UPPSALA
UNIVERSITET

UPTEC X 15 001
Examensarbete 30 hp
Januari 2015

Bayesian inference in aggregated hidden Markov models

Emil Marklund

UPPSALA UNIVERSITET

Degree Project in Molecular Biotechnology

Masters Programme in Molecular Biotechnology Engineering,
Uppsala University School of Engineering

UPTEC X 15 001	**Date of issue 2015-01**

Author

Emil Marklund

Title (English)

Bayesian inference in aggregated hidden Markov models

Title (Swedish)

Abstract

Single molecule experiments study the kinetics of molecular biological systems. Many such studies generate data that can be described by aggregated hidden Markov models, whereby there is a need of doing inference on such data and models. In this study, model selection in aggregated Hidden Markov models was performed with a criterion of maximum Bayesian evidence. Variational Bayes inference was seen to underestimate the evidence for aggregated model fits. Estimation of the evidence integral by brute force Monte Carlo integration theoretically always converges to the correct value, but it converges in far from tractable time. Nested sampling is a promising method for solving this problem by doing faster Monte Carlo integration, but it was here seen to have difficulties generating uncorrelated samples.

Keywords

Bayesian inference, aggregated hidden Markov models, model selection, variational Bayes, nested sampling, single molecule data

Supervisors

Dr. Martin Lindén
Uppsala University

Scientific reviewer

Prof. Mats Gustafsson
Uppsala University

Project name	Sponsors
Language **English**	Security
ISSN 1401-2138	Classification
Supplementary bibliographical information	Pages **54**

Biology Education Centre Biomedical Center Husargatan 3, Uppsala
Box 592, S-751 24 Uppsala Tel +46 (0)18 4710000 Fax +46 (0)18 471 4687

Bayesian inference in aggregated hidden Markov models

Emil Marklund

Populärvetenskaplig sammanfattning

Rörelse och interaktioner hos biologiska makromolekyler (såsom proteiner och DNA) kan idag undersökas på enmolekyl nivå på tillräckligt korta tidsskalor för att kunna studera processen i detalj. Sådana detaljstudier av biologin har potential att ge mycket gott till samhället i stort, om metodutvecklingen fortsätter gå framåt. Studier på molekylärnivå kan till exempel hjälpa oss att förstå mekanismer bakom antibiotikaresistens vilket kan leda till upptäckter av nya antibiotika, eller lära hos hur mikrober effektivast producerar förnybara energikällor, för att nämna två exempel.

Brist på bra beräkningsmetoder för dataanalys är idag en flaskhals när man vill tolka resultaten från enmolekyl-experiment. Arbetet under detta examensarbete har gått ut på att studera metoder som kan användas för att bygga matematiska modeller från enmolekyl-data och därmed bidra till att lösa problemen med dataanalys i forskningsfältet. Mer specifikt så har s.k. aggregerade dolda Markovmodeller studerats, en typ av modeller som kan användas för att beskriva många intressanta molekylära system som ribosomen och RNA-polymeraset. Kortfattat så grundar sig modellerna på att de beskriver de dolda bindingstillstånden hos den studerade molekylen samt övergångarna mellan dessa tillstånd. Till exempel så ser man från experimentdata från ribosomer endast om ribosomen är bunden till mRNA eller om den är fritt diffunderande i cellen. Ribosomen har dock egentligen ett stort antal dolda bindingsstillstånd som motsvarar varje kodon som den translaterar till en aminosyra under proteinsyntesen. Detta kan de aggregerade dolda Markovmodellerna beskriva, varför det är intressant att kunna konstruera sådana modeller.

Den mest grundläggande egenskapen hos en aggregerad dold Markovmodell är hur många dolda tillstånd den har och under detta arbete har vi undersökt hur man på ett korrekt sätt kan bestämma antalet dolda tillstånd utifrån experimentella data. Detta har vi gjort genom att ge poäng till olika modellstorlekar med det Bayesianska evidenset (beviset) och sen välja modellstorleken med högst poäng. Metoden ”variational Bayes inference” som visade sig approximera evidenset bra för icke aggregerade dolda Markovmodeller underskattade evidenset grovt för aggregerade modeller. Att beräkna evidenset exakt är möjligt när man har väldigt lite data men det tar däremot extremt lång tid när datamängden växer. Nested (nästlad) sampling är en metod som har potential att snabba upp den exakta beräkningen, men en algoritm som var tillräckligt effektiv hittades aldrig under detta projekt.

Examensarbete 30 hp

Civilingenjörsprogrammet i Molekylär bioteknik

Uppsala universitet, januari 2015

Table of contents

Abbreviations

BIC	Bayesian information criterion
BKU	Bauer Kienker uncoupled
FRET	Förster resonance energy transfer
HMM	hidden Markov model
KL	Kullback-Leibler
MCMC	Markov chain Monte Carlo
MIR	Manifest interconductance rank
SPT	single particle tracking
VB	variational Bayes
vbSPT	variational Bayes single particle tracking

Background

Analyzing single molecule data

Modern biological research is highly dependent on analyzing quantitative data measured from various experimentally studied systems. Single molecule experiments generate a huge amount of quantitative data and development of efficient methods for data analysis is therefore crucial for new insights and breakthroughs in the field. Simplistic or inappropriate approaches to data analysis leave a big amount of the information present in the single molecule data unused, which of course is not desirable.

The data generated from different types of single molecule experiments studying different kinds of systems often have many properties in common. When plotting the signal from the experiment against time the noisy readout is often seen to stay centered at some relatively constant mean before jumping to some other level and staying approximately constant for a period of time, then jumping to some other level, and so on. Real experimental data[1,2,3] demonstrating these properties can be found in Fig. 1.

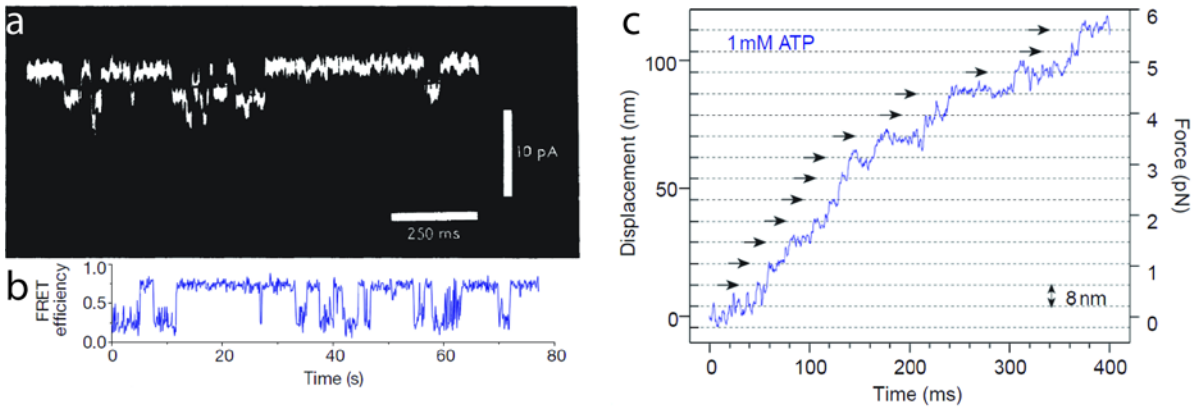


Figure 1. Data generated from different kinds of single molecule experiments have many properties in common. The signals of all three experiments are seen to stay approximately constant around a number of mean levels. Transitions between the levels are fast and the exact value of the mean at each level is obscured by noise in the signal. **(a)** Patch clamp measurement of the electric current through a membrane ion channel. Figure[1] used with permission from Nature Publishing Group. **(b)** A Förster resonance energy transfer (FRET) experiment measuring the FRET efficiency between two fluorophores situated on the h3 helix of 16S RNA and the ribosomal protein S4, respectively. This signal is a measure of the distance between the two components during assembly of the small ribosomal subunit. Figure[2] used with permission from Nature Publishing Group. **(c)** Measurement of the displacement along a microtubule of a single copy of the motor protein kinesin. Figure[3] used with permission from Nature Publishing Group.

Examples of experimental techniques that generate this kind of data are single particle tracking (SPT), patch clamp measurements of ion channels and Förster resonance energy transfer (FRET) experiments. The similarities in the data are no coincidence; the single molecule techniques all study associations, dissociations or conformational changes of proteins or other biological macromolecules. The transitions between the different mean

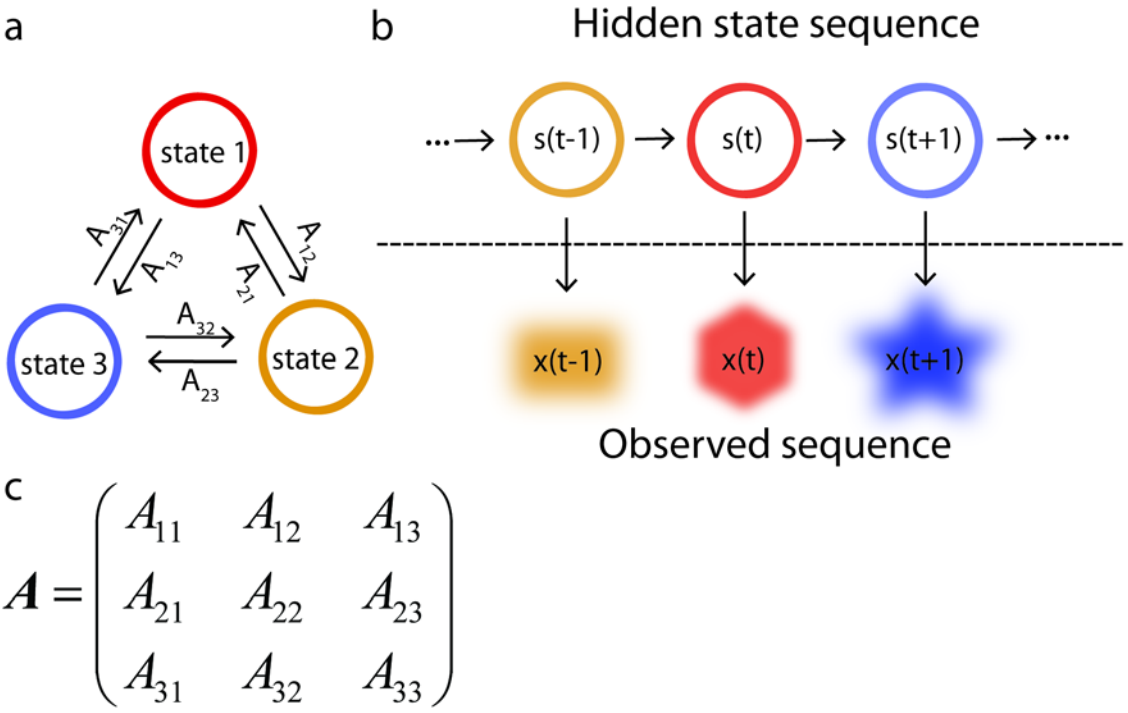
levels in the signal are explained by the chemical transitions that the molecule goes through. The noise in the signal can come from limitations in the experimental measurements, but can also be caused by inherent stochastic properties in the quantity that is measured.

Describing the data with hidden Markov models

The conformational changes in a macromolecule studied by a single molecule experiment are in many cases well described, or at least well approximated, by a unimolecular reaction scheme. In these situations the experimental single molecule data has great potential to be described by a hidden Markov model (HMM). In this section a brief introduction to HMMs are given along with explanations of notations used later in the report. The HMM has for a long time been a well used tool for statistical analysis and hence there exists many more thorough introductions than the one given here. For the interested reader see for example this text by Rabiner and Juang[4].

An HMM (Fig. 2a) is a statistical model consisting of a number of, say N, different hidden states. In discrete time the Markov process that is seen to obey the HMM is always found in one of these hidden states during each time step t. As time progresses so does the process and during each time step there is a probability to leave the current hidden state for any of the other hidden states in the model. The probabilities A_{ij} to leave state i for state j during a time step makes up the elements of the transition probability matrix $\boldsymbol{A}$. The process is *Markovian*, meaning that the transition probabilities only depends on the current hidden state, and not the state belonging at any previous or later time point. To generate the hidden state sequence of the process the initial state probabilities π_i, that is the probability for the process to start in state i, are also needed. With these definitions the row vector $\vec{\pi}$, containing all initial state probabilities, together with the transition probability matrix $\boldsymbol{A}$ fully describe the statistics generating the hidden state sequence $\vec{s}$.

The definitions in the previous paragraph do not only apply for HMMs but for all discrete-time Markov chains. What further defines a process following a *hidden* Markov model is that the states of the process are just that: hidden. For an outside observer the sequence of hidden states $\vec{s}$ is not visible. Instead an observation that is dependent on the hidden state is recorded by the observer (Fig. 2b). The observed sequence $\vec{x}$ obtained by observing the process for a period of time can be interpreted as the signal of the experiment when studying this system experimentally. The process generating each observation is often stochastic which contributes to noise in the signal.



c

$$\boldsymbol{A} = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix}$$

Figure 2. Schematic example of a three state HMM. (a) The three hidden states of the HMM and the possible transitions with probabilities A_{ij} connecting them. **(b)** A part of a hidden state sequence and an observed sequence generated by the three state HMM. At each time point t the hidden state s_t is not observed directly, but a noisy observation x_t dependent on the hidden state is. **(c)** The transition probability matrix $\boldsymbol{A}$ of a three state HMM.

Aggregated hidden Markov models

Aggregated hidden Markov models (Fig. 3a) are of special interest in this work. Following the naming convention of Fredkin and Rice[5], and Kienker[6], an aggregated HMM is here defined as an HMM where two or more hidden states have identical observable statistics. The hidden states are grouped into different aggregates where it is impossible to distinguish observations generated by different hidden states in the same aggregate. A given aggregate thus consists of one or more hidden states which are impossible to separate for an outside observer. In probabilistic terms, the observations from hidden states in the same aggregate are generated from identical statistical distributions (Fig. 3b).

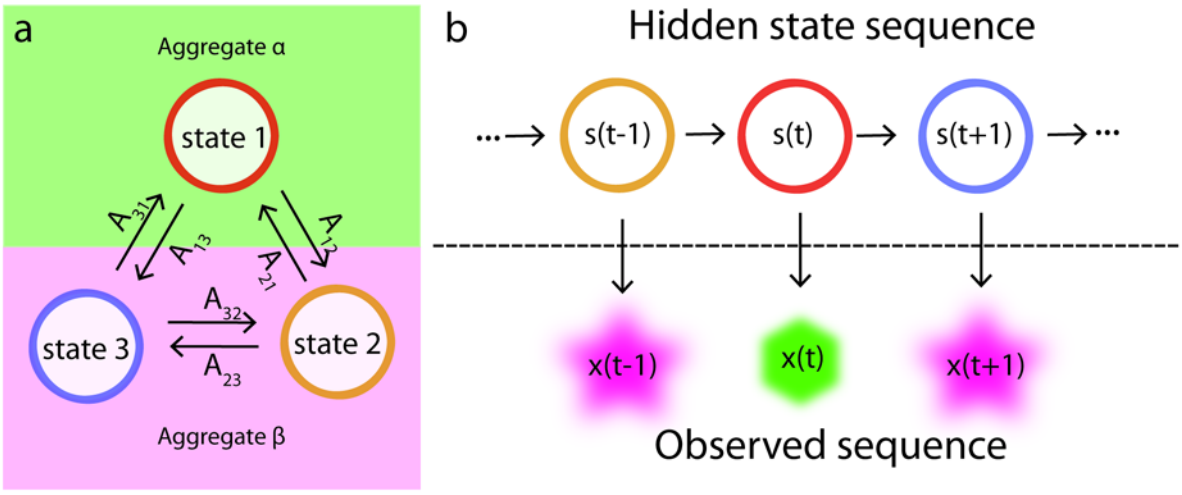


Figure 3. Schematic example of a three state aggregated HMM with two aggregates. (a) The three hidden states of the aggregated HMM and the possible transitions with probabilities A_{ij} connecting them. The hidden states 2 and 3 are aggregated giving the model two aggregates (α and β). **(b)** A part of a hidden state sequence and an observed sequence generated by the three state aggregated HMM. At each time point t the hidden state s_t is not observed directly, but a noisy observation x_t dependent on the hidden state is. The observations generated from states 2 and 3 are indistinguishable because these hidden states belong to the same aggregate.

In this work the parameters describing an aggregated HMM is for computational and pedagogical reasons dived into the model parameters $\theta = (\vec{\pi}, \boldsymbol{A})$, describing how the hidden state sequence is generated, and model structure $\eta = (N, \vec{f})$, describing the model size and aggregate belonging of the different hidden states. Here, $\vec{f}$ is a vector defining which aggregate each hidden state belongs to, such that $f(s_t)$ is the aggregate of state s_t. An assumption throughout this work has been that no noise is present in the observed data sequence $\vec{x}$, and thus there are no parameters included in θ that can generate noise in the signal. In this work the observed data is explicitly given by the aggregate belonging of the current hidden state according to

$$x_t = f(s_t). \qquad (0.1)$$

This simplification is made so that it is only general properties of aggregated HMMs that are studied here, and not some other special properties caused by noisy data. Put in other words, the hidden part of the hidden Markov models is here only caused by the aggregation of the hidden states, and not caused by difficulty to separate different aggregates from each other. The investigated analysis methods can however handle noisy data after small changes in the algorithms, which motivates the usefulness of this study for future works where noise is present. Examples on how the vector $\vec{f}$ can look in practise are $\vec{f} = (1 \quad 2)$ for an unaggregated model with two hidden states and $\vec{f} = (1 \quad 2 \quad 2)$ for an aggregated HMM with three hidden states were two hidden states belong to the second aggregate.

Because of the aggregation of the hidden states the concept of *equivalence* arises in aggregated HMMs. Equivalent models are defined to have identical observable statistics. This means that two equivalent models with different sets of parameters θ and θ' are both just as likely to have generated any observed sequence $\vec{x}$. In other words, it is impossible to separate the models based on experimental data, which means that is impossible to uniquely determine the most likely parameter set θ from an experiment studying an aggregated HMM. The concept of equivalence and how it is used in this work is further explained in the Methods chapter.

Bayesian inference

The goal of this study has been to investigate how to properly do *model selection* and *parameter inference* with aggregated HMMs. In this context, model selection means finding the optimal model structure η given some observed data sequence $\vec{x}$ and parameter inference is done when the optimal set of parameters θ are found given a model structure and some observed data. To do model selection some method to compare different model structures is needed, and here this has been done by scoring different models with the *evidence* obtained after doing Bayesian inference. The evidence $Z = p(\vec{x} \mid \eta)$ is defined as

$$Z = p(\vec{x} \mid \eta) = \int p(\vec{x} \mid \theta, \eta)\, p(\theta \mid \eta) d\theta = \int L(\theta)\, p_0(\theta) d\theta, \qquad (0.2)$$

Where $L(\theta) = p(\vec{x} \mid \theta, \eta)$ is the probability of $\vec{x}$ given θ and η, more commonly known as the *likelihood*, and $p_0(\theta) = p(\theta \mid \eta)$ is the *prior* defining the *a priori* probability of different parameter values before observing any data.

The evidence $p(\vec{x} \mid \eta)$ can with Bayes rule be seen to relate to $p(\eta \mid \vec{x})$ according to

$$p(\vec{x} \mid \eta) = \frac{p(\eta \mid \vec{x})\, p(\vec{x})}{p(\eta)} \qquad (0.3)$$

Given that all model structures and observed data are equally probable *a priori*, the maximization of the evidence is equivalent to the maximization of $p(\eta \mid \vec{x})$. This means that the probability of the model structure given the data $p(\eta \mid \vec{x})$ is maximized if models are selected with a criterion of maximum evidence, which intuitively is a good criterion for model selection.

This Bayesian criterion for model selection has previously been shown to solve the problem of overfitting for unaggregated HMMs, which traditionally has been a problem for direct maximum likelihood fits of the parameters[7]. Furthermore, even though the data used in this work was perfectly segmented (no noise) Bayesian model selection also has the advantage of being able to handle noisy data directly in the model selection algorithm. The Bayesian strategy of model selection should therefore be more generally applicable than for example a previously proposed method for model selection using causal states[8], which requires that the

data is segmented separately in an early step of the algorithm.

The integrand in the evidence integral is not difficult to compute when the likelihood is given by an HMM and a nice prior has been chosen (see the Methods chapter for details), but the integral itself is a whole other matter. Because of the complexity of the likelihood function there exists no analytical solution to the problem and some numerical method is therefore needed for the evidence calculation.

Three different strategies for numerically computing the evidence has been investigated during this study. The first method that was tested applies an algebraic approximation of the integrand when calculating the evidence. Doing inference with this approximation is commonly known as *Variation Bayes inference*[9,10] and the method is therefore referred to as variational Bayes (VB). The other two methods are based on direct Monte Carlo estimation of the integral. In these methods no algebraic approximation is used on the integrand and the methods are therefore called *exact* since the results they give theoretically always converge to the correct evidence. These two methods are referred to as *naive exact inference* and *nested sampling*[11,12]. These three computational methods are further explained in the Methods chapter.

Single particle tracking

As discussed previously there exists many different kinds of single molecule experiments that generate similar types of data, and if this data was generated from an experimental system with aggregated states, data analysis tools implementing aggregated HMMs are needed when doing inference and model selection on this data. Even though such an inference method could be applied generally to different types of single molecule experiments, the major motivation for this study has been to apply such a method on data from intracellular single particle tracking experiments. In these experiments subsequent pictures are taken of fluorescently labelled molecules inside a cell. The fluorescent spots generated from one single molecule in each individual frame in this film are connected to form a *trajectory* describing the motion of the particle. The step lengths between adjacent points in these trajectories are the data from the experiment and are thus interpreted as the observed sequences $\vec{x}$ when speaking in terms of an HMM. The hidden state sequences $\vec{s}$ are in this set up modelling the diffusive state of the single molecule (bound or not bound for example) and the model parameters θ are given by the reaction rates between these different diffusive states.

A variational Bayes method has previously been developed for efficiently doing inference and model selection based on SPT trajectories when the data is unaggregated[13]. That method was implemented in the MATLAB software variational Bayes single particle tracking (vbSPT) which is available for free online[14]. However, since aggregation of the diffusive states add complexity to the data and the model, this work was started to investigate how inference could be properly done in the aggregated case. Many interesting experimental systems are very likely to have aggregated diffusive states and proper analysis methods are thus needed if one wants to do model selection on these systems. Examples of molecular systems that are

expected to have aggregated diffusive states are the ribosomal subunits (translating mRNA one codon at the time or freely diffusing), the RNA-guided DNA endonuclease Cas9 (bound or not bound to DNA[15]) and RNA polymerase (transcribing DNA to mRNA or freely diffusing).

Methods

In this chapter the methods used in this work are presented, along with discussions of established concepts from previous studies that the work in this report builds upon and makes use of.

Discrete and continuous representations of an aggregated HMM

The HMMs in this report is to large extent represented in discrete time, where the hidden state transitions can be described by the transition probability matrix $\boldsymbol{A}$. HMMs and Markov chains in general can however just as well be considered in continuous time, and such a representation was also needed for some purposes during the work of this study.

In a continuous time HMM the state transitions are described by the generator matrix $\boldsymbol{Q}$ which can be seen to relate to the transition probability matrix $\boldsymbol{A}$ and the time step Δt in the discretization according to

$$\boldsymbol{A} = e^{\boldsymbol{Q}\Delta t}. \tag{0.4}$$

Here the generator matrix $\boldsymbol{Q}$ can be defined from the transition rate constants describing the state transitions of the model, and each row of $\boldsymbol{Q}$ sums to zero.

The generator matrix is here ordered in such a way that the states in the same aggregate β_k appear as a block in $\boldsymbol{Q}$. The generator matrix is then partitioned into submatrices and can be written on the form

$$\boldsymbol{Q} = \begin{pmatrix} \boldsymbol{Q}_{\beta_1\beta_1} & \cdots & \boldsymbol{Q}_{\beta_1\beta_n} \\ \vdots & \ddots & \vdots \\ \boldsymbol{Q}_{\beta_1\beta_n} & \cdots & \boldsymbol{Q}_{\beta_n\beta_n} \end{pmatrix}. \tag{0.5}$$

Here the submatrices $\boldsymbol{Q}_{\beta_k\beta_l}$ contains rate constants for transitions from states belonging to aggregate β_k to states belonging to aggregate β_l. As an example, with two aggregates α and β the generator matrix $\boldsymbol{Q}$ has the form

$$\boldsymbol{Q} = \begin{pmatrix} \boldsymbol{Q}_{\alpha\alpha} & \boldsymbol{Q}_{\alpha\beta} \\ \boldsymbol{Q}_{\beta\alpha} & \boldsymbol{Q}_{\beta\beta} \end{pmatrix}. \tag{0.6}$$

In discrete time, the transition probability matrix $\boldsymbol{A}$ is ordered on the same form, that is

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_{\beta_1\beta_1} & \cdots & \boldsymbol{A}_{\beta_1\beta_n} \\ \vdots & \ddots & \vdots \\ \boldsymbol{A}_{\beta_1\beta_n} & \cdots & \boldsymbol{A}_{\beta_n\beta_n} \end{pmatrix}, \tag{0.7}$$

which for a model with two aggregates becomes

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_{\alpha\alpha} & \boldsymbol{A}_{\alpha\beta} \\ \boldsymbol{A}_{\beta\alpha} & \boldsymbol{A}_{\beta\beta} \end{pmatrix}. \tag{0.8}$$

Dwell-time distributions in the different aggregates

A very central property for every aggregated HMM is the dwell-time distributions in the different aggregates, describing the probability to stay in the aggregates for different periods of time. Below is a definition of the dwell-time distribution in continuous time followed by a similar definition in discrete time.

Let a process generated by a continuous time aggregated HMM start by entering the aggregate α_0. If the process is observed to visit $r+1$ number of aggregates and the sequence of aggregates visited is $\vec{\alpha} = (\alpha_0, \alpha_1, ..., \alpha_r)$, the dwell-times $\vec{t} = (t_0, t_1, ..., t_r)$ that the aggregates in $\vec{\alpha}$ will be visited is distributed according to the probability density function[5,6,16]

$$f_{\vec{\alpha}}(\vec{t}) = \vec{\hat{\pi}}_{\alpha_0} e^{\boldsymbol{Q}_{\alpha_0\alpha_0}t_0} \boldsymbol{Q}_{\alpha_0\alpha_1} e^{\boldsymbol{Q}_{\alpha_1\alpha_1}t_1} ... e^{\boldsymbol{Q}_{\alpha_r\alpha_r}t_r} \vec{q}_{\alpha_r} = \vec{\hat{\pi}}_{\alpha_0} e^{\boldsymbol{Q}_{\alpha_0\alpha_0}t_0} \left(\prod_{m=1}^{r} \boldsymbol{Q}_{\alpha_{m-1}\alpha_m} e^{\boldsymbol{Q}_{\alpha_m\alpha_m}t_m} \right) \vec{q}_{\alpha_r}. \tag{0.9}$$

Here $\vec{\hat{\pi}}_\alpha$ is a row vector with elements being equal to the probability of aggregate α being entered via each of the hidden states in the aggregate, conditional on that the process is actually entering aggregate α. The column vector $\vec{q}_\alpha$ is defined according to

$$\vec{q}_\alpha = \sum_{\beta \neq \alpha} \boldsymbol{Q}_{\alpha\beta} \vec{u}_\beta, \tag{0.10}$$

where $\vec{u}_\beta$ is a column of N_β ones and N_β is the number of states in aggregate β.

From the multidimensional dwell-time distribution in Eq. (1.9) the one- and two- dimensional dwell-time distributions, that is the distribution of dwell-times when visiting one and two aggregates, can be read out as

$$f_\alpha(t_\alpha) = \vec{\hat{\pi}}_\alpha e^{\boldsymbol{Q}_{\alpha\alpha}t_\alpha} \vec{q}_\alpha, \tag{0.11}$$

and

$$f_{\alpha\beta}(t_\alpha, t_\beta) = \vec{\hat{\pi}}_\alpha e^{\boldsymbol{Q}_{\alpha\alpha}t_\alpha} \boldsymbol{Q}_{\alpha\beta} e^{\boldsymbol{Q}_{\beta\beta}t_\beta} \vec{q}_\beta, \tag{0.12}$$

where $\vec{\tilde{\pi}}_\alpha$, $\vec{q}_\alpha$ and $\vec{q}_\beta$ are defined correspondingly as for the multidimensional dwell-time distribution.

The above dwell-time distributions consist of sums of decaying exponentials, which can be seen after evaluation of the matrix exponentials and the matrix products in Eq. (1.9). For a truly Markovian process without aggregation, the dwell-time distribution simplifies to a single decaying exponential, which is expected for these memoryless processes. A very important connection between an aggregated HMM and its one- and two-dimensional dwell-time distributions has previously been proven by Fredkin and Rice[5]. It is there shown that the observable statistics of a continuous time aggregated HMM in steady state is completely determined by the one- and two-dimensional dwell-time distributions.

The one- and two-dimensional dwell-time distributions in discrete time has previously been written down on a similar form[17] as for the continuous counterpart described above. This expression can easily be expanded to higher dimensions to give the following definition for the discrete dwell-time distribution.

Let a discrete-time aggregated Markov process with state transition matrix $\boldsymbol{A}$ start by entering aggregate α_0. If the process is observed to visit $r+1$ number of aggregates and the sequence of aggregates visited is $\vec{\alpha} = (\alpha_0, \alpha_1, ..., \alpha_r)$, the time steps $\vec{t} = (t_0, t_1, ..., t_r)$ that the process dwells in the aggregates in $\vec{\alpha}$ is distributed according to the probability mass function

$$f_{\vec{\alpha}}(\vec{t}) = \vec{\tilde{\pi}}_\alpha \boldsymbol{A}_{\alpha_0\alpha_0}^{t_0-1} \boldsymbol{A}_{\alpha_0\alpha_1} \boldsymbol{A}_{\alpha_1\alpha_1}^{t_1-1} ... \boldsymbol{A}_{\alpha_r\alpha_r}^{t_r-1} \vec{q}_{\alpha_r} = \vec{\tilde{\pi}}_{\alpha_0} \boldsymbol{A}_{\alpha_0\alpha_0}^{t_0-1} \left(\prod_{\substack{m=1 \\ r>0}}^{r} \boldsymbol{A}_{\alpha_{m-1}\alpha_m} \boldsymbol{A}_{\alpha_m\alpha_m}^{t_m-1} \right) \vec{q}_{\alpha_r}. \tag{0.13}$$

Here $\vec{\tilde{\pi}}_\alpha$ is a row vector with elements being equal to the probability of aggregate α being entered via each of the states in the aggregate, conditional on that the process is actually entering aggregate α. $\vec{\tilde{\pi}}_\alpha$ can be calculated from the transition matrix $\boldsymbol{A}$ and the row vector $\vec{\pi}$ containing the initial state probabilities for all states as

$$\vec{\tilde{\pi}}_\alpha = \frac{\sum_{\beta \neq \alpha} \vec{\pi}_\beta \boldsymbol{A}_{\beta\alpha}}{\sum_{\beta \neq \alpha} \vec{\pi}_\beta \boldsymbol{A}_{\beta\alpha} \vec{u}_\alpha}, \tag{0.14}$$

where $\vec{u}_\alpha$ is a column of N_α ones and the denominator is a normalisation constant. The column vector $\vec{q}_\alpha$ in the dwell-time distribution arises due to the fact that the process can leave aggregate α for anyone of the other aggregates. This vector is defined according to

$$\vec{q}_\alpha = \sum_{\beta \neq \alpha} \boldsymbol{A}_{\alpha\beta} \vec{u}_\beta, \tag{0.15}$$

where $\vec{u}_\beta$ is a column of N_β ones.

From the multidimensional dwell-time distribution in Eq. (1.13) the one- and two-dimensional dwell-time distributions, that is the distribution of dwell-times when visiting one and two aggregates, can be read out as

$$f_\alpha(t_\alpha) = \vec{\tilde{\pi}}_\alpha \boldsymbol{A}_{\alpha\alpha}^{t_\alpha - 1} \vec{q}_\alpha, \tag{0.16}$$

and

$$f_{\alpha\beta}(t_\alpha, t_\beta) = \vec{\tilde{\pi}}_\alpha \boldsymbol{A}_{\alpha\alpha}^{t_\alpha - 1} \boldsymbol{A}_{\alpha\beta} \boldsymbol{A}_{\beta\beta}^{t_\beta - 1} \vec{q}_\beta, \tag{0.17}$$

where $\vec{\tilde{\pi}}_\alpha$, $\vec{q}_\alpha$ and $\vec{q}_\beta$ are defined correspondingly as for the multidimensional dwell-time distribution.

The continuous and discrete dwell-time distributions were in this study mostly used for plotting, but are also needed for a deeper understanding of the concept of equivalence.

Equivalence in aggregated HMMs

As briefly mentioned in the introduction, two aggregated HMMs are considered equivalent if and only if they have identical observable statistics. The concept for equivalence is tightly connected to the dwell-time distributions in the different aggregates. One established definition of equivalence is that two models are equivalent if and only if their dwell-time distributions of all dimensions are identical[5,6]. However, the definition of equivalence most useful for this work has been that of Ito et al.[18], where the dwell-time distributions are not used explicitly for the definition. There, equivalence is instead defined by looking at the likelihoods of the two different models, a definition that is here freely translated from Ito et al. to match our notations.

Definition 1. *Two aggregated HMMs with model parameters and model structures* (θ, η) *and* (θ', η') *are equivalent if and only if*

$$p(\vec{x} \mid \theta, \eta) = p(\vec{x} \mid \theta', \eta'), \tag{0.18}$$

for any observed data sequence $\vec{x}$*, where* $p(\vec{x} \mid \theta, \eta)$ *is the likelihood of the model described by* (θ, η).

Ito et al.[18] has also proven a very useful theorem that states that two discrete-time aggregated Markov models are equivalent if and only if there exists an linear map connecting the two parameter sets θ and θ'. Kienker[6] had earlier found a similar relationship concerning equivalent models in continuous time and under the constraint of steady state. Here, the results of Ito and notations of Kienker are used to state the following theorem about *similarity transformations* of equivalent models.

Theorem 1. *Let* $(\vec{\pi}, \boldsymbol{A}, \eta)$ *and* $(\vec{\pi}', \boldsymbol{A}', \eta)$ *describe two different aggregated hidden Markov models with the same model structure* η*. The models are then equivalent if and only if there exists a non-singular, similarity transformation matrix* $\boldsymbol{S}$ *such that*

$$\boldsymbol{A}' = \boldsymbol{S}^{-1}\boldsymbol{A}\boldsymbol{S}, \tag{0.19}$$

$$\vec{\pi}' = \vec{\pi}\boldsymbol{S}, \tag{0.20}$$

where $\boldsymbol{S}$ *has the form*

$$\boldsymbol{S} = \begin{pmatrix} \boldsymbol{S}_{\beta_1\beta_1} & 0 & \cdots & 0 \\ 0 & \boldsymbol{S}_{\beta_2\beta_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{S}_{\beta_n\beta_n} \end{pmatrix}, \tag{0.21}$$

the block matrices $\boldsymbol{S}_{\beta\beta}$ *are* N_β *by* N_β *matrices and all rows of* $\boldsymbol{S}$ *sums to 1.*

The original theorem as stated by Ito et al.[18] is somewhat more general than the one given here. The assumption about identical model structures is not made there, but is here made to make use of direct matrix multiplications with $\boldsymbol{S}^{-1}$ and $\boldsymbol{S}$ instead of abstract linear maps.

Theorem 1 makes it possible to group all models related through a similarity transformation into an *equivalence class*. Every set of model parameters $\theta = (\vec{\pi}, \boldsymbol{A})$ belonging to the same equivalence class is then by definition equivalent, and the likelihood of every member of the equivalence class is known after calculating it for one of the members. In this work this property was used when many likelihood evaluations were needed in the given inference algorithm and samples could be generated in a somewhat arbitrary way (which is the case for nested sampling). When possible, new samples were generated by a similarity transformation to reduce the number of calls to the computationally expensive likelihood function.

Canonical forms

The similarity transformations described in the previous section enables the use of canonical forms to uniquely represent each equivalence class, that is writing the θ on a specific form so that the θ uniquely labels one and only one equivalence class. After similarity transforming two equivalent models to the canonical form, both models are identical. Thus can canonical forms, among other things, be used to decide if two models are equivalent or not.

The canonical forms that have been used during this study are Bauer-Kienker uncoupled[6] (BKU) and Manifest Interconductance Rank[19] (MIR). Both of these canonical forms are found by disallowing some state transitions (setting elements in $\boldsymbol{A}'$ to 0) through a similarity transformation specified by the $\boldsymbol{A}$ before the similarity transformation. All transitions between hidden states in the same aggregate are disallowed after a model has been transformed to BKU form[6]. This is achieved by choosing $\boldsymbol{S}$ to diagonalize the diagonal

blocks in $\boldsymbol{A}$. The MIR canonical form is also based upon diagonalization but not of the same block matrices as for the BKU form. The MIR similarity transformation results in a model with the minimum number of non-zero transition probabilities possible for the given model structure[19], which might be desirable for some inference applications if the number of parameters is to be kept to a minimum.

These two canonical forms were originally defined in continuous time[6,19] with the generator matrix $\boldsymbol{Q}$ as a model parameter instead of the transition probability matrix $\boldsymbol{A}$. The proofs that the two canonical forms are identifiable, i.e. that each equivalence class has one and only one member on the canonical form, were also done with this representation[6,19]. It turns out however, that the similarity transformation matrices $\boldsymbol{S}$ used for transformation to the canonical forms can be found in discrete time with exactly the same algorithms as in continuous time (only replacing $\boldsymbol{Q}$ with $\boldsymbol{A}$ in the algorithms). Also, with the same arguments as for the continuous case as discussed by Bruno et al.[19], the canonical forms are identifiable in discrete time since there only exists one normalized similarity transformation that diagonalizes a non-degenerate matrix. The proof given by Bruno et al. showing that the canonical forms are identifiable can thus be directly applied on $\boldsymbol{A}$ instead of $\boldsymbol{Q}$, which shows that the canonical forms are identifiable also in discrete time. However, there exists a constraint for identifiability stating that $\boldsymbol{Q}$ (or $\boldsymbol{A}$ in this case) has to have non-degenerate eigenvalues. This is not a very tough constraint since matrices with degenerate eigenvalues are infinitely more uncommon than matrices with non-degenerate eigenvalues. Thus will these degenerate matrices be infinitely uncommon in real-life applications with the canonical forms. This is at least true in a numerical sense when generating $\boldsymbol{A}$ randomly according to some probability distribution that is not biased towards degenerate $\boldsymbol{A}$, and can thus also be considered likely for biological systems since nothing indicates that these rare $\boldsymbol{A}$ should occur more often in biology.

Intuitively, one might naively think that the invention of the canonical forms solves all problems regarding inference in aggregated HMMs, since they remove all degeneracy from the problem and might therefore be used for model selection with methods like VB that has been proven to work well for unaggregated models. This is however not the case since many aggregated HMMs have some negative transition probabilities in $\boldsymbol{A}$ when written on the canonical forms. Since VB is not able to handle such unphysical model representations, it is not possible to always use this method for approximating the evidence for fits to the canonical forms.

The algorithms described by Kienker[6] (BKU) and Bruno et al.[19] (MIR) to transform a model to the canonical forms were implemented in MATLAB, and then used to find the canonical forms when needed.

Generation of observed data sequences

All data that was analysed during this study was generated *in silico* and without any process or measurement noise. Given a set of model parameters $\theta = (\vec{\pi}, \boldsymbol{A})$ and a model structure $\eta = (N, \vec{f})$ an observed sequence $\vec{x}$ of length T was generated according to the following algorithm.

s_1 = random number generated by the probability distribution described by $\vec{\pi}$
$x_1 = f(s_1)$
for $t = 2:T$
  s_t = random number generated by the probability distribution described by row s_{t-1} in $\boldsymbol{A}$
  $x_t = f(s_t)$
end of for

This algorithm was implement in MATLAB and when multiple observed sequences were needed they were generated by putting multiple subsequent $\vec{x}$ in the same long vector, and by keeping track of their start and end based on their length.

The data used for the analysis results found in this report was generated from two different aggregated HMMs, designed to have different yet comparable dwell-time distributions in the aggregated states (see the Results chapter for more information). For both test models the model structure was set to $(N, \vec{f}) = (3, (1 \quad 2 \quad 2))$ and $\vec{\pi}$ to the steady state probabilities given implicitly by the chosen $\boldsymbol{A}$. The $\theta = (\vec{\pi}, \boldsymbol{A})$ were for Model 1 chosen as

$$\vec{\pi} = (0.2634 \quad 0.2557 \quad 0.4809), \tag{0.22}$$

$$\boldsymbol{A} = \begin{pmatrix} 0.9 & 0.018 & 0.082 \\ 0.024 & 0.91 & 0.066 \\ 0.042 & 0.038 & 0.92 \end{pmatrix}, \tag{0.23}$$

and for Model 2 as

$$\vec{\pi} = (0.2658 \quad 0.3531 \quad 0.3811), \tag{0.24}$$

$$\boldsymbol{A} = \begin{pmatrix} 0.9 & 0.095 & 0.005 \\ 0.017 & 0.922 & 0.061 \\ 0.054 & 0.006 & 0.94 \end{pmatrix}. \tag{0.25}$$

For Model 1, the BKU and MIR canonical representations are physically valid, meaning that all elements of the canonical $\boldsymbol{A}$ are positive. For Model 2 on the other hand, the BKU and MIR canonicals forms are unphysical with some negative elements in $\boldsymbol{A}$. Because of this, it was only possible to do VB model selection on the canonical forms for Model 1, since VB is unable to handle unphysical $\boldsymbol{A}$.

Likelihood and prior

All three proposed methods for numerically estimating the evidence according to the integral in Eq. (1.2) is dependent on evaluating the likelihood $p(\vec{x}\,|\,\theta,\eta)$ and the prior $p(\theta\,|\,\eta)$ for a given set of $\vec{x},\theta$, and η. As it happens, there exists standard methods for such evaluations which are explained in this section.

An analytic expression of the likelihood can be obtained after writing down the probability of a hidden state sequence conditional on the model parameters, and the probability of an observed sequence conditional on the hidden state sequence and the model structure. These two probabilities are given by

$$p(\vec{s}\,|\,\theta) = p(\vec{s}\,|\,\boldsymbol{A},\vec{\pi}) = \pi_{s_1}\prod_{t=1}^{T-1} A_{s_t,s_{t+1}}, \tag{0.26}$$

and

$$p(\vec{x}\,|\,\vec{s},\eta) = p(\vec{x}\,|\,\vec{s},\vec{f}) = \prod_{t=1}^{T}\delta_{f(s_t),x_t}, \tag{0.27}$$

where δ_{ij} is the Kronecker delta defined by

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}. \tag{0.28}$$

Because the joint distribution of the hidden and observed data conditional on the model parameters and structure can be factorized into a product of the two above given probabilities, the likelihood for an HMM can be calculated as a sum of such products over all possible hidden state sequences according to

$$p(\vec{x},|\,\theta,\eta) = \sum_{\vec{s}} p(\vec{x},\vec{s}\,|\,\theta,\eta) = \sum_{\vec{s}} p(\vec{s}\,|\,\boldsymbol{A},\vec{\pi})\,p(\vec{x}\,|\,\vec{s},\vec{f}). \tag{0.29}$$

Direct evaluation of this expression is in almost all cases computationally infeasible since the number of terms in the sum grows exponentially with the length of $\vec{x}$. The likelihood in Eq. (1.29) can instead be evaluated by sweeping through all possible hidden state sequences $\vec{s}$ by a well established implementation of dynamic programming called the forward-backward algorithm[4], which computes the likelihood in tractable time.

The function implementing the forward-backward algorithm was here taken directly from vbSPT[13]. This code is written in C to speed up the computation[14] and this function was called during this work when evaluation of the likelihood was needed in the higher level MATLAB code implementing VB, naive exact inference or nested sampling.

The priors used during this study were all chosen to be Dirichlet distributed, mainly because the Dirichlet distribution is its own conjugate[10], which has nice effects on the variational

distributions used during the VB estimation of the evidence (see Appendix 1). More precisely, $\vec{\pi}$ and the rows of $\boldsymbol{A}$ were chosen to be distributed on this form where the prior $p(\theta \mid \eta)$ can be written as

$$p(\theta \mid \eta) = p(\boldsymbol{A}, \vec{\pi} \mid \eta) = p(\boldsymbol{A} \mid \eta)\, p(\vec{\pi} \mid \eta), \tag{0.30}$$

where,

$$p(\boldsymbol{A} \mid \eta) = \prod_{i=1}^{N} p(A_{i,:} \mid \eta) = \prod_{i=1}^{N} \mathrm{Dir}(A_{i,:} \mid \tilde{w}_{i,:}^{(\boldsymbol{A})}), \tag{0.31}$$

$$p(\vec{\pi} \mid \eta) = \mathrm{Dir}(\vec{\pi} \mid \tilde{w}^{(\vec{\pi})}), \tag{0.32}$$

$A_{i,:}$ denotes the ith row of $\boldsymbol{A}$, and $\tilde{w}_{i,:}^{(\boldsymbol{A})}$ and $\tilde{w}^{(\vec{\pi})}$ are parameters of the Dirichlet distributions.

The results reported here were all obtained with flat prior distributions, i.e. priors where $\tilde{w}_{i,:}^{(\boldsymbol{A})}$ and $\tilde{w}^{(\vec{\pi})}$ were chosen so that all $A_{i,:}$ and $\vec{\pi}$ were equally probable *a priori*. A flat prior is invariant under similarity transformations of the parameters, which might be desirable to get an unbiased evidence estimation. Flat Dirichlet distributions have previously been used for successful VB inference on unaggregated FRET data[7], while the standard choice in vbSPT is a prior that makes it possible to make assumptions about the mean dwell-time in the hidden states[13].

For evidence calculations with the model structure $(N, \vec{f}) = (2, (1 \quad 2))$ the prior parameters were set to

$$\tilde{w}_{i,:}^{(\boldsymbol{A})} = \tilde{w}^{(\vec{\pi})} = (1 \quad 1), \tag{0.33}$$

and for the model structure $(N, \vec{f}) = (3, (1 \quad 2 \quad 2))$ the corresponding choice of parameters were

$$\tilde{w}_{i,:}^{(\boldsymbol{A})} = \tilde{w}^{(\vec{\pi})} = (1 \quad 1 \quad 1). \tag{0.34}$$

The only exception for a truly flat prior was when model fitting was done to the canonical BKU and MIR forms for the model structure $(N, \vec{f}) = (3, (1 \quad 2 \quad 2))$. Since these canonical forms are based upon setting some of the elements in the transition probability matrix $\boldsymbol{A}$ to zero, the $\tilde{w}_{i,:}^{(\boldsymbol{A})}$ were in these cases chosen so that the prior $p(\theta \mid \eta)$ was zero for $\boldsymbol{A}$ that were not written on the canonical form. When fitting models to the BKU canonical form $\tilde{\boldsymbol{w}}^{(\boldsymbol{A})}$ was chosen as

$$\underset{\sim}{\boldsymbol{w}}^{(\boldsymbol{A})} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \tag{0.35}$$

when model fitting was done on the MIR canonical form $\underset{\sim}{\boldsymbol{w}}^{(\boldsymbol{A})}$ was set to

$$\underset{\sim}{\boldsymbol{w}}^{(\boldsymbol{A})} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \tag{0.36}$$

while $\underset{\rightleftharpoons}{w}^{(\vec{\pi})}$ was still chosen according to Eq. (1.34) in both cases.

The algebraic expression of the Dirichlet probability density function (see Eq. (2.40), Appendix 1) was for this study implemented in a MATLAB function, and this function was later called when evaluation of the prior was needed.

Variational Bayes inference

Variational Bayes inference has previously been shown to work well for model selection on unaggregated HMMs from FRET[7] and SPT[13] data. The invention[9] of this method has been very significant since it in general converges much faster than exact methods and because other methods for approximately maximizing the evidence, such as the Bayesian information criterion (BIC), make many assumptions that are not applicable for HMMs. The BIC assumes totally uncorrelated data[7], which is not true here since the elements in the observed sequence $\vec{x}$ depend on the hidden state at the previous time step. Also, BIC assumes that the likelihood $p(\vec{x} \mid \theta, \eta)$ is sharply and uniquely peaked[7], which is not true for aggregated HMMs because of the equivalence of different parameter sets θ. For aggregated HMMs the likelihood function is not only not uniquely peaked; it is not peaked at all. Instead a multidimensional ridge of maximum likelihood is expected for the aggregated HMMs, representing an equivalence class of maximum likelihood. Even though VB inference has been proven to work well for unaggregated models, the question if it can or cannot be used for model selection on aggregated HMMs has before this study been an unresolved problem.

The VB framework relies upon the maximization of a lower bound of the evidence with the respect to two separable variational distributions $q(\theta)$ and $q(\vec{s})$ using the calculus of variations. These two distributions are obtained through a mean field approximation of the distribution $p(\theta, \vec{s}_{1:T} \mid \vec{x}_{1:T}, \eta)$ given by

$$p(\theta, \vec{s} \mid \vec{x}, \eta) = q(\theta, \vec{s}) \approx q(\theta) q(\vec{s}). \tag{0.37}$$

Maximizing the lower bound of the evidence with this approach is equivalent to minimizing the Kullback-Leibler (KL) divergence of $p(\theta, \vec{s} \mid \vec{x}, \eta)$ from $q(\theta) q(\vec{s})$. Since this KL-

divergence is a measure of the distance between $p(\theta,\vec{s}\,|\,\vec{x},\eta)$ and $q(\theta)q(\vec{s})$, the product of the variational distributions is by this definition certain to be an approximation of the real distribution, as long as the factorisation in Eq. (1.37) is a good approximation. The obtained maximized lower bound of the evidence can then be used as a valid approximation of the real evidence as long as the mean field assumption is a good one for the current problem at hand.

For this study the VB algorithm was implemented in MATLAB, an implementation that greatly resembles and builds upon the one used by vbSPT[13] (see Appendix 1 for a derivation of the algorithm). The start guess for the VB algorithm (used to give the start guess for $q(\theta)$) was in all cases chosen as the correct θ that the data was generated with. More precisely, since $q(\theta)$ is a product of Dirichlet distributions (see Appendix 1), the start guess for $q(\theta)$ was given by Dirichlet parameters set to $10^5 \cdot \theta$, where θ were the parameters that generated the data.

Naive exact inference

The evidence was also calculated by a naive implementation of Monte Carlo integration. In this approach the arithmetic mean of the likelihood was calculated when θ was sampled from the prior. At the limit of infinitely many samples this mean can be seen to be equal to the evidence according to

$$Z = \int p(\vec{x}\,|\,\theta,\eta)\,p(\theta\,|\,\eta)\,d\theta = \left\langle p(\vec{x}\,|\,\theta,\eta)\right\rangle_{p(\theta|\eta)} = \left\langle L(\theta)\right\rangle_{p_0(\theta)}. \tag{0.38}$$

In the implementation used here, the maximum likelihood value found by the Baum-Welch algorithm[4] was factorized out of the integral to avoid numerical underflow. What was actually computed was therefore

$$Z_{exact} = L_{\max}\left\langle \frac{L(\theta)}{L_{\max}}\right\rangle_{p_0(\theta)} = L_{\max}\sum_{i=1}^{N_{samples}} \frac{1}{N_{samples}}\frac{L(\theta_i)}{L_{\max}}. \tag{0.39}$$

Convergence of the calculation was checked by both looking at a plot of the evidence as a function of the number of samples used and then terminating the algorithm when the plot of the function had visually converged, but also by looking at the mean and standard error of the evidence calculated from sixteen independent workers running in parallel and thereby getting an estimation how much the results from different independent calculations varied. With this approach it became possible to vary the number of samples used in each evidence estimation to ensure sufficient convergence for the given number of observed sequences included in $\vec{x}$.

Nested sampling

An alternative method for doing exact inference with nested sampling[11,12] was also investigated. In nested sampling the prior is sampled in a more clever way than in the naive approach described above. For big datasets, only a very small fraction of the prior has a corresponding likelihood value that significantly contributes to the evidence integral, giving

that this form of more efficient sampling is needed when the data amount increases. The nested sampling algorithm relies upon the transformation of the multidimensional, difficult to evaluate evidence integral in Eq. (1.2) into a one dimensional integral with a well behaving integrand. This is done by first defining X as the fraction of prior mass where the likelihood $L(\theta)$ is larger than some lower bound L^* so that

$$dX = p_0(\theta)d\theta, \tag{0.40}$$

and

$$X(L^*) = \int_{L(\theta)>L^*} p_0(\theta)d\theta. \tag{0.41}$$

The evidence integral in Eq. (1.2) can then be re-written as

$$Z = \int_0^1 L(X)dX, \tag{0.42}$$

where $L^* \equiv L(X)$ is the inverse of the function given by equation (1.41), a monotonic decreasing function that after sampling can be integrated over by standard methods for numerical integration in one dimension (Fig. 4.)

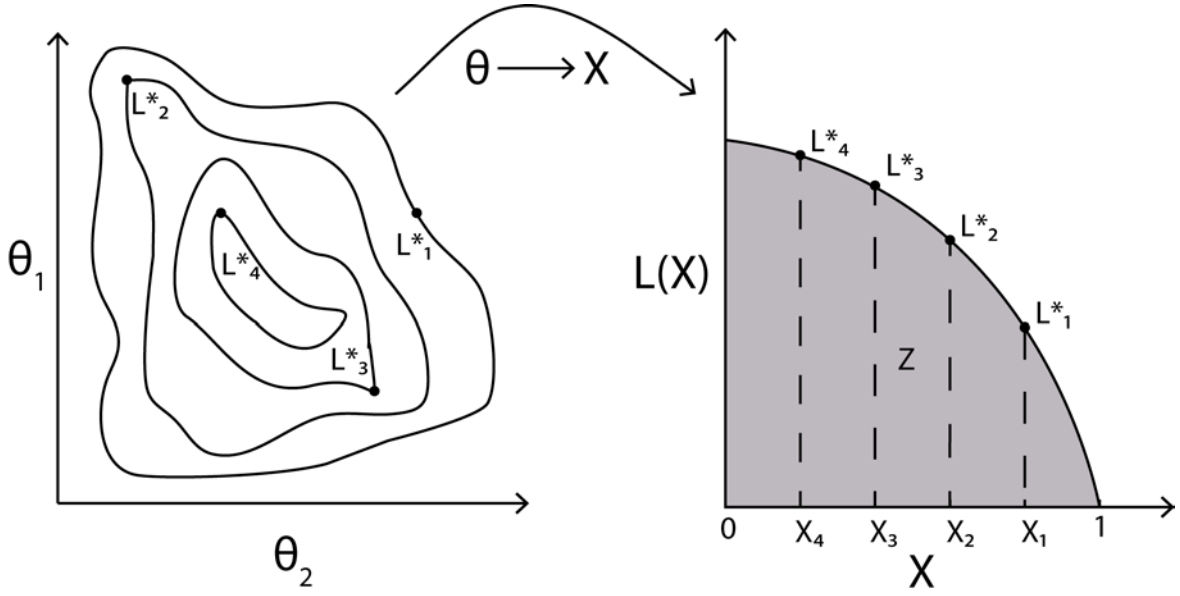


Figure 4. A two dimensional likelihood function is transformed to one dimension. The plot to the left shows a contour plot of the likelihood for four different nested shells L_i^*. The likelihood can be transformed from the vector domain θ to the scalar domain X so that the evidence Z can be estimated as a weighted sum over all $L(X_i)$. $L(X)$ is a monotonically decreasing function that typically has a large magnitude for very small X and much smaller magnitude for bigger X. This means that it is only the left most parts of the plot of the function that contributes to the integral Z (not demonstrated in the cartoon above).

The whole point of the nested sampling algorithm is to generate many samples in the regions where $L(\theta)$ is significantly larger than zero i.e. where the contribution to the evidence integral is big. This is achieved by generating new samples of θ from the prior with the constraint $L(\theta) > L^*$ and by increasing L^* for every new sample that is generated.

The algorithm is initially given M different objects that are sets of model parameters θ_i generated from the prior. This corresponds to sampling from the prior with the constraint $L > 0$. Since the objects are generated from the prior the corresponding prior masses X_i are uniformly distributed between zero and one. If sample number j is the current active object with the lowest likelihood, the algorithm then sets $L^* = L_j$ and $X^* = X_j$, discards object number j (but keeps track of it as a sample for the evidence estimation) and replaces it with a new sample of θ generated from the prior with the constraint $L(\theta) > L^*$, and continues like this until the evidence converges. The values of X_i for the samples are actually not known, but the distribution from which the X of a new active object was generated are always known (uniform between zero and X^*). This distribution enables the X_i to be estimated and the evidence to be calculated as the area under the graph of the function $L(X)$.

The step in the nested sampling algorithm of generating new, independent samples of θ from the prior with the constraint $L(\theta) > L^*$ is easier said than done. The major work done with nested sampling during this study has been trying to get an algorithm that can generate such samples to work. This is also the part of the algorithm where the special properties of the aggregated HMMs has to be considered most thoroughly, while other parts of the algorithm such as estimating the evidence from $L(X_i)$ and X_i values can be implemented similarly as in previous works[11,20,21]. This is why the results given in this report mainly focus on the part of generating new samples and not so much on other higher level parts of the algorithm.

The nested sampling implementations used during this work all use different Markov chain Monte Carlo (MCMC) methods for sampling of the prior with a likelihood constraint. All algorithms tested centred around starting from one of the current objects in the nested sampling algorithm and from this θ then randomly navigating the prior with a combination of small Gaussian steps and longer steps in θ achieved by similarity transformations. An acceptance ratio given by the Metropolis choice[22] was used in all implementations and to obey the likelihood constraint new samples θ were only accepted if $L(\theta) > L^*$. The step type of a given step (Gaussian or similarity transformation) was determined stochastically with a certain predefined probability and the step lengths of the Gaussian steps (effectively determined by the variance of the distribution that generated the step) were adaptively changed at checkpoints in the nested sampling algorithm when the acceptance ratio of an Gaussian step deviated to much from 50 %. The different implementations of MCMC tested here varied mostly in how the Gaussian steps were taken (perturbing the entire $\boldsymbol{A}$ matrix in the same step, perturbing an entire row in $\boldsymbol{A}$ in the same step or perturbing the elements pair wise etc.). However, all tested algorithms showed similar results, and none of them generated

uncorrelated θ sufficiently fast. This is why more detailed descriptions of the algorithms are left out from this report. The attempt to MCMC sample the prior did however give some useful insight about the properties of the problem which can be useful for future works (see Results chapter for more information).

Comparing the evidence of different model structures

To compare the evidence calculated for different model structures, the difference between the natural logarithm of the evidence of interest Z and some reference evidence Z_{ref} (often the evidence for the model structure $(N, \vec{f}) = (2, (1 \quad 2))$) was calculated according to

$$\text{score} = \ln Z - \ln Z_{ref}. \tag{0.43}$$

To determine the data dependency of this score, it was in many cases plotted against the total number of time steps included in the observed sequences in the data. The break-point (intersection with x-axis) of selecting the model structure of interest instead of the reference model structure was then estimated by linear interpolation in log log scale between the two points adjacent to the x-axis.

Results

Varying the amount of data in VB model selection

To test how much data VB needs to predict the correct model size for some given test model, the VB algorithm was run with different model structures η on data generated *in silico* with the model structure $(N, \vec{f}) = (3, (1 \quad 2 \quad 2))$. When this was done for different numbers of observed sequences $\vec{x}$ of length eleven (ten time steps in each sequence) generated from Model 1, about $3 \cdot 10^6$ time steps were needed in the data before VB gave a higher evidence score to the correct model structure $(N, \vec{f}) = (3, (1 \quad 2 \quad 2))$ than to the model structure $(N, \vec{f}) = (2, (1 \quad 2))$ (Fig. 5). The corresponding break-point for choosing the correct model size of three was somewhat lower when doing model fitting to the canonical BKU and MIR forms (about $1.5 \cdot 10^6$ time steps in the data).

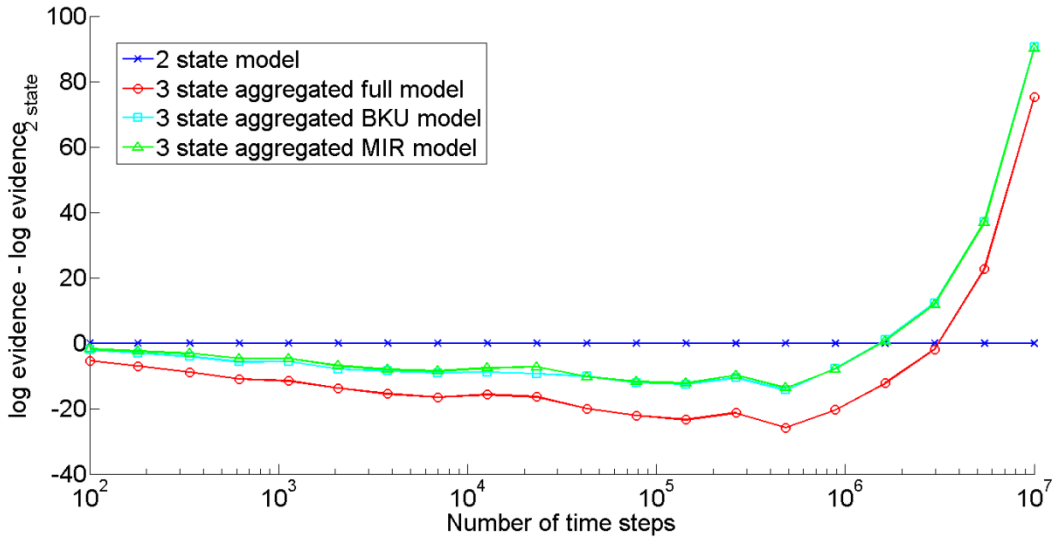


Figure 5. Evidence estimated by VB for different number of observed sequences included in the data. The VB log evidence of different model fits minus the VB log evidence of the two state model fit, calculated for data generated from Model 1 and plotted against the number of time steps included in the data. Ten time steps were present in each observed sequence included in the data and the mean dwell-times in the aggregates were 10 (aggregate 1) and 28 (aggregate 2) time steps respectively.

Effect of different dwell-time distributions on VB model selection

To test how the dwell-time in the aggregated states affects the VB model selection, the Models 1 and 2 were designed to have comparable yet different one dimensional dwell-time distributions in the aggregate with two hidden states. The distribution for both models gives a mean dwell-time of 28 time steps, but for Model 1 the dwell-time probability mass function is monotonically decreasing while it for Model 2 has a clear peak around 10 time steps (Fig. 6a). Because the dwell-time distribution in the unaggregated hidden state (hidden state 1) is identical for both models, the dwell-time distribution in the aggregated states is the only statistical difference between the models. When data then was generated with these models and analysed with VB, the effect of the different dwell-time distributions on VB model selection could be investigated. The results show that VB needs approximately 100 times less data to correctly chose the correct three state aggregated model instead of the two state unaggregated model when the data was generated from the model with a peaked dwell-time distribution (Model 2) compared to when it was generated from the model that has a monotonically decreasing dwell-time distribution (Model 1) (Fig. 6b).

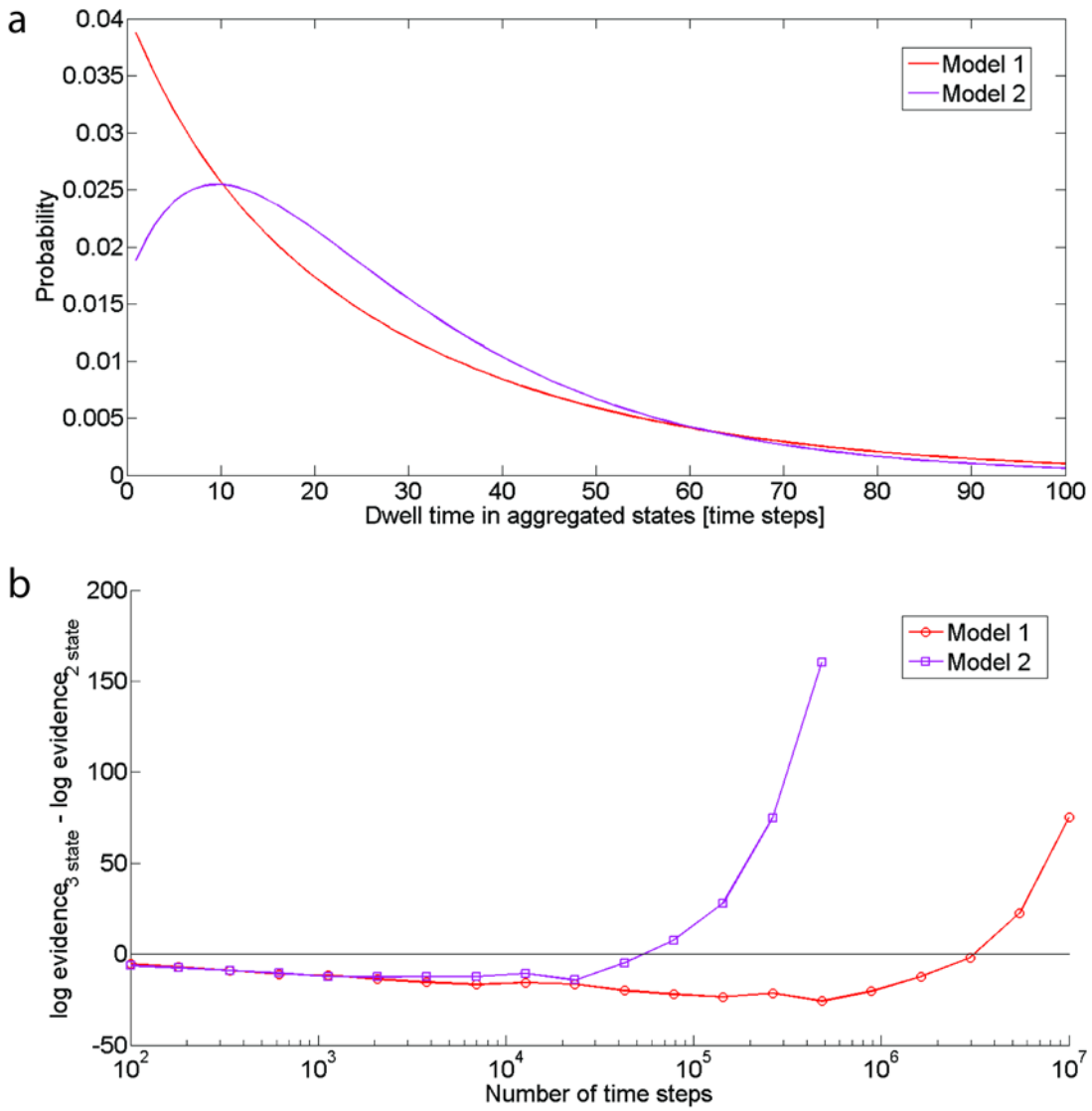


Figure 6. VB needs less data for correct model selection when the aggregated dwell-time distribution is peaked. (a) The discrete dwell-time distribution in aggregate 2 (aggregated hidden states 2 and 3) for Model 1 and 2. Both distributions have an expected value (mean dwell-time) of 28 time steps. **(b)** The VB log evidence of the three state full model fit minus the VB log evidence of the two state model fit, calculated for data generated from Model 1 and 2 and plotted against the number of time steps in the data. Ten time steps were present in each observed sequence included in the data and the mean dwell-times in the aggregates were for both models 10 (aggregate 1) and 28 (aggregate 2) time steps respectively.

The intuitive cause of this result, that VB needs less data to do correct model selection on an aggregated HMM when the dwell-time distribution is peaked, is that a peaked dwell-time distribution clearly is not a single exponential. For unaggregated HMMs, all one dimensional dwell-time distributions are by definition single exponentials. When it is easy to see that a dwell-time distribution consists of two or more exponentials, it is also easy to determine that the model is aggregated. This is a welcome result for future experimental applications since many interesting molecular biological systems (ribosome and RNA-polymerase etc.) are

likely to have peaked dwell-time distributions in the slowly diffusing aggregate (see the Discussion chapter for more information).

Varying the number of time steps per sequence in VB model selection

All results reported so far were obtained after analysing observed sequences with ten time steps in each sequence. However, to investigate how the sequence length affects the VB model selection, this property was varied in other numerical experiments. The VB algorithm was run on datasets with five different numbers of time steps per observed sequence $\vec{x}$ and the evidence of the three state model relative to that of the two state model was plotted against the number of time steps in the data (Fig. 7) as demonstrated in the previous sections. The results show that the VB algorithm needs less data for correct model selection when the observed sequences $\vec{x}$ are long compared to when they are shorter. This trend becomes even clearer when looking at the results from 50 independent experiments performed with the same procedure as described above but instead plotting the mean of all 50 experiment against the number of time steps in the data. Such experiments were performed and the break-point (intersection with x-axis) of correct model selection was determined for all sequence lengths (Table 1). For Model 2, about 20 times less data was needed for doing correct model selection when all the data was present in one long observed sequences compared to when the data had two time steps per observed sequence.

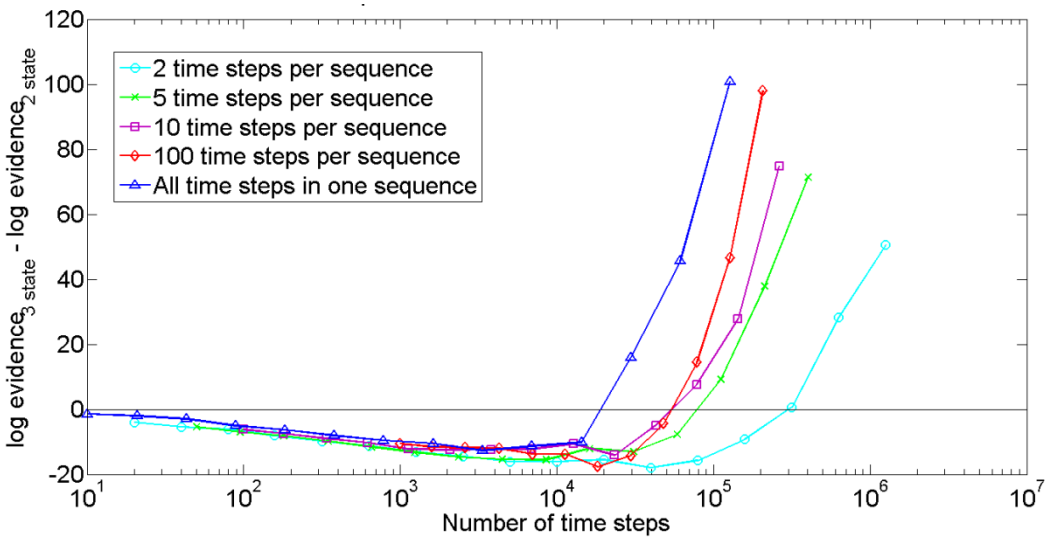


Figure 7. VB needs less data for correct model selection when the observed sequences are long. The VB log evidence of three state model fits minus the log evidence of the corresponding two state model fits, calculated for data generated from Model 2 and plotted against the number of time steps in the data. Five different curves are shown representing five different lengths of the observed sequences included in the data.

Table 1. Amount of data needed for correct model selection for different lengths of the observed sequences. The experiment generating the plot in Fig. 7 was repeated 50 times with 50 independently generated datasets and the intersection with the x-axis for each mean curve was determined by linear interpolation between the two points adjacent to the x-axis. All points were generated with the same number of time steps as in Fig. 7 and every curve did only cross the x-axis at one point with the plotting resolution used in the experiment.

Number of time steps per observed sequence	Break-point of correct model selection (intersection with x-axis)
2	$4 \cdot 10^5$
5	$1 \cdot 10^5$
10	$6 \cdot 10^4$
100	$3 \cdot 10^4$
All steps in the same observed sequence	$2 \cdot 10^4$

Comparing exact and VB evidence estimations

The most important property for any method that is estimating some quantity like the evidence is probably that the calculated estimate is relatively correct and close to the real value. To test if this is true for VB evidence estimations for aggregated HMMs, the estimated evidences from VB was compared with the evidence calculated with direct Monte Carlo integration using naive exact inference (Fig. 8). The results show that the evidence for a three state aggregated HMM is heavily underestimated by VB and that the error increases when more data is added to the evidence calculation. On the other hand, the VB evidence estimation for a two state unaggregated Markov model is very close to the correct value as calculated by naive exact inference, which supports that VB evidence estimates are accurate for unaggregated HMMs.

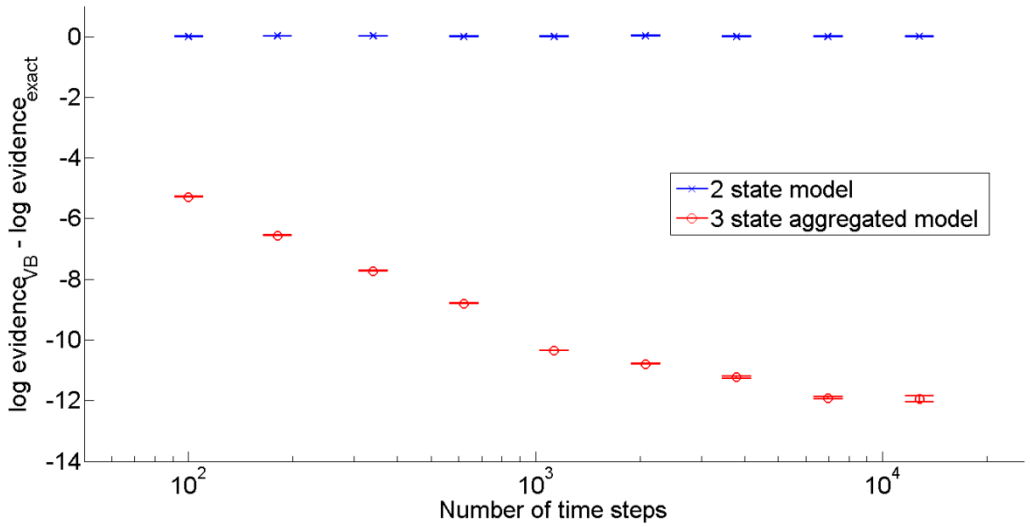


Figure 8. VB is underestimating the evidence for aggregated model fits. The VB log evidence of two and three state model fits minus the corresponding log evidence estimated by naive exact inference, calculated for data generated from Model 2 and plotted against the number of time steps in the data. The error bars represent the standard error of sixteen independent calculations. Ten time steps were present in each observed sequence included in the data.

Corresponding experiments were performed on data generated from Model 1 (Table 2), giving results that were very similar to those found for Model 2 (Fig. 8). For Model 1 it was however also possible to compare the model selection methods for fits to the BKU and MIR canonical forms. The results show that the VB evidence estimation is closer to the evidence calculated by naive exact inference for fits to the canonical forms than for fits to a full three state model. The VB evidence estimations for fits to the aggregated canonical forms are however still not at all as accurate as the VB evidence estimation for an unaggregated two state model. The relevance of calculating the evidence by naive exact inference for the canonical forms without further validation could (and maybe should) also be questioned. Only physically valid samples of θ are generated from the prior in the Monte Carlo integration giving the evidence, meaning that all equivalence classes where the canonical forms have negative elements in $\boldsymbol{A}$ are neglected in the integration. The integrand in the evidence integral is not expected to take a maximum for such unphysical $\boldsymbol{A}$ for data generated from Model 1, but one cannot with certainty say that the contribution from these equivalence classes to the integral is negligible before investigating the matter further. However, it is probably safe to say that VB is underestimating the evidence for the canonical fits. $\ln Z_{exact}$ should only be able to increase if more equivalence classes are considered in the parameter space that is sampled, giving an even bigger underestimation.

Table 2. VB is underestimating the evidence for fits to canonical aggregated models. The VB log evidence of different model fits minus the corresponding log evidence estimated by naive exact inference. The reported values are means and standard errors from sixteen independent calculations. 113 observed sequences were used in the data and ten time steps were present in each observed sequence.

Data fitted to	$\ln Z_{VB} - \ln Z_{exact}$	Standard error
2 state model	0.00136	0.00206
3 state aggregated model on BKU form	-3.83922	0.00992
3 state aggregated model on MIR form	-5.57410	0.00144
3 state aggregate full model	-9.47487	0.00860

Comparing exact and VB model selection

The results from the VB and naive exact inference were also plotted in the same way as in the experiment demonstrated in Fig. 5, to show how well VB determines the break-point in data amount needed for correct model selection. The results show that VB needs a larger amount of data for correct model selection of an aggregated three state model than the corresponding exact evidence calculation needs. For an experiment with observed sequences generated from Model 2 with ten time steps per sequence, about 7 times more data was needed by VB for correct model selection compared to the exact inference calculation (Fig. 9).

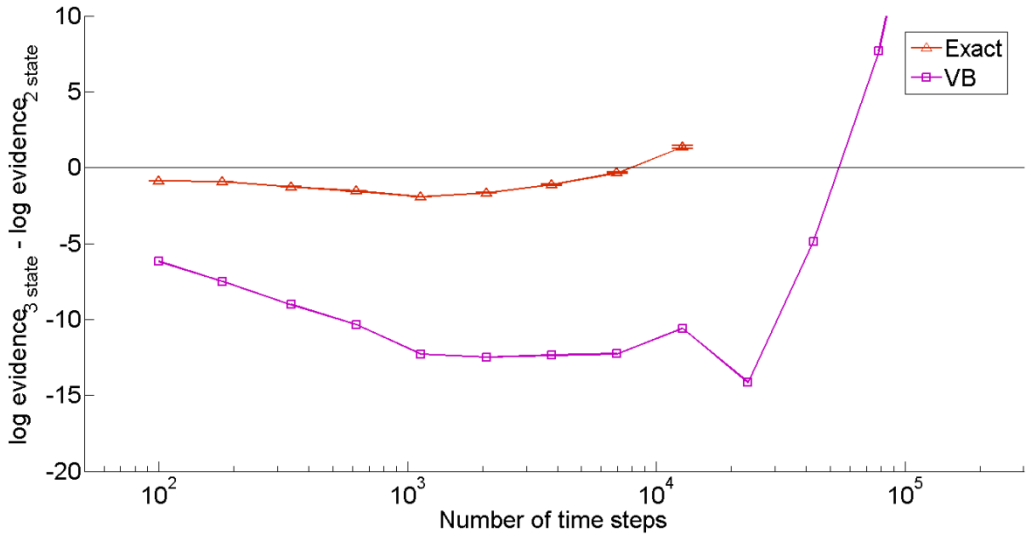


Figure 9. VB is underfitting aggregated data. The exact and VB log evidence of an aggregated three state model fit minus the corresponding log evidence of a two state model fit, calculated for data generated from Model 2 and plotted against the number of time steps included in the data. The error bars (on the exact graph) represent the standard error of sixteen independent calculations of the difference in log evidence. Ten time steps were present in each observed sequence included in the data.

Sampling of the prior in nested sampling

In the trials of doing nested sampling calculations of the evidence integral for aggregated HMMs, the prior proved to be difficult to sample by MCMC for high values of the likelihood lower bound L^*. In all MCMC sampling methods tested, the correlation time for some of the elements in the transition probability matrix $\boldsymbol{A}$ was too high to give a sufficient improvement in the rate of convergence for the nested algorithm compared to naive exact inference. The correlation between the MCMC samples was often even worse when looking at the elements of $\boldsymbol{A}$ transformed to the BKU or MIR canonical form, which might suggest that the tested methods for sampling the prior has trouble finding all equivalence classes sufficiently fast.

To further investigate what the long correlations in the sampling might be caused by, the elements of $\boldsymbol{A}$ for the MCMC samples were plotted in pairs for visualization (Fig. 10a). The plotting of this projection down to the plain of pairs of variables showed complex dependencies in the data points, suggesting that the constrained prior is a complex looking hypervolume for high likelihood constraints. Since the strategy in the tested implementations of MCMC is to perform a random walk in this parameter space, it is perhaps not surprising that the algorithms did not navigate the entire hypervolume sufficiently fast.

However, when the elements of the MIR representation of $\boldsymbol{A}$ was plotted in the corresponding way as described above, the resulting two dimensional projections did not look as complicated as for non-canonical $\boldsymbol{A}$ (Fig. 10b). For high likelihood constraints the resulting surfaces were typically approximately round or elliptical, with high variance in one direction and smaller variance in the orthogonal direction. The results suggests that it might be easier to perform a random walk and do MCMC sampling in the MIR parameter space compared to the non-canonical parameter space. Similar plots were also made for the BKU canonical form, the resulting surfaces were not as complex as the non-canonical ones, but did still have more dependencies in the data than the plots for the MIR canonical form.

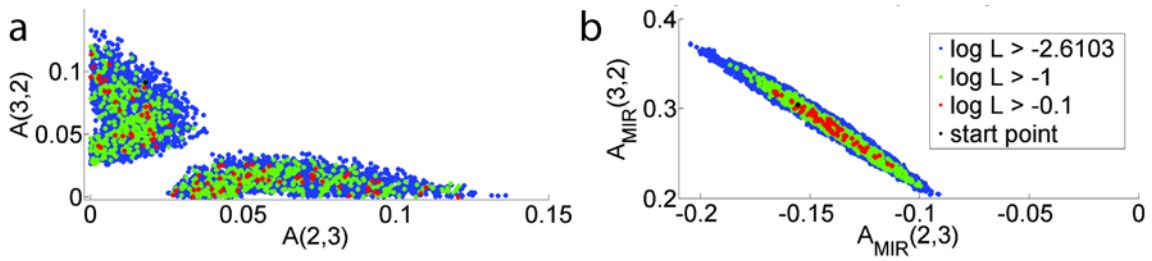


Figure 10. MCMC samples of the prior with a high likelihood constraint. 5,000 MCMC sweeps with 120 steps per sweep generating 5,000 MCMC samples from a start point given by one of the active objects in a nested sampling evidence calculation. The Gaussian step probability was set to 0.1 and the similarity transformation probability step was correspondingly set to 0.9. The likelihood constraint was set to $\ln L^* = -2.6103$ and the samples were plotted with different colours dependent on their likelihood value. The likelihood was normalized to the maximum likelihood calculated by the Baum-Welch algorithm[4] so that $\max(\ln L) = 0$. The data used consisted of 88,587 observed sequences with 10 time steps in each sequence, generated from Model 2. The model structure $(N, \vec{f}) = (3, (1 \quad 2 \quad 2))$ was used in the nested sampling and MCMC algorithms. **(a)** The values of the elements A_{32} and A_{23} for the MCMC samples **(b)** The values of the corresponding MIR transformed elements for the same samples.

The posterior approximated by VB

For further insights to why VB is bad at approximating the evidence for aggregated HMMs (Fig. 8) the posterior $q(\theta)$ approximated by VB was compared with MCMC samples of the prior generated with a high likelihood constraint. The posterior is the normalized product of the likelihood and the prior (the integrand in the evidence integral in Eq. (1.2)). With a flat prior like the one used here, the largest values of the posterior are found at the same set of parameters as the largest values of the likelihood function. With a high likelihood constraint, this is the same area in parameter space as the MCMC samples of the prior should be found. When samples from the VB posterior were plotted together with the MCMC samples of the prior in a corresponding way as in Fig. 10a, the resulting plots showed surfaces with very few properties in common (Fig. 11). As described earlier, the region of high likelihood generated by the MCMC samples is complex with many dependencies in the data, while the samples from the VB posterior are centred around the inferred parameter set with little correlations and dependencies between the parameters. This suggests that the posterior approximated by VB is a bad approximation of the real posterior, causing the KL-divergence of $p(\theta, \vec{s} \mid \vec{x}, \eta)$ from $q(\theta)q(\vec{s})$ to be big which in turns give that evidence estimated by VB is a bad approximation of the real evidence. In the implementation of VB used here, the posterior is assumed to be Dirichlet distributed. The plots shown here clearly demonstrate that this is not a good approximation for posteriors of aggregated HMMs.

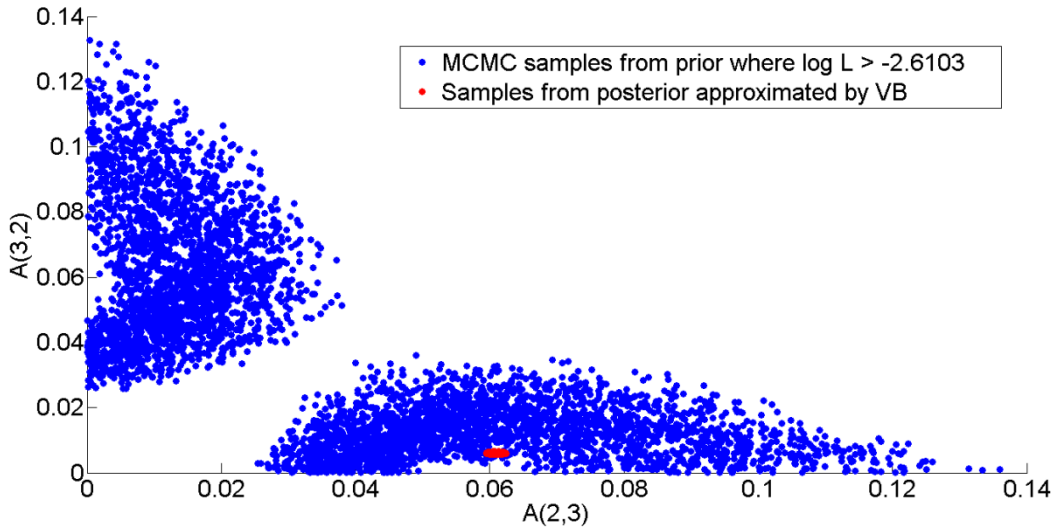


Figure 11. VB is bad at approximating the posterior of aggregated HMMs. MCMC samples of the prior (blue) generated and plotted as described in Fig. 10. Plotted together with 5,000 samples of the posterior approximated by VB inference (red) run on the same dataset.

Discussion

The major conclusion and trend found by this work is perhaps that VB inference seems to be very inappropriate to use when doing model selection on aggregated HMMs. VB is both heavily underestimating the evidence (Fig. 8) and, which is perhaps even worse, underfitting the data for much bigger data amounts than an exact method (Fig. 9). Furthermore, naive exact inference is also not a sustainable method for frequently calculating the evidence for intermediate to big datasets. For example, the exact evidence of the three state model used for calculating the final point in Fig. 9 required sixteen CPU workers running in parallel for 50 days to give a result with the standard error represented in the figure. This point had just over 10,000 time steps present in the data, which is not at all an extreme amount when considering the amount of data that single molecule experiments can produce. The time complexity of the algorithm has not been thoroughly estimated. It is however with certainty far worse than linear since the time for convergence of the points in Fig. 9 goes from approximately 10 seconds for the first point with 100 time steps in the data, to about 100,000 seconds for the next to the last point with 6,950 time steps, to over 4,000,000 seconds for the ninth and final point with 12,740 time steps. The total increase factor in the amount of data is about 100 while the corresponding increase in the computation time is over 40,000. Automation and exact determination of the time complexity is however difficult to do for the naive exact inference algorithm since convergence of the algorithm is somewhat of a subjective measure. All calculations with this algorithm reported here did seem to converge nicely when looking at a plot of the evidence as a function of the number of samples used, and the small standard error between the independent workers is also a sign of convergence. However, while most of the calculations were terminated when the standard error had decreased to just under 100 times of the mean of the evidence, such a hard constraint was not possible to use for the points using the largest amounts of data, demonstrated by their larger standard error in the plot in

Fig. 9. It is difficult to directly compare the convergence times for different points since the absolute error tolerance is smaller for earlier ones.

Even though the results show that VB approximates the evidence badly for aggregated HMMs, some of the results found here for VB are expected to be generally applicable also for valid methods for computing the evidence. The result showing that it is easier to correctly determine a model to be aggregated if the dwell-time distribution is peaked (Fig. 6) is predicted to apply also for exact calculations of the evidence, for the same reasons mentioned in the Results chapter. If this is true, it may have nice effects for the model selection on experimental data generated from many interesting molecular biological systems. The reason for this is that many experimental systems are predicted to disobey detailed balance, giving them the possibility to have peaked dwell-time distributions, and will thus probably require less data for proper model selection. A chemical process at true equilibrium is defined to obey detailed balance, meaning that all elementary reactions in the process are equilibrated by their reverse reaction. All non energy consuming processes thus obey detailed balance at equilibrium. It has previously been shown[23] that the one dimensional dwell-time distributions are always monotonically decreasing for aggregated HMMs that obey detailed balance, meaning that aggregated HMMs with peaked dwell-time distributions are not obeying detailed balance. Since many biological processes (transcription by RNA polymerase and translation by the ribosome etc.) are energy consuming, they are disobeying detailed balance, and can thus give rise to dwell-time distributions that are non-monotonically decreasing.

As mentioned briefly in the Background chapter, model selection by Bayesian inference is expected to not overfit the data, which has previously been shown for model fitting to unaggregated HMMs[7]. No evidence calculation results are reported here for model sizes larger than the correct three state model, but VB evidence estimations were also performed on larger models for a few of the datasets. However, the evidence for larger model sizes was in all cases lower than the evidence for the three state models. This result is very consistent with the expectation that Bayesian inference does not overfit data, and a more thorough investigation with larger model sizes was therefore omitted during this work to focus on the more relevant problem of underfitting with smaller models.

The implementation of nested sampling tested here did not generate samples of the prior sufficiently fast to be convenient to use in practice, but the general method still has promise to solve the problem of accurately calculating the evidence of aggregated HMMs much faster than naive exact inference. The methods tested here for generating samples from the prior were still very simplistic and basic, giving that other more refined methods might be able to solve the problem with greater success. For example, it would be very interesting to apply the sampling method of MultiNest[20] for nested sampling on aggregated HMMs. This method has been able to compute the evidence for other problems with multimodal and degenerate posteriors and might therefore also work for the posterior of aggregated HMMs. Another possible solution is to make a change of variables in the evidence integral, and perform sampling on the canonical MIR form. Since the likelihood constrained prior did not look as complex on the MIR form (Fig. 10), it might be easier to navigate the prior after the variable change. In either way, sampling methods with pure MCMC like the ones used here are

probably not the most effective. Instead a better strategy is probably, in one way or another, to try to apply some form of ellipsoidal sampling. For example, the trick behind MultiNest[20] is to enclose the prior in multiple ellipsoids that approximate the hard likelihood boundary, followed by generating samples within these ellipsoids.

Acknowledgements

First of all I would like to thank my supervisor Martin Lindén for his excellent supervision and guidance during the work of this project. I am also very grateful to Johan Elf and the entire Elf lab for all of their help and support. I would also like to give gratitude to my scientific reviewer Mats Gustafsson and to my opponent Staffan Arvidsson for their useful comments when reviewing this report. Finally, I would also like to thank my friends, girlfriend and family for always being there for me.

References

1. Neher, E. & Sakmann, B. Single-channel currents recorded from membrane of denervated frog muscle fibres. *Nature* **260,** 799–802 (1976).
2. Kim, H., Abeysirigunawarden, S.C., Chen, K., Mayerle, M., Ragunathan, K., Luthey-Schulten, Z., Ha, T. & Woodson, S.A. Protein-guided RNA dynamics during early ribosome assembly. *Nature* **506,** 334–338 (2014).
3. Nishiyama, M., Higuchi, H. & Yanagida, T. Chemomechanical coupling of the forward and backward steps of single kinesin molecules. *Nat Cell Biol* **4,** 790–797 (2002).
4. Rabiner, L. & Juang, B. H. An introduction to hidden Markov models. *IEEE ASSP Magazine* **3,** 4–16 (1986).
5. Fredkin, D. R. & Rice, J. A. On Aggregated Markov Processes. *Journal of Applied Probability* **23,** 208–214 (1986).
6. Kienker, P. Equivalence of Aggregated Markov Models of Ion-Channel Gating. *Proc. R. Soc. Lond. B* **236,** 269–309 (1989).
7. Bronson, J. E., Fei, J., Hofman, J. M., Gonzalez, R. L. & Wiggins, C. H. Learning Rates and States from Biophysical Time Series: A Bayesian Approach to Model Selection and Single-Molecule FRET Data. *Biophys J* **97,** 3196–3205 (2009).
8. Kelly, D., Dillingham, M., Hudson, A. & Wiesner, K. A New Method for Inferring Hidden Markov Models from Noisy Time Sequences. *PLoS ONE* **7,** e29703 (2012).
9. MacKay, D. J. C. *Ensemble Learning for Hidden Markov Models*. (1997).
10. MacKay, D. J. C. *Information theory, inference, and learning algorithms*. Version 7.2. (Cambridge University Press, 2003).
11. Skilling, J. Nested sampling for general Bayesian computation. *Bayesian Anal.* **1,** 833–859 (2006).
12. Sivia, D. & Skilling, J. *Data Analysis: A Bayesian Tutorial*. 2nd edition. (Oxford University Press, 2006).
13. Persson, F., Lindén, M., Unoson, C. & Elf, J. Extracting intracellular diffusive states and transition rates from single-molecule tracking data. *Nat Meth* **10,** 265–269 (2013).
14. Persson, F., Linden, M., Unoson, C. & Elf, J. vbSPT (Variational Bayes for Single Particle Tracking) - User-guide and Documentation. <http://vbspt.sourceforge.net/ref/vbSPT_userguide.pdf> (2013).
15. Sternberg, S. H., Redding, S., Jinek, M., Greene, E. C. & Doudna, J. A. DNA interrogation by the CRISPR RNA-guided endonuclease Cas9. *Nature* **507,** 62–67 (2014).

16. Colquhoun, D. & Hawkes, A. G. On the Stochastic Properties of Bursts of Single Ion Channel Openings and of Clusters of Bursts. *Phil. Trans. R. Soc. Lond. B* **300,** 1–59 (1982).
17. Li, C.-B. & Komatsuzaki, T. Aggregated Markov Model Using Time Series of Single Molecule Dwell Times with Minimum Excessive Information. *Phys. Rev. Lett.* **111,** 058301 (2013).
18. Ito, H., Amari, S.-I. & Kobayashi, K. Identifiability of hidden Markov information sources and their minimum degrees of freedom. *IEEE Transactions on Information Theory* **38,** 324–333 (1992).
19. Bruno, W. J., Yang, J. & Pearson, J. E. Using independent open-to-closed transitions to simplify aggregated Markov models of ion channel gating kinetics. *Proc. Natl. Acad. Sci. U.S.A.* **102,** 6326–6331 (2005).
20. Feroz, F., Hobson, M. P. & Bridges, M. MultiNest: an efficient and robust Bayesian inference tool for cosmology and particle physics. *Monthly Notices of the Royal Astronomical Society* **398,** 1601–1614 (2009).
21. Burkoff, N. S., Várnai, C., Wells, S. A. & Wild, D. L. Exploring the Energy Landscapes of Protein Folding Simulations with Bayesian Computation. *Biophysical Journal* **102,** 878–886 (2012).
22. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics* **21,** 1087–1092 (1953).
23. Kijima, S. & Kijima, H. Statistical analysis of channel current from a membrane patch I. Some stochastic properties of ion channels or molecular systems in equilibrium. *Journal of Theoretical Biology* **128,** 423–434 (1987).

Appendix 1 – Derivation of variational Bayes inference

The aggregated hidden Markov model

$\vec{x}_{1:T}$ is the observed data sequence in this case explicitly stating which aggregate is observed.

$\vec{s}_{1:T}$ is the hidden state sequence.

$\eta = (N, \vec{f})$ is the model structure where N is the number of hidden states and $\vec{f}$ is a vector defining which aggregate each hidden state belongs to such that $f(s_t)$ is the aggregate of state s_t.

$\theta = (\boldsymbol{B}, \vec{a}, \vec{\pi})$ are the model parameters. In this appendix the transition probability matrix $\boldsymbol{A}$ has been reparameterized into the matrix $\boldsymbol{B}$ and the row vector $\vec{a}$. Here $a_i = p(s_{t+1} \neq i \mid s_t = i)$ is the probability to leave state i and $B_{ij} = p(s_{t+1} = j \mid s_t = i, i \neq j)$ is the probability to go from state i to j, conditional on a transition occurring. This reparameterization makes it easier to chose priors that are non flat[1] and by following this convention it became possible to re-use code from previous works[1,2] when the algorithms found here were implemented in practice. The elements of transition probability matrix $\boldsymbol{A}$ can be seen to relate to the elements of $\boldsymbol{B}$ and $\vec{a}$ according to

$$A_{ij} = \delta_{ij}(1 - a_i) + (1 - \delta_{ij}) a_i B_{ij} \tag{2.1}$$

The distribution of the hidden state sequence according to an HMM is

$$p(\vec{s}_{1:T} \mid \theta) = p(\vec{s}_{1:T} \mid \boldsymbol{B}, \vec{a}, \vec{\pi}) = \pi_{s_1} \prod_{t=1}^{T-1} (1 - a_{s_t})^{\delta_{s_t, s_{t+1}}} (a_{s_t} B_{s_t, s_{t+1}})^{1 - \delta_{s_t, s_{t+1}}}. \tag{2.2}$$

Or written in another way to simplify some calculations later

$$p(\vec{s}_{1:T} \mid \theta) = p(\vec{s}_{1:T} \mid \boldsymbol{B}, \vec{a}, \vec{\pi}) = \prod_{m=1}^{N} \pi_m^{\delta_{m, s_1}} \prod_{t=1}^{T-1} \prod_{i,j=1}^{N} \left((1 - a_i)^{\delta_{ij}} (a_i B_{ij})^{1 - \delta_{ij}} \right)^{\delta_{i, s_t} \delta_{j, s_{t+1}}}. \tag{2.3}$$

The distribution of the observed aggregate sequence, conditional on the hidden state sequence is

$$p(\vec{x}_{1:T} \mid \vec{s}_{1:T}, \eta) = p(\vec{x}_{1:T} \mid \vec{s}_{1:T}, \vec{f}) = \prod_{t=1}^{T} \delta_{f(s_t), x_t}. \tag{2.4}$$

The joint distribution of the hidden and observed data conditional on the model parameters and structure can then be factorized as

$$p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) = p(\vec{s}_{1:T} \mid \boldsymbol{B}, \vec{a}, \vec{\pi}) \, p(\vec{x}_{1:T} \mid \vec{s}_{1:T}, \vec{f}). \tag{2.5}$$

Finding the optimal model structure by maximum evidence

The most probable model structure η given some observed data $\vec{x}_{1:T}$ is found by maximizing the probability $p(\eta \mid \vec{x}_{1:T})$. This probability can be obtained by marginalizing over the model parameters in the joint probability $p(\theta, \eta \mid \vec{x}_{1:T})$ so that

$$p(\eta \mid \vec{x}_{1:T}) = \int d\theta p(\theta, \eta \mid \vec{x}_{1:T}). \tag{2.6}$$

This probability can be seen to relate to the joint distribution of the hidden and observed data conditional on the model in Eq. (2.5). Marginalizing over the hidden state sequence in Eq. (2.5) gives

$$p(\vec{x}_{1:T} \mid \theta, \eta) = \sum_{s} p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta). \tag{2.7}$$

The integrand on the right side in Eq. (2.6) is the posterior probability of the likelihood $p(\vec{x}_{1:T} \mid \theta, \eta)$ given by Eq. (2.7). This posterior probability and the likelihood relates through Bayes rule as

$$p(\theta, \eta \mid \vec{x}_{1:T}) = \frac{p(\vec{x}_{1:T} \mid \theta, \eta) p(\theta, \eta)}{p(\vec{x}_{1:T})}, \tag{2.8}$$

where the likelihood $p(\vec{x}_{1:T} \mid \theta, \eta)$ is the probability of the observed data conditional on the model, $p(\vec{x}_{1:T})$ is a normalization constant and $p(\theta, \eta)$ is the prior of the model parameters and structure describing our beliefs of these quantities before observing any data. After performing the marginalization according to Eq. (2.6) the sought probability $p(\eta \mid \vec{x}_{1:T})$ becomes

$$p(\eta \mid \vec{x}_{1:T}) = \int d\theta \frac{p(\vec{x}_{1:T} \mid \theta, \eta) p(\theta, \eta)}{p(\vec{x}_{1:T})} = \frac{1}{p(\vec{x}_{1:T})} \int d\theta \sum_{s} p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) p(\theta, \eta). \tag{2.9}$$

Since the prior probability of the model structure η is independent of the model parameters θ, the prior in Eq. (2.9) can be written as $p(\theta, \eta) = p(\theta \mid \eta) p(\eta)$ which gives

$$p(\eta \mid \vec{x}_{1:T}) = \frac{1}{p(\vec{x}_{1:T})} \int d\theta \sum_{s} p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) p(\theta \mid \eta) p(\eta). \tag{2.10}$$

Under the assumption that $p(\eta)$ is constant for a reasonable interval of model sizes N, which means that there is no difference in the prior beliefs of the model structure in this interval, and by noting that the normalization constant $p(\vec{x}_{1:T})$ is independent of model parameters and structure, maximization of $p(\eta \mid \vec{x}_{1:T})$ can equivalently be done by maximizing

$$p(\vec{x}_{1:T} \mid \eta) = \int d\theta \sum_s p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) p(\theta \mid \eta). \tag{2.11}$$

Here the marginal likelihood $p(\vec{x}_{1:T} \mid \eta)$ is called the *evidence* of the model structure. In the next section this quantity will be sought by a mean field methodology enabling tractable estimation of the evidence.

Variational maximum evidence

To find the model structure η that maximizes the evidence, the evidence has to be calculated for each η. In this and similar works[2,3] the evidence is approximated by first determining a mean field approximation of the lower bound of the evidence and subsequent maximization of this approximation using the calculus of variations. Hence the name *variational* Bayes has been used to describe this kind of inference methods[4]. The derivation of the variational Bayes algorithm made here follows the derivation made for vbSPT, where the variational maximum evidence for unaggregated HMMs are found based on single particle tracking data[1,2].

The functions that the lower bound of the evidence is to be maximized with respect to are obtained through a mean field approximation of the distribution $p(\theta, \vec{s}_{1:T} \mid \vec{x}_{1:T}, \eta)$ given by

$$p(\theta, \vec{s}_{1:T} \mid \vec{x}_{1:T}, \eta) = q(\theta, \vec{s}_{1:T}) \approx q(\theta) q(\vec{s}_{1:T}), \tag{2.12}$$

where $q(\theta)$ and $q(\vec{s}_{1:T})$ are separable variational distributions that the evidence will be maximized with respect to.

The evidence in Eq. (2.11) is re-written as a functional that can make use of the mean field approximation in Eq. (2.12). This functional is obtained by multiplying and dividing the logarithm of the evidence with $q(\theta, \vec{s}_{1:T})$. Jensen's inequality is then used on this expression to get a lower bound of the evidence, which gives

$$\begin{aligned} \ln p(\vec{x}_{1:T} \mid \eta) &= \ln \int d\theta \sum_s q(\theta, \vec{s}_{1:T}) \frac{p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) p(\theta \mid \eta)}{q(\theta, \vec{s}_{1:T})} \\ &\geq \int d\theta \sum_s q(\theta, \vec{s}_{1:T}) \ln \frac{p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) p(\theta \mid \eta)}{q(\theta, \vec{s}_{1:T})} \equiv F\left[q(\theta, \vec{s}_{1:T}), \vec{x}_{1:T} \right]. \end{aligned} \tag{2.13}$$

The expression of F in Eq. (2.13) gives a lower bound for the logarithm of the evidence. To get a tractable problem, the ansatz is here to find an approximation of the evidence by maximizing the lower bound F and to do this with respect to separable variational distributions where the mean field approximation $q(\theta, \vec{s}_{1:T}) \approx q(\theta) q(\vec{s}_{1:T})$ is used.

Maximizing the lower bound of the evidence F with the respect to $q(\theta)$ and $q(\vec{s}_{1:T})$ can in fact be interpreted as minimizing a measure of the distance between the distributions

$p(\theta, \vec{s}_{1:T} \mid \vec{x}_{1:T}, \eta)$ and $q(\theta)q(\vec{s}_{1:T})$, which is what is wanted in the mean field approximation in Eq. (2.12). This can be seen if the logarithm of the evidence $\ln p(\vec{x}_{1:T} \mid \eta)$ is both added and subtracted from the expression of F in Eq. (2.13).

$$
\begin{aligned}
F\left[q(\theta, \vec{s}_{1:T}), \vec{x}_{1:T}\right] &= \ln p(\vec{x}_{1:T} \mid \eta) - \ln p(\vec{x}_{1:T} \mid \eta) + \int d\theta \sum_s q(\theta, \vec{s}_{1:T}) \ln \frac{p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) p(\theta \mid \eta)}{q(\theta, \vec{s}_{1:T})} \\
&= \ln p(\vec{x}_{1:T} \mid \eta) - \ln p(\vec{x}_{1:T} \mid \eta) - \int d\theta \sum_s q(\theta, \vec{s}_{1:T}) \ln \frac{q(\theta, \vec{s}_{1:T})}{p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) p(\theta \mid \eta)} \\
&= \ln p(\vec{x}_{1:T} \mid \eta) - \int d\theta \sum_s q(\theta, \vec{s}_{1:T}) \ln \frac{q(\theta, \vec{s}_{1:T})}{\frac{p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) p(\theta \mid \eta)}{p(\vec{x}_{1:T} \mid \eta)}} \\
&= \ln p(\vec{x}_{1:T} \mid \eta) - \int d\theta \sum_s q(\theta, \vec{s}_{1:T}) \ln \frac{q(\theta, \vec{s}_{1:T})}{p(\theta, \vec{s}_{1:T} \mid \vec{x}_{1:T}, \eta)} \\
&= \ln p(\vec{x}_{1:T} \mid \eta) - D_{KL}(q(\theta, \vec{s}_{1:T}) \,||\, p(\theta, \vec{s}_{1:T} \mid \vec{x}_{1:T}, \eta)).
\end{aligned}
\tag{2.14}
$$

Here the distribution $p(\theta, \vec{s}_{1:T} \mid \vec{x}_{1:T}, \eta) = \frac{p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) p(\theta \mid \eta)}{p(\vec{x}_{1:T} \mid \eta)}$ is introduced in the second last line and $D_{KL}(q \,||\, p)$ represent the Kullback-Leibler divergence (KL-divergence) of p from q. The KL-divergence gives a non-symmetric measure of the difference between the distributions p and q and can thus be interpreted as a semi-norm. By comparing the the expressions of the lower bound of the evidence in the Eqs. (2.13) and (2.14) it becomes clear that maximizing F with the respect to q is equivalent to minimizing $D_{KL}(q \,||\, p)$. From the properties of the KL-divergence described above this verifies that the optimal separable variational distribution $q(\theta)q(\vec{s}_{1:T})$ obtained from maximization of F is an approximation of $p(\theta, \vec{s}_{1:T} \mid \vec{x}_{1:T}, \eta)$.

The problem of finding the evidence have now been simplified to maximizing the functional F with respect to the variational distributions $q(\theta)$ and $q(\vec{s}_{1:T})$. This problem will be solved by optimization with the calculus of variations using Lagrange multipliers to enforce the constraints of normalization for the probability distributions $q(\theta)$ and $q(\vec{s}_{1:T})$. The lower bound F can equivalently be written as

$$
\begin{aligned}
F\left[q(\theta), q(\vec{s}_{1:T}), \vec{x}_{1:T}\right] &= \int d\theta \sum_s q(\theta) q(\vec{s}_{1:T}) \ln \frac{p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta)\, p(\theta \mid \eta)}{q(\theta) q(\vec{s}_{1:T})} \\
&= \int d\theta \sum_s q(\theta) q(\vec{s}_{1:T}) \ln\, p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) p(\theta \mid \eta) \\
&\quad - \int d\theta \sum_s q(\theta) q(\vec{s}_{1:T}) \ln q(\theta) - \int d\theta \sum_s q(\theta) q(\vec{s}_{1:T}) \ln q(\vec{s}_{1:T}) \\
&= \int d\theta \sum_s q(\theta) q(\vec{s}_{1:T}) \ln\, p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) p(\theta \mid \eta) \\
&\quad - \int d\theta q(\theta) \ln q(\theta) - \sum_s q(\vec{s}_{1:T}) \ln q(\vec{s}_{1:T}).
\end{aligned}
\tag{2.15}
$$

The expression of the lower bound F in Eq. (2.15) will now in turn be optimized with the respect to $q(\theta)$ and $q(\vec{s}_{1:T})$. Optimization with the respect to $q(\theta)$ while keeping $q(\vec{s}_{1:T})$ fixed can equivalently be done on the functional

$$
\begin{aligned}
F^*\left[q(\theta)\right] &= \int d\theta \sum_s q(\theta) q(\vec{s}_{1:T}) \ln\, p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) p(\theta \mid \eta) - \int d\theta q(\theta) \ln q(\theta) \\
&= \int d\theta \left[\sum_s q(\theta) q(\vec{s}_{1:T}) \ln\, p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) p(\theta \mid \eta) - q(\theta) \ln q(\theta) \right],
\end{aligned}
\tag{2.16}
$$

since the last term in Eq. (2.15) does not depend on $q(\theta)$. The problem have now been narrowed down to optimizing F^* with the constraint

$$
\int d\theta q(\theta) = 1, \tag{2.17}
$$

since $q(\theta)$ is a normalized probability density function. This problem can be solved with the Lagrange-Euler equation while using a Lagrange multiplier to ensure the constraint in Eq. (2.17). The auxiliary function G^* is therefore introduced so that

$$
G^*\left[q(\theta)\right] = G\left[q(\theta)\right] + \lambda q(\theta), \tag{2.18}
$$

where G is the integrand in Eq. (2.16)

$$
G\left[q(\theta)\right] = \sum_s q(\theta) q(\vec{s}_{1:T}) \ln\, p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) p(\theta \mid \eta) - q(\theta) \ln q(\theta), \tag{2.19}
$$

and the term $\lambda q(\theta)$ is the product between a Lagrange multiplier and the variational distribution $q(\theta)$, which enforces the normalization constraint in Eq. (2.17). According to the Lagrange-Euler equation for variational problems (here with an integral constraint) a necessary condition for an optimum is[5]

$$\frac{\partial G^*}{\partial q(\theta)} - \frac{d}{d\theta}\frac{\partial G^*}{\partial q'(\theta)} = 0, \tag{2.20}$$

and since G^* has no dependence of $q'(\theta)$ this is equivalent to

$$\frac{\partial G^*}{\partial q(\theta)} = 0. \tag{2.21}$$

With the auxiliary function G^* according to the Eqs. (2.18) and (2.19) the derivative in Eq. (2.21) can be re-written as

$$\begin{aligned}
&\frac{\partial G^*}{\partial q(\theta)} = \frac{\partial}{\partial q(\theta)}\left[G + \lambda q(\theta)\right] = \frac{\partial G}{\partial q(\theta)} + \lambda \\
&= \frac{\partial}{\partial q(\theta)}\left[\sum_s q(\theta) q(\vec{s}_{1:T}) \ln p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) p(\theta \mid \eta) - q(\theta) \ln q(\theta)\right] + \lambda \\
&= \frac{\partial}{\partial q(\theta)}\left[q(\theta)\left\{\sum_s q(\vec{s}_{1:T}) \ln p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) + \ln p(\theta \mid \eta)\sum_s q(\vec{s}_{1:T}) - \ln q(\theta)\right\}\right] + \lambda \\
&= \frac{\partial}{\partial q(\theta)}\left[q(\theta)\left(\sum_s q(\vec{s}_{1:T}) \ln p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) + \ln p(\theta \mid \eta) - \ln q(\theta)\right)\right] + \lambda \\
&= \left(\sum_s q(\vec{s}_{1:T}) \ln p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) + \ln p(\theta \mid \eta) - \ln q(\theta)\right) + q(\theta)\left(-\frac{1}{q(\theta)}\right) + \lambda \\
&= \left\langle \ln p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta)\right\rangle_{q(\vec{s}_{1:T})} + \ln p(\theta \mid \eta) - \ln q(\theta) - 1 + \lambda,
\end{aligned} \tag{2.22}$$

where the normalization constraint $\sum_s q(\vec{s}_{1:T}) = 1$ has been used on the fourth line and the product rule has been used for derivation on the fifth line. This result means that the Lagrange-Euler Eq. (2.21) can be written as

$$\ln q(\theta) = \left\langle \ln p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta)\right\rangle_{q(\vec{s}_{1:T})} + \ln p(\theta \mid \eta) - 1 + \lambda. \tag{2.23}$$

Here, the constant $-1+\lambda$ enforces normalization, and can instead be renamed

$$-1 + \lambda = -\ln \int d\theta q(\theta) = -\ln Z_\theta, \tag{2.24}$$

where Z_θ is a normalization constant. Eq. (2.24) inserted into Eq. (2.23) gives the final expression for the optimization criterion for $q(\theta)$,

$$\ln q(\theta) = -\ln Z_\theta + \left\langle \ln p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta)\right\rangle_{q(\vec{s}_{1:T})} + \ln p(\theta \mid \eta). \tag{2.25}$$

The corresponding derivation must also be done for the variational distribution $q(\vec{s}_{1:T})$, that is

optimizing F with the respect to $q(\vec{s}_{1:T})$ while keeping $q(\theta)$ fixed. The modified functional F^* that has the same stationary points (with the respect to $q(\theta)$ when $q(\vec{s}_{1:T})$ is fixed) as F is now chosen as

$$F^*\left[q(\vec{s}_{1:T})\right] = \int d\theta \sum_s q(\theta) q(\vec{s}_{1:T}) \ln p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) p(\theta \mid \eta) - \sum_s q(\vec{s}_{1:T}) \ln q(\vec{s}_{1:T})$$
$$= \sum_s \left[\int d\theta q(\theta) q(\vec{s}_{1:T}) \ln p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) p(\theta \mid \eta) - q(\vec{s}_{1:T}) \ln q(\vec{s}_{1:T}) \right]. \tag{2.26}$$

Solving the problem of optimizing F^* with the respect to $q(\vec{s}_{1:T})$ in the corresponding way as shown for $q(\theta)$ gives

$$\ln q(\vec{s}_{1:T}) = -\ln Z_s + \left\langle \ln p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) \right\rangle_{q(\theta)}. \tag{2.27}$$

The two optimization criteria for $q(\theta)$ and $q(\vec{s}_{1:T})$ are then

$$\ln q(\theta) = -\ln Z_\theta + \left\langle \ln p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) \right\rangle_{q(\vec{s}_{1:T})} + \ln p(\theta \mid \eta), \tag{2.28}$$

$$\ln q(\vec{s}_{1:T}) = -\ln Z_s + \left\langle \ln p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) \right\rangle_{q(\theta)}, \tag{2.29}$$

where Z_θ and Z_s are constants that ensure normalization of the variational distributions and $\langle ... \rangle_q$ denotes an expected value with respect to the probability distribution q.

What has been shown in this section are necessary (but not always sufficient) conditions for finding the maxima of F with the respect to $q(\theta)$ and $q(\vec{s}_{1:T})$. This means that variational distributions calculated according to the update Eqs. (2.28) and (2.29) represent stationary points of F, but it has not been shown here that they always represent a maximum.

The computational implementations and simulations done with these optimization criterion do however imply that the update Eqs. (2.28) and (2.29) really do give a maximum to F. Iterative updating of the variational distributions according to the Eqs. (2.28) and (2.29) while keeping the other variational distribution fixed, gives increasing values of F. These iterations are run until F converges and as a result the optimal $q(\theta)$ and $q(\vec{s}_{1:T})$ are also found for the current η. This iterative updating of the variational distributions can in many ways be compared with the traditional expectation-maximization algorithm (EM-algorithm) used for maximum likelihood estimation of model parameters. In the following sections it is shown how the quantities on the right side in the update Eqs (2.28) and (2.29) can be calculated, thus giving a recipe for the iterative updating of $q(\theta)$ and $q(\vec{s}_{1:T})$.

M-step of the VBEM algorithm – updating q(θ)

In the M-step of the variational Bayes expectation-maximization (VBEM) algorithm, the variational distribution $q(\theta)$ is updated in such a way that it maximises the lower bound of the evidence. This is achieved when $q(\theta)$ follows the update equation

$$\ln q(\theta) = -\ln Z_\theta + \left\langle \ln p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) \right\rangle_{q(\vec{s}_{1:T})} + \ln p(\theta \mid \eta). \tag{2.28}$$

The prior distribution $p(\theta \mid \eta)$ is chosen to factorize in a certain way which will lead to a variational distribution $q(\theta)$ that also factorizes nicely. The prior distribution is therefore chosen as

$$p(\theta \mid \eta) = p(\boldsymbol{B}, \vec{a}, \vec{\pi} \mid \eta) = p(\boldsymbol{B} \mid \eta)\, p(\vec{a} \mid \eta)\, p(\vec{\pi} \mid \eta) = p(\vec{\pi} \mid \eta) \prod_{i=1}^{N} p(B_{i,:} \mid \eta)\, p(a_i \mid \eta). \tag{2.30}$$

Here $B_{i,:}$ notes row i of the matrix parameter $\boldsymbol{B}$. With Eq. (2.5) as the joint distribution of the hidden and observed data conditional on the model according to the aggregated HMM and the above prior distribution the update Eq. (2.28) becomes

$$\begin{aligned}
\ln q(\theta) &= -\ln Z_\theta + \left\langle \ln p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) \right\rangle_{q(\vec{s}_{1:T})} + \ln p(\theta \mid \eta) \\
&= -\ln Z_\theta + \left\langle \ln p(\vec{s}_{1:T} \mid \boldsymbol{B}, \vec{a}, \vec{\pi})\, p(\vec{x}_{1:T} \mid \vec{s}_{1:T}, \vec{f}) \right\rangle_{q(\vec{s}_{1:T})} + \ln p(\vec{\pi} \mid \eta) \prod_{i=1}^{N} p(B_{i,:} \mid \eta)\, p(a_i \mid \eta) \\
&= -\ln Z_\theta + \left\langle \ln p(\vec{s}_{1:T} \mid \boldsymbol{B}, \vec{a}, \vec{\pi}) \right\rangle_{q(\vec{s}_{1:T})} + \left\langle \ln p(\vec{x}_{1:T} \mid \vec{s}_{1:T}, \vec{f}) \right\rangle_{q(\vec{s}_{1:T})} \\
&\quad + \ln p(\vec{\pi} \mid \eta) + \sum_{i=1}^{N} \ln p(B_{i,:} \mid \eta) + \sum_{i=1}^{N} \ln p(a_i \mid \eta).
\end{aligned} \tag{2.31}$$

With $p(\vec{s}_{1:T} \mid \boldsymbol{B}, \vec{a}, \vec{\pi})$ given by the definition of an HMM in Eq. (2.3), the second term of the last line in Eq. (2.31) is

$$\begin{aligned}
&\left\langle \ln p(\vec{s}_{1:T} \mid \boldsymbol{B}, \vec{a}, \vec{\pi}) \right\rangle_{q(\vec{s}_{1:T})} = \left\langle \ln \prod_{m=1}^{N} \pi_m^{\delta_{m,s_1}} \prod_{t=1}^{T-1} \prod_{i,j=1}^{N} \left((1-a_i)^{\delta_{ij}} (a_i B_{ij})^{1-\delta_{ij}} \right)^{\delta_{i,s_t}\delta_{j,s_{t+1}}} \right\rangle_{q(\vec{s}_{1:T})} \\
&= \sum_{m=1}^{N} \left\langle \delta_{m,s_1} \ln \pi_m \right\rangle_{q(\vec{s}_{1:T})} + \sum_{t=1}^{T-1} \sum_{i,j=1}^{N} \left\langle \delta_{i,s_t}\delta_{j,s_{t+1}} (1-\delta_{ij}) \ln B_{ij} \right\rangle_{q(\vec{s}_{1:T})} \\
&+ \sum_{t=1}^{T-1} \sum_{i,j=1}^{N} \left\langle \delta_{i,s_t}\delta_{j,s_{t+1}} \left((1-\delta_{ij}) \ln a_i + \delta_{ij} \ln(1-a_i) \right) \right\rangle_{q(\vec{s}_{1:T})} \\
&= \sum_{m=1}^{N} \left\langle \delta_{m,s_1} \right\rangle_{q(\vec{s}_{1:T})} \ln \pi_m + \sum_{t=1}^{T-1} \sum_{i,j=1}^{N} \left\langle \delta_{i,s_t}\delta_{j,s_{t+1}} \right\rangle_{q(\vec{s}_{1:T})} (1-\delta_{ij}) \ln B_{ij} \\
&+ \sum_{t=1}^{T-1} \sum_{i,j=1}^{N} \left\langle \delta_{i,s_t}\delta_{j,s_{t+1}} \right\rangle_{q(\vec{s}_{1:T})} \left((1-\delta_{ij}) \ln a_i + \delta_{ij} \ln(1-a_i) \right).
\end{aligned}$$

(2.32)

The third term in the last line of update Eq. (2.31) is constant for all θ given a fixed variational distribution $q(\vec{s}_{1:T})$ and can thus be integrated into the normalization constant Z_θ. Insertion of Eq. (2.32) into update Eq. (2.31) and collecting of the terms that depend on the different parameters then gives a separable variational distribution so that

$$\ln q(\theta) = -\ln Z_\theta + \ln q(\boldsymbol{B}) + \ln q(\vec{a}) + \ln q(\vec{\pi}), \tag{2.33}$$

where the logarithms of the factors in the distribution are given by

$$\ln q(\boldsymbol{B}) = \sum_{t=1}^{T-1}\sum_{i,j=1}^{N} \left\langle \delta_{i,s_t}\delta_{j,s_{t+1}} \right\rangle_{q(\vec{s}_{1:T})} (1-\delta_{ij}) \ln B_{ij} + \sum_{i=1}^{N} \ln p(B_{i,:} \mid \eta), \tag{2.34}$$

$$\ln q(\vec{a}) = \sum_{t=1}^{T-1}\sum_{i,j=1}^{N} \left\langle \delta_{i,s_t}\delta_{j,s_{t+1}} \right\rangle_{q(\vec{s}_{1:T})} \left((1-\delta_{ij}) \ln a_i + \delta_{ij} \ln(1-a_i)\right) + \sum_{i=1}^{N} \ln p(a_i \mid \eta), \tag{2.35}$$

$$\ln q(\vec{\pi}) = \sum_{m=1}^{N} \left\langle \delta_{m,s_1} \right\rangle_{q(\vec{s}_{1:T})} + \ln p(\vec{\pi} \mid \eta). \tag{2.36}$$

Direct algebraic expressions of the factors $q(\boldsymbol{B})$, $q(\vec{a})$ and $q(\vec{\pi})$ in the variational distribution $q(\theta)$ can be obtained by choosing priors that are Dirichlet and beta distributed. More specifically $p(B_{i,:} \mid \eta)$ and $p(\vec{\pi} \mid \eta)$ are chosen to be Dirichlet probability density functions and $p(a_i \mid \eta)$ is chosen to be a beta probability density function. Since the priors are chosen to be products of Dirichlet or beta distributions, the distributions $q(\boldsymbol{B})$, $q(\vec{a})$ and $q(\vec{\pi})$ are, as will be seen, distributed on the same form as the prior. This is because the conjugate prior of a Dirichlet and beta distribution is a Dirichlet and beta distribution, respectively. The priors are therefore chosen as

$$p(\boldsymbol{B} \mid \eta) = \prod_{i=1}^{N} p(B_{i,:} \mid \eta) = \prod_{i=1}^{N} Dir(B_{i,:} \mid \tilde{w}_{i,:}^{(\boldsymbol{B})}), \tag{2.37}$$

$$p(\vec{a} \mid \eta) = \prod_{i=1}^{N} p(a_i \mid \eta) = \prod_{i=1}^{N} \beta(a_i \mid \tilde{w}_{i1}^{(\vec{a})}, \tilde{w}_{i2}^{(\vec{a})}), \tag{2.38}$$

$$p(\vec{\pi} \mid \eta) = Dir(\vec{\pi} \mid \tilde{\vec{w}}^{(\vec{\pi})}), \tag{2.39}$$

where the variables $\tilde{w}$ are hyperparameters of the priors. The probability density functions of the Dirichlet distributions in Eqs. (2.37) and (2.39) can be written as

$$p(\vec{\pi} \mid \eta) = Dir(\vec{\pi} \mid \tilde{\vec{w}}^{(\vec{\pi})}) = \frac{1}{Z} \prod_{\substack{j=1 \\ \pi_j \neq 0}}^{N} \pi_j^{(\tilde{w}_j^{(\vec{\pi})} - 1)}, \tag{2.40}$$

$$p(B_{i,:}\,|\,\eta) = Dir(B_{i,:}\,|\,\tilde{w}_{i,:}^{(\boldsymbol{B})}) = \frac{1}{Z}\prod_{\substack{j=1\\ B_{ij}\neq 0}}^{N} B_{ij}^{(\tilde{w}_{ij}^{(\boldsymbol{B})}-1)}, \tag{2.41}$$

where the normalisation constant Z (here for $p(\vec{\pi}\,|\,\eta)$ as an example) is given by

$$Z = B(\tilde{\vec{w}}^{(\vec{\pi})}) = \prod_j \Gamma(\tilde{w}_j^{(\vec{\pi})})\frac{1}{\Gamma(\sum_k \tilde{w}_k^{(\vec{\pi})})}. \tag{2.42}$$

The beta distribution is a special case with a Dirichlet distribution with only two hyperparameters. The probability density function of the beta distribution in Eq. (2.38) can be written as

$$p(a_i\,|\,\eta) = \beta(a_i\,|\,\tilde{w}_{i1}^{(\vec{a})}, \tilde{w}_{i2}^{(\vec{a})}) = \frac{\Gamma(\tilde{w}_{i1}^{(\vec{a})} + \tilde{w}_{i2}^{(\vec{a})})}{\Gamma(\tilde{w}_{i1}^{(\vec{a})})\Gamma(\tilde{w}_{i2}^{(\vec{a})})} a_i^{(\tilde{w}_{i1}^{(\vec{a})}-1)}(1-a_i)^{(\tilde{w}_{i2}^{(\vec{a})}-1)}. \tag{2.43}$$

With the form of the priors known, direct expressions of the factors of the variational distribution can be determined from the logarithm of the factors in the Eqs. (2.34), (2.35) and (2.36). Using Eq. (2.34) and doing the this derivation for $q(\boldsymbol{B})$ gives

$$\begin{aligned}
\ln q(\boldsymbol{B}) &= \sum_{t=1}^{T-1}\sum_{i,j=1}^{N}\left\langle \delta_{i,s_t}\delta_{j,s_{t+1}}\right\rangle_{q(\vec{s}_{1:T})}(1-\delta_{ij})\ln B_{ij} + \sum_{i=1}^{N}\ln p(B_{i,:}\,|\,\eta) \\
&= \sum_{t=1}^{T-1}\sum_{i,j=1}^{N}\left\langle \delta_{i,s_t}\delta_{j,s_{t+1}}\right\rangle_{q(\vec{s}_{1:T})}(1-\delta_{ij})\ln B_{ij} + \sum_{i=1}^{N}\ln\frac{1}{Z_i}\prod_{\substack{j=1\\ B_{ij}\neq 0}}^{N} B_{ij}^{(\tilde{w}_{ij}^{(\boldsymbol{B})}-1)} \\
&= \sum_{t=1}^{T-1}\sum_{\substack{i,j=1\\ i\neq j}}^{N}\left\langle \delta_{i,s_t}\delta_{j,s_{t+1}}\right\rangle_{q(\vec{s}_{1:T})}\ln B_{ij} + \sum_{i=1}^{N}\ln\frac{1}{Z_i}\prod_{\substack{j=1\\ i\neq j}}^{N} B_{ij}^{(\tilde{w}_{ij}^{(\boldsymbol{B})}-1)} \\
&= \sum_{t=1}^{T-1}\sum_{\substack{i,j=1\\ i\neq j}}^{N}\left\langle \delta_{i,s_t}\delta_{j,s_{t+1}}\right\rangle_{q(\vec{s}_{1:T})}\ln B_{ij} + \sum_{\substack{i,j=1\\ i\neq j}}^{N}(\tilde{w}_{i,:}^{(\boldsymbol{B})}-1)\ln B_{ij} + \sum_{i=1}^{N} ln\frac{1}{Z_i} \\
&= \sum_{\substack{i,j=1\\ i\neq j}}^{N}\left[\left(\sum_{t=1}^{T-1}\left\langle \delta_{i,s_t}\delta_{j,s_{t+1}}\right\rangle_{q(\vec{s}_{1:T})} + \tilde{w}_{ij}^{(\boldsymbol{B})} - 1\right)\ln B_{ij}\right] - \ln Z_{\boldsymbol{B}}
\end{aligned}$$

$$\Leftrightarrow q(\boldsymbol{B}) = \frac{1}{Z_{\boldsymbol{B}}}\prod_{i=1}^{N}\prod_{\substack{j=1\\ i\neq j}}^{N} B_{ij}^{\sum_{t=1}^{T-1}\left\langle \delta_{i,s_t}\delta_{j,s_{t+1}}\right\rangle_{q(\vec{s}_{1:T})} + \tilde{w}_{ij}^{(\boldsymbol{B})} - 1}. \tag{2.44}$$

With the hyperparameters of $q(\boldsymbol{B})$ defined as

$$w_{ij}^{(\boldsymbol{B})} = \tilde{w}_{ij}^{(\boldsymbol{B})} + \sum_{t=1}^{T-1}\left\langle \delta_{i,s_t}\delta_{j,s_{t+1}}\right\rangle_{q(\vec{s}_{1:T})}, \quad i \neq j, \tag{2.45}$$

Eq. (2.44) describes a product Dirichlet distributions such that

$$q(\boldsymbol{B}) = \prod_{i=1}^{N} Dir(B_{i,:} \mid \vec{w}_{i,:}^{(\boldsymbol{B})}). \tag{2.46}$$

The corresponding derivation can be done for $q(\vec{a})$ and $q(\vec{\pi})$ which gives the following expressions for the factors of the variational distribution $p(\theta)$

$$q(\boldsymbol{B}) = \prod_{i=1}^{N} Dir(B_{i,:} \mid \vec{w}_{i,:}^{(\boldsymbol{B})}), \quad w_{ij}^{(\boldsymbol{B})} = \tilde{w}_{ij}^{(\boldsymbol{B})} + \sum_{t=1}^{T-1} \left\langle \delta_{i,s_t} \delta_{j,s_{t+1}} \right\rangle_{q(\vec{s}_{1:T})}, \quad i \neq j, \tag{2.47}$$

$$\begin{aligned} q(\vec{a}) = \prod_{i=1}^{N} \beta(a_i \mid w_{i1}^{(\vec{a})}, w_{i2}^{(\vec{a})}), \quad w_{i1}^{(\vec{a})} &= \tilde{w}_{i1}^{(\vec{a})} + \sum_{t=1}^{T-1} \left\langle \delta_{i,s_t} (1 - \delta_{i,s_{t+1}}) \right\rangle_{q(\vec{s}_{1:T})}, \\ w_{i2}^{(\vec{a})} &= \tilde{w}_{i2}^{(\vec{a})} + \sum_{t=1}^{T-1} \left\langle \delta_{i,s_t} \delta_{i,s_{t+1}} \right\rangle_{q(\vec{s}_{1:T})}, \end{aligned} \tag{2.48}$$

$$q(\vec{\pi}) = Dir(\vec{\pi} \mid \vec{w}^{(\vec{\pi})}), \quad w_i^{(\vec{\pi})} = \tilde{w}_i^{(\vec{\pi})} + \left\langle \delta_{i,s_1} \right\rangle_{q(\vec{s}_{1:T})}. \tag{2.49}$$

Some averages with the respect to these distributions are used in the subsequent E-step. These are

$$\left\langle \ln B_{ij} \right\rangle_{q(B_{i,:})} = \psi(w_{ij}^{(\boldsymbol{B})}) - \psi(w_{i0}^{(\boldsymbol{B})}), \quad w_{i0}^{(\boldsymbol{B})} = \sum_{\substack{j=1 \\ i \neq j}}^{N} w_{ij}^{(\boldsymbol{B})}, \tag{2.50}$$

$$\left\langle \ln a_i \right\rangle_{q(\vec{a})} = \psi(w_{i1}^{(\vec{a})}) - \psi(w_{i0}^{(\vec{a})}), \quad w_{i0}^{(\vec{a})} = w_{i1}^{(\vec{a})} + w_{i2}^{(\vec{a})}, \tag{2.51}$$

$$\left\langle \ln(1 - a_i) \right\rangle_{q(\vec{a})} = \psi(w_{i2}^{(\vec{a})}) - \psi(w_{i0}^{(\vec{a})}), \tag{2.52}$$

$$\left\langle \ln \pi_i \right\rangle_{q(\vec{\pi})} = \psi(w_i^{(\vec{\pi})}) - \psi(w_0^{(\vec{\pi})}), \quad w_0^{(\vec{\pi})} = \sum_{i=1}^{N} w_i^{(\vec{\pi})}, \tag{2.53}$$

where ψ denotes the digamma function.

The described hyperparameters of $q(\theta)$ are calculated in each M-step of the VBEM algorithm and $q(\theta)$ is in that way updated. The algorithm can then go on to the E-step where the averages and modes are used in the update of $q(\vec{s}_{1:T})$.

E-step of the VBEM algorithm – updating q(s)

In the E-step of the VBEM algorithm, the variational distribution $q(\vec{s}_{1:T})$ is updated in such a way that it maximises the lower bound of the evidence. This is achieved when $q(\vec{s}_{1:T})$ follows the update equation

$$\ln q(\vec{s}_{1:T}) = -\ln Z_s + \left\langle \ln p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) \right\rangle_{q(\theta)}. \tag{2.29}$$

Since the variational distribution $q(\theta)$ factorizes according to the Eq. (2.33) and with the joint distribution of the hidden and observed data conditional on the model according to Eq. (2.5), the update Eq. (2.29) can be re-written as

$$\begin{aligned}
\ln q(\vec{s}_{1:T}) &= -\ln Z_s + \left\langle \ln p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) \right\rangle_{q(\theta)} \\
&= -\ln Z_s + \left\langle \ln p(\vec{s}_{1:T} \mid \boldsymbol{B}, \vec{a}, \vec{\pi})\, p(\vec{x}_{1:T} \mid \vec{s}_{1:T}, \vec{f}) \right\rangle_{q(\boldsymbol{B})q(\vec{a})q(\vec{\pi})} \\
&= -\ln Z_s + \left\langle \ln p(\vec{s}_{1:T} \mid \boldsymbol{B}, \vec{a}, \vec{\pi}) \right\rangle_{q(\boldsymbol{B})q(\vec{a})q(\vec{\pi})} + \left\langle \ln p(\vec{x}_{1:T} \mid \vec{s}_{1:T}, \vec{f}) \right\rangle_{q(\boldsymbol{B})q(\vec{a})q(\vec{\pi})} \\
&= -\ln Z_s + \left\langle \ln p(\vec{s}_{1:T} \mid \boldsymbol{B}, \vec{a}, \vec{\pi}) \right\rangle_{q(\boldsymbol{B})q(\vec{a})q(\vec{\pi})} + \ln p(\vec{x}_{1:T} \mid \vec{s}_{1:T}, \vec{f}).
\end{aligned} \tag{2.54}$$

To determine the variational distribution $q(\vec{s}_{1:T})$, the terms on the right side of Eq. (2.54) has to be determined .The third term is determined by using the expression for the probability distribution of the observed aggregated sequence from Eq. (2.4) which gives

$$\ln p(\vec{x}_{1:T} \mid \vec{s}_{1:T}, \vec{f}) = \ln \prod_{t=1}^{T} \delta_{x_t, f(s_t)} = \sum_{t=1}^{T} \ln \delta_{x_t, f(s_t)},$$

$$\Leftrightarrow \ln p(\vec{x}_{1:T} \mid \vec{s}_{1:T}, \vec{f}) = \begin{cases} 0 & \text{if } \forall t,\, x_t = f(s_t) \\ -\infty & \text{otherwise} \end{cases} \tag{2.55}$$

For the second term in Eq. (2.54) the factors $q(\boldsymbol{B})$, $q(\vec{a})$ and $q(\vec{\pi})$ from the variational distribution $q(\theta)$ are needed. These are obtained from the previous M-step as

$$q(\boldsymbol{B}) = \prod_{i=1}^{N} Dir(B_{i,:} \mid w_{i,:}^{(\boldsymbol{B})}), \tag{2.47}$$

$$q(\vec{a}) = \prod_{i=1}^{N} \beta(a_i \mid w_{i1}^{(\vec{a})}, w_{i2}^{(\vec{a})}), \tag{2.48}$$

$$q(\vec{\pi}) = Dir(\vec{\pi} \mid \vec{w}^{(\vec{\pi})}), \tag{2.49}$$

where the hyperparameters w are calculated in the previous M-step in the VBEM algorithm. With the distribution of the hidden state sequence according to Eq. (2.2) the second term in the update Eq. (2.54) becomes

$$
\begin{aligned}
\left\langle \ln p(\vec{s}_{1:T} \mid \boldsymbol{B}, \vec{a}, \vec{\pi}) \right\rangle_{q(\boldsymbol{B})q(\vec{a})q(\vec{\pi})} &= \left\langle \ln \pi_{s_1} \prod_{t=1}^{T-1} (1-a_{s_t})^{\delta_{s_t,s_{t+1}}} (a_{s_t} B_{s_t s_{t+1}})^{1-\delta_{s_t,s_{t+1}}} \right\rangle_{q(\boldsymbol{B})q(\vec{a})q(\vec{\pi})} \\
&= \left\langle \ln \pi_{s_1} \right\rangle_{q(\boldsymbol{B})q(\vec{a})q(\vec{\pi})} + \sum_{t=1}^{T-1} (1-\delta_{s_t,s_{t+1}}) \left\langle \ln B_{s_t s_{t+1}} \right\rangle_{q(\boldsymbol{B})q(\vec{a})q(\vec{\pi})} \\
&\quad + \sum_{t=1}^{T-1} \delta_{s_t,s_{t+1}} \left\langle \ln(1-a_{s_t}) \right\rangle_{q(\boldsymbol{B})q(\vec{a})q(\vec{\pi})} + \sum_{t=1}^{T-1} (1-\delta_{s_t,s_{t+1}}) \left\langle \ln a_{s_t} \right\rangle_{q(\boldsymbol{B})q(\vec{a})q(\vec{\pi})} \\
&= \left\langle \ln \pi_{s_1} \right\rangle_{q(\vec{\pi})} + \sum_{t=1}^{T-1} (1-\delta_{s_t,s_{t+1}}) \left\langle \ln B_{s_t s_{t+1}} \right\rangle_{q(B_{s_t:})} \\
&\quad + \sum_{t=1}^{T-1} \delta_{s_t,s_{t+1}} \left\langle \ln(1-a_{s_t}) \right\rangle_{q(\vec{a})} + \sum_{t=1}^{T-1} (1-\delta_{s_t,s_{t+1}}) \left\langle \ln a_{s_t} \right\rangle_{q(\vec{a})}.
\end{aligned}
$$

(2.56)

The right side of Eq. (2.56) is now a sum of expected values with the respect to Dirichlet and beta distributions which can be calculated with the Eqs. (2.50), (2.51), (2.52) and (2.53). Eq. (2.56) then becomes

$$
\begin{aligned}
&\left\langle \ln p(\vec{s}_{1:T} \mid \boldsymbol{B}, \vec{a}, \vec{\pi}) \right\rangle_{q(\boldsymbol{B})q(\vec{a})q(\vec{\pi})} = \sum_{t=1}^{T-1} (1-\delta_{s_t,s_{t+1}}) \left[\psi(w^{(\boldsymbol{B})}_{s_t,s_{t+1}}) - \psi(w^{(\boldsymbol{B})}_{s_t,0}) \right], \\
&+ \sum_{t=1}^{T-1} \left\{ (1-\delta_{s_t,s_{t+1}}) \left[\psi(w^{(\vec{a})}_{s_t,1}) - \psi(w^{(\vec{a})}_{s_t,0}) \right] + \delta_{s_t,s_{t+1}} \left[\psi(w^{(\vec{a})}_{s_t,2}) - \psi(w^{(\vec{a})}_{s_t,0}) \right] \right\} + \psi(w^{(\vec{\pi})}_{s_1}) - \psi(w^{(\vec{\pi})}_{0}).
\end{aligned}
$$

(2.57)

By inserting the Eqs. (2.55) and (2.57) as terms in the update Eq. (2.54) of the variational distribution $q(\vec{s}_{1:T})$, and introducing the variables H_{t,s_t} and $Q_{s_t,s_{t+1}}$, the update Eq. (2.54) becomes

$$
\ln q(\vec{s}_{1:T}) = -\ln Z_s + \sum_{t=1}^{T} \ln H_{t,s_t} + \sum_{t=1}^{T-1} \ln Q_{s_t,s_{t+1}}. \tag{2.58}
$$

Here, $\ln H_{t,s_t}$ and $\ln Q_{s_t,s_{t+1}}$ are given by inspection of the Eqs. (2.55) and (2.57) as

$$
\ln Q_{s_t,s_{t+1}} = \begin{cases} \psi(w^{(\vec{a})}_{s_t 2}) - \psi(w^{(\vec{a})}_{s_t 0}) & s_t = s_{t+1} \\ \psi(w^{(\boldsymbol{B})}_{s_t s_{t+1}}) - \psi(w^{(\boldsymbol{B})}_{s_t 0}) + \psi(w^{(\vec{a})}_{s_t 1}) - \psi(w^{(\vec{a})}_{s_t 0}) & s_t \neq s_{t+1} \end{cases} \tag{2.59}
$$

$$
\ln H_{t,s_t} = \delta_{t,1} \left[\psi(w^{(\vec{\pi})}_{s_1}) - \psi(w^{(\vec{\pi})}_{0}) \right] + \begin{cases} 0 & f(s_t) = x_t \\ -\infty & f(s_t) \neq x_t \end{cases} \tag{2.60}
$$

Or

$$H_{t,s_t} = \begin{cases} \delta_{x_t, f(s_t)} e^{\psi(w_{s_1}^{(\vec{\pi})}) - \psi(w_0^{(\vec{\pi})})} & t = 1 \\ \delta_{x_t, f(s_t)} & t > 1 \end{cases} \tag{2.61}$$

H_{t,s_t} and $Q_{s_t,s_{t+1}}$ are updated according to these equations in each E-step of the VBEM algorithm. With these expressions known the variational distribution $q(\vec{s}_{1:T})$ can be calculated for all possible values of the hidden state sequence $\vec{s}_{1:T}$. The naïve implementation of the calculation of all possible $q(\vec{s}_{1:T})$ would however require an enormous amount of calculations. Instead all possible hidden state sequences can be swept through an implementation of dynamic programming called the forward-backward algorithm. The normalization constant Z_s can be obtained through this algorithm along with two expectation values that will be used in the next M-step. These expectation values are

$$\left\langle \delta_{j,s_t} \right\rangle_{q(s)} = p(s_t = j), \tag{2.62}$$

$$\left\langle \delta_{j,s_t} \delta_{k,s_{t+1}} \right\rangle_{q(s)} = p(s_{t+1} = k \mid s_t = j), \tag{2.63}$$

With the averages $p(s_t = j)$ and $p(s_{t+1} = k \mid s_t = j)$ calculated the M-step is finished. These averages are then used in the next E-step of the VBEM algorithm (see the Eqs. (2.47) , (2.48) and (2.49)).

Calculating the lower bound F

The goal of this inference method is to find the model structure η with optimized parameters θ that maximises the lower bound F of the evidence. A way of calculating this lower bound is therefore needed, both to check for convergence for each η and to compare the converged lower bound for different η to find the maximum. This approximation of the evidence is sought by first looking at the expression of the lower bound found in Eq. (2.15) and utilizing that $q(\theta)$ and $q(\vec{s}_{1:T})$ are normalized probability distributions. This gives

$$\begin{aligned} F\left[q(\theta), q(\vec{s}_{1:T}), \vec{x}_{1:T}\right] &= \int d\theta \sum_s q(\theta) q(\vec{s}_{1:T}) \ln p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) p(\theta \mid \eta) \\ &- \int d\theta q(\theta) \ln q(\theta) - \sum_s q(\vec{s}_{1:T}) \ln q(\vec{s}_{1:T}) \\ &= \sum_s q(\vec{s}_{1:T}) \left[\left\langle \ln p(\vec{x}_{1:T}, \vec{s}_{1:T} \mid \theta, \eta) \right\rangle_{q(\theta)} - \ln q(\vec{s}_{1:T}) \right] + \int d\theta q(\theta) \left[\ln p(\theta \mid \eta) - \ln q(\theta) \right]. \end{aligned} \tag{2.64}$$

$q(\vec{s}_{1:T})$ is given by the update Eq. (2.29) just after the E-step, and when this is inserted to the expression above, Eq. (2.64) simplifies to

$$F\left[q(\theta), q(\vec{s}_{1:T}), \vec{x}_{1:T}\right] = \ln Z_s - \int d\theta q(\theta) \ln \frac{q(\theta)}{p(\theta \mid \eta)} \tag{2.65}$$

The first term in Eq. (2.65) is the normalization constant for $q(\vec{s}_{1:T})$ (i.e. the likelihood of the current θ). This value is obtained through the forward-backward algorithm and has thus just been calculated after the E-step. The second term is also easy to compute because of the separable $q(\theta)$ and $p(\theta \mid \eta)$. The second term can be expanded according to

$$\int d\theta q(\theta) \ln \frac{q(\theta)}{p(\theta \mid \eta)} = \sum_i \int dB_{i,:} q(B_{i,:}) \ln \frac{q(B_{i,:})}{p(B_{i,:} \mid \eta)} + \int d\vec{a} q(\vec{a}) \ln \frac{q(\vec{a})}{p(\vec{a} \mid \eta)} + \int d\vec{\pi} q(\vec{\pi}) \ln \frac{q(\vec{\pi})}{p(\vec{\pi} \mid \eta)} \tag{2.66}$$

All of the three terms in Eq.(2.66) can be calculated as expected values of the variational distributions given by the Eqs. (2.50), (2.51) and (2.53). The terms are thus calculated according to

$$\begin{aligned}
\int dB_{i,:} q(B_{i,:}) \ln \frac{q(B_{i,:})}{p(B_{i,:} \mid \eta)} &= \left\langle \ln \left(\frac{\Gamma(w_{i0}^{(\boldsymbol{B})})}{\Gamma(\tilde{w}_{i0}^{(\boldsymbol{B})})} \prod_{\substack{j=1 \\ j \neq i}}^{N} \frac{\Gamma(w_{ij}^{(\boldsymbol{B})})}{\Gamma(\tilde{w}_{ij}^{(\boldsymbol{B})})} B_{ij}^{w_{ij}^{(\boldsymbol{B})} - \tilde{w}_{ij}^{(\boldsymbol{B})}} \right) \right\rangle_{q(\boldsymbol{B})} \\
&= \ln \frac{\Gamma(w_{i0}^{(\boldsymbol{B})})}{\Gamma(\tilde{w}_{i0}^{(\boldsymbol{B})})} - \left(w_{i0}^{(\boldsymbol{B})} - \tilde{w}_{i0}^{(\boldsymbol{B})} \right) \psi(w_{i0}^{(\boldsymbol{B})}) - \sum_{\substack{j=1 \\ j \neq i}}^{N} \left[\ln \frac{\Gamma(w_{ij}^{(\boldsymbol{B})})}{\Gamma(\tilde{w}_{ij}^{(\boldsymbol{B})})} - \left(w_{ij}^{(\boldsymbol{B})} - \tilde{w}_{ij}^{(\boldsymbol{B})} \right) \psi(w_{ij}^{(\boldsymbol{B})}) \right],
\end{aligned} \tag{2.67}$$

$$\begin{aligned}
\int d\vec{a} q(\vec{a}) \ln \frac{q(\vec{a})}{p(\vec{a} \mid \eta)} &= \sum_{i=1}^{N} \left\langle \ln \left(\frac{\Gamma(w_{i0}^{(\vec{a})})}{\Gamma(\tilde{w}_{i0}^{(\vec{a})})} \prod_{\substack{j=1 \\ j \neq i}}^{2} \frac{\Gamma(w_{ij}^{(\vec{a})})}{\Gamma(\tilde{w}_{ij}^{(\vec{a})})} a_i^{w_{ij}^{(\vec{a})} - \tilde{w}_{ij}^{(\vec{a})}} \right) \right\rangle_{q(\vec{a})} \\
&= \sum_{i=1}^{N} \left\{ \ln \frac{\Gamma(w_{i0}^{(\vec{a})})}{\Gamma(\tilde{w}_{i0}^{(\vec{a})})} - \left(w_{i0}^{(\vec{a})} - \tilde{w}_{i0}^{(\vec{a})} \right) \psi(w_{i0}^{(\vec{a})}) - \sum_{\substack{j=1 \\ j \neq i}}^{2} \left[\ln \frac{\Gamma(w_{ij}^{(\vec{a})})}{\Gamma(\tilde{w}_{ij}^{(\vec{a})})} - \left(w_{ij}^{(\vec{a})} - \tilde{w}_{ij}^{(\vec{a})} \right) \psi(w_{ij}^{(\vec{a})}) \right] \right\}
\end{aligned} \tag{2.68}$$

$$\begin{aligned}
\int d\vec{\pi} q(\vec{\pi}) \ln \frac{q(\vec{\pi})}{p(\vec{\pi} \mid \eta)} &= \left\langle \ln \left(\frac{\Gamma(w_{i0}^{(\vec{\pi})})}{\Gamma(\tilde{w}_{i0}^{(\vec{\pi})})} \prod_{\substack{j=1 \\ j \neq i}}^{N} \frac{\Gamma(w_{ij}^{(\vec{\pi})})}{\Gamma(\tilde{w}_{ij}^{(\vec{\pi})})} a_i^{w_{ij}^{(\vec{\pi})} - \tilde{w}_{ij}^{(\vec{\pi})}} \right) \right\rangle_{q(\vec{\pi})} \\
&= \ln \frac{\Gamma(w_{i0}^{(\vec{\pi})})}{\Gamma(\tilde{w}_{i0}^{(\vec{\pi})})} - \left(w_{i0}^{(\vec{\pi})} - \tilde{w}_{i0}^{(\vec{\pi})} \right) \psi(w_{i0}^{(\vec{\pi})}) - \sum_{\substack{j=1 \\ j \neq i}}^{N} \left[\ln \frac{\Gamma(w_{ij}^{(\vec{\pi})})}{\Gamma(\tilde{w}_{ij}^{(\vec{\pi})})} - \left(w_{ij}^{(\vec{\pi})} - \tilde{w}_{ij}^{(\vec{\pi})} \right) \psi(w_{ij}^{(\vec{\pi})}) \right]
\end{aligned} \tag{2.69}$$

The lower bound F can now finally be calculated, which is done as a last step in the algorithm after the VBEM iterations have converged. This maximized lower bound is used as an approximation of the logarithm of the evidence and can then be used for model selection when comparing the evidence of different model structures η with each other.

Multiple observed sequences

When multiple observed sequences $\vec{x}$ are used in one call to the VBEM algorithm, the sum of the lower bound from each observed sequence is maximized with one single model. This is effectively done by summing over all observed sequences M when calculating the parameters of the variational distribution $q(\theta)$ in the M-step. For example, $w_{ij}^{(\boldsymbol{B})}$ which is calculated according to Eq. (2.45) for one observed sequence, is for multiple sequences calculated according to

$$w_{ij}^{(\boldsymbol{B})} = \tilde{w}_{ij}^{(\boldsymbol{B})} + \sum_{m=1}^{M}\sum_{t=1}^{T-1}\left\langle \delta_{i,s_t^m}\delta_{j,s_{t+1}^m}\right\rangle_{q(\vec{s}_{1:T})}, \quad i \neq j, \tag{2.70}$$

where s_t^m is the hidden state at time t for the mth observed sequence in the dataset. Similarily, contributions from additional observed sequences are just added as new rows to the matrices $\ln \boldsymbol{H}$ and $\ln \boldsymbol{Q}$, which serve as the input to the forward-backward algorithm. This means that the normalization constant Z_s (i.e. likelihood) also has contributions from multiple observed sequences when such are given, meaning that Z_s is the likelihood of the entire dataset.

References

1. Persson, F., Linden, M., Unoson, C. & Elf, J. vbSPT (Variational Bayes for Single Particle Tracking) - User-guide and Documentation. <http://vbspt.sourceforge.net/ref/vbSPT_userguide.pdf> (2013).
2. Persson, F., Lindén, M., Unoson, C. & Elf, J. Extracting intracellular diffusive states and transition rates from single-molecule tracking data. *Nat Meth* **10,** 265–269 (2013).
3. MacKay, D. J. C. *Ensemble Learning for Hidden Markov Models*. (1997).
4. MacKay, D. J. C. *Information theory, inference, and learning algorithms*. Version 7.2. (Cambridge University Press, 2003).
5. Logan, J. D. *Applied Mathematics*. (Wiley-Interscience, 2006).