vbTPM user guide and manual

Martin Lindén, Stephanie Johnson, Jan-Willem van de Meent, Rob Phillips, and Chris H Wiggins

September 6, 2013

Contents

1 Getting started with vbTPM

1.1 Installation

Get the source code, and make sure VB7, HMMcore, and tools are in your matlab path.

Make sure that HMMcore/ contains binaries for your systems. If not, a simple matlab compilation script can be found in HMMcore/.

1.2 Hardware requirements

Tethered particle motion often produces large data sets of many long trajectories, which makes the HMM analysis computer intensive. As an example, one parameter point in our test data sets, about 90 trajectories averaging 45 min, downsampled to 10 Hz, took about 24 h to go through on two 6 core Intel Xeon E5645 2.40GHz processors. The analysis time increases sharply with the number of states (including spurious ones, like transient sticking events).

1.3 A small test problem

A small test problem, which runs on a (fast) laptop in about one hour, with examples of data and runinput files, can be found in example1/, with the actual data in example1/lacdata/. The data set has one calibration (cal) and one production (trj) trajectory for each bead.

Runinput files contain all parameters to run the analysis and access the results. The meaning of the parameters are documentet in the help text of `VB7_batch_run.m`, and commented in those files. `runinput1.m` refers to an already completed analysis (results in example1/HMMresults1/), while `runinput2.m` has not yet run.

Run basic analysis. To start analyzing the test data set, type `VB7_batch_run('runinput2')` in the Matab command prompt. Since the runinput file has `one_at_a_time=true;` this will analyze one trajectory in the data set. Several calls are needed to complete the analysis, in our experience, matlab tend to hoard memory if several large data sets are analyzed consecutively.

To work around that, one can use scripts that starts several matlab sessions with a single trajectory in each. An example for the bash shell is `runscript1.sh`, which calls `runinput1.m`. To parallellize, run several instances of the script at once. Since the program keeps track of which trajectories have already been 'checked out', it is also possible to run on several computers, it the results folder is synced regularly.

Manage the analysis. `VB7_batch_manage` is a tool for managing the basic analysis. It can collect the results and write them to a file, count how many trajectories in a data set has been analyzed, and also clean up temporary files from unfinished trajectories, which is useful if an analysis run is interrupted.

Look at the results

`VB7_batch_postprocess()` starts the GUI for manual state classification and inspection of the results, and can also convert the simple HMM models to factorial models for further analysis. To try it out, use the runinput file `runinput1.m`, which is already analyzed. To access the fitted models directly, use `VB7_batch_manage` with the collect option.

1.4 Other useful scripts

2 Diffusive model for TPM

2.1 Diffusive hidden Markov model

We model the looping dynamics by a discrete Markov process s_t with N states, a transition probability matrix $\mathbf{A}$, and initial state distribution $\vec{\pi}$,

$$p(s_t|s_{t-1},\mathbf{A}) = A_{s_{t-1}s_t}, \quad p(s_1) = \pi_{s_t}. \qquad (1)$$

This is the standard hidden part of an HMM, and the physics of TPM goes into the emission model, that describes the restricted Brownian motion of the bead. We use a discrete time model of over-damped 2D diffusion in a harmonic potential, that has been suggested as a simplified model for TPM (1, 2),

$$\vec{x}_t = K_{s_t}\vec{x}_{t-1} + \vec{w}_t/(2B_{s_t})^{1/2}, \qquad (2)$$

where the index s_t indicate parameters that depend on the hidden state. Thermal noise enters through the uncorrelated Gaussian random vectors $\vec{w}_t$ with unit variance. The unintuitive parametrization is chosen for computational convenience; K_j and B_j are related to the spring and diffusion constant of the bead, and some insight into their physical meaning can be gained by noting that with a single hidden state, Eq. 2

describes a Gaussian process with zero mean and

$$RMS = \sqrt{\langle \vec{x}^2 \rangle} = (B(1-K^2))^{-1/2},$$
$$\frac{\langle \vec{x}_{t+m} \cdot \vec{x}_t \rangle}{\langle \vec{x}^2 \rangle} = K^m \equiv e^{-m\Delta t/\tau}, \quad (3)$$

where Δt is the sampling time, and τ is a bead correlation time. This model thus captures the diffusive character of the bead motion, while still retaining enough simplicity to allow efficient variational algorithms (3, 4).

2.2 Factorial model

(?)

3 The VB-algorithm

3.1 Model selection by maximum evidence

Our analysis aims not only to extract parameter values from TPM data, but also to learn the number of hidden states N, corresponding to different DNA-protein conformations. This means that we need to compare models with different number of unknown parameters. We take a Bayesian approach to this problem.

A distinguishing feature of Bayesian data analysis is the treatment of random variables and unknown parameters on an equal footing (5, 6). Hence, given some data $\vec{x}_{1:T}$ and a set of competing models with different number of states $N = 1, 2, \ldots$ (and *1:T* is a compact way to denote a whole time series), we can use the laws of probability to express our confidence about those models in terms of conditional probabilities,

$$p(N|\vec{x}_{1:T}) = p(\vec{x}_{1:T}|N)p(N)/p(\vec{x}_{1:T}), \quad (4)$$

where $p(N)$ expresses our beliefs about the different models prior to seeing the data, and $p(\vec{x}_{1:T})$ is a normalization constant. A Bayesian rule for model selection is therefore to prefer the model that maximizes $p(\vec{x}_{1:T}|N)$, a quantity known as the evidence. For our more complex model, parameters and hidden states will have to be integrated out,

$$p(\vec{x}_{1:T}|N) = \int d\theta \sum_{s_{1:T}} p(\vec{x}_{1:T}, s_{1:T}|\theta)p(\theta|N), \quad (5)$$

where the first factor in the integrand describes the model, and the second expresses our prior beliefs about the parameters (see below).

The integrand in the evidence, Eq. (5), requires an explicit expression for the probability of a sequence of bead positions and hidden states. This expression can be written down based on the above model, and factorizes in the usual HMM fashion, as

$$\begin{aligned} p(\vec{x}_{1:T}, s_{1:T}|\theta)p(\theta|N) &= p(\vec{x}_1)p(s_1|\vec{\pi}) \\ &\times \prod_{t=2}^{T} p(\vec{x}_t|\vec{x}_{t-1}, s_t, \vec{K}, \vec{B})p(s_t|s_{t-1}, \mathbf{A}) \\ &\times p(\vec{\pi}|N) \prod_{j=1}^{N} p(K_j, B_j|N)p(A_{j,:}|N), \quad (6) \end{aligned}$$

where $A_{j,:}$ denote row j of the matrix $\mathbf{A}$. The first right hand side line in Eq. (6) describes the initial state and bead position. We will neglect the factor $p(\vec{x}_1)$ from now on, but the initial state $p(s_1|\vec{\pi})$ and transition probabilities $p(s_t|s_{t-1}, \mathbf{A})$ are given by Eq. (1), and the bead motion follows from Eq. (2),

$$p(\vec{x}_t|\vec{x}_{t-1}, s_t, \vec{K}, \vec{B}) = \frac{B_{s_t}}{\pi} e^{-B_{s_t}(\vec{x}_t - K_{s_t}\vec{x}_{t-1})^2}. \quad (7)$$

Finally, the last line of Eq. (6) contains prior distributions over parameters conditional on the number of states. We use conjugate priors, parameterized to have minimal impact on the inference results (see SI).

3.2 The variational approximation

An exact computation of the Bayesian evidence is impractical or impossible for most interesting models, and clever approximations are needed. The approximation we use here is variously known as ensemble learning, variational Bayes, or (in statistical physics jargong) mean field theory (4, 6), has previously been applied to biophysical time-series of FRET data (7–9) and *in vivo* single particle tracking (10). The idea is to approximate the log evidence by a lower bound, $\ln p(x|N) \geq F_N$, with

$$F_N = \int d\theta \sum_s q(s)q(\theta) \ln \frac{p(x,s|\theta)p(\theta|N)}{q(s)q(\theta)}, \quad (8)$$

where $q(s)$ and $q(\theta)$ are arbitrary probability distributions over the hidden states and parameters respectively. These are optimized to make the bound as tight as possible for each model, the model that achieves the highest lower bound wins, and the corresponding optimal distributions $q(s)q(\theta)$ can be used for approximate inference about parameter values and hidden states. In particular, optimizing F_N with respect to the variational distributions leads to

$$\ln q(\theta) = -\ln Z_\theta + \ln p(\theta|N) + \langle \ln p(x,s|\theta) \rangle_{q(s)}, \quad (9)$$

$$\ln q(s) = -\ln Z_s + \langle \ln p(x,s|\theta) \rangle_{q(\theta)}, \quad (10)$$

where the Z's are Lagrange multipliers to enforce normalization, and $\langle \cdot \rangle_{q(\cdot)}$ denotes an average over $q(\cdot)$. We solve these equations iteratively until the lower bound converges, repeating the analysis many times with independent initial conditions in order to find a global maximum. The iterative solution approach results in an EM-type variational algorithm, detailed below. We refer to Refs. (3, 10, 11) for details on how to derive variational algorithms for HMMs, and Refs. (4, 6, 11) for more general discussion of variational inference methods.

3.2.1 Parameter distributions

The results of plugging our diffusinve HMM into the parameter update equation (9) are as follows. The initial state probability vector, and each row in the transition matrix (denoted $A_{j,:}$), are Dirichlet distributed,

$$q(\vec{\pi}) = \mathrm{Dir}(\vec{\pi}|\vec{w}^{(\vec{\pi})}), \quad (11)$$

$$w_j^{(\vec{\pi})} = \tilde{w}_j^{(\vec{\pi})} + \langle \delta_{j,s_1} \rangle_{q(s_{1:T})}, \quad (12)$$

$$q(A_{i,:}) = \mathrm{Dir}(A_{i,:}|\vec{w}^{(\mathbf{A})}), \quad (13)$$

$$w_{ij}^{(\mathbf{A})} = \tilde{w}_{ij}^{(\mathbf{A})} + \sum_{t=2}^{T} \langle \delta_{i,s_{t-1}} \delta_{j,s_t} \rangle_{q(s_{1:T})}. \quad (14)$$

Here, variables under tilde's (˜) are hyperparameters that parameterize the prior distributions, and can be interpreted as pseudo-observations. The Dirichlet density function is

$$\mathrm{Dir}(\vec{\pi}|\vec{u}) = \frac{\Gamma(u_0)}{\prod_j \Gamma(u_j)} \prod_j \pi_j^{u_j - 1}, \quad u_j > 1, \quad (15)$$

where $u_0 = \sum_j u_j$ is called the strength, and the density is non-zero in the region $0 \leq \pi_j \leq 1$, $\sum_j \pi_j = 1$. Before moving on, we quote some useful expectation values for future reference,

$$\langle \ln \pi_i \rangle_{q(\vec{\pi})} = \psi(w_i^{(\vec{\pi})}) - \psi(w_0^{(\vec{\pi})}), \quad (16)$$

$$\langle \ln A_{ij} \rangle_{q(\mathbf{A})} = \psi(w_{ij}^{(\mathbf{A})}) - \psi(w_{i0}^{(\mathbf{A})}), \quad (17)$$

where $\psi(x)$ is the digamma function, and

$$\langle \pi_i \rangle_{q(\vec{\pi})} = \frac{w_i^{(\vec{\pi})}}{w_0^{(\vec{\pi})}}, \tag{18}$$

$$\mathrm{Var}[\pi_i]_{q(\vec{\pi})} = \frac{w_i^{(\vec{\pi})}\big((1-w_i^{(\vec{\pi})}\big)}{(w_0^{(\vec{\pi})})^2\big(1+w_0^{(\vec{\pi})}\big)}, \tag{19}$$

$$\langle A_{ij} \rangle_{q(\mathbf{A})} = \frac{w_{ij}^{(\mathbf{A})}}{w_{i0}^{(\mathbf{A})}}, \tag{20}$$

$$\mathrm{Var}[A_{ij}]_{q(\mathbf{A})} = \frac{w_{ij}^{(\mathbf{A})}\big(1-w_{ij}^{(\mathbf{A})}\big)}{(w_{i0}^{(\mathbf{A})})^2\big(1+w_{i0}^{(\mathbf{A})}\big)}. \tag{21}$$

The bead motion parameters have the following variational distributions

$$q(K_j, B_j) = \frac{B_j^{n_j}}{W_j} e^{-B_j\left(v_j(K_j-\mu_j)^2+c_j\right)}, \tag{22}$$

$$W_j = \frac{c^{-(n_j+\frac{1}{2})}\Gamma(n_j+\frac{1}{2})}{\sqrt{v_j/\pi}}, \tag{23}$$

with the range $B_j \geq 0$, $-\infty < K_j < \infty$. Physically, we might rather expect $0 < K_j < 1$, but the extended range for K_j simplifies the calculations a lot. The VBM equations are

$$n_j = n_j^0 + M_j, \tag{24}$$

$$c_j = c_j^0 + C_j + v_j^0(\mu_j^0)^2 - \frac{\left(v_j^0\mu_j^0 + U_j\right)^2}{v_j^0 + V_j}, \tag{25}$$

$$v_j = v_j^0 + V_j, \tag{26}$$

$$\mu_j = \frac{v_j^0\mu_j^0 + U_j}{v_j^0 + V_j}, \tag{27}$$

with

$$M_j = \sum_{t=2}^{T} \langle \delta_{s_t,j} \rangle, \tag{28}$$

$$C_j = \sum_{t=2}^{T} \langle \delta_{s_t,j} \rangle \vec{x}_t^2, \tag{29}$$

$$U_j = \sum_{t=2}^{T} \langle \delta_{s_t,j} \rangle \vec{x}_t \cdot \vec{x}_{t-1}, \tag{30}$$

$$V_j = \sum_{t=2}^{T} \langle \delta_{s_t,j} \rangle \vec{x}_{t-1}^2. \tag{31}$$

Some useful expectation values for future reference are

$$\langle \ln B_j \rangle_{q(\vec{B},\vec{K})} = \psi\big(n_j + \frac{1}{2}\big) - \ln c_j, \tag{32}$$

$$\langle B_j \rangle_{q(\vec{B},\vec{K})} = \frac{n_j + \frac{1}{2}}{c_j}, \tag{33}$$

$$\langle B_j K_j^2 \rangle_{q(\vec{B},\vec{K})} = \frac{1}{2v_j} + \mu_j^2 \frac{n_j + \frac{1}{2}}{c_j}, \tag{34}$$

$$\langle B_j K_j \rangle_{q(\vec{B},\vec{K})} = \mu_j \frac{n_j + \frac{1}{2}}{c_j}, \tag{35}$$

$$\mathrm{Var}[B_j]_{q(\vec{B},\vec{K})} = \frac{n_j + \frac{1}{2}}{c_j^2}, \tag{36}$$

$$\langle K_j \rangle_{q(\vec{B},\vec{K})} = \mu_j, \tag{37}$$

$$\mathrm{Var}[K_j]_{q(\vec{B},\vec{K})} = \frac{c_j}{2v_j(n_j - \frac{1}{2})}. \tag{38}$$

3.2.2 Hidden state distribution

The variational distribution has a simple form,

$$\ln q(s_{1:T}) = -\ln Z + \sum_{t=1}^{T} \ln h_{s_t}(t) + \sum_{t=2}^{T} \ln J_{s_{t-1},s_t}, \tag{39}$$

i.e., an initial state distribution, a point-wise term that depends on the initial conditions and the data, and a transition probability. The point-wise

$$\ln q(s_{1:T}) = -\ln Z + \sum_{t=1}^{T} \ln h_{s_t}(t) + \sum_{t=2}^{T} \ln J_{s_{t-1},s_t}, \tag{40}$$

i.e., an initial state distribution, point-wise terms that depends on the initial conditions and the data, and transition terms. The mathematical form of this expression is the same as encountered in maximum-likelihood optimization of hidden Markov Models, and hence the normalization constant and expectation values needed for the parameter update equations can be computed by the Baum-Welch algorithm (12), which resembles the transfer matrix solution for spin models in statistical physics.

Similarly, and the most likely sequence of hidden states can be computed by the Viterbi algorithm (13).

Specifically, the initial term is given by

$$\ln h_j(1) = \langle p(s_1 = j|\vec{\pi})\rangle_{q(\vec{\pi})} = \psi(w_j^{(\vec{\pi})}) - \psi(w_0^{(\vec{\pi})}), \tag{41}$$

the point-wise contributions for $t > 1$ are

$$\begin{aligned}\ln h_j(t) = {}& \psi\big(n_j + \frac{1}{2}\big) - \ln(\pi c_j) - \frac{\vec{x}_{t-1}^2}{2v_j} \\ &- \frac{n_j + \frac{1}{2}}{c_j}\Bigg(\vec{x}_{t-1}^2\Big(\mu_j - \frac{\vec{x}_t \cdot \vec{x}_{t-1}}{\vec{x}_{t-1}^2}\Big)^2 \\ &+ \vec{x}_t^2 - \frac{\big(\vec{x}_t \cdot \vec{x}_{t-1}\big)^2}{\vec{x}_{t-1}^2}\Bigg), \end{aligned} \tag{42}$$

and the transition terms are given by

$$\ln J_{ji} = \psi\big(w_{j,i}^{(\mathbf{A})}\big) - \psi\big(\sum_{k=1}^{N} w_{j,k}^{(\mathbf{A})}\big). \tag{43}$$

3.3 VBEM iterations and model search

The iterative optimization of the variational distributions are done as follows. To start with, an initial guess for the variational parameter distributions are generated. We then alternate between VBE step, in which we construct the hidden state distribution and compute the averages $\langle \delta_{j,s_t} \rangle_{q(s)}$ and $\langle \delta_{j,s_t}\delta_{k,s_{t+1}} \rangle_{q(s)}$ in a Baum-Welch forward-backward sweep, and a VBM step, in which we use these averages to update the parameter variational distributions, until the lower bound converges.

The variational approach has the additional useful tendency to penalizing overfitting already during the VBEM iterations, by depopulating superfluous states (3, 10, 11). We exploit this property by using a greedy search algorithm to explore the model space. The basic strategy is to start by fitting a model with many states from random initial conditions, and then exploring less complex models by gradually removing the least populated states. This saves computing time by supplying good initial guesses for the low complexity models (which therefore converge quickly), and by lowering the number of independent restarts, since it is easier to construct a good initial guess for a model with many states.

3.4 The lower bound

has an especially simple form just after the VBE step (3, 10, 11), given by the normalization constant $\ln Z$ of the variational hidden state distribution, minus the Kullback-Leibler divergences between the variational and prior parameter distributions,

$$
\begin{aligned}
F = \ln Z - \int d\vec{\pi} q(\vec{\pi}) \ln \frac{q(\vec{\pi})}{p(\vec{\pi})} \\
- \sum_{j=1}^{N} \Bigg[\int d^N A_{j,:}\; q(A_{j,:}) \ln \frac{q(A_{j,:})}{p_0(A_{j,:})} \\
+ \int dB_j dK_j\; q(B_j, K_j) \ln \frac{q(B_j, K_j)}{p_0(B_j, K_j)} \Bigg]. \qquad (44)
\end{aligned}
$$

The Kullback-Leibler terms can be expressed in terms of the expectation values computed above. For the initial state distribution, we get

$$
\begin{aligned}
\int d\vec{\pi} q(\vec{\pi}) \ln \frac{q(\vec{\pi})}{p_0(\vec{\pi})} = \ln \tilde{w}_0^{(\vec{\pi})} - \psi(\tilde{w}_0^{(\vec{\pi})}) - \frac{1}{\tilde{w}_0^{(\vec{\pi})}} \\
+ \sum_{j=1}^{N} \left[\big(w_j^{(\vec{\pi})} - \tilde{w}_j^{(\vec{\pi})}\big) \psi(w_j^{(\vec{\pi})}) - \ln \frac{\Gamma(w_j^{(\vec{\pi})})}{\Gamma(\tilde{w}_j^{(\vec{\pi})})} \right]. \\
(45)
\end{aligned}
$$

To get this simple form, we used that $w_0^{(\vec{\pi})} = 1 + \tilde{w}_0^{(\vec{\pi})}$ (since $\sum_j \langle \delta_{j,s_1} \rangle = 1$), and the identities $\Gamma(x+1) = x\Gamma(x)$ and $\psi(x+1) = \psi(x) + \frac{1}{x}$. Furthermore, each row of the transition probability matrix contributes

$$
\begin{aligned}
\int d^N A_{j,:}\; q(A_{j,:}) \ln \frac{q(A_{j,:})}{p_0(A_{j,:})} \\
= \ln \frac{\Gamma(w_{j0}^{(\mathbf{A})})}{\Gamma(\tilde{w}_{j0}^{(\mathbf{A})})} - (w_{j0}^{(\mathbf{A})} - \tilde{w}_{j0}^{(\mathbf{A})}) \psi(w_{j0}^{(\mathbf{A})}) \\
- \sum_{k=1}^{N} \left[\ln \frac{\Gamma(w_{jk}^{(\mathbf{A})})}{\Gamma(\tilde{w}_{jk}^{(\mathbf{A})})} - \big(w_{jk}^{(\mathbf{A})} - \tilde{w}_{jk}^{(\mathbf{A})}\big) \psi(w_{jk}^{(\mathbf{A})}) \right]. \\
(46)
\end{aligned}
$$

Finally, the emission parameter of each state contributes

$$
\begin{aligned}
\int dB_j \int d^N K_j\; q(B_j, K_j) \ln \frac{q(B_j, K_j)}{p(B_j, K_j)} = \dots \\
= - \frac{n_j + \frac{1}{2}}{c_j} \Big(c_j - \tilde{c}_j - \tilde{v}_j (\mu_j - \tilde{\mu}_j)^2 \Big) \\
+ \frac{1}{2} \ln \frac{v_j}{\tilde{v}_j} + (\tilde{n}_j + \frac{1}{2}) \ln \frac{c_j}{\tilde{c}_j} - \ln \frac{\Gamma\big(n_j + \frac{1}{2}\big)}{\Gamma\big(\tilde{n}_j + \frac{1}{2}\big)} \\
+ (n_j - \tilde{n}_j) \psi\big(n_j + \frac{1}{2}\big) + \frac{\tilde{v}_j}{2 v_j} - \frac{1}{2}. \qquad (47)
\end{aligned}
$$

3.5 Two types of states

The above algorithm is readily extended to treat the model where genuine and spurious states are separated into two different hidden processes. We implemented a brute force approach to this problem, where we define new composite hidden states $\hat{s}_t = (s_t, c_t)$ and run the above algorithm on this composite model. This has a significant computational cost, since a simple model with $N_{gen.}$ genuine states and $N_{sp.}$ spurious ones gets $N_{gen.} \times (1 + N_{sp.})$ states after conversion. However, since we do not perform exhaustive model search in this representation and can utilize the simpler model to make good initial guesses, this is not a significant problem.

3.6 Choice of prior distributions

We would like to choose uninformative prior distributions in order to let the data speak for itself as much as possible. This is unproblematic for the emission parameters K, B, since the amount of data in all states is large enough to overwhelm any prior influence. We use

$$
\tilde{\mu}_j = 0.6, \qquad \tilde{n}_j = 1, \qquad (48)
$$

$$
\tilde{v}_j = 5.56 \text{ nm}^2, \qquad \tilde{c}_j = 30000 \text{ nm}^2, \qquad (49)
$$

which corresponds to

$$\langle K_j \rangle = 0.6, \qquad \langle B_j \rangle = 5 \times 10^{-5}\ \mathrm{nm}^{-2}, \quad (50)$$

$$\mathrm{std}(K_j) = 0.3, \quad \mathrm{std}(B_j) = 141.4 \times 10^{-5}\ \mathrm{nm}^{-2}. \quad (51)$$

The initial state prior is unproblematic for the opposite reason: the long length of the trajectories makes the initial state relatively unimportant to describe the data. We use a constant prior strength of 5,

$$\tilde{w}_j^{(\vec{\pi})} = 5/N, \quad (52)$$

where N is the number of hidden states.

The transition probabilities needs more care, because the potentially low number of transitions per trajectory makes the prior relatively more influential. Following Persson et al. (10), we parameterize this prior in terms of an expected mean dwell time and an overall number of pseudocounts (prior strength) for each hidden state. In particular, we define a transition *rate* matrix Q with mean dwell time t_D,

$$Q_{ij} = \frac{1}{t_D}\left(-\delta_{ij} + \frac{1-\delta_{ij}}{N-1}\right), \quad (53)$$

and then construct the prior based on the transition probability propagator per unit time step,

$$\tilde{w}_{ij}^{(\mathbf{A})} = \frac{t_A f_{sample}}{n_{downsample}} e^{\Delta t Q}. \quad (54)$$

Here, t_A is the prior strength; both t_A and t_D is specified in time units to be invariant a change of sampling frequency. Further, the timestep is given by $\Delta t = n_{downsample}/f_{sample}$, where f_{sample} is the sampling frequency (30 Hz in our case), and $n_{downsample}$ is the downsampling factor (we use 3).

Numerical experiments by Persson et al. (10) show that choosing the strength too low compared to the mean dwell time produces a bias towards sparse transition matrices. This is not desireable in our case, and we therefore use $t_D = 1$ s, and $t_A = 5$ s throughout this work.

Prior for factorial model: TBA.(**?**)

3.7 Empirical Bayes update equations

The empirical Bayes update equations optimizes the lower bound with respect to the hyperparameters in the prior distribution. This means optimizing sums of Kullback-Leibler divergence terms.

The initial state probability, and the rows of the transition probability matrix, are both Dirichlet distributed. Thus, for M trajectories with Dirichlet parameters $u_j^{(i)}$, $i = 1, 2, \dots, M$, and hyperparameters $\tilde{u}_j$ ($u = w^{(\vec{\pi})}, u^{(\mathbf{A})}$), we need to solve

$$\frac{d}{d\tilde{u}_j} \sum_i \Bigg(\ln \frac{\Gamma(u_0^{(i)})}{\Gamma(\tilde{u}_0)} - (u_0^{(i)} - \tilde{u}_0)\psi(u_0^{(i)}) - \sum_{k=1}^{N} \left[\ln \frac{\Gamma(u_k^{(i)})}{\Gamma(\tilde{u}_k^{(i)})} - \big(u_k^{(i)} - \tilde{u}_k\big)\psi(u_k^{(i)}) \right] \Bigg) = 0, \quad (55)$$

where $u_0^{(i)} = \sum_k u_k^{(i)}$ and similar for $\tilde{u}_0$. This leads to the update equations

$$\psi(\tilde{u}_0) - \psi(\tilde{u}_j) = \frac{1}{M} \sum_i \Big(\psi(u_0^{(i)}) - \psi(u_j^{(i)}) \Big). \quad (56)$$

A numerical solution turned out to be easier using the variables $\tilde{U}_j = \ln \tilde{u}_j$ (to numerically enforce $\tilde{u}_j > 0$).

For the emission parameters, the update equations are instead derived from minimizing

Eq. (47) summed over M trajectories,

$$f_{KB} = \sum_i \Bigg(-\frac{n^{(i)}+\frac{1}{2}}{c^{(i)}}\Big(c^{(i)} - \tilde{c} - \tilde{v}(\mu^{(i)} - \tilde{\mu})^2\Big) + \frac{1}{2}\ln\frac{v^{(i)}}{\tilde{v}} + (\tilde{n}+\frac{1}{2})\ln\frac{c^{(i)}}{\tilde{c}} - \ln\frac{\Gamma\big(n^{(i)}+\frac{1}{2}\big)}{\Gamma\big(\tilde{n}+\frac{1}{2}\big)} + (n^{(i)}-\tilde{n})\psi\big(n^{(i)}+\frac{1}{2}\big) + \frac{1}{2}\Big(\frac{\tilde{v}}{v^{(i)}} - 1\Big)\Bigg). \tag{57}$$

Minimizing with respect to $\tilde{\mu}$ and $\tilde{v}$ leads to

$$\tilde{\mu} = \frac{1}{M}\sum_i \mu^{(i)}, \tag{58}$$

$$\frac{1}{\tilde{v}} = \frac{1}{M}\sum_i \Big(\frac{1}{v^{(i)}} + 2(\tilde{\mu} - \mu^{(i)})^2\Big). \tag{59}$$

The remaining $\tilde{c}$ and $\tilde{n}$ lead to

$$\frac{\tilde{n}+\frac{1}{2}}{\tilde{c}} = \frac{1}{M}\sum_i \frac{n^{(i)}+\frac{1}{2}}{c^{(i)}}, \tag{60}$$

$$\ln\tilde{c} - \psi\big(\tilde{n}+\frac{1}{2}\big) = \frac{1}{M}\sum_i \Bigg(\ln c^{(i)} - \psi\big(n^{(i)}+\frac{1}{2}\big)\Bigg), \tag{61}$$

which we solve numerically. This gets easier by defining $\alpha = \frac{\tilde{n}+\frac{1}{2}}{\tilde{c}}$, then solve the second equation for $\tilde{c}$ numerically, and finally compute $\tilde{n} = \alpha\tilde{c} - \frac{1}{2}$.

3.8 Factorial model

3.9 Notation and symbols

3.10 Large data sets

Parallellization is achieved by running external scripts that starts multiple independent Matlab runs, and use the presence of various .mat files to coordinate the computations. Repeated restarts of Matlab also avoids problems with memory performance.

3.11 Doing empirical Bayes

Our empirical Bayes analysis of hierarchical models used tailored scripts...

References

[1] John F. Beausang, Chiara Zurla, Laura Finzi, Luke Sullivan, and Philip C. Nelson. Elementary simulation of tethered brownian motion. American Journal of Physics, 75 (6):520–523, 2007. doi: 10.1119/1.2710484.

[2] Moshe Lindner, Guy Nir, Anat Vivante, Ian T. Young, and Yuval Garini. Dynamic analysis of a diffusing particle in a trapping potential. Phys. Rev. E, 87 (2), 2013. doi: 10.1103/PhysRevE.87.022716. URL http://link.aps.org/doi/10.1103/PhysRevE.87.022716.

[3] D. J. C. MacKay. Ensemble learning for hidden Markov models. accessed Feb 15 2011, 1997. URL http://www.inference.phy.cam.ac.uk/mackay/abstracts/ensemblePaper.html.

[4] Christopher Bishop. Pattern recognition and machine learning. Springer, New York, 2006. ISBN 9780387310732.

[5] Sean R Eddy. What is Bayesian statistics? Nat. Biotech., 22(9):1177–1178, 2004. doi: 10.1038/nbt0904-1177.

[6] David MacKay. Information theory, inference, and learning algorithms. Cambridge University Press, Cambridge UK ;;New York, 2003. ISBN 9780521642989.

[7] Jonathan E. Bronson, Jingyi Fei, Jake M. Hofman, Ruben L. Gonzalez Jr., and

Chris H. Wiggins. Learning rates and states from biophysical time series: A bayesian approach to model selection and Single-Molecule FRET data. Biophysical Journal, 97(12):3196–3205, 2009. doi: 10.1016/j.bpj.2009.09.031.

[8] Jan-Willem van de Meent, Jonathan E. Bronson, Ruben L. Gonzalez Jr., and Chris H. Wiggins. Learning biochemical kinetic models from single-molecule data with hierarchically-coupled hidden markov models. 2012. manuscript in preparation.

[9] Kenji Okamoto and Yasushi Sako. Variational bayes analysis of a photon-based hidden markov model for single-molecule FRET trajectories. Biophys. J., 103(6):1315–1324, 2012. doi: 10.1016/j.bpj.2012.07.047.

[10] Fredrik Persson, Martin Lindén, Cecilia Unoson, and Johan Elf. Extracting intracellular diffusive states and transition rates from single-molecule tracking data. Nat. Meth., 10(3):265–269, 2013. doi: 10.1038/nmeth.2367.

[11] Matthew Beal. Variational Algorithms for approximate Bayesian inference. PhD thesis, University of Cambridge, UK, 2003.

[12] L.E. Baum, T. Petrie, G. Soules, , and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. The Annals of Mathematical Statistics, 41:164–171, 1970.

[13] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory, 13(2):260–269, 1967.