cpHMM (compound-state Hidden Markov Model) – User guide and Documentation

Nicholas C Lammers, Vahe Galstyan, Armando Reimer, Sean A Medin, Chris H Wiggins, and Hernan G Garcia

December 13, 2019

This document describes the workflow of the cpHMM method which can be used to infer the promoter state kinetics from MS2 live imaging data. It starts with discussions of raw MS2 data pre-processing, the inputs and outputs of the inference algorithm written in Matlab, and concludes with directions for interpreting and validating the cpHMM inference results. The mathematical aspects of the method development and graphical representations of the different facets of its performance are present in the SI Appendix and Extended Materials and Methods of the reference paper, and we encourage the users to consult as a complementary source. Additionally, the scripts related to the cpHMM method are available in https://github.com/GarciaLab/cpHMM.

Reference: Lammers, Galstyan, Reimer, Madin, Wiggins, Garcia – "Multimodal transcriptional control of pattern formation in embryonic development," PNAS (forthcoming Dec 2019)

1 Data pre-processing

The cpHMM method assumes that the input time series data have a uniform time resolution ΔT, meaning that consecutive data points correspond to fluorescence readouts which are ΔT apart (20 sec in our inference). Therefore, the raw MS2 trace data first needs to be interpolated with the specified time resolution ΔT. This resolution should be close to that of the experimental data acquisition in order to preserve most of the original information. At the same time, its value needs to be such that the integer memory of the system, which represent the number of ΔT intervals that fit within the polymerase elongation time window τ_{elong}, is an integer, i.e. $w = \tau_{\text{elong}}/\Delta T \in \mathbb{N}$ ($w = 7$ in most of our inference runs).

After the raw data is pre-processed to have uniform time resolution, the subsets of traces which are believed to represent promoter activities with identical kinetics (e.g. from nuclei within the same AP bin) should be collected in a cell array in order to be passed as inputs to the pooled inference function.

2 Running cpHMM

The pre-processed MS2 trace data in the form of cell arrays is then passed to the cpHMM algorithm for a pooled parameter inference. If the data represent full traces, i.e. traces that capture the start of promoter activity with no fluorescence readouts prior to the $t = 0$ point, then the default inference function `local_em_MS2_reduced_memory.m` should be used. On the other hand, if the data represent truncated traces from a specified time window, then one of these two inference functions should be used: `local_em_MS2_reduced_memory_truncated.m` or `local_em_MS2_reduced_memory_truncated_ss.m`. The first one makes no assumptions about the promoter activity prior to the start of the trace, while the second one assumes that the promoter has reached steady state, with effective state occupancies dictated by the kinetic rates inferred for the given time window. These inference options on truncated traces can be used to study temporal variations in the promoter's kinetic parameters by conducting inferences with a sliding time window.

In additional to the trace data, the inference functions need to be provided the set of parameters listed in Table 1. These include: 1) a set of constant system parameters, 2) initialization values for the transition and emission parameters, 3) parameters specifying the termination conditions of the cpHMM algorithm.

The number of effective promoter states (`K`) is specified by the user and needs to be an integer greater than 1 (values 2 and 3 were used in our analysis). Parameters related to the features of the gene construct and polymerase elongation (`w` and `kappa`) need to be calculated based on the knowledge of the whole gene and MS2 segment lengths, as well as the polymerase elongation rate (refer to see SI Appendix, sections D and J for details).

Initial guesses for the model parameters which serve as starting points for optimization are made using the function `local_em_iid_reduced_memory.m` (or

System constants

`K`	Number of effective promoter states.
`w`	Integer memory of the system equal to the ratio of the elongation time (τ_{elong}) and the time resolution (ΔT).
`kappa`	Ratio of the time it takes a polymerase to transcribe the MS2 loops (τ_{MS2}) and the time resolution (ΔT). Does not have to be an integer.

Initialization parameters

`v`	Fluorescence emission values per time step for each of the `K` effective promoter states.
`noise`	Gaussian noise in the fluorescence readout.
`pi0_log`	Log values of the initial promoter state probabilities.
`A_log`	Log values of the transition probabilities between each pair of states.

Algorithm parameters

`n_steps_max`	Highest number of expectation-maximization (EM) iterations before the algorithm is terminated (the value 500 was used in our inferences).
`eps`	Threshold value for the relative change in the Euclidean norms of the model parameters needed for terminating the optimization procedure (the value 10^{-4} was used in our inferences).

Table 1: Additional input parameters required for running the cpHMM algorithm in Matlab.

`local_em_iid_reduced_memory_truncated.m`), which provides a crude estimate of the parameters with some added randomness by ignoring the temporal correlation between data points. Additional details on the initializstion procedure can be found in the "Execution of the cpHMM method" subsection of SI Appendix, Section D.

3 cpHMM outputs

The cpHMM algorithm performs a local optimization of an objective function which represents a lower bound of the log-likelihood of observing the entire trace data (see SI Appendix, section D for details). Its output includes the set of optimal model parameters and the value of the optimized objective function. The inferred parameters are:

- initial promoter state probabilities ($\vec{\pi}$)
- fluorescence emission values per time step at each state ($\vec{v}$), which can be converted into RNAP loading rates using the fluorescence calibration

- fluorescence readout precision parameter ($\lambda = 1/\sigma^2$, where σ is the noise)
- transition probabilities between states (A_{ij}), which can be converted into transition rates (R_{ij}) using the relation $\mathbf{A} = e^{\mathbf{R}\Delta T}$

Note that because the optimization procedure is local, cpHMM algorithm should to be run for multiple random initialization conditions in order to find the set of model parameters that globally maximizes the objective function (10-20 such local runs were performed in our analyses).

4 Viterbi trace reconstruction

The set of model parameters inferred using the cpHMM algorithm can then be used to obtain the mostly likely promoter state trajectories that led to corresponding fluorescence readouts. This is done using the `viterbi.m` function which implements the Viterbi decoding algorithm and outputs the most likely promoter state trajectory along with the fluorescence series corresponding to this trajectory.

5 Error estimation and validation

The pooled inference algorithm returns a single set of optimal parameters when applied on the entire trace data. To obtain error bars on these inferred parameters, the bootstrapping method is used. Specifically, inference is conducted on multiple randomly sampled subsets of the full data set, and the variability in the inferred parameters is taken as a proxy for the algorithm's inference precision (see "Extended Materials and Methods" for additional details).

Lastly, the accuracy of the cpHMM algorithm is confirmed using statistical validation studies. Since in the cpHMM formalism the transitions between effective promoter states take place at discrete time points spaced ΔT apart and an averaged RNAP loading rate is assigned to each promoter states, effects related to continuous-time promoter switching and discrete RNAP loading statistics may, at certain parameter regimes, lower the accuracy of parameter inference. For that reason, the accuracy of the method needs to be confirmed on data sets generated synthetically using the parameters inferred from experimental MS2 data. We have demonstrated that for the inferred kinetic regime of the *eve* promoter the cpHMM algorithm is indeed highly accurate, and refer the reader to SI Appendix, section E for further details on the implementation of the validation studies.