SMACKS Manual

Sonja Schmid & Markus Götz

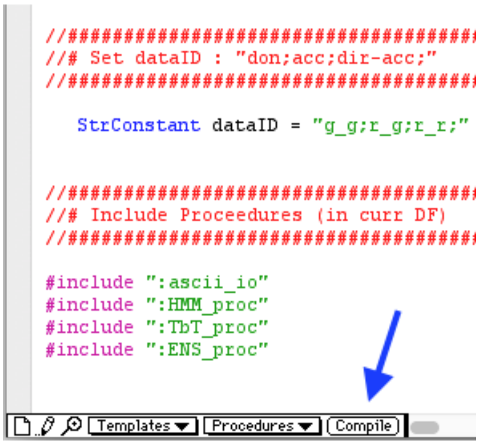
Version 1.4

SMACKS (Single Molecule Analysis of Complex Kinetic Sequences) is a maximum likelihood approach to extract kinetic rate models from noisy single molecule data. While fine-tuned for FRET data, SMACKS is applicable to single molecule time traces of any kind. It optimizes hidden Markov models with a chosen number of states n based on input data of a given dimensionality d. Up to 3 dimensions were tested.
This manual explains the functionalities of the SMACKS software tool. For a complete description of the SMACKS approach please refer to Schmid et al. [1]. This manual follows the terminology of the original publication, which is itself close to Rabiner [2] & Fink [3].

Contents

1 Installation

SMACKS is implemented in Igor Pro (Wavemetrics). If your institution does not have a license for Igor Pro, you can still test SMACKS on your own data using the free 30 days trial of the full software package available at `wavemetrics.com`. After downloading SMACKS' source code and example data from `singlemolecule.uni-freiburg.de/SMACKS`, double click `startSMACKS.ipf` and SMACKS is ready to run. In case the SMACKS menu does not show up automatically, click the tiny compile button indicated in Fig. 1.
IMPORTANT UPDATE: If you are using Igor Pro v7, v8, or higher, you will have to first load (= double click) the new `PREP_Igor7andHigher.ipf`. This prepares the compiler for `startSMACKS.ipf`, and everything runs smoothly.



```
//####################################
//# Set dataID : "don;acc;dir-acc;"
//####################################

   StrConstant dataID = "g_g;r_g;r_r;"


//####################################
//# Include Proceedures (in curr DF)
//####################################

#include ":ascii_io"
#include ":HMM_proc"
#include ":TbT_proc"
#include ":ENS_proc"
```

Figure 1: The tiny compile button. It disappears after compilation.

The dataID holds the names of individual dimensions of your input data (e.g. fluorescence channels). They are used by the importer as base names for the supplied input data (see next section). By default the dataID is set to `"g_g;r_g;r_r;"`. But you can modify it to fit your own data. For example set it to `"force;"` or `"blue;green;"` or any other semicolon separated input list. Alphabetic characters and "_" are allowed.

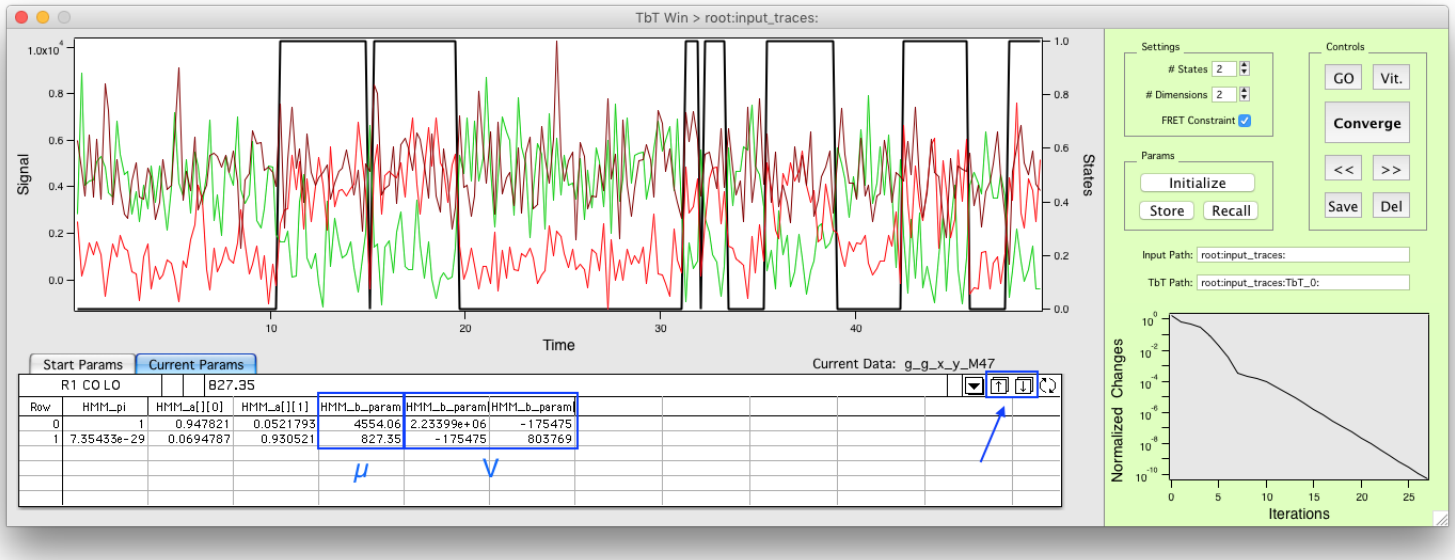
2 Data Import

For maximum compatibility, SMACKS comes with an ascii importer. For a quick test, we further include experimental example data at `SMACKS_tool/exampleData/`.
To load your data into SMACKS, go to the SMACKS menu → **Import ascii**. Press *shift* and select multiple ascii files (.dat or .txt) to import the desired dataset into SMACKS. The importer accepts files with one, two or three data columns (tab- or space-separated) representing multiple dimensions of one time trace. File names are arbitrary, as the imported trajectories are renamed as specified by the dataID plus suffix. (The original names are stored in Igor's "wavenotes" and recovered during results export. See respective section.)

Input data must not include NaNs or INFs. It should only contain the range that is relevant for analysis (e.g. no after-bleach tail).
All calculations are performed in user-supplied time units. E.g. if your sampling rate is 10 frames per second, all transition probabilities a_{ij} are specified per time interval of $0.1s$. They are converted to rate constants in Hertz by $k_{ij} = 10 \cdot a_{ij}$ (neglecting multiple transitions per time interval). Therefore, constant time intervals are required. But specific time information is not needed.
If the optional "FRET constraint" is going to be used, fluorescence data must be corrected for experimental offsets, crosstalk etc. as detailed in Hellenkamp & Schmid et al. [4].

3 Step 1: Trace-by-Trace HMM (TbT)

As a first step, individual HMMs are optimized for each trajectory separately. The TbT workflow can be called from the SMACKS menu → **Init TbT**. The resulting user interface is shown in Figure 2.


TbT Win > root:input_traces:

Current Data: g_g_x_y_M47

Start Params | Current Params

R1 C0 L0 | 827.35

Row	HMM_pi	HMM_a[][0]	HMM_a[][1]	HMM_b_param	HMM_b_param	HMM_b_param
0	1	0.947821	0.0521793	4554.06	2.23399e+06	-175475
1	7.35433e-29	0.0694787	0.930521	827.35	-175475	803769

States 2

Dimensions 2

FRET Constraint ☑

Input Path: root:input_traces:

TbT Path: root:input_traces:TbT_0:

Figure 2: The Trace-by-Trace HMM interface, TbT Win.

The data plot (top left of Fig. 2) shows all dimensions of the current input trace (e.g. FRET donor in green, acceptor in red, directly excited acceptor in dark red). Once determined, the states assigned by the Viterbi algorithm (subsequently referred to as *Viterbi path*) are displayed in black. Time is displayed in the user-supplied units (i.e. time bins of the measurement). The name of the current input trace is shown below the plot on the right.

The parameter tabs (bottom left of Fig. 2) display the **Initial Params** or the **Current Params**. HMM_pi, HMM_a and HMM_b_param denote the π, $\mathbf{A}$ and $\mathbf{B}$ parameters, respectively.
The π vector comprises one start probability per state. The matrix $\mathbf{A}$ holds all transition probabilities, i.e. $n \times n$ entries. The rows specify the initial state and the columns the final state. To preclude a certain transition, set the respective matrix element to 0. The Gaussian probability densities (one per state) are parametrized by $\mathbf{B}$, which holds the Gauss positions μ (in d dimensions) in the first column and the co-variance matrix V in the adjacent $d \times d$ dimensions as indicated for two states in Fig. 2. The μ and V of individual states are stored in n individual layers. These are accessible in Igor by the up/down arrows indicated in Fig. 2 (active after clicking the corresponding column).

SMACKS provides typical initial parameters for TIRF experiments. Customized initial $\mathbf{B}$ parameters can be obtained from Gaussian fits to your data histogram. For multi-dimensional data, a diagonal co-variance matrix V is usually a good starting point and possible correlations are fit during optimization.

Click the **Initial Params** tab to adjust the default parameters for your needs. Don't forget to confirm modifications by clicking **Initialize**. Existing **Current Params** are overwritten. You can store and recall current parameters with the respective buttons, **Store** or **Recall**.

The control panel at the right lets you proceed through the TbT workflow: Start by telling SMACKS the number of states (**# States**) that you discern in your data by eye, subsequently referred to as *apparent states*. It is no problem if not every trace reaches all of these states (see below). Next set the number of dimensions (**# Dimensions**). For 2D input, there is the **FRET Constraint** option detailed in Schmid et al. [1].

Now you are ready to run: click **Converge** to optimize an HMM based on the current input. The convergence can be followed by the **Normalized Changes** plot below (see the Miscellaneous section for details). You can always interrupt SMACKS by clicking the red **Stop** button that appears when SMACKS is busy. Alternatively, use **Go** for one iteration at a time and **Vit.** to calculate the most probable state sequence given the current input (data & parameters).

If you are happy with the optimized parameters (i.e. they have converged and yield reasonable state allocation), save them by clicking **Save** or delete them again by **Del**. If the initial parameters are not appropriate for the input data, the Forward Backward algorithm will diverge. SMACKS will let you know by printing a "numerical error" message.

Once appropriate *initial* parameters were found, you can speed up the TbT procedure by calling **TbT Batch Converge** from the SMACKS menu. This will optimize individual parameters for each trace in the dataset. Only the parameters that converged properly will be saved. Next browse through the traces (using $<<$, $>>$) and delete parameters that cause inappropriate Viterbi paths. For traces that do not reach all states, no reasonable parameters can be found. Therefore, those parameters should be deleted even if the optimization converged.

To get around this issue, call **TbT Apply Means** once the inappropriate parameters have been deleted. Thereby, the mean of all saved $\mathbf{B}$ parameters is applied to the remaining trajectories (i.e. those without saved parameters). For simplicity only these "remaining" trajectories will be displayed in the TbT Win. If at this stage, there are many "good quality" traces that are not well fit, you should reconsider your initial definition of the apparent states. Otherwise, sort out unrepresentative traces by deleting their parameters again. These are not considered in the next step: the ENS run.

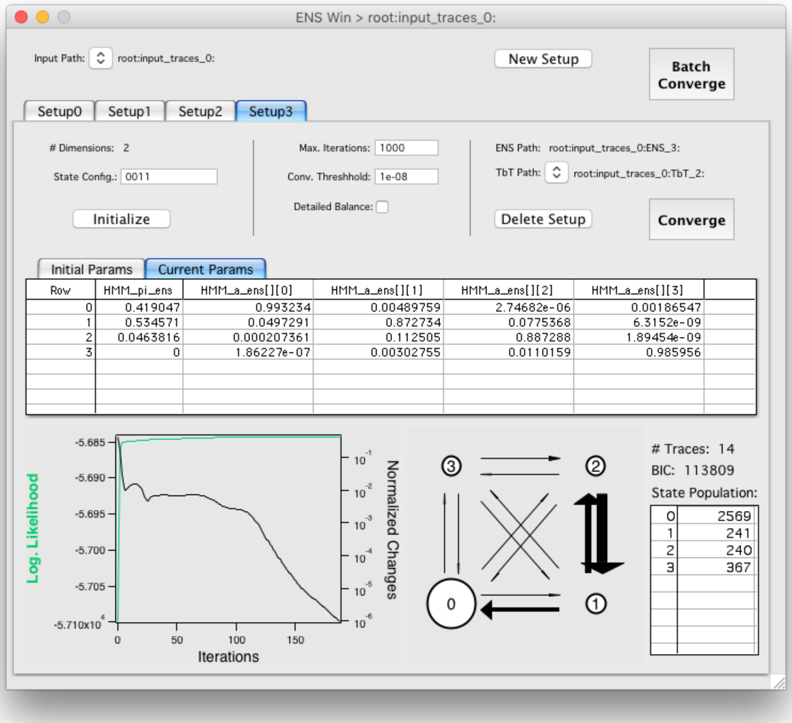
The current **Input Path** and **TbT Path** are both displayed below the controls. The former denotes Igor's data folder holding the original input data, which is not modified by SMACKS. The latter holds all TbT related data, i.e. parameters, Viterbi paths, auxiliaries.

4 Step 2: Semi-Ensemble HMM (ENS)

The ENS workflow is called from the SMACKS menu $\rightarrow$ **Init ENS**, which displays the **ENS Win** shown in Figure 3 and closes the **TbT Win**. (Both windows can be recreated by the respective entry of the SMACKS menu.)
Based on the apparent states and the optimized $\mathbf{B}$ parameters of the TbT workflow, the ENS interface lets you analyze different state models assembled in the setup tabs (**Setup0** etc.).

So tell SMACKS the desired state model configuration, short **State Config.** E.g. "0011" is interpreted as a 4-state model including twice the apparent state0 and twice the apparent state1. Analogously, "0120" denotes twice the apparent state 0 and once the apparent states 1 and 2. If required, adjust the default **Initial Params**. Either way, confirm by clicking **Initialize**.

Further settings are the maximal number of iterations (**Max. Iterations**) per run and the maximal "Normalized Changes" threshold serving as convergence criterion, **Conv. Threshold**. Generally applicable default values are provided. An optional detailed balance constraint, adapted from Greenfeld et al. [5], conserves the model at thermodynamic equilibrium. This



ENS Win > root:input_traces_0:

Input Path: root:input_traces_0:

Dimensions: 2

State Config.: 0011

Max. Iterations: 1000

Conv. Threshhold: 1e-08

Detailed Balance: □

ENS Path: root:input_traces_0:ENS_3:

TbT Path: root:input_traces_0:TbT_2:

Row	HMM_pi_ens	HMM_a_ens[][0]	HMM_a_ens[][1]	HMM_a_ens[][2]	HMM_a_ens[][3]
0	0.419047	0.993234	0.00489759	2.74682e-06	0.00186547
1	0.534571	0.0497291	0.872734	0.0775368	6.3152e-09
2	0.0463816	0.000207361	0.112505	0.887288	1.89454e-09
3	0	1.86227e-07	0.00302755	0.0110159	0.985956

Traces: 14

BIC: 113809

State Population:

0	2569
1	241
2	240
3	367

Figure 3: The ensemble HMM interface, ENS Win.

constraint is adequate for experiments without an external energy source (e.g. ATP).

After submitting the desired **Initial Params** by clicking **Initialize** (see Step 1 for parameter details), **Converge** will optimize the model until either the **Conv. Threshold** or **Max. Iterations** are reached. As in TbT mode, a red **Stop** button shows up, which lets you stop the calculations at any time.

The **Log. Likelihood** plot provides visual feedback on the state of convergence. A cartoon illustrates the current rate model (center bottom). Additionally, the number of considered traces (**# Traces**), the current value of the Bayesian information criterion (**BIC**) and the relative **State Population** deduced from the Viterbi paths are displayed on the right.

Among several model configurations, the most appropriate model can be identified using parsimony criteria, such as **BIC**. Click **New Setup** to create an additional setup or **Delete Setup** to delete a specific setup. The latter does not affect original input and TbT data. **Batch Converge** (top right) runs over all predefined setups. This is useful to perform calculations over night. You can stop and resume calculations at will.

The data of each setup is stored in a specific **ENS Path**. Ensemble HMM makes use of the TbT information stored in the displayed **TbT Path**.

5 Export Results

To export the results, call the SMACKS menu → **Export Results**. The ensemble optimized start and transitions probabilities together with additional results will be stored in SMACKS_summary.dat. The individual $\mathbf{B}$ parameters and Viterbi paths are reported as separate files specified by the original filenames: "original name"_b_param.dat or "original name"_viterbi.dat , respectively.

6 Miscellaneous

- **Normalized Changes** of the diagonal entries of the transition matrix have proven useful for monitoring convergence of the HMM:

  $$\textbf{Normalized Changes} = \sum_{i=0}^{n-1} \frac{|a_{ii} - a'_{ii}|}{a_{ii}}$$

  where a'_{ii} are the diagonal matrix elements of the previous iteration and the sum goes over all states. While the likelihood increases monotonically, the **Normalized Changes** may not.

- The **Viterbi Browser** called from the SMACKS menu lets you review all saved trajectories. Both the **TbT Win** and the **ENS Win** are recreated by the respective entry of the SMACKS menu.

- SMACKS' data hierarchy inside Igor:
  Input data is stored at `root:input_traces_0:` .
  The TbT step is performed in `root:input_traces_0:TbT_0:` .
  The ENS step is performed in `root:input_traces_0:ENS_0:` .
  Higher suffix numbers (`_1`, `_2`, ...) are used for additional folders of the same kind, such as additional ENS setups.
  Call the SMACKS menu → **Set Input Path** to change between different input paths. The complete data folder hierarchy is displayed by Igor's Data Browser (Data menu → Data Browser).

- Calculations run multi-threaded on the cpu, ergo: the more cpu cores, the faster.

- In response to many requests, we include as an additional functionality the confidence interval (CI) computation after [5]:
  SMACKS menu → **Confidence Intervals**. It requires a previously converged model, and considers the selected setup in the **ENS Win**. The output contains a plot of the transition probabilities with CI's as error bars, and a table with all numerical values: CIp, CI in *positive* direction; CIm, CI in *negative* direction; MLE (maximum likelihood estimator), the transition probability; cat, category, e.g. 01 meaning transition 0 → 1.
  CI's are computationally costly; their calculation may take a while.

- In general, don't forget to convert into units of Hertz. SMACKS does not know the sampling rate / exposure time of your experiment. The simple conversion in discrete time (see Section 2) works well in most cases. Only for very fast transitions (i.e. probabilities $\gtrsim$ 10% per time step), it leads to $\gtrsim$10% underestimation of the rate constant [5].

References

[1] S. Schmid, M. Götz, and T. Hugel. Single-molecule analysis beyond dwell times: Demonstration and assessment in and out of equilibrium. *Biophysical Journal*, 111(7):1375 – 1384, OCT 4 2016.

[2] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb 1989.

[3] Gernot A Fink. Markov models for pattern recognition: from theory to applications. *Springer Science & Business Media*, , 2014.

[4] Björn Hellenkamp, Sonja Schmid, Olga Doroshenko, Oleg Opanasyuk, Ralf Kühnemuth, Soheila Rezaei Adariani, Benjamin Ambrose, Mikayel Aznauryan, Anders Barth, Victoria Birkedal, Mark E. Bowen, Hongtao Chen, Thorben Cordes, Tobias Eilert, Carel Fijen, Christian Gebhardt, Markus Götz, Giorgos Gouridis, Enrico Gratton, Taekjip Ha, Pengyu Hao, Christian A. Hanke, Andreas Hartmann, Jelle Hendrix, Lasse L. Hildebrandt, Verena Hirschfeld, Johannes Hohlbein, Boyang Hua, Christian G. Hübner, Eleni Kallis, Achillefs N. Kapanidis, Jae-Yeol Kim, Georg Krainer, Don C. Lamb, Nam Ki Lee, Edward A. Lemke, Brié Levesque, Marcia Levitus, James J. McCann, Nikolaus Naredi-Rainer, Daniel Nettels, Thuy Ngo, Ruoyi Qiu, Nicole C. Robb, Carlheinz Röcker, Hugo Sanabria, Michael Schlierf, Tim Schröder, Benjamin Schuler, Henning Seidel, Lisa Streit, Johann Thurn, Philip Tinnefeld, Swati Tyagi, Niels Vandenberk, Andrés Manuel Vera, Keith R. Weninger, Bettina Wünsch, Inna S. Yanez-Orozco, Jens Michaelis, Claus A. M. Seidel, Timothy D. Craggs, and Thorsten Hugel. Precision and accuracy of single-molecule fret measurements - a multi-laboratory benchmark study. *Nature Methods*, 15(9):669–676, 2018. doi: 10.1038/s41592-018-0085-0. URL https://doi.org/10.1038/s41592-018-0085-0.

[5] Max Greenfeld, Dmitri S. Pavlichin, Hideo Mabuchi, and Daniel Herschlag. Single molecule analysis research tool (SMART): An integrated approach for analyzing single molecule data. *PLoS ONE*, 7(2):e30024, 2012.