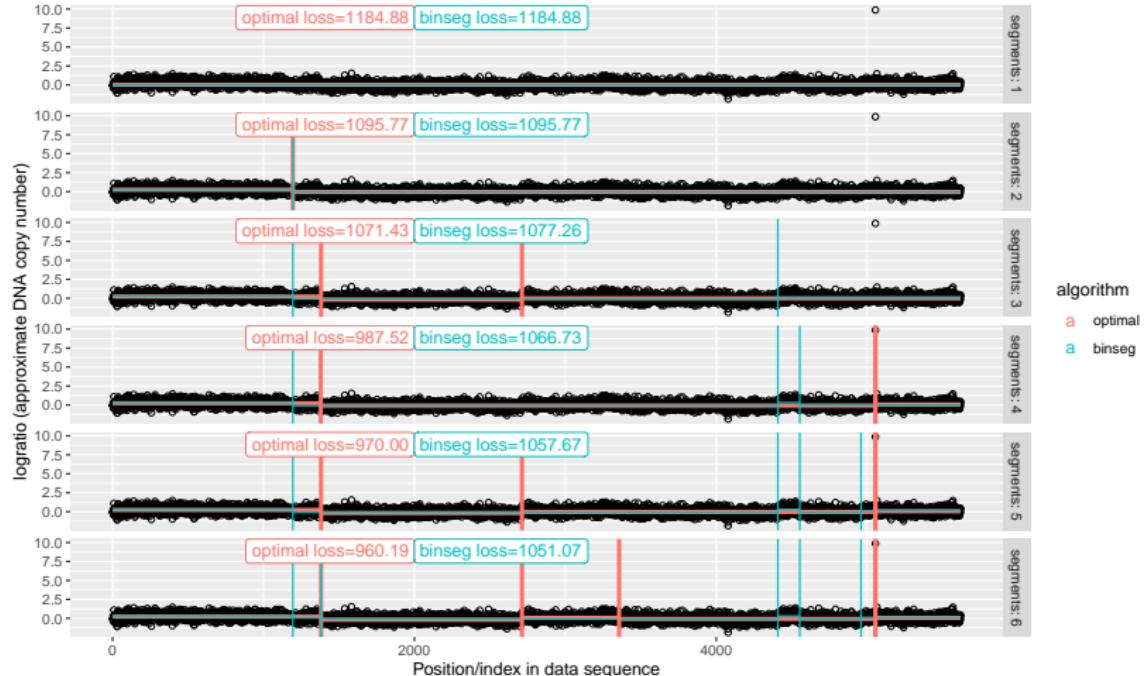
Zoom to center



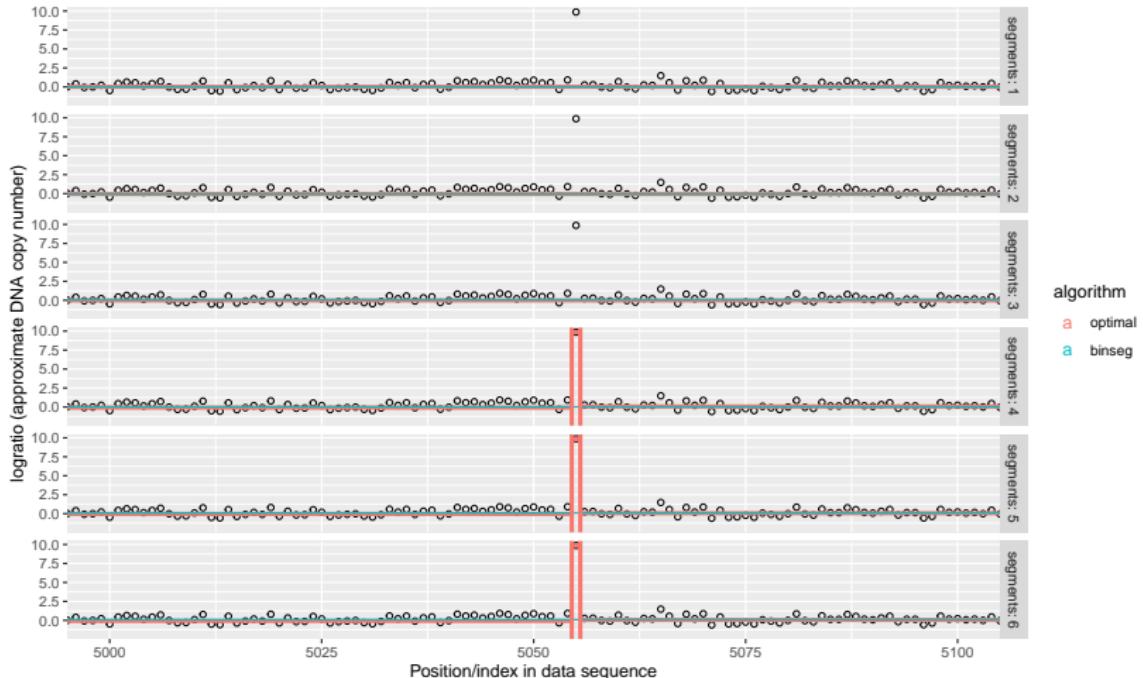
Zoom to end



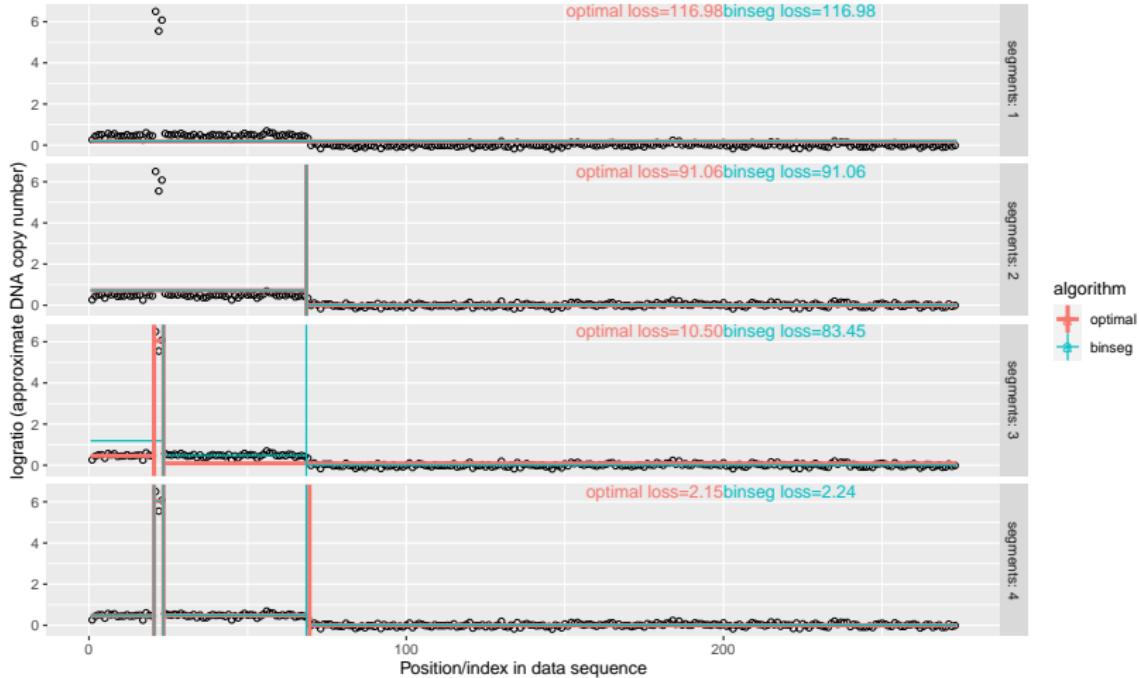
Example 3: difficult to detect an outlier



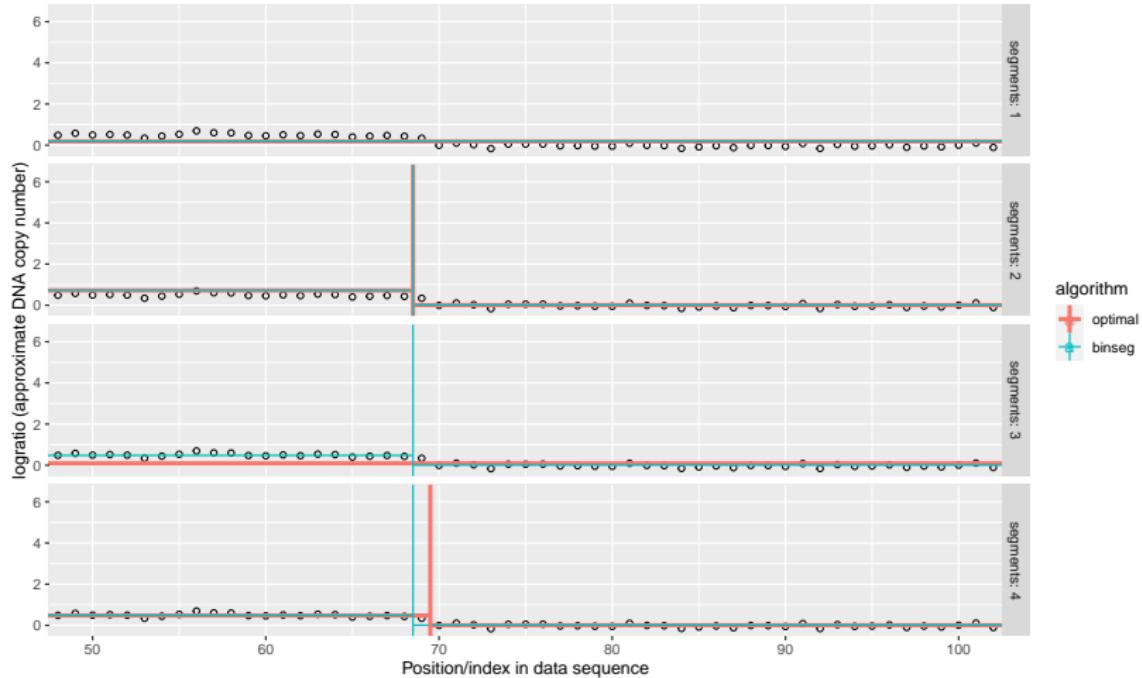
Zoom X axis to outlier



Example 4: stuck with sub-optimal change



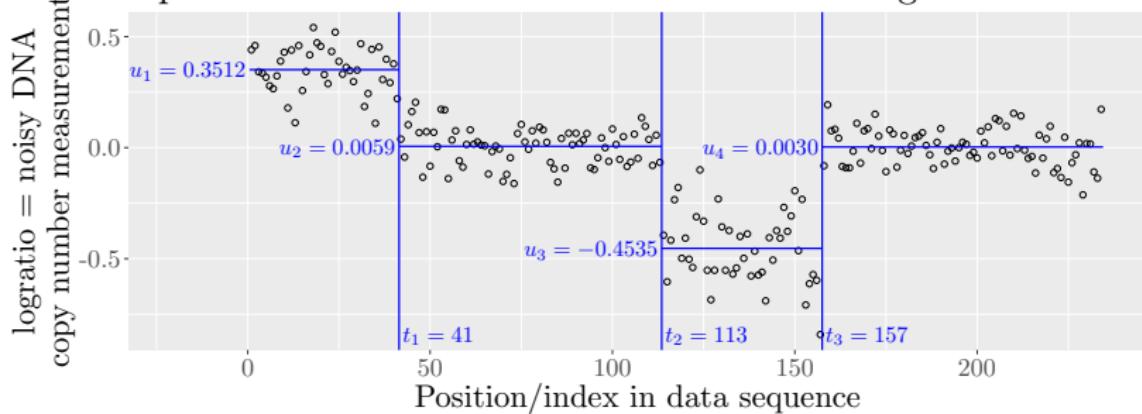
Zoom to changepoint



Dynamic programming for optimal changepoint detection

- ▶ We have n data z_1, \dots, z_n .
- ▶ Fix the number of segments $S \in \{1, 2, \dots, n\}$.
- ▶ Optimization variables: $S - 1$ changepoints $t_1 < \dots < t_{S-1}$ and S segment means $u_1, \dots, u_S \in \mathbb{R}$ ($t_0 = 0$, $t_S = n$).
- ▶ Statistical model: for every segment $s \in \{1, \dots, S\}$,
 $z_i \stackrel{\text{iid}}{\sim} N(u_s, \sigma^2)$ for every data point $i \in (t_{s-1}, t_s]$ implies square loss function $\ell(u_s, z_i) = (u_s - z_i)^2$ to minimize.

Optimal model for $n = 234$ data and $S = 4$ segments



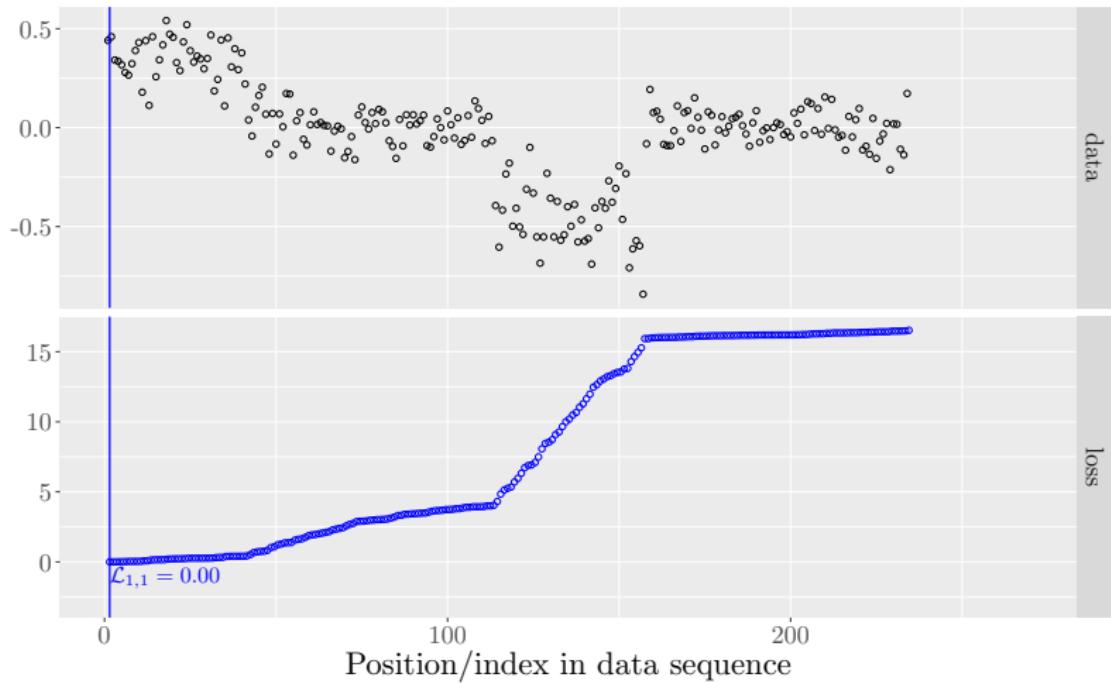
Maximum likelihood inference for S segments and n data

The best loss for S segments and n data is

$$\begin{aligned} \mathcal{L}_{S,n} &= \min_{\substack{\mathbf{u} \in \mathbb{R}^S \\ 0 = t_0 < t_1 < \dots < t_{S-1} < t_S = n}} \\ &\quad \sum_{s=1}^S \sum_{i=t_{s-1}+1}^{t_s} \ell(u_s, z_i) \\ &= \min_{t_{S-1}} \underbrace{\min_{\substack{\mathbf{u}_1, \dots, \mathbf{u}_{S-1} \\ t_1 < \dots < t_{S-2}}} \sum_{s=1}^{S-1} \sum_{i=t_{s-1}+1}^{t_s} \ell(u_s, z_i)}_{\mathcal{L}_{S-1, t_{S-1}}} \end{aligned}$$

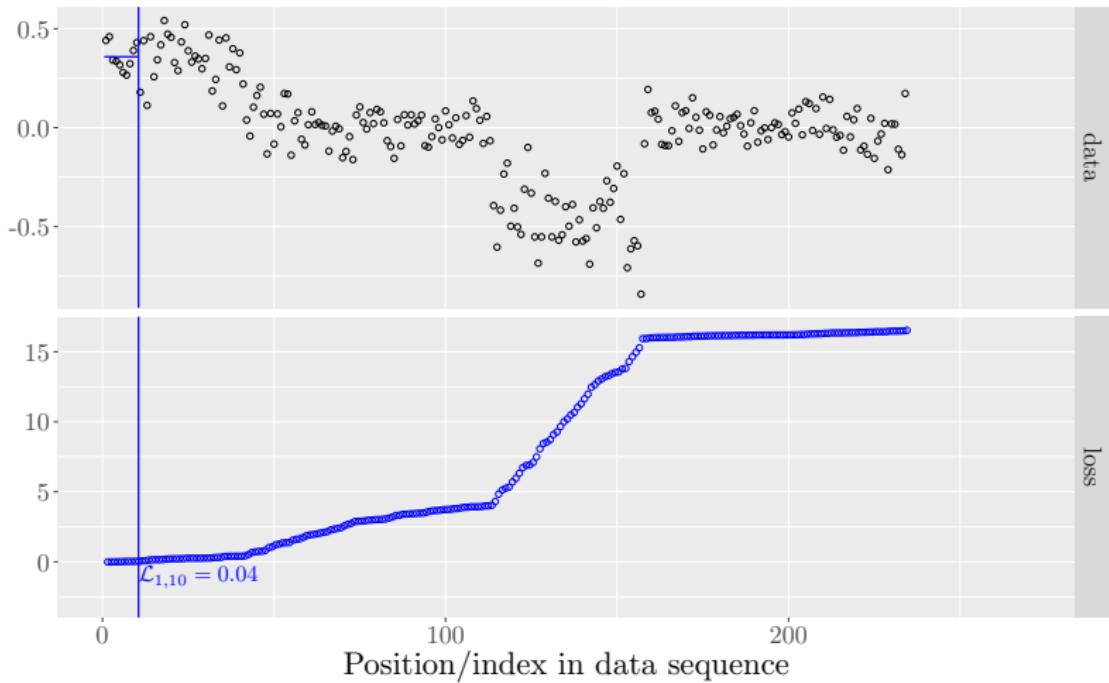
Initialization of dynamic programming

$$\mathcal{L}_{1,t} = c_{(0,t]}$$



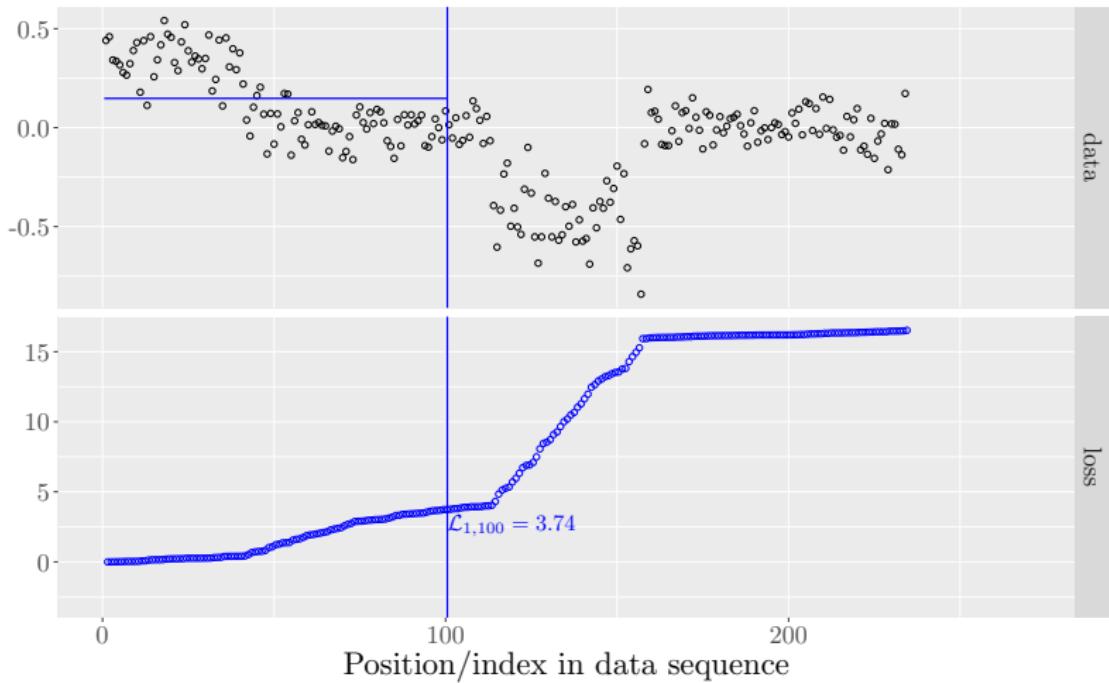
Initialization of dynamic programming

$$\mathcal{L}_{1,t} = c_{(0,t]}$$



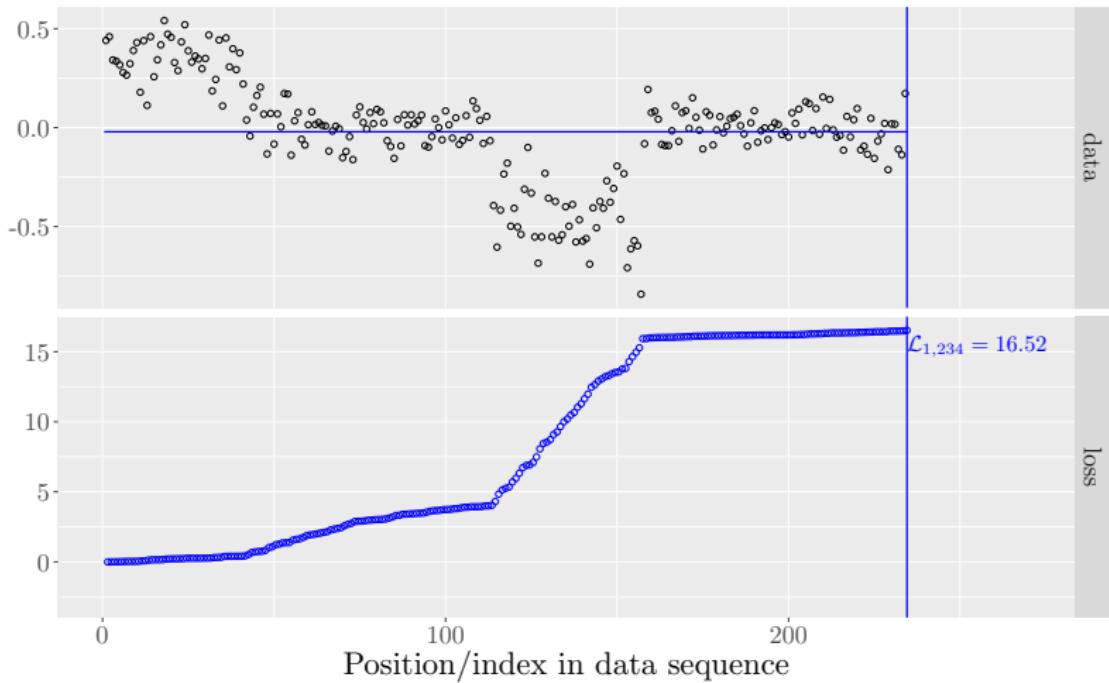
Initialization of dynamic programming

$$\mathcal{L}_{1,t} = c_{(0,t]}$$



Initialization of dynamic programming

$$\mathcal{L}_{1,t} = c_{(0,t]}$$



Efficient computation of c values

The best square loss up to t is

$$c_{(0,t]} = \min_u \sum_{i=1}^t (u - z_i)^2 = t \hat{u}_t^2 - 2\hat{u}_t \sum_{i=1}^t z_i + \sum_{i=1}^t z_i^2.$$

We can use cumulative sum to compute S_t for all t in linear $O(t)$ time, $S_t = \sum_{i=1}^t z_i$.

As can all best means up to t , $\hat{u}_t = S_t/t$.

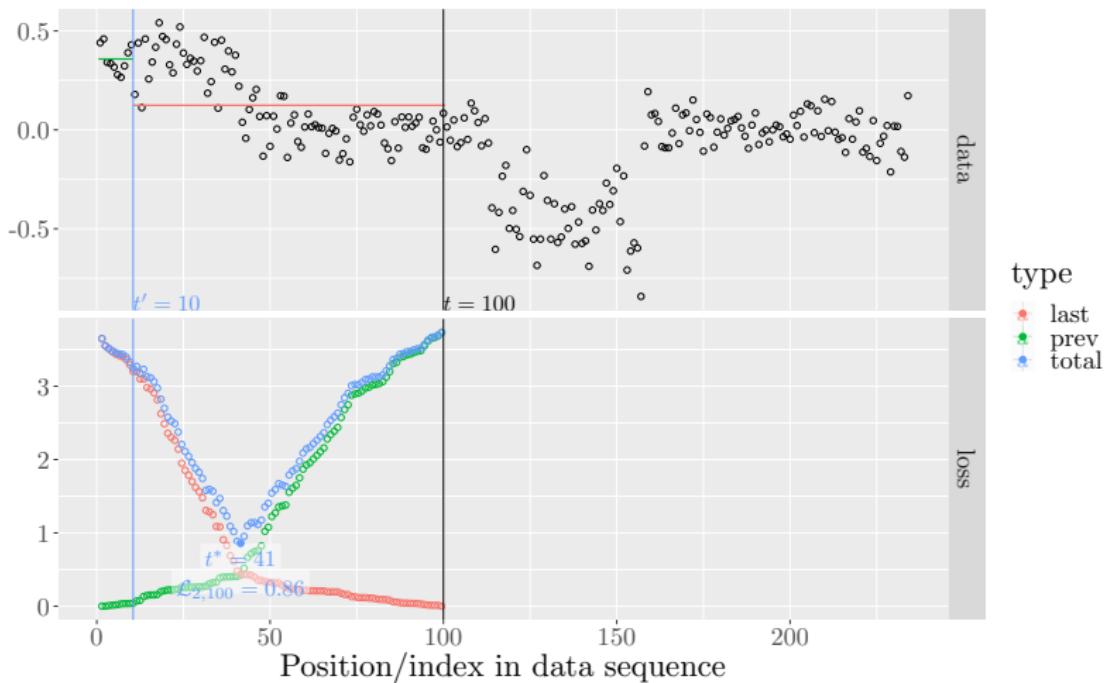
As can cumulative sum of squares, $Q_t = \sum_{i=1}^t z_i^2$.

All best loss values up to t can also be computed in linear time,

$$t \hat{u}_t^2 - 2\hat{u}_t \sum_{i=1}^t z_i + \sum_{i=1}^t z_i^2 = t(S_t/t)^2 - 2(S_t/t)(S_t) + Q_t = Q_t - S_t^2/t.$$

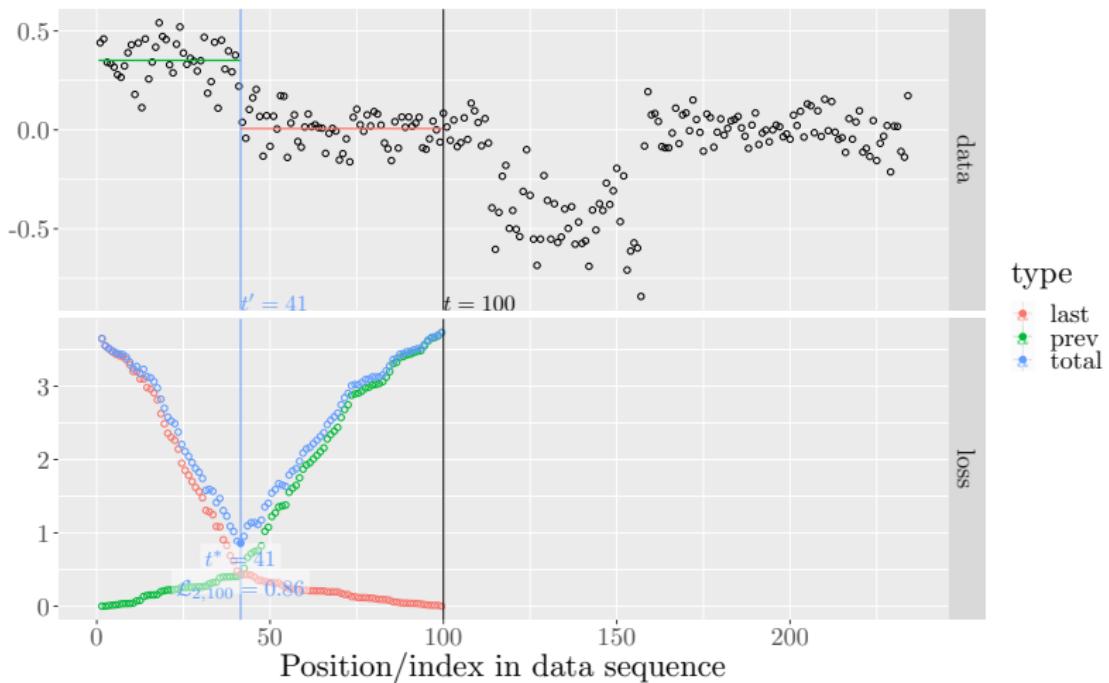
Second step of dynamic programming

$$\mathcal{L}_{2,t} = \min_{t' < t} \mathcal{L}_{1,t'} + c_{(t',t]}$$



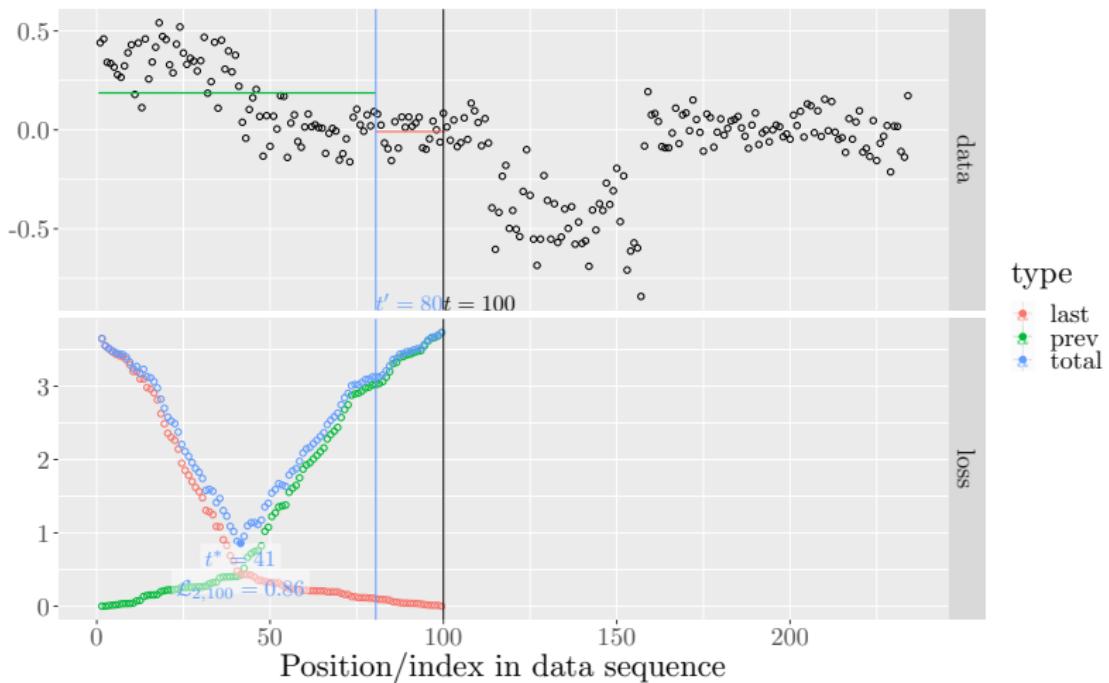
Second step of dynamic programming

$$\mathcal{L}_{2,t} = \min_{t' < t} \mathcal{L}_{1,t'} + c_{(t',t]}$$



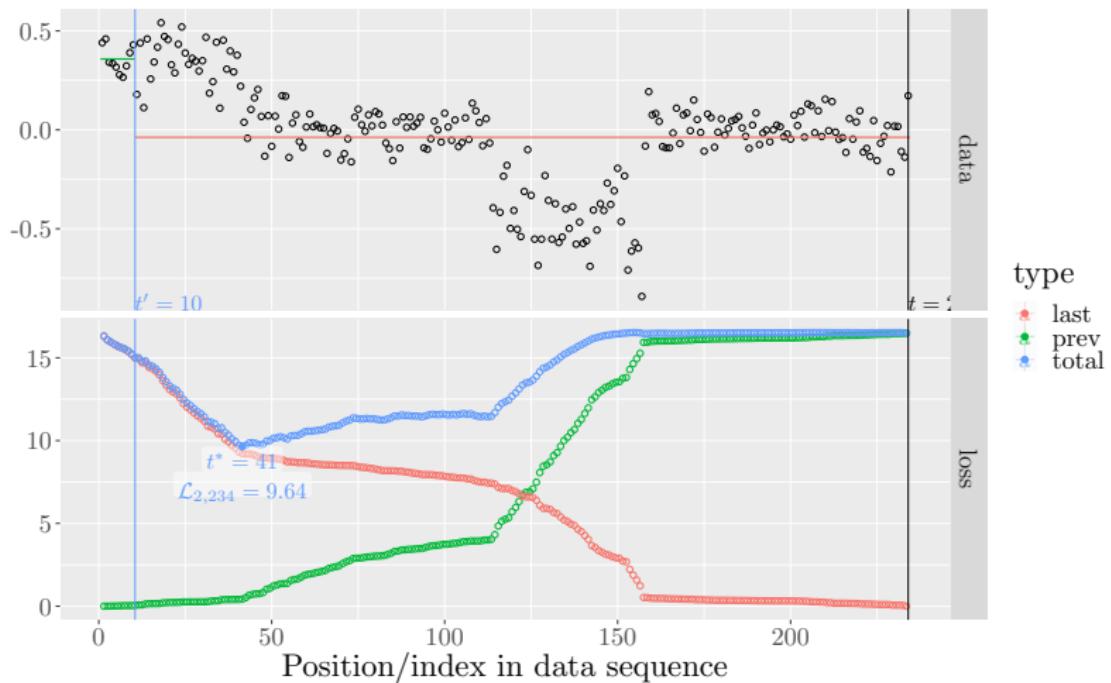
Second step of dynamic programming

$$\mathcal{L}_{2,t} = \min_{t' < t} \mathcal{L}_{1,t'} + c_{(t',t]}$$



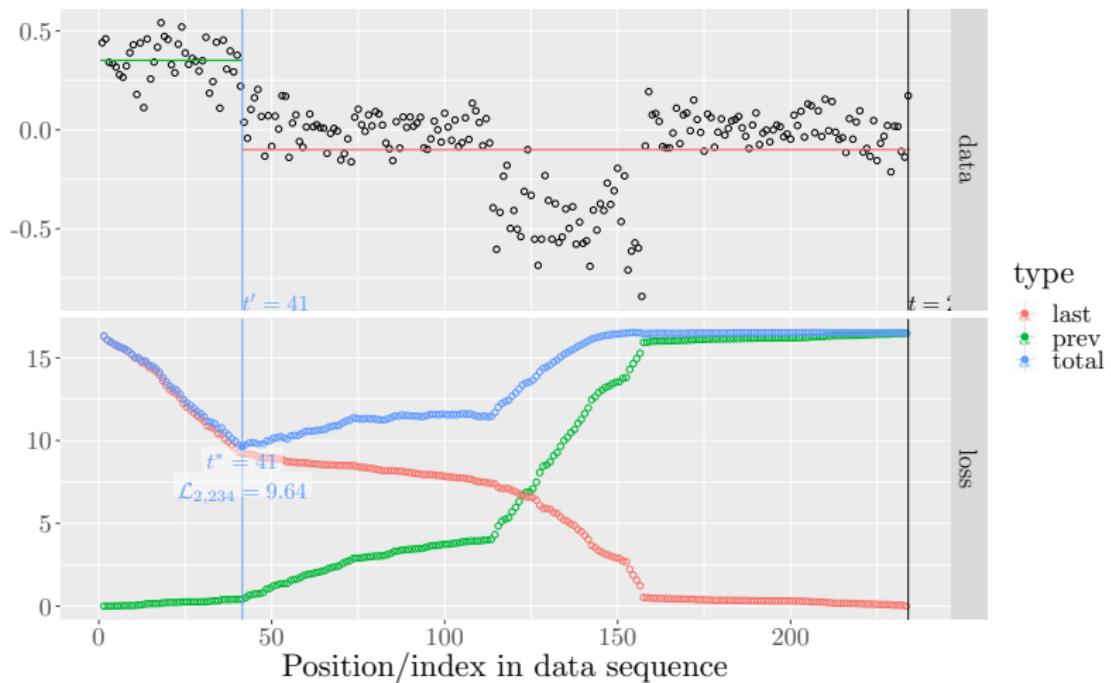
Second step of dynamic programming

$$\mathcal{L}_{2,t} = \min_{t' < t} \mathcal{L}_{1,t'} + c_{(t',t]}$$



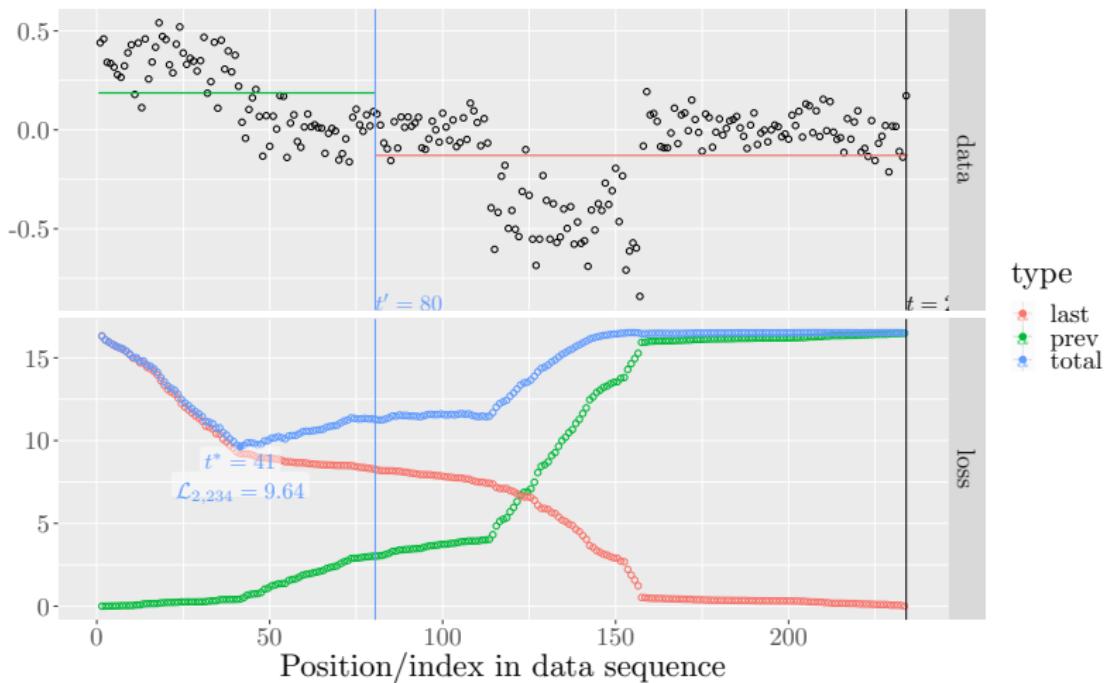
Second step of dynamic programming

$$\mathcal{L}_{2,t} = \min_{t' < t} \mathcal{L}_{1,t'} + c_{(t',t]}$$



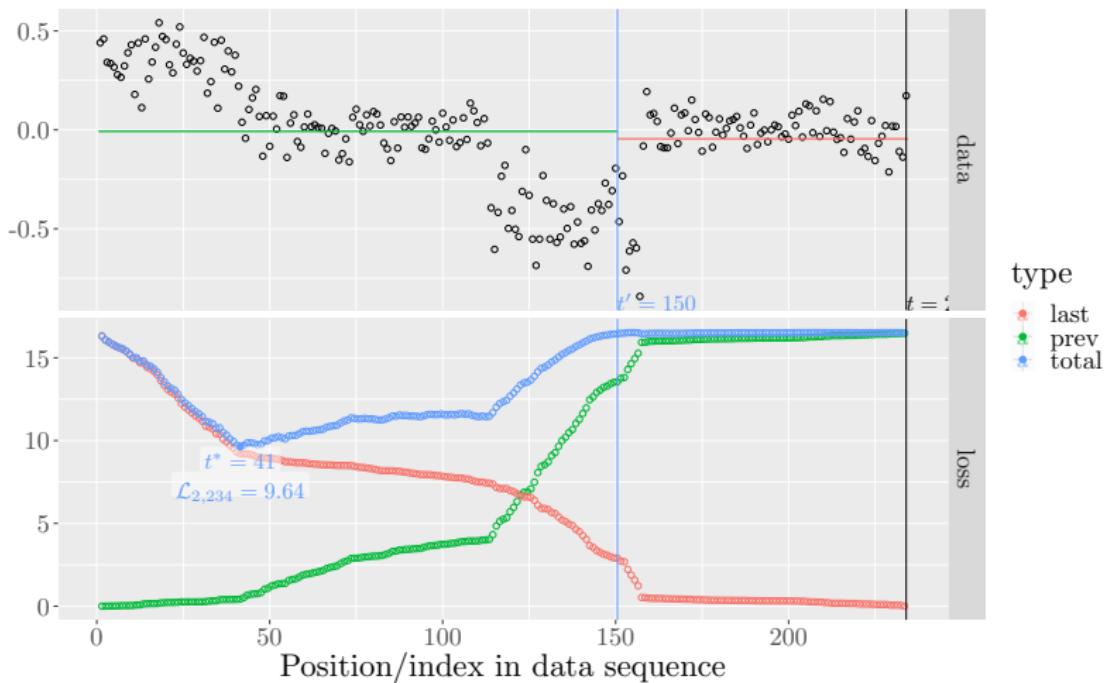
Second step of dynamic programming

$$\mathcal{L}_{2,t} = \min_{t' < t} \mathcal{L}_{1,t'} + c_{(t',t]}$$



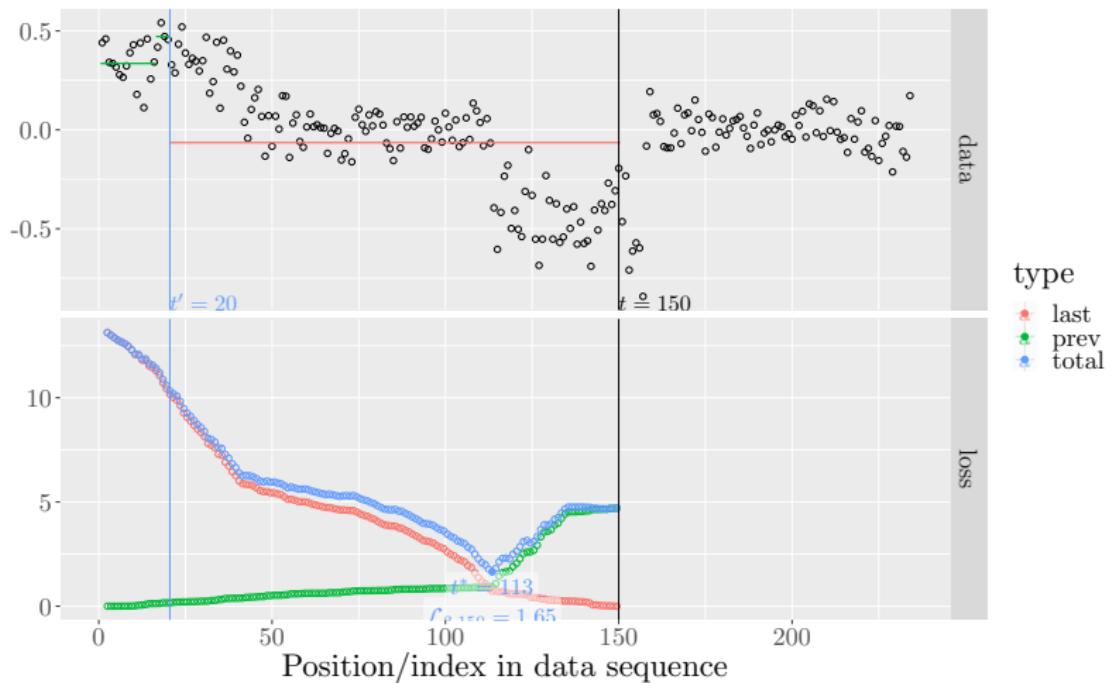
Second step of dynamic programming

$$\mathcal{L}_{2,t} = \min_{t' < t} \mathcal{L}_{1,t'} + c_{(t',t]}$$



Second step of dynamic programming

$$\mathcal{L}_{3,t} = \min_{t' < t} \mathcal{L}_{2,t'} + c_{(t',t]}$$



Second step of dynamic programming

$$\mathcal{L}_{3,t} = \min_{t' < t} \mathcal{L}_{2,t'} + c_{(t',t]}$$

