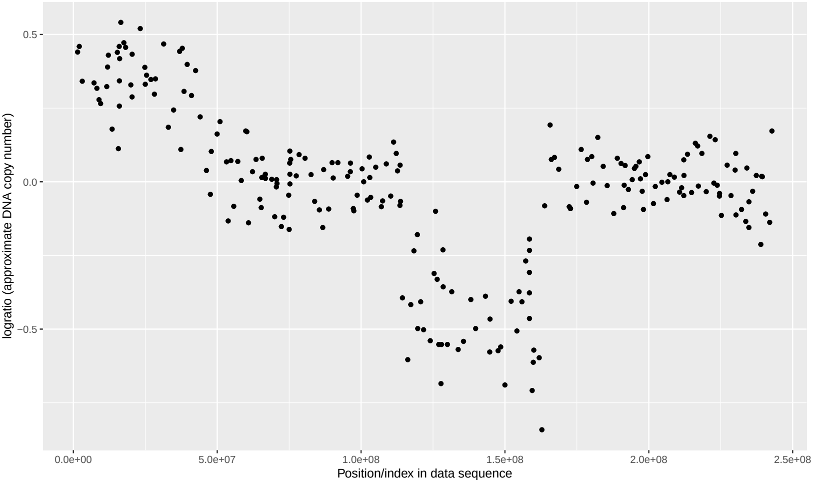
Hidden Markov Models

Toby Dylan Hocking

Background: detecting abrupt changes is important

Example from cancer diagnosis: breakpoints are associated with aggressive disease in neuroblastoma.

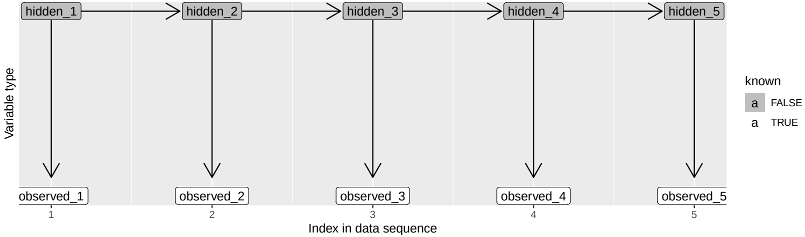


Motivation for Hidden Markov Models (HMMs)

- Sometimes we have an interpretation / expectation of what the segments/clusters mean.
- For example in DNA copy number data the logratio=0 means normal copy number (two copies – one from each parent), whereas higher logratio values indicate gain/amplification and lower values indicate loss/deletion.

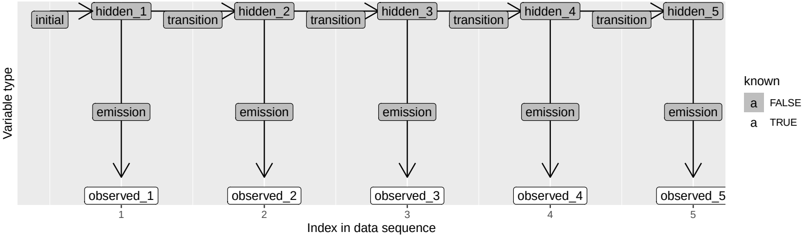
HMM ideas

- Each observed data variable in the sequence has a corresponding un-observed (hidden) state variable.
- There are typically a finite number of possible values for each hidden state variable, $k \in \{1, \dots, K\}$.
- Markov assumption: first-order dependency (each hidden variable only depends on the previous hidden variable in the sequence).



Parameters of HMM

- transition matrix: $A \in [0,1]^{K \times K}$ for K clusters. Each entry a_{ij} is the probability of transitioning from state i to state j.
- initial state distribution: $\pi \in [0,1]^K$ (prior weights).
- emission: define the probablity of observing data y in state k, $b_j(y) \in [0,1]$ (mean, covariance).
- These parameters are unknown in advance and must be learned from the data.
- Comparison with Gaussian Mixture Models: HMM has all of GMM parameters, plus transition matrix.

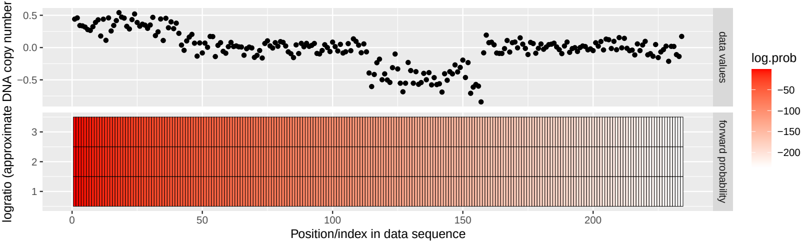


Three problems and algorithms

- Evaluation of likelihood of a given data sequence and model parameters => forward algorithm.
- Decoding most likely sequence of hidden states given observed data => Viterbi algorithm.
- Learning model parameters that maximize likelihood of a given data set => Baum-Welch algorithm.

Forward algorithm

- Assume each state $k \in \{1, \dots, K\}$ has an emission probability function $b_k(o_t)$ for observed data o_t at time t.
- Let a_{kj} be a transition parameter (probability of going from state k to state j).
- Then we can recursively compute the forward path probability, $\alpha_t(j) = \sum_{k=1}^{K} \alpha_{t-1}(k) a_{kj} b_j(o_t)$.
- Total log likelihood is summed over all states at the last data point in the sequence.

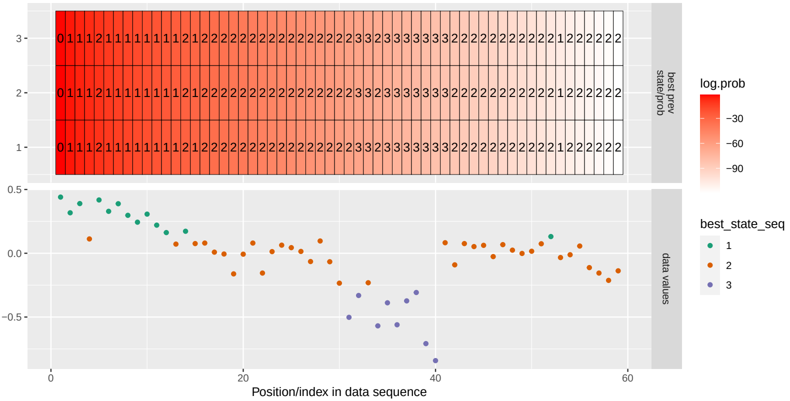


Viterbi algorithm

- Assume each state $k \in \{1, \ldots, K\}$ has an emission probability function $b_k(o_t)$ for observed data o_t at time t.
- Let a_{kj} be a transition parameter (probability of going from state k to state j).
- Then we can recursively compute the probability of the best sequence of hidden variables that ends at data point t in state j, $v_t(j) = \max_{k \in \{1,\ldots,K\}} v_{t-1}(k) a_{kj} b_j(o_t)$.

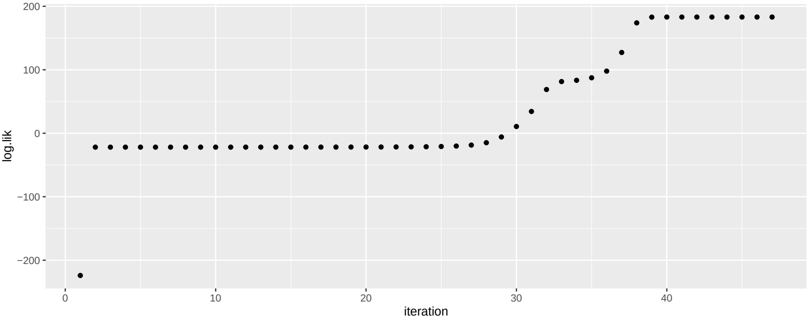
Viterbi algorithm example

```
## List of 4
##  $ mean      : num [1:3] 0.25 0 -0.5
##  $ sd        : num 1
##  $ transition: num [1:3, 1:3] 0.333 0.333 0.333 0.333 0.
##  $ initial   : num [1:3] 0.333 0.333 0.333
```

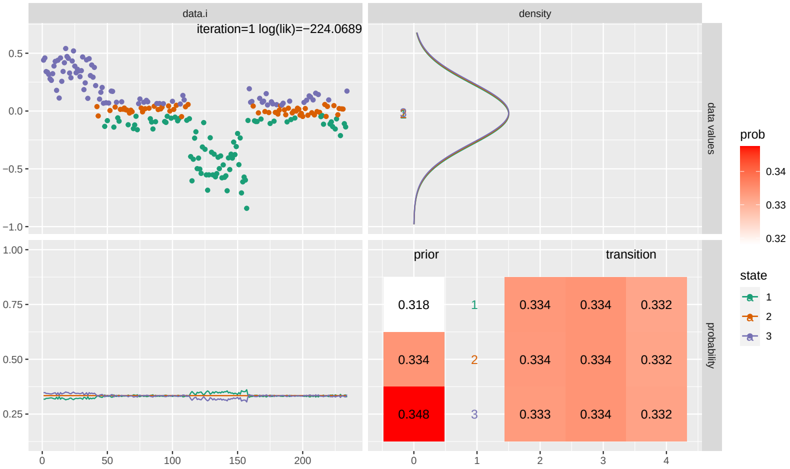


Baum-Welch learning algorithm

- Is an instance of Expectation-Maximization (EM), like the Gaussian Mixture Model learning algorithm.
- E step involves forward/backward passes over data sequences, to compute probability of each data point in each state.
- M step involves re-computing model parameters.
- Repeat until the log likelihood stops increasing.

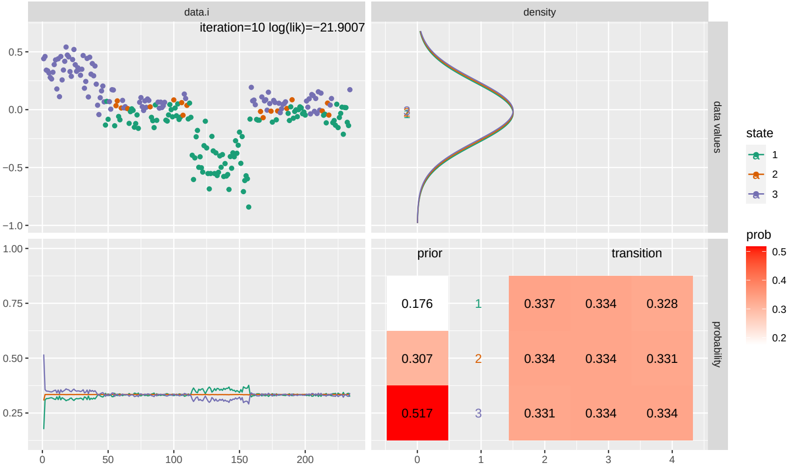


Visualization of learning iterations

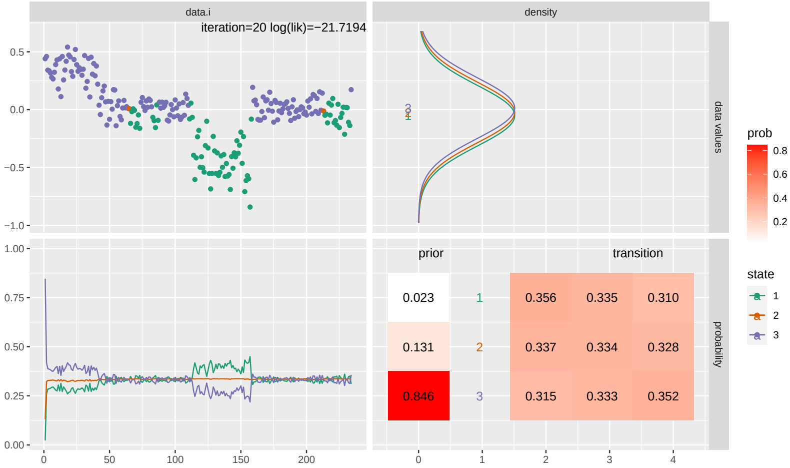


prior		transition		
0.318	1	0.334	0.334	0.332
0.334	2	0.334	0.334	0.332
0.348	3	0.333	0.334	0.332

Visualization of learning iterations

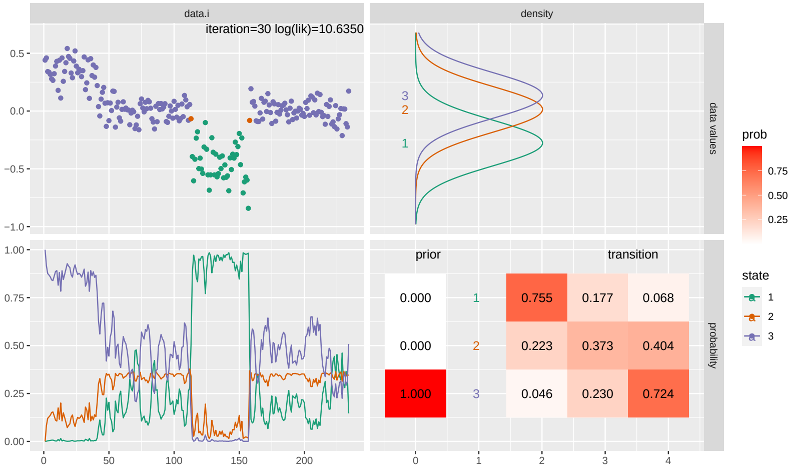


Visualization of learning iterations



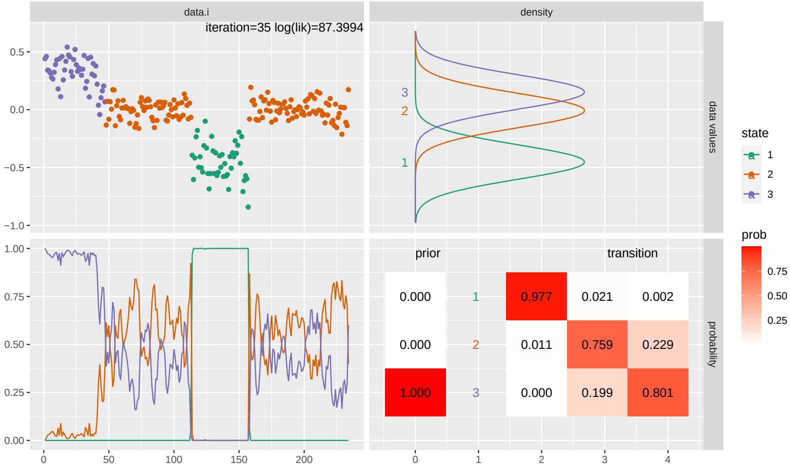
prior		transition		
0.023	1	0.356	0.335	0.310
0.131	2	0.337	0.334	0.328
0.846	3	0.315	0.333	0.352

Visualization of learning iterations

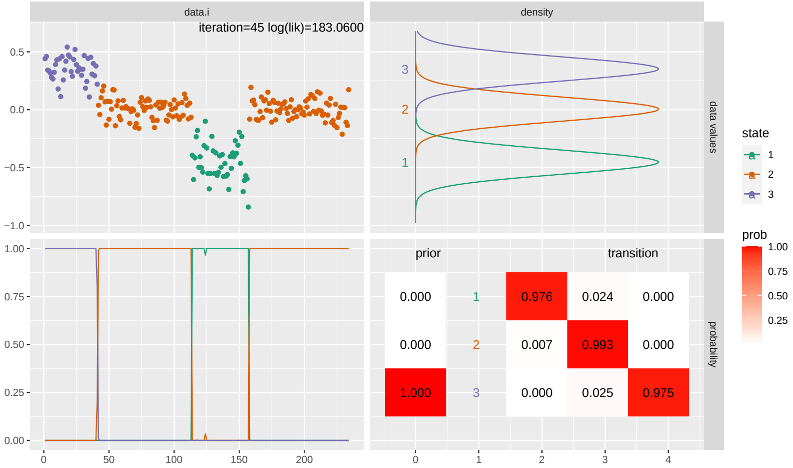


	prior		transition		
	0.000	1	0.755	0.177	0.068
	0.000	2	0.223	0.373	0.404
	1.000	3	0.046	0.230	0.724

Visualization of learning iterations

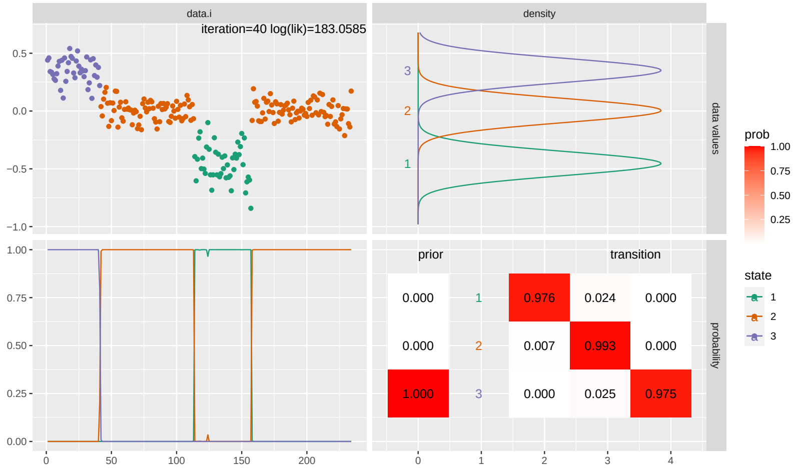


Visualization of learning iterations



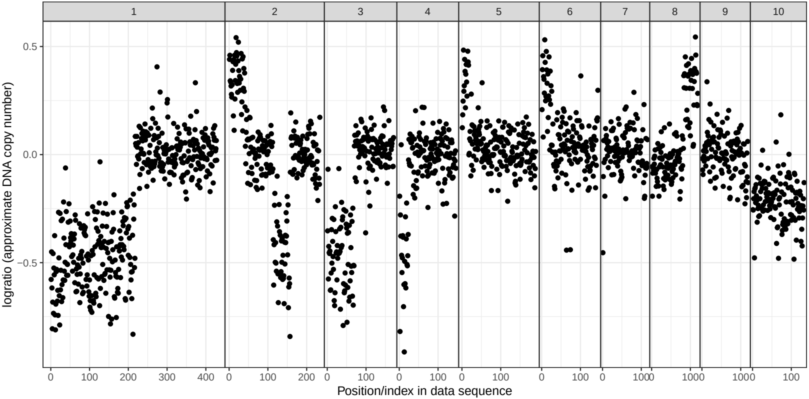
prior	state	transition		
0.000	1	0.976	0.024	0.000
0.000	2	0.007	0.993	0.000
1.000	3	0.000	0.025	0.975

Visualization of learning iterations

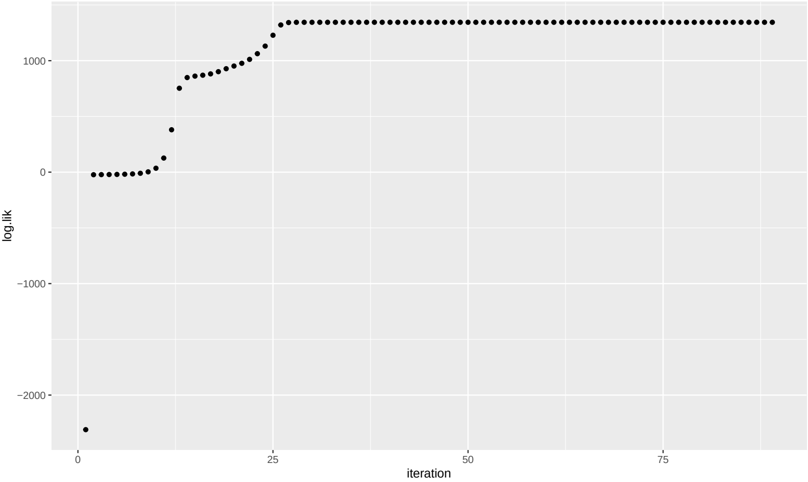


Shared states between chromosomes on a profile?

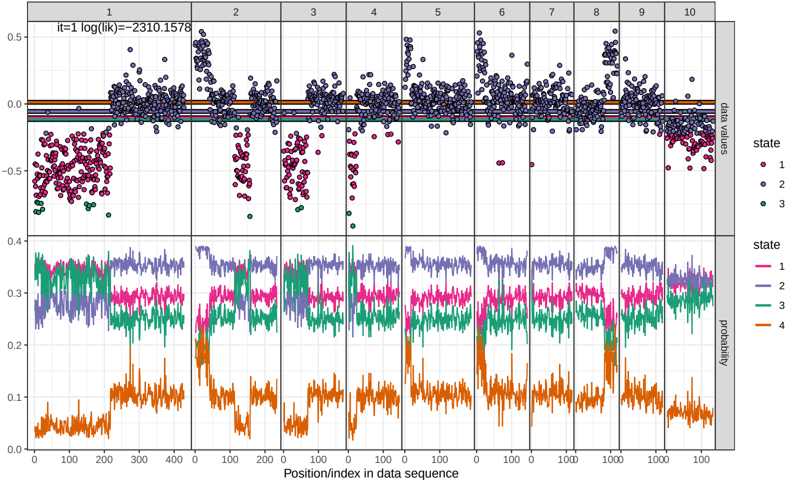
- Sometimes there are several data sequences which are assumed to have the same set of hidden states.
- In this case Baum-Welch can be used to fit HMM to all data at the same time (total log likelihood is summed over all data sequences).



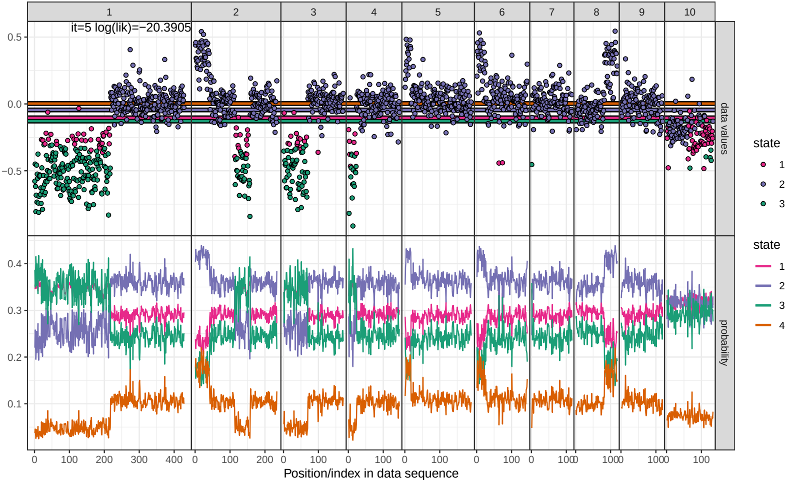
HMM on whole profile



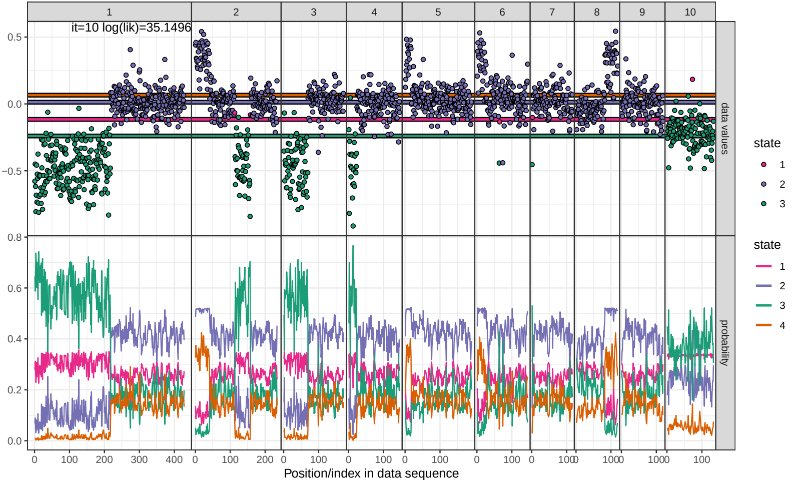
Learning iterations using multiple sequences



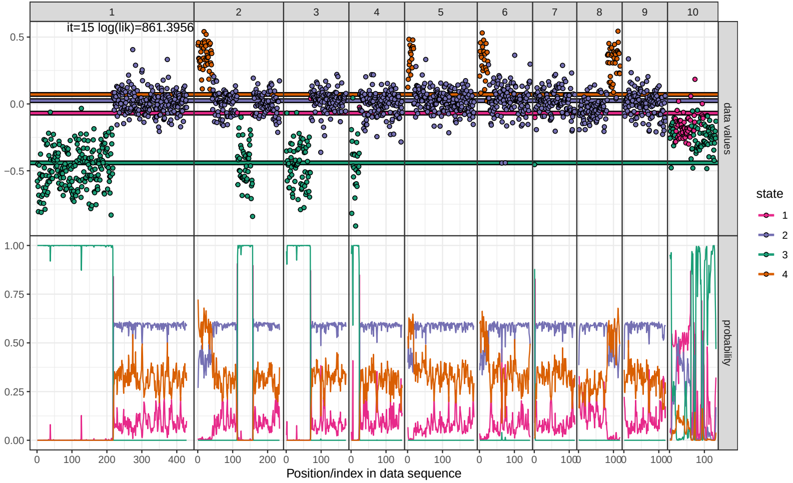
Learning iterations using multiple sequences



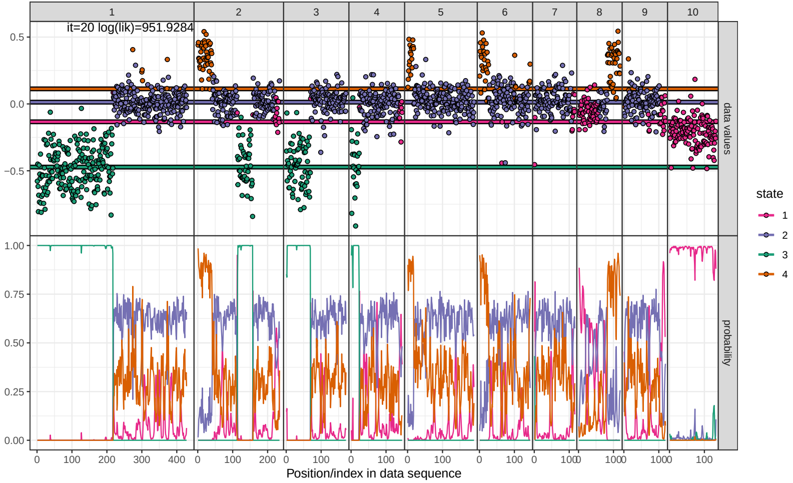
Learning iterations using multiple sequences



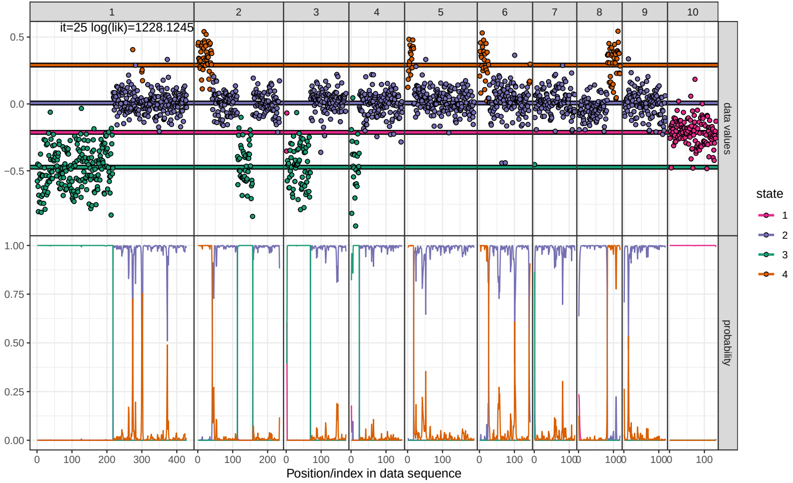
Learning iterations using multiple sequences



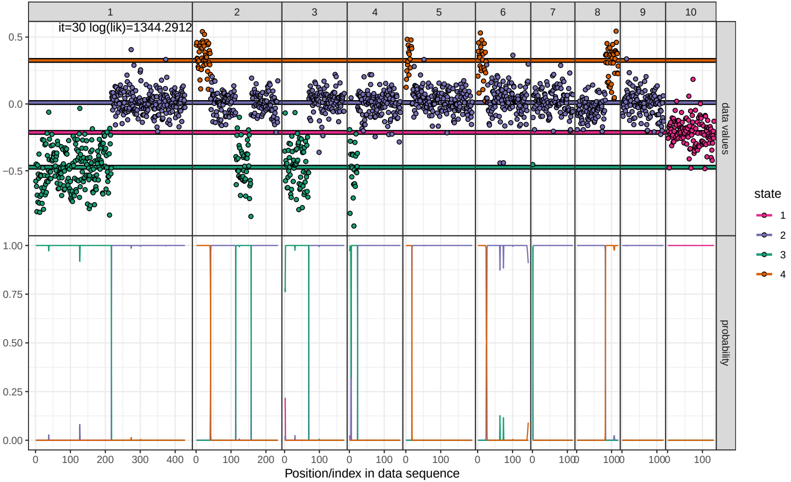
Learning iterations using multiple sequences



Learning iterations using multiple sequences



Learning iterations using multiple sequences



Time/space complexity TODO

Possible exam questions TODO