SMART: Single Molecule Analysis Research Tool
Version 1.0

Max Greenfeld
Dmitri Pavlichin
Hideo Mabuchi
Daniel Herschlag

Contents

List of Figures

List of Tables

Preface

The Single Molecule Analysis Research Tool (SMART) package provides a comprehensive set of tools for analyzing single molecule data. This instruction manual aims to provide a detailed account of all the functionality built into SMART so that users can get the most from the package. To aid in presenting these topics video tutorials have been recorded to further supplement this manual. To Begin using SMART only a few things are needed:

1. Data from a single molecule experiment
2. Desktop computer running Matlab 2008a or higher
3. SMART software package downloaded from the website maintained by the Simbios (The NIH Center for Physics-based Simulations of Biological Structures) at Stanford University

With these items in hand you will quickly be analyzing your data with hidden markov analysis and powerful statistical tools that allow the behavior of individual molecules to be faithfully analyzed. The graphical interface of SMART enables rapid, sorting, inspecting and plotting of data, which allows you to spend more time focused on analyzing the data. The standardized data format used in SMART will enable sharing of data within a lab and among colleagues. Finally the SMART package is easy to modify, so advanced users can improve on SMART to meet their particular needs.

The developers of SMART anticipate being able to integrate user feedback into future versions of the package so please contact us with suggestions or details of improvements you have made.

Stanford, CA

M.G.

D.P.

H.M.

D.H.

Chapter 1

Analyzing Data with SMART

SMART has extensive functionality all of which is accessible via a graphical interface. This chapter illustrates how to analyze data using SMART by following the analysis of example data that is provided with the SMART software package.

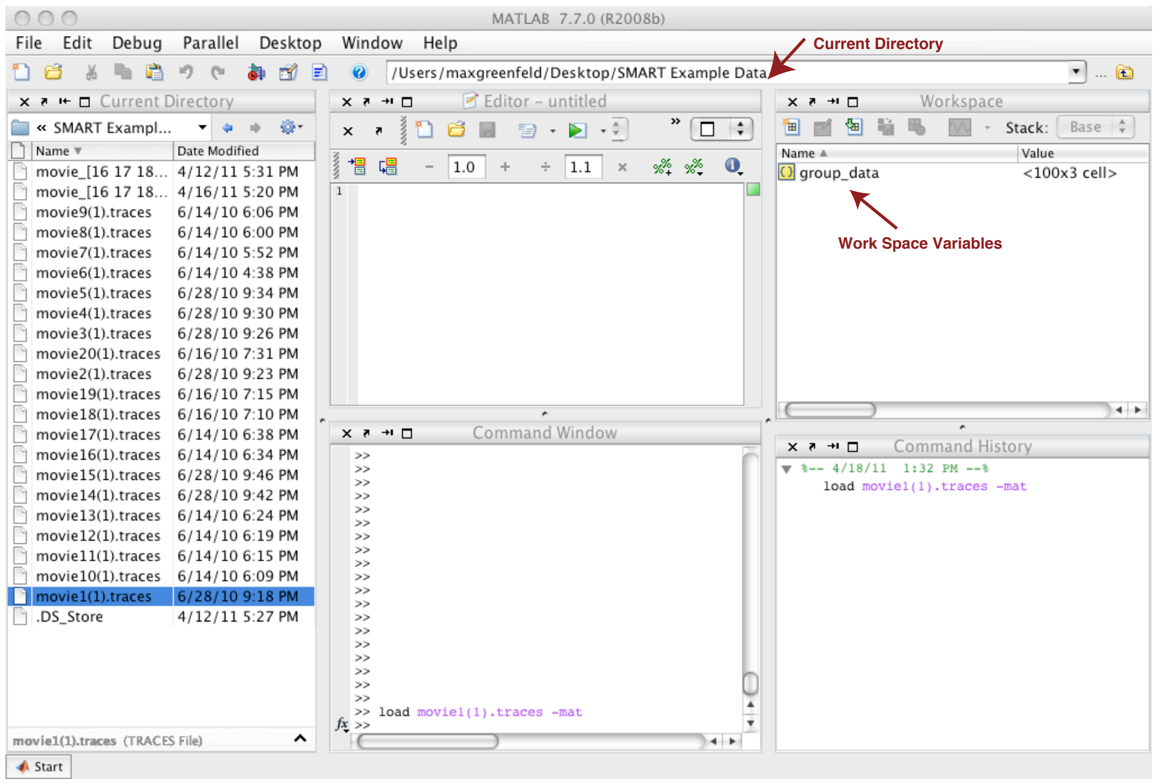
The SMART software package is implemented in the MATLAB programing language and requires MATLAB 2008a or latter to run. SMART can be run without programing experience or experience with MATLAB. If your organization does not all ready have a MATLAB license one can be purchased from MathWorks. First MATLAB should be installed on the computer you will be using for data analysis; this can be accomplished by following the instructions accompanying MATLAB. Second the folder containing the SMART scripts should be added to your MATLAB path, from the default MATLAB window (See Fig 1.1) this can be done by:

1. Go to File and select Set Path
2. In the Set Path window select Add with Subfolders
3. Navigate to the location where you saved the SMART folder and select Open
4. Save this path by selecting Save in the Set Path window

If you work in a lab that already uses SMART you are ready to begin analyzing your data. If you are using SMART for the first time you, you will have to put your raw data files into the .traces format SMART requires which is described in the section 1.1.3.

1.1 Raw Data Files in SMART & .traces Files

To access data in the SMART graphical interface your raw data needs to be put in a form that is recognizable by SMART. No standardized data format currently



Name	Date Modified
movie_[16 17 18...	4/12/11 5:31 PM
movie_[16 17 18...	4/16/11 5:20 PM
movie9(1).traces	6/14/10 6:06 PM
movie8(1).traces	6/14/10 6:00 PM
movie7(1).traces	6/14/10 5:52 PM
movie6(1).traces	6/14/10 4:38 PM
movie5(1).traces	6/28/10 9:34 PM
movie4(1).traces	6/28/10 9:30 PM
movie3(1).traces	6/28/10 9:26 PM
movie20(1).traces	6/16/10 7:31 PM
movie2(1).traces	6/28/10 9:23 PM
movie19(1).traces	6/16/10 7:15 PM
movie18(1).traces	6/16/10 7:10 PM
movie17(1).traces	6/14/10 6:38 PM
movie16(1).traces	6/14/10 6:34 PM
movie15(1).traces	6/28/10 9:46 PM
movie14(1).traces	6/28/10 9:42 PM
movie13(1).traces	6/14/10 6:24 PM
movie12(1).traces	6/14/10 6:19 PM
movie11(1).traces	6/14/10 6:15 PM
movie10(1).traces	6/14/10 6:09 PM
movie1(1).traces	6/28/10 9:18 PM
.DS_Store	4/12/11 5:27 PM

Name	Value
group_data	<100x3 cell>

```
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>> load movie1(1).traces -mat
>>
```

```
%-- 4/18/11  1:32 PM --%
   load movie1(1).traces -mat
```

Figure 1.1: Default MATLAB window. This screenshot highlights the Command Window, Workspace, Work Space Variables and Current Directory referenced throughout descriptions in the text

exists for saving raw single molecule data. The SMART package uses a .traces file for this task. The .traces is a standard MATLAB .MAT file type that has a uniform data structure that is convenient for storing raw single molecule data. A quick way to inspect the format of a .traces file is to load one of the example traces supplied in the "SMART Example Data" folder downloaded with SMART.

This can be done by navigating to the "SMART Example Data" downloaded with the SMART package, then load an example .traces file by typing in the Command Window

```
>> load -mat movie1(1).traces
```

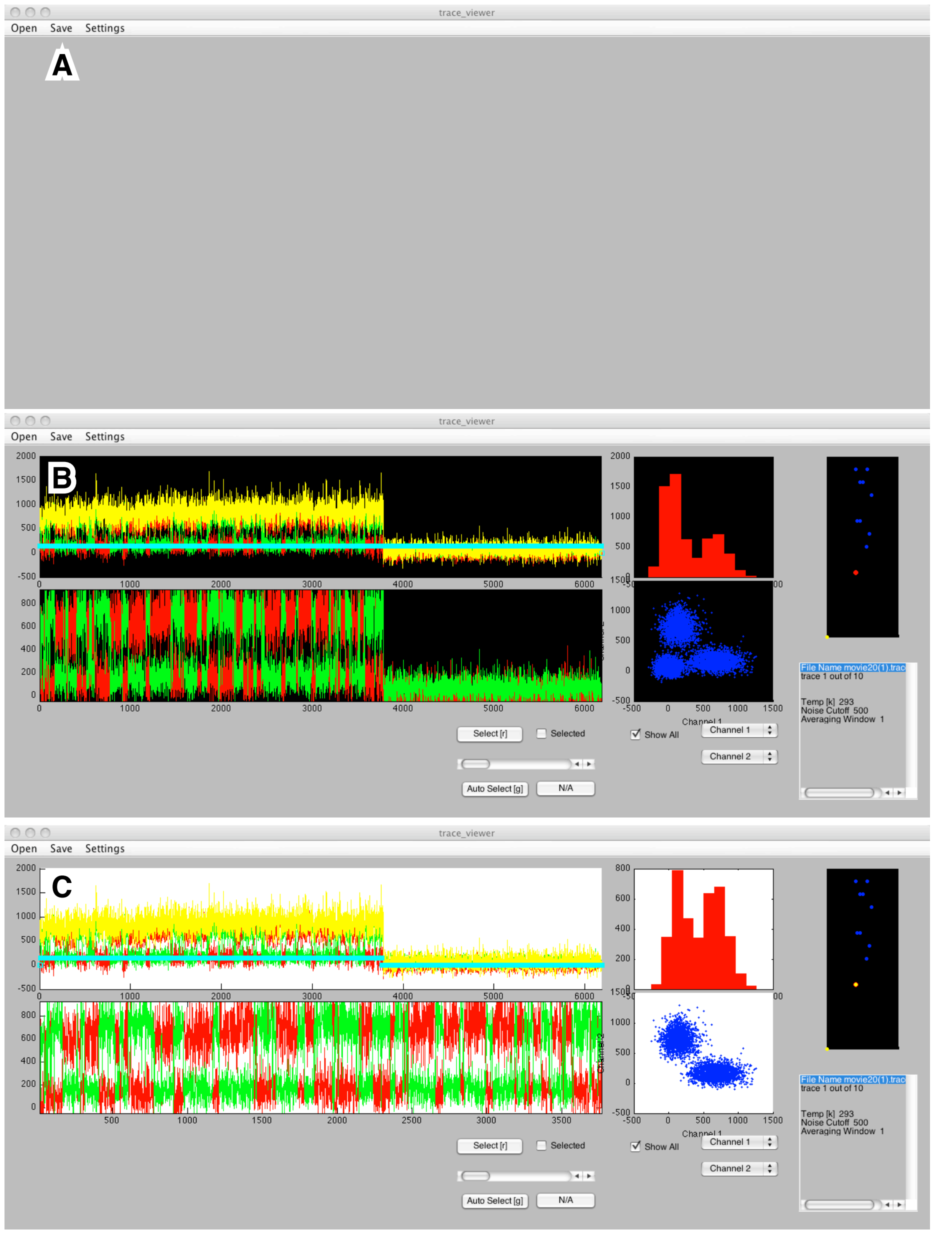
The raw data has now been loaded into the Workspace under the variable group_data. The variable group_data is a cell array where each row corresponds to a trace (in this example there are 10 traces in the example data) and the three columns are used to store different aspects of the raw data. Tables 1.1 and 1.2 details what is stored in each column. A user of SMART must put their data in this format. This will require writing a simple matlab script that opens your current raw data files and saves them as .traces files, See section 1.1.3.

1.1.1 Viewing and Selecting Traces

This data format allows for .traces files to contain traces which will only have limited regions analyzed and for some traces to be skipped in the analysis. This feature can be clearly seen by using the interface that allows the viewing, inspecting and selection of .traces files. Type into the command line

```
>> trace_viewer
```

this will bring up a window that looks like, Fig. 1.2A. From the Open menu select one of the .traces files in the "SMART Example Data" folder. If the backgrounds of the trace being displayed are black (e.g. 1.2B) the trace is not selected and will not be analyzed as described in Section 1.2. However, the entire trace or a region of the trace can be manually selected using the cursor to grab a region of interest and then pressing Select. Once the trace is selected, the background will be white as shown in 1.2C. Once all the traces of interest have been selected from the Save menu select Save Data, this will overwrite the original .traces file with a new one where the only thing that has changed are the differences in the selected regions.



File Name movie20(1).trac
trace 1 out of 10
Temp [k] 293
Noise Cutoff 500
Averaging Window 1

File Name movie20(1).trac
trace 1 out of 10
Temp [k] 293
Noise Cutoff 500
Averaging Window 1

Figure 1.2: The trace_viewer interface allows for .traces files to be inspected and for regions of traces to be selected for analysis. **(A)** The trace_viewer interface immediately after being opened. **(B)** The trace_viewer displaying the first trace in a .traces file prior to selecting a region of interest **(C)** The trace_viewer after a region of interest was selected

1.1.2 Combine .traces Files

SMART is set up to allow data from different types of experiments to be analyzed simultaneously. Raw data from different types of experiments (e.g. from different days, different types of molecules or different solution conditions) in the .traces format and with the file naming scheme movieX(Y).traces can be easily combined using the numbering indices included in the file names using the proc_traces window. The X and Y in the file name correspond to variables known as movie_num and Y is the movie_ser (See Section 1.1.3 and Table1.2 for detailed descriptions) in the .traces files.

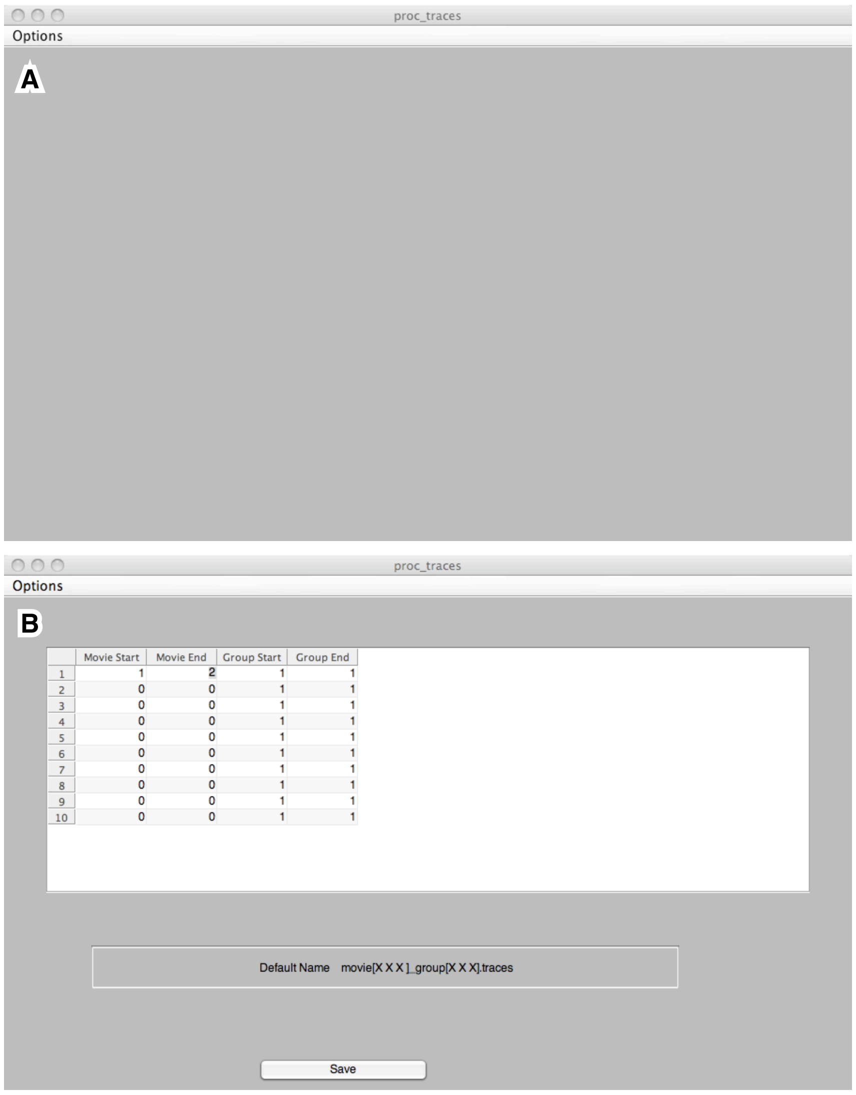
This numbering scheme was settled upon because the developers of SMART initially conducted experiments where the most basic form of the raw data were groups of about 100 traces extracted from movies that monitored a wide-field image of molecules sparsely attached to a surface of a slide. The movie number (movie_num) is sufficient to identify most experiment types. However, in some instances multiple movies are taken of the same field of view of molecules, leading to multiple movies that have to be subsequently aligned and molecules colocalized. To deal with this type of data it is convenient to have an orthogonal index for each movie number (i.e. movie_ser). A user of SMART can choose to use the indices in the manner that there were initially intended or for a scheme the better fits their experimental design.

To begin set the current path of MATLAB to the directory that contains your raw data. To begin grouping .traces files in this directory type into the command line

```
>> proc_traces
```

This will load the proc_traces window. Select from the “Options” pulldown “Combine Raw Data” to bring up a field that will allow a range of movie numbers and movie series to be input. For the field shown in Fig. 1.3 each row corresponds to a sequential set of .traces files to be combined. For the example shown movie1(1).traces and movie2(1).traces will be combined into one file. If one or more movies in the series is not in the directory the files that are present will still be combined into one file. If discontinuous movie numbers need to be used simply move to the next row.

A default concatenated file name will be made that shows the movie number and movie series number that were input. Alternatively a user specified name can be typed into the box indicating the default name format. After all values have been input click Save. This will combine all the data and save a .traces file in the current directory.



	Movie Start	Movie End	Group Start	Group End
1	1	2	1	1
2	0	0	1	1
3	0	0	1	1
4	0	0	1	1
5	0	0	1	1
6	0	0	1	1
7	0	0	1	1
8	0	0	1	1
9	0	0	1	1
10	0	0	1	1

Default Name movie[X X X]_group[X X X].traces

Figure 1.3: To combine .traces files the proc_traces interface can be used. **(A)** The default proc_traces window immediately after opening. **(B)** The proc_traces window set to group movie1(1).traces and movie2(1).traces as described in the text.

1.1.3 Generating .traces Files

MATLAB can open almost any data type and a number of built in functions allow data to be loaded by running one line of code. Once your data is open you can copy the data into a group_data cell array that has the format described in Tables 1.1 and 1.2 . The group_data variable should them be save with a file name that ends with a .traces extension. The file name can be anything however, the format movieX(Y).traces where X is the movie_num and Y is the movie_ser described in Table 1.2 has proven to be a useful way to categorize raw data for the individuals who created SMART and is required to access all the functionality described in section 1.1.2 and 1.4 .

To help with this conversion process we have provided two scripts the convert two different types of raw data into .traces files. The simplest form of the raw data that can be converted to a .traces formatwith these scrips is a single trace that has been saved in a Tab Delimited Text (.txt) file. To see how this script is used set the current directory to the SMART Example Data folder, in this there is a text file titled 'simplest_raw_data.txt' this can be opened and inspected with any standard text editor. This file can be converted to a .traces file by entering into the command window of MATLAB

```
>> generate_dot_traces('simplest_raw_data.txt')
```

Running this script will generate a .traces file title 'simplest_raw_data.traces' which will be saved in the current directory. This .traces file can then be used in all the graphical interfaces of SMART.

Modifying of the generate_dot_traces script would likely be a good starting point for converting an arbitrary data format to the .traces format which is compatible with SMART. If you have limited programing experience consider finding an individual in your organization who does have experience. If that resource is not available there are many online tutorials that provide a good introduction to the basics of MATLAB programing and are a good starting point for writing this conversion script. If you have trouble with this step feel free to contact the developers of SMART for assistance.

We have also provided a script that allows traces that have been saved in a specific type of binary file to be opened and converted into the SMART format. An example file ('movie3.traces' has been included which allows data that has been generated by extracting traces with scripts distributed from Taekjip Ha's group (University of Illinois at Urbana-Champaign) from wide field movie fluorescent images of single molecules to be analyzed with SMART. This example can be run if the current

Column 1	Column 3	Column 3
struct with fields detailed in Table 1.2	raw data m×n matrix	empty cell or m×1

Table 1.1: Cell array structure of group_data. Each row contains a trace. For each trace the first column cell contains a structure array with the fields listed in Table 1.2. The second column cell containes the raw data. The data should be an m × n matrix where m is a unique time in the time series data n corresponds to data from 1 to an arbitrary number of channels. The graphical interface has been generalized to handle up to 4 channels, as graphical display of more than this becomes cumbersome, the HMM algorithms are generalized to handle an arbitrary number of channels. The third column cell is either empty, indicating SMART should not analyze that trace or is a m × 1 matrix where values in the column vector are either 1 or 0. Values of 1 indicate that part of the trace should be analyzed values of 0 indicate that part should not be analyzed.

directory is set to SMART Example Data folder by typing

```
>> btraces_to_SMART_traces('movie3.traces')
```

Modification of this script is also a useful means for converting your data into a SMART compatible format.

variable name	variable description
name	string containing file name
gp_num	is reserved for future use and should be set to NaN
movie_num	an index for grouping experiment set to an integer 1 or greater
movie_ser	an index for grouping traces that have the same movie_num typically set to 1
trace_num	an index for traces within particular experiment
spots_in_movie	how many traces were initially identified in the expermeint or are in the .traces file
positions_x	if molecules have an identifiable x coordinate set to that coordinate
positions_y	if molecules have an identifiable y coordinate set to that coordinate
positions	stores the position of all other potential traces within a movie
fps	indicates the data acquisition rate or is set to NaN
len	indicates the length of the entire trace
nchannels	indicates the number of data channels in the trace

Table 1.2: Fields in first column of .traces files

1.2 Process Data

Once a user has a .traces file they are interested in analyzing with an HMM they need to load that file into the proc_traces interface. From the proc_traces select "Open" from the pulldown menu. This will bring up the "Select File to Open" window that allows the user to navigate to and select a .traces file they want to process (i.e. fit to an HMM). Once a .traces file is selected (in the example the movie[1 2]_group[1].traces which was created in Section 1.1.2 was selected) the proc_traces window will configure to look like that shown in the screen shot depicted in Fig. 1.4.

This interface allows users to specify the model types to be fit to the data, details on how those variables should be fit and some additional summary statistics to be fit. This interface is also formated so that future iterations of this package can be expanded to accommodate calculating other summary statistics or alternative model types or fitting algorithms.

The screen shots shown in Fig. 1.4A and B depict fields that take on a range of values that will specify the model to be fit and how that fit is to be completed. The views shown are the default values and it is set up to analyze a FRET trace (or any two channel data) with a two state HMM and by determining confidence values on gaussian emission parameters. For the example the movie[1 2]_group[1].traces that is loaded all that is required to analyze the data with this model is inputing the FPS field, which can be set to a place holder of 1 since this is simulated data. Pressing Fit will run the script and the Command Window will look as in Fig. 1.7. When the script is done running a file with the same name as the .traces file but now with a .proc extension will be save in your current directory.

The section below describes the meaning of each field where changes from the default values are common. Table 1.3 provides an additional summary of these parameters. For fields that are not described See Tables 2.1, 2.2, 2.3 and 2.4 for a detailed descriptions. Two examples of how these parameters typically change are shown in Fig. 1.5, see the caption for a description of the HMM model being depicted.

1. **HMM** This field indicates if HMM fitting should be done. In some instances a user of SMART might want to use the graphical interface to calculate a summary statistic that does not involve fitting an HMM, so fitting an HMM model would be a waste of time. Input yes to fit HMMs or no to skip the HMM fitting.

2. **Einital** Sets the initial conditions for the emission model parameters. In most cases this should be set to auto. This setting will use an algorithm to determine

different initial conditions for each trace to be fit. In rare instances these values do not allow for the algorithm to converge to a good fit. This will typically be determined by inspecting traces that end up having outlier fitted parameters. If a user wants to fit a trace using a user specified initial starting condition they can input a string that conforms to the syntax indicated in Table 2.2.

3. **Ainitial** This has an identical role as Einitial except this is used to initialize the fits for the transition matrix (rate parameters). In most cases this should be set to auto. However, a string that specifies the initial conditions can be input; see Table 2.1 for the string syntax.

4. **auto_confint** This field indicates the confidence bound that will be calculated. This value can range from 0 to 1. The 0.97 shown for this example indicates the 97% confidence bound.

5. **Detailed Balance** This field indicates if the fit should be completed imposing a constraint of detailed balance. Input 1 to constrain the fit or 0 to skip the constraint.

6. **Channel Spec** C1, C2, C3 and C4 indicate 4 input signal channels each of which can be fit in the SMART interface. If the column is left blank like C3 and C4 in the example shown the interface will not expect data for than channel. For each channel the 2 parameters that need to be specified are indicated below (See Fig. 1.6)B.

    (a) **fitChanellType** This specification indicates the noise model that is to be fit to each state in a model. Currently SMART can fit to poissonian or gaussian noise models. These are specified by inputing the strings gauss or poisson.

    (b) **Error State mr** This allows the user to specify if a confidence interval should be calculated for the λ parameter in a poissonian noise model and the λ and μ (mean) parameters in a gaussian noise model. To calculate a confidence bound enter 1 to skip this calculation enter 0.

7. **nStates** This field indicates the number of states in the model being fit. This value can take on integer values of ≤ 10. This will create a generic HMM where all transitions are allowed (See Fig. 1.6A). When this value is changed it reinitializes the graphically displayed HMM shown next to the table, the discStates field, the state Spec field and the Param 1, Param 2 and Param 3 fields.

8. **discStates** This field is used to specify if a state should have a unique emission parameter. If you want two states to have the same emission intensity assign

them to be the same number. Numbering begins with 1 and ends once all states have an assignment. The graphical representation of HMM indicates which states have identical emissions by assigning those states to have the same color.

9. **State Spec** This field changes depending on the value of nStates. This field can be read as a generic HMM transition matrix. State, a and k are used synonymously at this time, but future versions will consistently use the a notation. Allowed transitions are set to 1. To disallow transition set the transition values to 0. To calculate a confidence interval on an inferred transition rate set the value to 2. Use the graphical representation of the HMM as a guide for the HMM being specified.

10. **Param 1, Param 2 and Param 3** These pull down menus can be used to choose groups of parameters that will be clustered. Up to 3 parameters can be jointly clustered. Add parameters to be clustered from left to right. Up to 10 different groups of parameters can be specified in one run of fitting.

Explanations of the fields in Fig. 1.4B are as follows.

1. **FPS** The field allows the user to input the instrument acquisition rate. This is necessary for converting to continuous time transition rates. This can be any integer value. If no conversion is necessary input 1.

2. **temp** The field allows you to input the temperature an experiment was conducted at. This is useful for indexing experiments conducted at different temperatures and can be used for calculating thermodynamic parameters i.e. ΔG which SMART can do for two state traces.

3. **FRET Spec** For users conducting FRET experiment SMART has some built in functionality to calculate and visualize FRET using simple methods. This functionality results from older unpublished versions of SMART. Since it might be useful for some users we left it in the user interface. To run this section enter yes to skip the section enter no.

4. **Cross Talk** This field defines the amount of cross talk between the donor and acceptor fluorophores. This can take on any value between 0 and 1 but for most dye systems is about 0.05. An investigator needs to determine this value for their system of study.

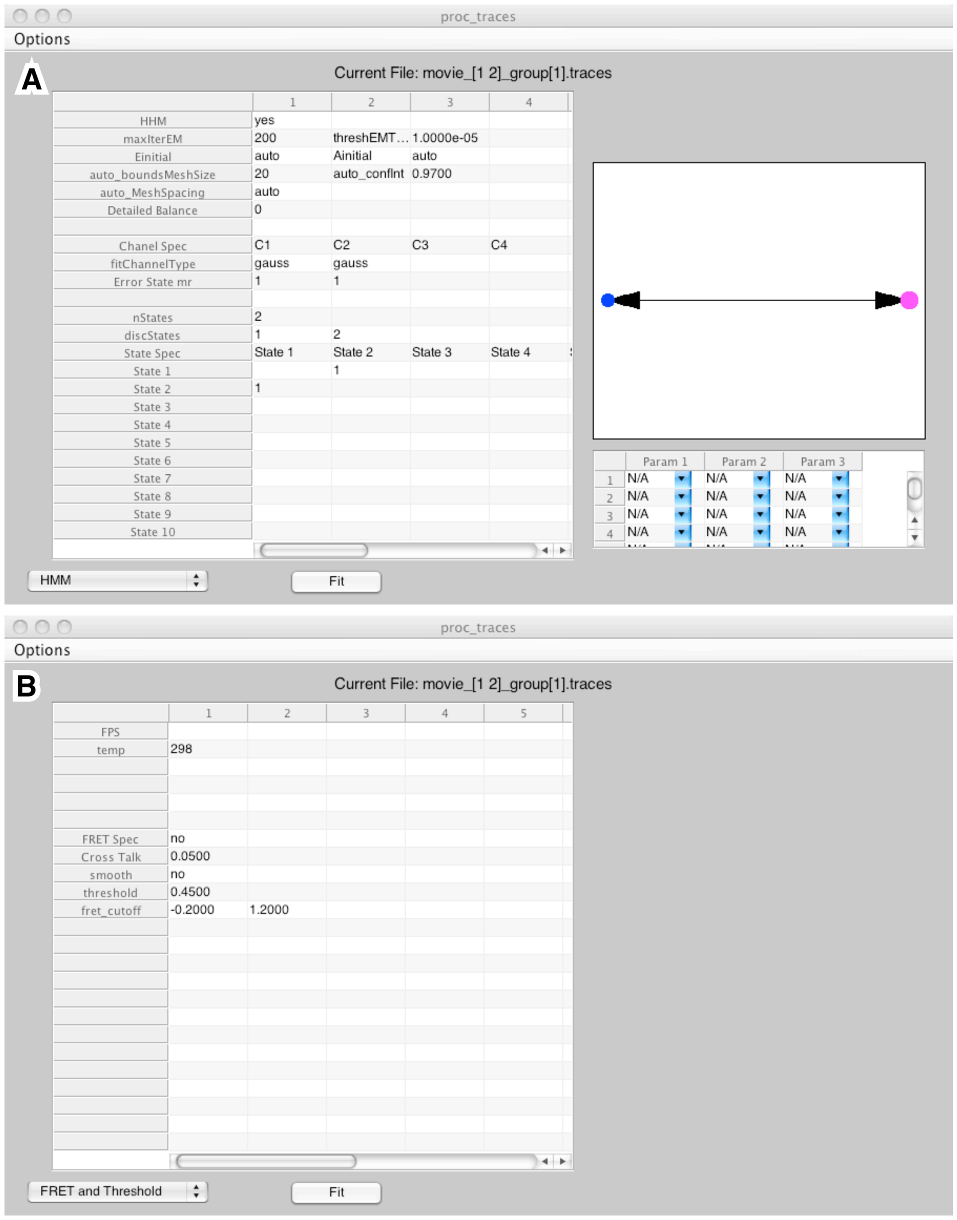
5. **smooth** Smooth uses the MATLAB function smooth to put traces through a box car averaging filter. The default window is no this indicates no filter will be used. Alternatively input an integer value larger than 1 to indicate the averaging window to be used.

6. **threshold** This indicates the threshold level that will be used to determine the point of transition between the low and high FRET states. This can take on any value between 0 to 1.

7. **fret_cutoff** When calculating FRET it is common to have some values that are anomalously high or low. An easy way to deal with this limitation is to set values beyond a certain level to a thresholded value. Enter the lower threshold value in the column labeled 1 and the upper threshold value in the column labeled 2.

GUI Label	Functional Description	Selection Options
HMM	Use HMM fitting algorithm	yes or no
maxIterEM	Use the default or see Table 2.2	200
threshEMT	Use the default or see Table 2.2	10^{-3}
Einital	Use the default or see Table 2.2	'auto'
Ainitial	Use the default or see Table 2.1	'auto'
auto_confint	Indicates what confidence bound should be calculated	0 to 1
auto_boundsMeshSize	Use the default or see Table 2.3	10
auto_MeshSpacing	Use the default or see Table 2.3	'square'
Detailed Balance	Fit model with the detailed balance constraint	0 or 1
Channel Spec	model parameters for 4 channels (C1/C2/C3/C4) can be specified in the SMART interface	
fitChanellType	Emission distribution to be fit	poisson or gauss
Error State mr	Calculate confidence int ervals for the emissions parameter of the specified chanell	1 (yes) or 0 (no)
nStates	Number of states in the HMM	an integer ≤ 10
discStates	Which states have a unique emission intensity	1 (yes) or 0 (no)
State Spec	Specified allowed transition for the nStates model an if confidence intervals should be calculated	0 no transition, 1 transition, 2 transition + confidence interval

Table 1.3: Summary of how to define HMMs in SMART. All parameters in this table correspond to fields depicted in Fig. ??A.



Current File: movie_[1 2]_group[1].traces

	1	2	3	4
HHM	yes			
maxIterEM	200	threshEMT…	1.0000e-05	
Einitial	auto	Ainitial	auto	
auto_boundsMeshSize	20	auto_confInt	0.9700	
auto_MeshSpacing	auto			
Detailed Balance	0			
Chanel Spec	C1	C2	C3	C4
fitChannelType	gauss	gauss		
Error State mr	1	1		
nStates	2			
discStates	1	2		
State Spec	State 1	State 2	State 3	State 4
State 1		1		
State 2	1			
State 3				
State 4				
State 5				
State 6				
State 7				
State 8				
State 9				
State 10				

	Param 1	Param 2	Param 3
1	N/A	N/A	N/A
2	N/A	N/A	N/A
3	N/A	N/A	N/A
4	N/A	N/A	N/A

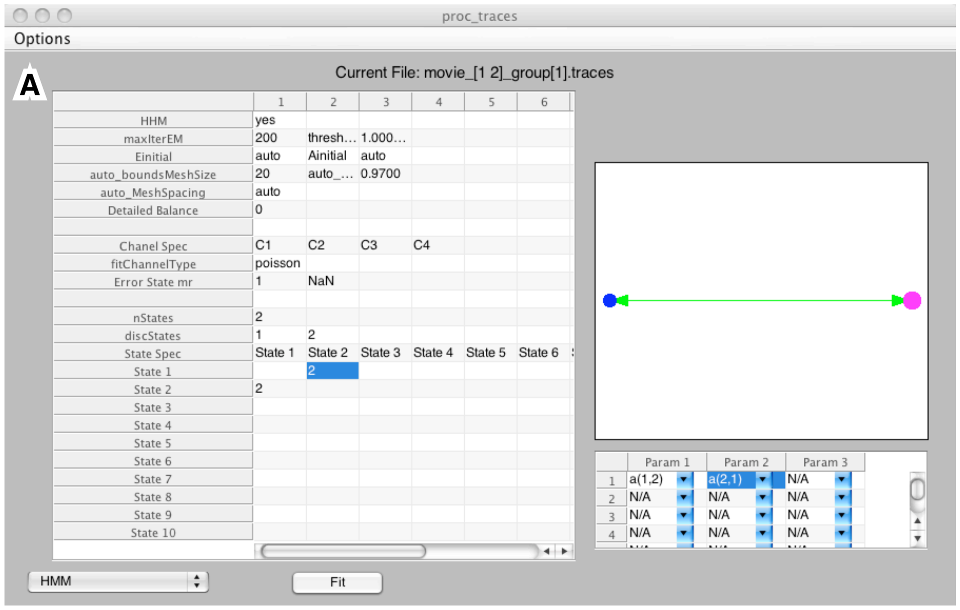
HMM

Current File: movie_[1 2]_group[1].traces

	1	2	3	4	5
FPS					
temp	298				
FRET Spec	no				
Cross Talk	0.0500				
smooth	no				
threshold	0.4500				
fret_cutoff	-0.2000	1.2000			

FRET and Threshold

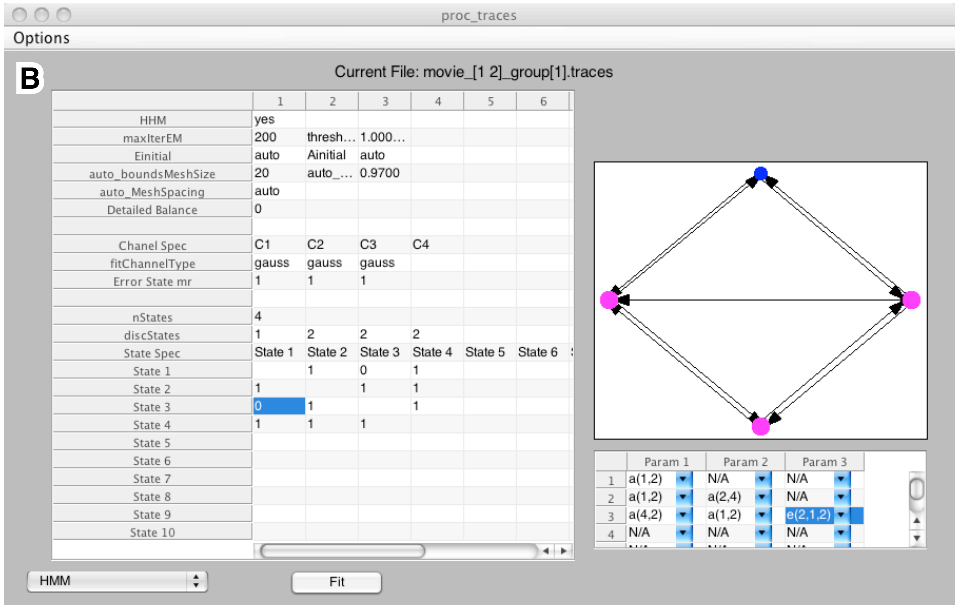
Figure 1.4: Depiction of the default fields for defining fitting parameters. **(A)** All fields for the HMM fitting. **(B)** Fields for inputing FPS, temperature and FRET parameters.



Current File: movie_[1 2]_group[1].traces

	1	2	3	4	5	6
HHM	yes					
maxIterEM	200	thresh…	1.000…			
Einitial	auto	Ainitial	auto			
auto_boundsMeshSize	20	auto_…	0.9700			
auto_MeshSpacing	auto					
Detailed Balance	0					
Chanel Spec	C1	C2	C3	C4		
fitChannelType	poisson					
Error State mr	1	NaN				
nStates	2					
discStates	1	2				
State Spec	State 1	State 2	State 3	State 4	State 5	State 6
State 1		2				
State 2	2					
State 3						
State 4						
State 5						
State 6						
State 7						
State 8						
State 9						
State 10						

	Param 1	Param 2	Param 3
1	a(1,2)	a(2,1)	N/A
2	N/A	N/A	N/A
3	N/A	N/A	N/A
4	N/A	N/A	N/A



Current File: movie_[1 2]_group[1].traces

	1	2	3	4	5	6
HHM	yes					
maxIterEM	200	thresh…	1.000…			
Einitial	auto	Ainitial	auto			
auto_boundsMeshSize	20	auto_…	0.9700			
auto_MeshSpacing	auto					
Detailed Balance	0					
Chanel Spec	C1	C2	C3	C4		
fitChannelType	gauss	gauss	gauss			
Error State mr	1	1	1			
nStates	4					
discStates	1	2	2	2		
State Spec	State 1	State 2	State 3	State 4	State 5	State 6
State 1		1	0	1		
State 2	1		1	1		
State 3	0	1		1		
State 4	1	1	1			
State 5						
State 6						
State 7						
State 8						
State 9						
State 10						

	Param 1	Param 2	Param 3
1	a(1,2)	N/A	N/A
2	a(1,2)	a(2,4)	N/A
3	a(4,2)	a(1,2)	e(2,1,2)
4	N/A	N/A	N/A

Figure 1.5: Example of how the HMM fitting window will look for different types of HMMs. **(A)** A 2 state HMM being fit to 1 channel data with poisson emissions and with confidence intervals being calculated for all fitted parameters and with the covariation being calculated on the two inferred rate constants to allow clustering latter on. **(B)** A 4 state HMM fit to 3 channel data with gaussian emissions in each channel, The fit is constrained to only allow two unique emission levels and transitions are not allowed between States 1 and 3 and 3 and 1. Confidence intervals will be calculated on the inferred emission parameters and some of the inferred parameters will have their covariation calculated to allow clustering of those parameters latter on.

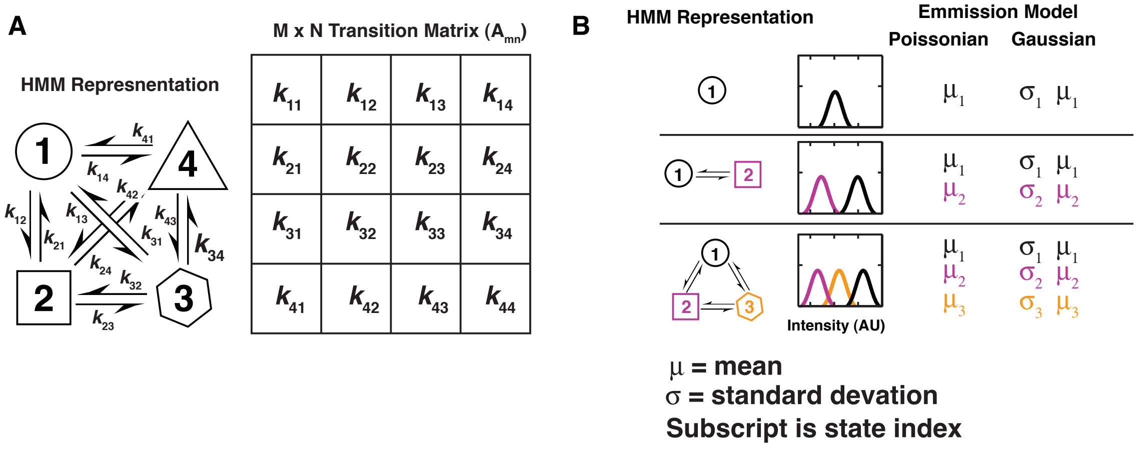
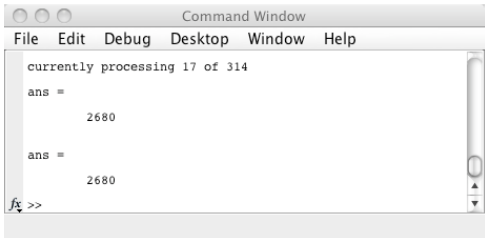

A
HMM Represnentation
M x N Transition Matrix (A_{mn})
k_{11} k_{12} k_{13} k_{14}
k_{21} k_{22} k_{23} k_{24}
k_{31} k_{32} k_{33} k_{34}
k_{41} k_{42} k_{43} k_{44}
B
HMM Representation
Emmission Model
Poissonian
Gaussian
Intensity (AU)
μ = mean
σ = standard devation
Subscript is state index

Figure 1.6: HMM transition matrix and emissions notation. **(A)** A general 4 state HMM is depicted on the right with all transition rates shown. On the right the corresponding 4 × 4 transition matrix is shown with the position of each of the inferred rate constants. **(B)** For each channel an emission model is fit. Depending on if a Poissonian or Gaussian emission model is chosen one (μ) or two (μ and σ) fitted parameters is defined for each state. The HMM representation of 1 to 3 state models is shown on the left. For a single channel the preliminary (before fits) emission distributions are shown in the middle and the fitted parameters for Poissonian or Gaussian models are shown on the right.



```
currently processing 17 of 314
ans =

        2680


ans =

        2680

>>
```

Figure 1.7: Command window while traces are being processed. This will appear after Fit in Fig. 1.4 has been pressed.

1.3 Loading .proc Files

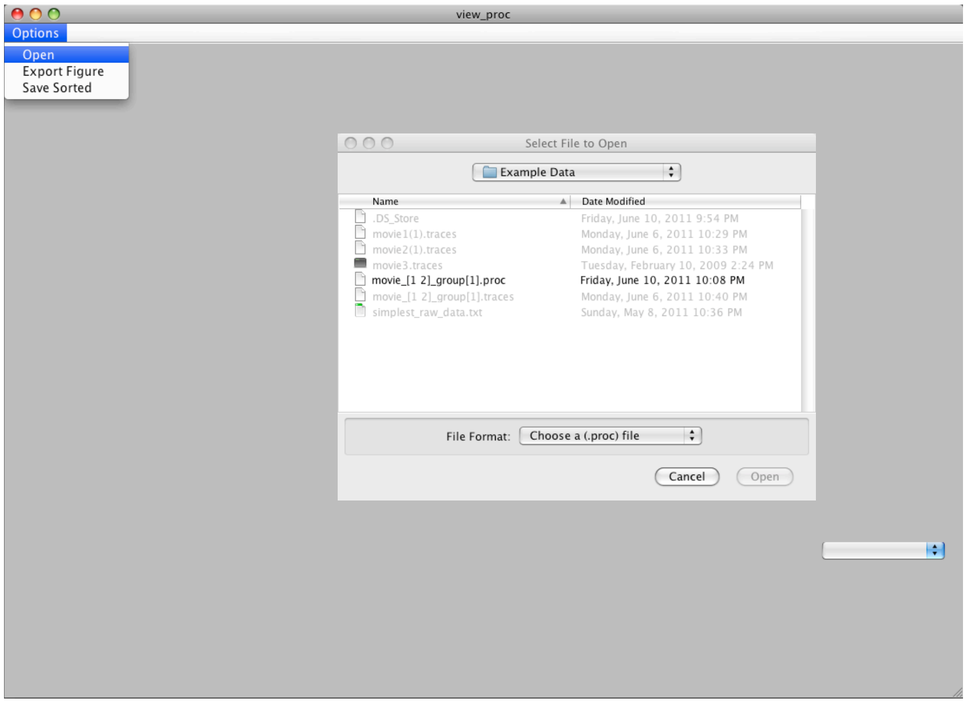
Once the traces have finished being processed resulting in a .proc file being generated the data is ready to be analyzed. This is done in the view_proces interface. This interface allows molecules to be sorted based on user specified properties, for individual molecules to be inspected, for summary plots of groups of molecules to be generated and for molecules to be clustered.

To begin enter into the command window of MATLAB

```
>> view_proc
```

This will load the view_proc window. As shown in Fig. 1.8 select from the "Options" pulldown "Open". This will bring up a window (depicted in the figure) that will allow you to navigate to and select the .proc file you are interested in analyzing. For the running example throughout this manual the movie[1 2]_group[1].proc is selected to be opened.

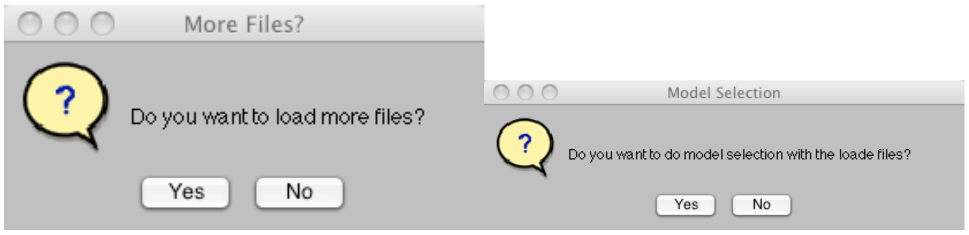
Before the data is completely opened the user will have to specify whether they want to load more than one .proc file and whether they want to do model selection on the data. Two popup windows are given to specify this. The first popup is shown in Fig. 1.9a. If you want to analyze data from multiple .proc files select Yes and this will take you back to the field shown in Fig. 1.8. If you select No the popup shown in Fig. 1.9b will appear and give the user the option of doing model selection with the loaded data. Selecting Yes will take the user to the model selection interface (see Section 1.15) selecting No will bring up the sorting interface (see Section 1.4).



Name	Date Modified
.DS_Store	Friday, June 10, 2011 9:54 PM
movie1(1).traces	Monday, June 6, 2011 10:29 PM
movie2(1).traces	Monday, June 6, 2011 10:33 PM
movie3.traces	Tuesday, February 10, 2009 2:24 PM
movie_[1 2]_group[1].proc	Friday, June 10, 2011 10:08 PM
movie_[1 2]_group[1].traces	Monday, June 6, 2011 10:40 PM
simplest_raw_data.txt	Sunday, May 8, 2011 10:36 PM

File Format: Choose a (.proc) file

Figure 1.8: Opening .proc files in SMART. **(A)** In the view_proc window select Open from the Options tab. **(B)** Navigate to and select the desired .proc file



Do you want to load more files?

Do you want to do model selection with the loade files?

(a) Popup 1 (b) Popup 2

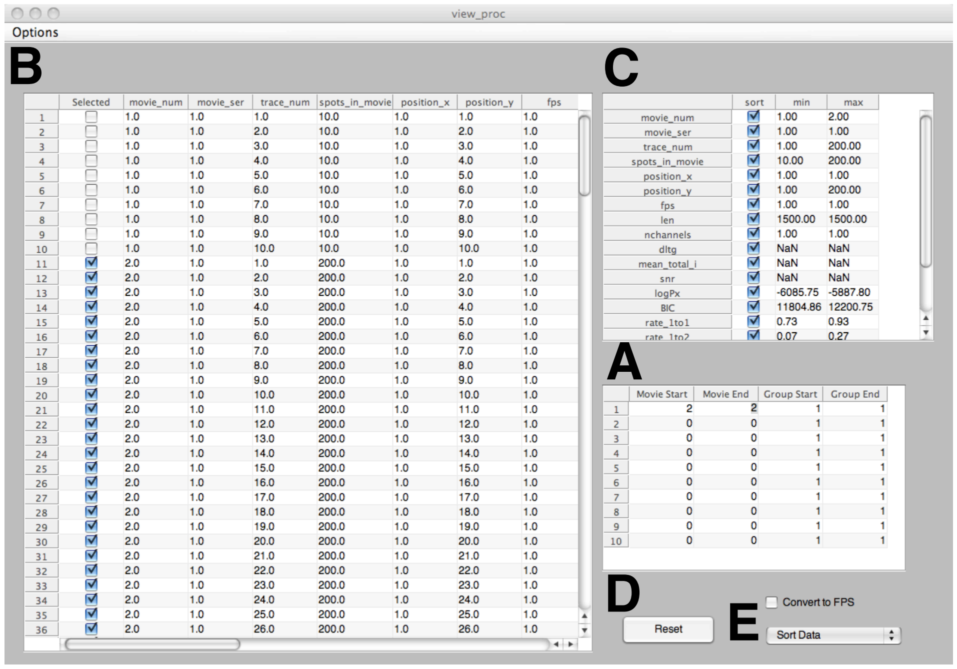
Figure 1.9: Once a .proc file is selected a second can be loaded if desired. The user then needs to decide if they want to do model selection with their loaded data.

1.4 Sort Processed Data

To aid in data analysis SMART has the ability to rapidly select subsets of imported data to be analyzed. This feature aids in finding subgroups of molecules within a particular type of experiment and allows for the rapid analysis of multiple experimental types; as this feature eliminates potential presorting steps prior to processing the data in SMART. To accomplish this SMART allows the user to group molecules based on two criteria, indices of a particular experiment or by defining a limited range of fitted or calculated experimental observables to be selected.

Fig. 1.10 depicts a screen shot of the molecule sorting interface. Fig. 1.10B summarizes for every imported molecule unique identifiers and fitted parameters. This field can be used to visually inspect the molecules that have been selected and for manually adding or removing select molecules to be used in downstream analysis steps (selected molecules have the check box selected). Fig. 1.10C summarizes a high and a low range of select fitted parameters that a user can use to limit the range of selected molecules. Fig. 1.10C Molecules in SMART are typically indexed by two parameters a movie number and group number. This indexing is used to select groups of molecules. Inputing starting and ending points will inclusively select all molecules within the range. If the user wants to select a discontinuous group of molecules multiple rows in this field can be used. Fig. 1.10D allows all of the selection parameters to be reset to the default values. If nothing is changed from the default values all the inputed traces will be carried on to subsequent steps.

Once the user has selected molecules they are interested in analyzing further the pull down window shown in Fig. 1.10E is used to move on to downstream analysis steps. Selecting HMM or FRET will bring up fitted and calculated parameters as described in Section 1.5. Selecting Cluster will bring up a window for clustering the molecules as described in 1.7. Checking the box “Convert to FPS” will convert all transition rates from discrete time to continuous using the FPS value displayed in 1.10C.



	Selected	movie_num	movie_ser	trace_num	spots_in_movie	position_x	position_y	fps
1	☐	1.0	1.0	1.0	10.0	1.0	1.0	1.0
2	☐	1.0	1.0	2.0	10.0	1.0	2.0	1.0
3	☐	1.0	1.0	3.0	10.0	1.0	3.0	1.0
4	☐	1.0	1.0	4.0	10.0	1.0	4.0	1.0
5	☐	1.0	1.0	5.0	10.0	1.0	5.0	1.0
6	☐	1.0	1.0	6.0	10.0	1.0	6.0	1.0
7	☐	1.0	1.0	7.0	10.0	1.0	7.0	1.0
8	☐	1.0	1.0	8.0	10.0	1.0	8.0	1.0
9	☐	1.0	1.0	9.0	10.0	1.0	9.0	1.0
10	☐	1.0	1.0	10.0	10.0	1.0	10.0	1.0
11	☑	2.0	1.0	1.0	200.0	1.0	1.0	1.0
12	☑	2.0	1.0	2.0	200.0	1.0	2.0	1.0
13	☑	2.0	1.0	3.0	200.0	1.0	3.0	1.0
14	☑	2.0	1.0	4.0	200.0	1.0	4.0	1.0
15	☑	2.0	1.0	5.0	200.0	1.0	5.0	1.0
16	☑	2.0	1.0	6.0	200.0	1.0	6.0	1.0
17	☑	2.0	1.0	7.0	200.0	1.0	7.0	1.0
18	☑	2.0	1.0	8.0	200.0	1.0	8.0	1.0
19	☑	2.0	1.0	9.0	200.0	1.0	9.0	1.0
20	☑	2.0	1.0	10.0	200.0	1.0	10.0	1.0
21	☑	2.0	1.0	11.0	200.0	1.0	11.0	1.0
22	☑	2.0	1.0	12.0	200.0	1.0	12.0	1.0
23	☑	2.0	1.0	13.0	200.0	1.0	13.0	1.0
24	☑	2.0	1.0	14.0	200.0	1.0	14.0	1.0
25	☑	2.0	1.0	15.0	200.0	1.0	15.0	1.0
26	☑	2.0	1.0	16.0	200.0	1.0	16.0	1.0
27	☑	2.0	1.0	17.0	200.0	1.0	17.0	1.0
28	☑	2.0	1.0	18.0	200.0	1.0	18.0	1.0
29	☑	2.0	1.0	19.0	200.0	1.0	19.0	1.0
30	☑	2.0	1.0	20.0	200.0	1.0	20.0	1.0
31	☑	2.0	1.0	21.0	200.0	1.0	21.0	1.0
32	☑	2.0	1.0	22.0	200.0	1.0	22.0	1.0
33	☑	2.0	1.0	23.0	200.0	1.0	23.0	1.0
34	☑	2.0	1.0	24.0	200.0	1.0	24.0	1.0
35	☑	2.0	1.0	25.0	200.0	1.0	25.0	1.0
36	☑	2.0	1.0	26.0	200.0	1.0	26.0	1.0

	sort	min	max
movie_num	☑	1.00	2.00
movie_ser	☑	1.00	1.00
trace_num	☑	1.00	200.00
spots_in_movie	☑	10.00	200.00
position_x	☑	1.00	1.00
position_y	☑	1.00	200.00
fps	☑	1.00	1.00
len	☑	1500.00	1500.00
nchannels	☑	1.00	1.00
dltg	☑	NaN	NaN
mean_total_i	☑	NaN	NaN
snr	☑	NaN	NaN
logPx	☑	-6085.75	-5887.80
BIC	☑	11804.86	12200.75
rate_1to1	☑	0.73	0.93
rate_1to2	☑	0.07	0.27

	Movie Start	Movie End	Group Start	Group End
1	2	2	1	1
2	0	0	1	1
3	0	0	1	1
4	0	0	1	1
5	0	0	1	1
6	0	0	1	1
7	0	0	1	1
8	0	0	1	1
9	0	0	1	1
10	0	0	1	1

Figure 1.10: view_proc window for selecting molecules based on fitted parameters and/or experiment type. **(A)** Summary of all molecules loaded in to SMART. **(B)** Field for selecting molecules based on fitted parameters. **(C)** Field for selecting molecules based on experiment type. **(D)** Reset button to restore all visualized fields to the default values. **(E)** Menu for switching to interfaces that allow visualization and clustering of molecules.

1.5 View Processed Data

Once traces have been fitted to HMMs an experimentalist is interested in being able to quickly visualized the fits of an individual molecule and of many molecules. When HMM is selected from the pulldown menu shown in Fig. 1.10E the view_proc window will appear as shown in Fig. 1.11. This window allows all the fitted parameters determined in the HMM fits to be displayed for an individual molecule and for plots to be made displaying all the molecules that were selected prior to choosing this window.

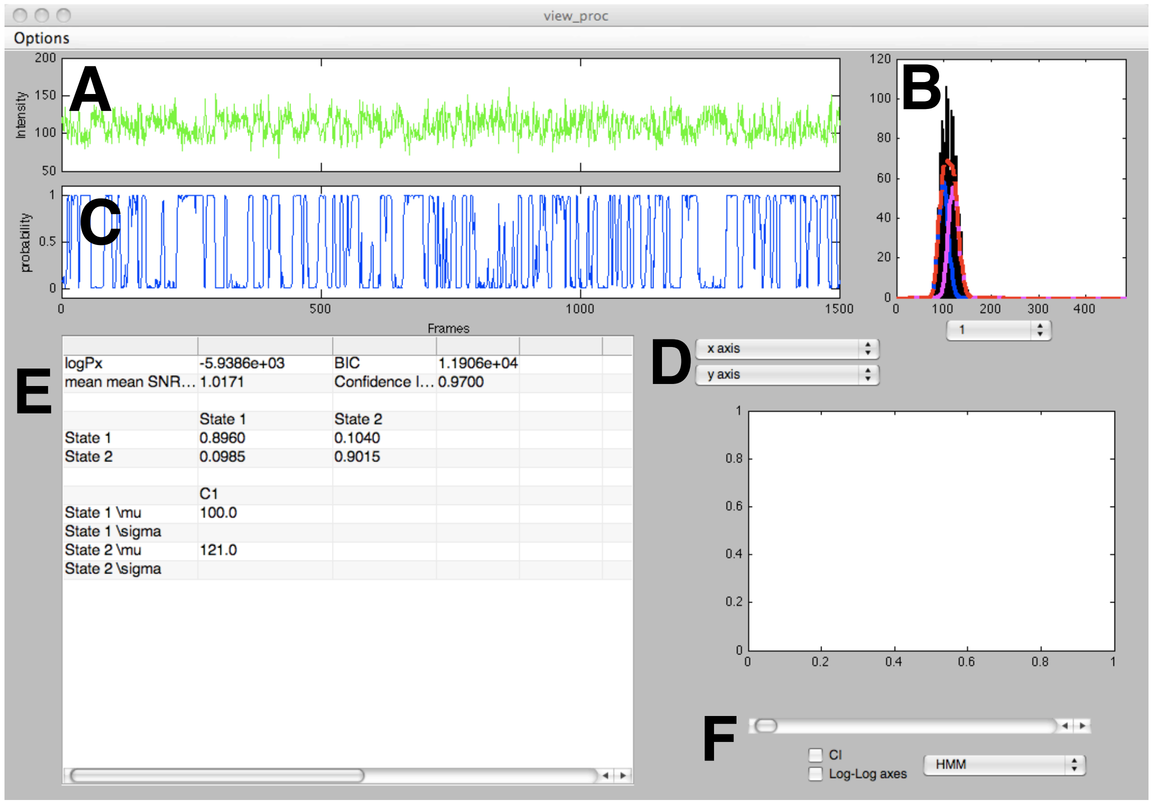
Fig. 1.11A displays the raw data for a single trace. A two channel FRET trace is shown and this axes can display up to four independent channels. Fig. 1.11B displays the inferred state probabilities for each if the states. Inferred state probabilities are bounded between 0 and 1 and indicate for every point in time of the trace the probability that state is occupied. Fig. 1.11C displays the cumulative intensity histograms for any one of the channels in the experiment (using the pull-down menu below the plot, see Fig. 1.13A the displayed channel can be changed). Fig. 1.11D allows for any of the parameters displayed in Fig. 1.11E to be plotted. Fig. 1.12 provides a detailed description of the different plots that can be generated in this window. Any plot displayed in this window can be exported and save for future use.

To export the plot displayed in Fig. 1.11D from the "Options" menu select the "Export Figure". This will create a standard MATLAB figure that can be edited and saved like any figure in MATLAB. Fig. 1.13E displays parameters that were determined during the HMM fitting for a single trace. The notation for how the fitted emission and kinetic parameters are displayed is gone over in Fig. 1.6 . The region adjacent to 1.11F has four functions. The horizontal scroll bar is used to move between all the different traces. The CI check box can be selected to display the confidence intervals as shown in Fig. 1.13B. The Log-Log axes check box can be selected to have the displayed plot have Log-Log axes. Finally the pulldown menu can be used to go between the sorting, FRET or clustering interfaces.

To enable zooming into particular parts of the data plots press z on the key board. This will change the cursor to allow zooming. After zooming is complete press z again.

Sometimes in SMART it is useful to save subsets of a larger data set. Once a subset of data has been selected using the tools just described in Fig. 1.10 the data can be save by selecting the "Save Sorted" or "Export Table" under the "Options" menu in the HMM visualization interface. If "Save Sorted" is selected in the Command Window of the MATLAB work environment a prompt will appear with a suggested name (press Enter to accept) or type in a desired name and then press Enter. A new .proc file will then be saved in the current directory. If "Export Table" is selected a file title current_data.mat will be save in the current directory. This will save most of the data that was selected in a very simple format that can be easily navigated so a user can access the date for plotting in a program like Excel if they desire.

Additional data summaries can be view by selecting FRET from the pulldown menu shown in Fig 1.11F and 1.10E. Selecting FRET will bring up the interface depicted in Fig. 1.14. This interface primarily serves to display FRET data and parameters that were derived from fitting thresholded FRET traces as described in Section 1.2. The exception to this is the top part of the table shown in Fig. 1.14D. This part of the table is used to display a number of parameters that are convenient for tracking or summarizing molecules that are not determined by HMM fits such as the file name, trace number and molecule position.



logPx	-5.9386e+03	BIC	1.1906e+04
mean mean SNR...	1.0171	Confidence I...	0.9700
	State 1	State 2	
State 1	0.8960	0.1040	
State 2	0.0985	0.9015	
	C1		
State 1 \mu	100.0		
State 1 \sigma			
State 2 \mu	121.0		
State 2 \sigma			

Figure 1.11: view_proc window for HMM. **(A)** Raw experimental data. **(B)** Cumulative intensity histograms overlaid with fitted state emissions. **(C)** Fitted HMM state probabilities. **(D)** Regions for plotting many inferred parameters, see Fig. 1.12. **(E)** Table for displaying inferred parameters. **(F)** Slider bar for changing what molecules are displayed and check box that is used to indicate if confidence intervals should be displayed or if plotting axes should have a Log-Log scale.

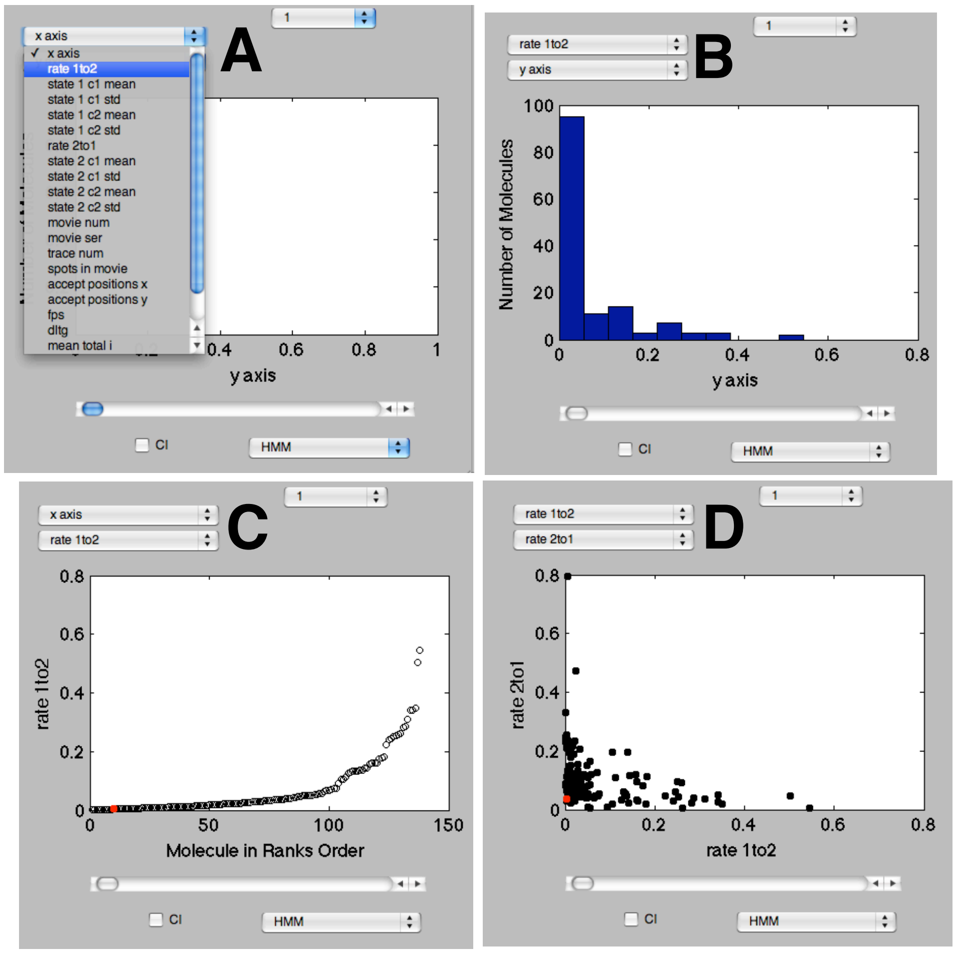


Figure 1.12: Multiple types of summary plots can be generated in the view_proc window. **(A)** As depicted the drop down windows can be used to select fitted parameters that will be plotted in the axes in the view_proc window. **(B)** Using only the x axis pull down will produce a histogram of the selected parameters. **(C)** Using only the y axis pull down will plot the selected parameter in rank order from the lowest to highest value. **(D)** Using both variables will produce a scatter plot on the specified axes. The red point in **(C)** and **(D)** indicates the molecule that is being displayed in other fields. Clicking on other data points will cause the selected data point to update.

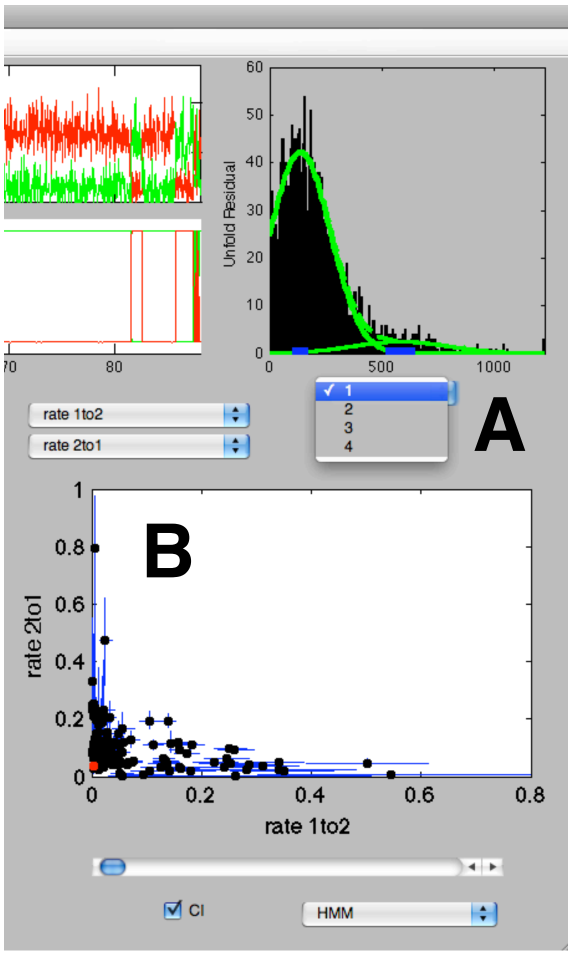
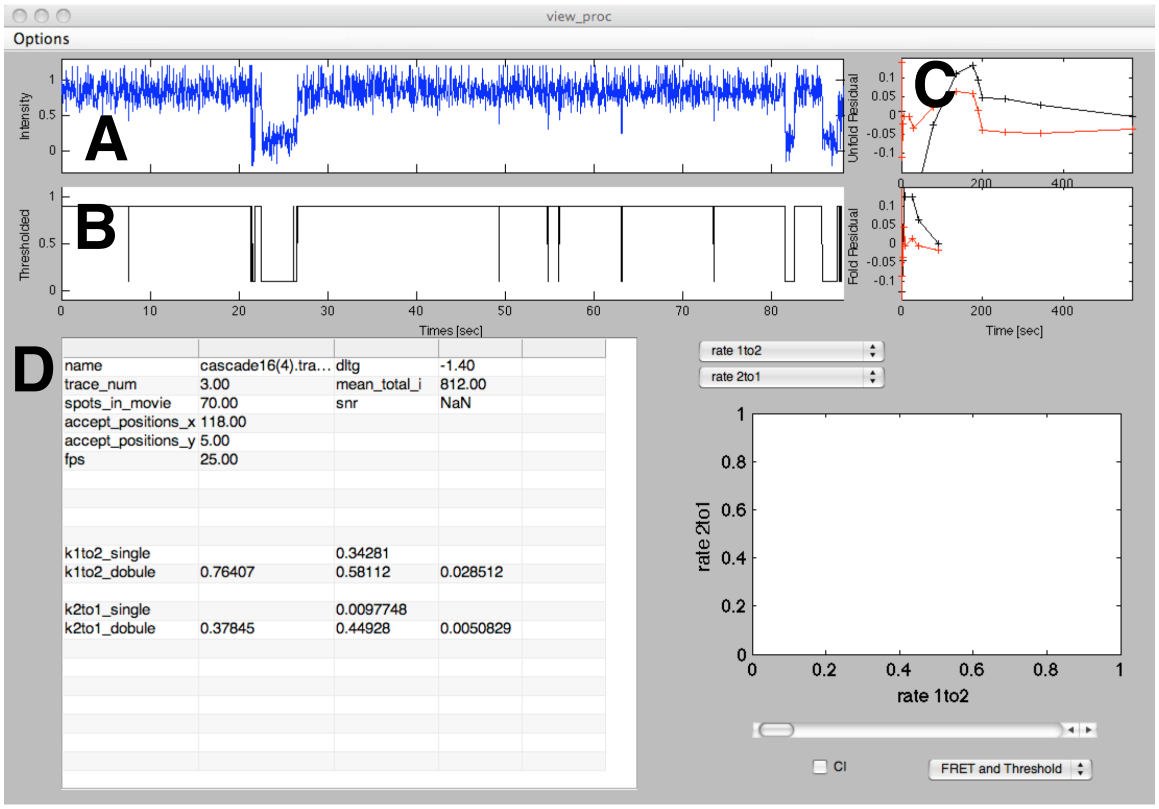


Figure 1.13: Depiction of how confidence intervals and fitted emissions distributions are displayed in the view_proc window. **(A)** The pull down menu below the emission histogram allows the displayed channel to be updated.**(B)** Scatter plot as in Fig. 1.12D but in this instance the CI check box is checked indicating that the confidence intervals should be drawn for the inferred parameters. The confidence intervals are depicted by the blue bars.



name	cascade16(4).tra…	dltg	-1.40	
trace_num	3.00	mean_total_i	812.00	
spots_in_movie	70.00	snr	NaN	
accept_positions_x	118.00			
accept_positions_y	5.00			
fps	25.00			
k1to2_single		0.34281		
k1to2_dobule	0.76407	0.58112	0.028512	
k2to1_single		0.0097748		
k2to1_dobule	0.37845	0.44928	0.0050829	

Figure 1.14: view_proc window for FRET and other summary statistics. **(A)** Window showing the current FRET trace. **(B)** Window showing transitions determined by thresholding. **(C)** Residuals for the fits to cumulative dwell time distributions for folding and unfolding rates. Single exponential fits are in black and double exponential fits are in red. **(D)** Table showing kinetic parameters determined by thresholding and other summary statistics.

1.6 Model Selection

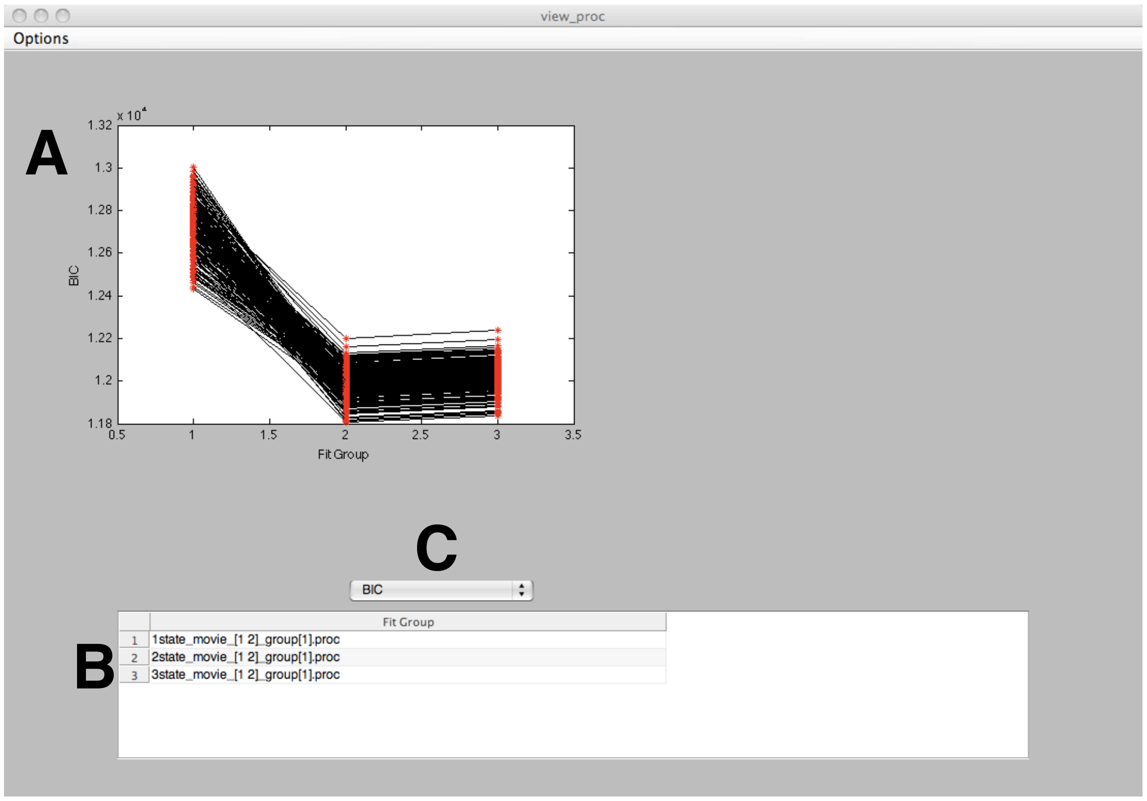
An important component of analyzing single molecule data is determining the appropriate model to fit to the data. SMART provides a tool which allows the BIC and log p (data | model) (logPx) statistics determined for fits of a given trace to be compared.

To begin this process you need to fit a raw data set to different HMMs as described in Section 1.2. In the SMART Example Data folder the movie_[1 2]_group[1].traces data was fit to 1, 2 and 3 state HMMs (after each fit the name of the resulting .proc file was manually appended to denote what model was fit to the data e.g. 1state_movie_[1 2]_group[1].proc. To compare the model fits for different traces the .proc files need to be opened in the view_proc interface. This is done in an analogous fashion as described 1.3 but the answers to the two popups shown in Fig. 1.9a and Fig. 1.9b will be opposite.

The .proc files should be loaded in to the SMART interface in order of increasing model complexity. In this example this means loading the 1 state fit first and the 3 state fit last. After loading the first .proc file select Yes from the popup shown in Fig. 1.9a, this will allow additional files to be imported. Once you have loaded all of the different model fits select No from the popup shown in Fig. 1.9a. This will bring up the second popup shown in Fig. 1.9b, which asks if you want to do model selection with the data, the answer to this will be Yes.

Once the data has been loaded this will bring up the interface shown in Fig. 1.15. This interface is relatively simple compared to the others in SMART and serves to plot 4 different parameters in the axes shown in Fig. 1.15A. In these axes the BIC, LogPx or these values normalized for trace length can be plotted. Both the BIC and logPx scale with trace length so to compare traces of different length on the same figure it is convenient to plot normalized values. The ordinate values i.e. 1, 2, and 3 correspond to the order in which the data was loaded and is shown in Fig. 1.15B. The pulldown menu in 1.15C allows the user to change what is being plotted in this interface. The data shown in Fig. 1.15 can be exported using the“Export Table” function described in section 1.5. When this is used the saved file contains all the data needed to easily recreate the figure displayed in Fig. 1.15A using a different plotting software.

Note: Zooming does not work perfectly in this interface, so for the time it is best to export the figure and then use the zooming function on the exported figure.



	Fit Group
1	1state_movie_[1 2]_group[1].proc
2	2state_movie_[1 2]_group[1].proc
3	3state_movie_[1 2]_group[1].proc

Figure 1.15: view_proc window model selection interface. **(A)** Figure axes for plotting models selection criteria. **(B)** List of the data files that were loaded and the order in which they are being ploted in (A). **(C)** Pull down menu to switch what model selection criteria is being plotted. Options include the BIC, logPx or those values normalized for trace length.

1.7 Clustering

SMART has the ability to cluster molecules based on up to three jointly inferred parameters. To cluster molecules two separate calculations must be completed. First, when the HMM fitting is completed the covarience between inferred parameters must be determined. This is done by specifying a single parameter or a group of up to three parameters with the Param 1, 2 or 3 fields as described in Section 1.2. If this has not been done the clustering algorithm that is run in this section can not be used.

Assuming covariance parameters were calculated and prior to clustering your data can be loaded, selected and visualized as described in Section 1.3, 1.4 and 1.5. To begin clustering use the pulldown menu depicted in Fig 1.10E to select Cluster. This will bring up the default clustering window shown in Fig. 1.16. This interface allows for the user to choose which variable groups should be clustered, specify the maximum number of clusters to be fit to the data, cluster the data and visualize the clustering output.

Once the default clustering window has been brought up use the top pull down menu shown next to Fig. 1.16A to select the parameters that should be clustered. Next use the middle pull down menu to select the maximum number of clusters to be fit to the data. For example if 4 clusters is picked the data will be fit with 1, 2, 3 and 4 clusters. After these values have been selected press the Cluster button. This will run the clustering script. The length of time it takes to run the script depends on the number of molecules and number of clusters being fit. For most situations we have encountered the clustering is completed in 1 to 2 minutes.

When the clustering is complete a file titled "cluster_outputs.mat" will be saved in the current directory. This contains the output from the clustering result and can be used to generate cluster plots using MATLAB directly or a different plotting software. the clustering interface will also change to the view shown in Fig. 1.17.

Once the clustering window has changed you can use the pull down menu below the plotting axes to select a range of plots that summarize the clustering outputs in various ways. Summary plots include Bar plots of cluster size (e.x. Fig. 1.17A), scatter plots of cluster fits (e.x. Fig. 1.17B) and plot of the BIC and Log p (date | model) (logPx) of the fit. These summary plots can be used to assess how many clusters well fit the data given the uncertainties in the inferred parameters.

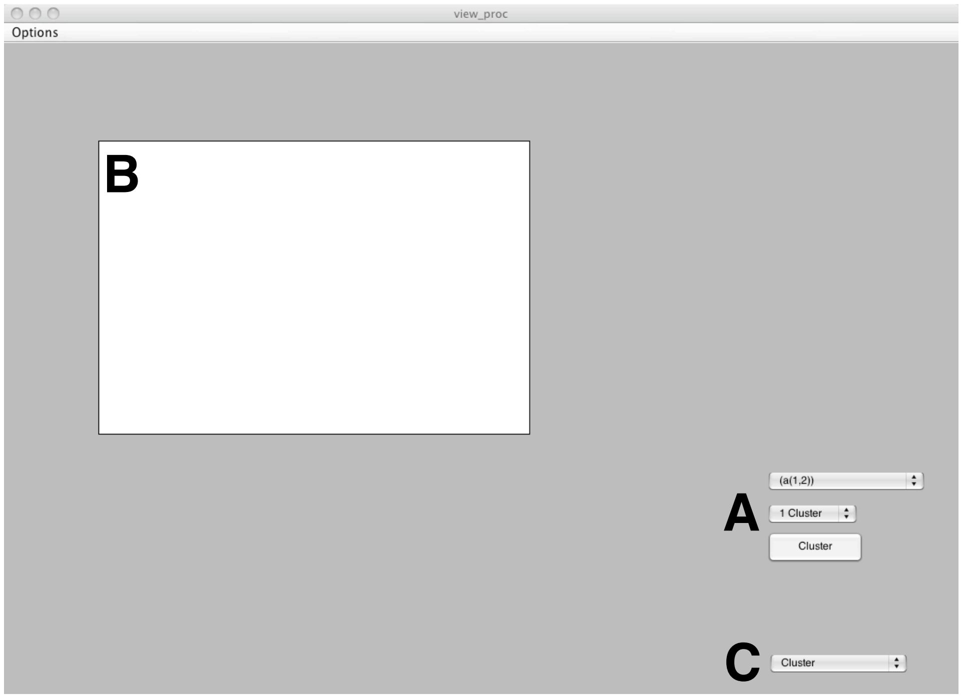


Figure 1.16: The default view_proc window prior to clustering molecules **(A)** To cluster molecules select the parameters that can be clustered (i.e. had covariance matrices calculated as specified in Section 1.2) then select the maximum number of clusters you want to fit to the data and then press Cluster. **(B)** Axes used for displaying different cluster plots. **(C)** This pull down menu still allows for navigation to other functions of the view_proc window.

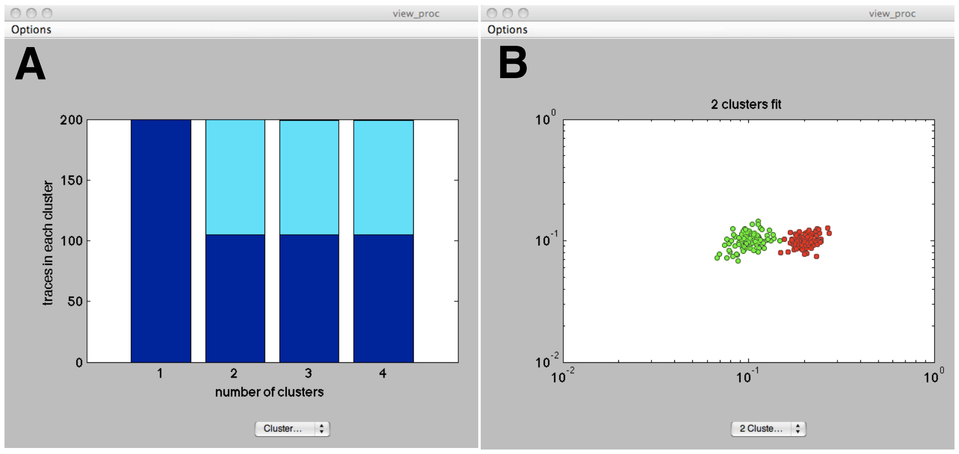


Figure 1.17: Examples of cluster plots **(A)** Stacked bar plot displaying how many molecules reside in each cluster for the 1, 2, 3 and 4 cluster fits (third and 4th clusters are either 1 or 0 making them difficult or impossible to see in this example) **(A)** Scatter plot of rate constants determined for a two state fit. Molecules are color coded according to which cluster they reside in for this two cluster fit.

Chapter 2

Using SMART Functions from the Command Line

The SMART graphical interface can access all key functionality of the HMM algorithms. However, advanced users might desire to access the HMM function directly in their own MATLAB scripts. This section provides additional descriptions on the implementation of the HMM algorithms in SMART and the syntax for accessing them directly.

2.1 Fitting a trace from the command line

This section walks through the demofit.m example script provided in the SMART HMM folder. This script demonstrates simulation of a single trace, setting up fit parameters, performing the fit, and viewing the results. To run the script, go to the HMM folder and enter

```
>> demofit
```

into the command window of MATLAB.

2.2 Generating a simulated trace

To simulate a trace we must specify the length N of the trace, the transition matrix $(A)_{ij} = \mathrm{P}(x_{t+1} = j|x_t = i)$ and any parameters necessary to sample from the emissions dstributions. This is done in lines 1-15 of demofit.m.

The signal parameters are specified by a single MATLAB cell array of vectors E indexed by $E\{n,c\}(p)$, where n is the index of the hidden state, c is the index of the channel, and p is the index of the emissions parameter. $p=1$ corresponds to the mean for both Poisson and Gaussian channels. $p=2$ corresponds to the standard deviation for a Gaussian channel (and is disregarded for Poisson channels in the fits). We generate a sample trace by entering (demofit.m line 21)

```
>> [trueStates,y] = HMMNoisy3_(A,E,{'gauss','gauss'},N);
```

where trueStates is the N by 1 sequence of true hidden states and y is the N by number of channels matrix of observations.

2.3 Setting up fit parameters

Next we must set up the fitting parameters. This is done in the demofit_initialize_parameters.m subscript of demofit.m (invoked at line 24 of demofit.m). We must create a MATLAB structure (named params in this example) with the fields in table 2.1 and then enter (demofit.m line 36)

```
>> output = TrainPostDec_(params);
```

2.3.1 Specifying levels with distinct emissions

The discStates field of the fitting parameters structure specifies which states have the same emissions distribution as other states, corresponding to states that may have the same FRET level, but may have different lifetimes. This variable is a vector of integers that must add up to the total number of hidden states. If all states have possibly distinct emissions distributions, set discStates = $[1\dots1]$ or discStates = [] (empty).

To specify that k states have identical emissions distributions, enter k as a component of discStates. States are numbered in order of increasing emissions mean. For example, if our model has 5 states, setting discStates = [2 1 2] means the first two states (lowest mean) have an identical emissions distribution, the third state has a unique emissions distribution, and the last two states (highest mean) have an identical emissions distribution. Setting discStates = [1 2 2] means the first (lowest

parameters field	description	suggested default value
data	N by C matrix of observations where N is the number of time points and C is the number of channels corresponding to fitChannelType below	
pi	N by 1 vector of true hidden states (for use in plotting outputs). Leave empty for real data	[] (empty)
nStates	Integer number of HMM states to fit	
nChannels	Integer number of channels in trace (number of columns in data)	
fitChannelType	cell array that lists channel types, e.g. {'gauss','gauss','poisson'} means the first two channels are normally-distributed and the third is Poisson-distributed	
discStates	List of integers that adds up to nStates. Specifies which states have distinct emissions model (see 2.3.1)	[] (empty)
noHops	List of forbidden transitions, e.g. [1 2 ; 3 4 ; 2 1] means $a_{12} = a_{34} = a_{21} = 0$ (see 2.3.2 for further discussion)	[] (empty)
sameHops	List of transitions same as othe rtransitions, e.g. [1 2 3 4 ; 2 1 1 3] means $a_{12} = a_{34}$ and $a_{21} = a_{13}$	[] (empty)
imposeDetailedBalance	true or false to guarantee that output satisfies detailed balance or not	false
tryPerms	true or false to try all possible permutations of discStates distinct states vector (see 2.3.1)	false
Ainitial	set to 'auto' or a K by K initial guess for transition matrix in doing fit. Set to desired value if wish to only train emissions distribution and set trainA = false below	'auto'

Table 2.1: Fields in params structure for fitting a single trace (continued on next page)

parameters field	description	suggested default value
Einitial	set to 'auto' or an initial guess for emissions distribution parameters in doing fit (see [ABOVE] for format of emissions structures). Set to desired value if wish to only train transition matrix and set trainE = false below	'auto'
trainA	true or false if wish to do maximum likelihood estimation of transition matrix or not. If set to false, user must provide a transition matrix in the Ainitial field; otherwise the program will choose some initial guess for the transition matrix and never update it	true
trainE	true or false if wish to do maximum likelihood estimation of emissions distribution parameters or not. If set to false, user must provide an emissions distribution cell array in the Einitial field; otherwise the program will choose some initial guess for the emissions distribution cell array and never update it	true
maxIterEM	Integer > 0. Maximum number of iterations over which to perform maximum likelihood estimation of model parameters	200
threshEMToConverge	minimum exp log-likelihood difference between consecutive iterations minus one before declare opimization to have converged	10^{-3}
SNRwarnthresh	threshold value of inferred SNR before issuing warning to user	1
returnFit	true or false to return or not return fit of probability to be in each state at each time. Setthing this to 'true' appends a N by K structure to your output, which dominates the size in memory of your output for sufficiently long traces	false

Table 2.2: Fields in params structure for fitting a single trace (continued on next page)

parameters field	description	suggested default value
paramsErrorToBound-Auto	String that specifies for which parameters to compute likelihood ratio confidence bounds in an automatically chosen region (chosen to include threshold value of likelihood ratio, specified in auto_confInt). e.g. 'a(1,2),a(2,1),a(3,2)' means find 1-D confidence bounds for a_{12}, a_{21}, and a_{32}, '(a(1,2),a(2,1))' means find a 2-D joint confidence bound for a_{12} and a_{21}, and 'e(3,2,1)' means find a 1-D confidence bound for hidden state 3, channel 2, parameter 1 (the mean for a Poisson- or Gaussian-distributed channel; hidden states are sorted in order of increasing mean)	" (empty string)
auto_confInt	fraction between 0 and 1. $1 -$ auto_confInt gives the minimum likelihood ratio to include in computing likelihood ratio confidence bounds for parameters specified in paramsErrorToBoundAuto	
auto_boundsMeshSize	Integer > 0 number of positions at which to sample data likelihood in automatically chosen region around MLE. For joint (2-D) confidence bounds, computation time scales as the square of this number	10
auto_MeshSpacing	'square' or 'auto'. Setting this to 'auto' ensures an equal number (half of auto_boundsMeshSize) of data likelihood samples below and above the MLE estimator for a model parameter. Setting this to 'square' spaces out the auto_boundsMeshSize samples equally in the data likelihood sampling range and may not be desirable if the data likelihood is skewed near the MLE	'square'

Table 2.3: Fields in params structure for fitting a single trace (continued from previous page)

parameters field	description	suggested default value
paramsErrorToBound-Manual	String that specifies for which parameters to compute likelihood ratio confidence bounds in a manually chosen region (specified in ManualBoundRegions below). Same string format as for paramsErrorToBoundAuto	” (empty string)
ManualBoundRegions	Cell array of row vectors specifying values of model parameters at which to evaluate data likelihood. Must contain one entry per set of parameters specified in paramsErrorTo-BoundManual (above), e.g. if paramsError-ToBoundManual = ’(a(1,2),a(2,1)),e(3,2,1)’ we could set ManualBoundRegions{1} = $\{0.01 : 0.01 : 0.2, 0.01 : 0.01 : 0.2\}$, ManualBoundRegions{2} = 50 : 10 : 150 to sample values of a_{12} and a_{21} in the 2-D rectangular region specified by the first entry, and sample values of $E\{3,2\}(1)$ (the mean in state 3, channel 2) in the 1-D region from 50 to 150 in increments of 10. Limited to 1-D and 2-D regions	[] (empty)
plotPFits	’true’ or ’false’ to plot inferred hidden states at each iteration of optimization of model parameters. Can be handy for debugging or for choosing initial parameter values	false
showProgressBar	’true’ or ’false’ to show progress bar during optimization of model parameters. Might not work on a mac	false
pause	’true’ or ’false’ to pause or not at each iteration of optimization of model parameters. Only pauses if plotPFits is set to ’true’. Pauses after plotting inferred hidden states	false

Table 2.4: Fields in params structure for fitting a single trace (continued from previous page)

mean) state has a unique emissions distribution, the next two states (highest mean) have identical emissions distributions, and the last two states have identical emissions distributions.

To try fitting all permutations of the discStates vector (e.g. try to fit both [1 2] and [2 1] for a three-state model that has only two distinct emissions distributions associated with its three states) and return the best fit, set the parameters field tryPerms = true.

The number of free parameters in the model is reduced to calculate the BIC if not all states have unique emissions distributions.

2.3.2 Specifying forbidden transitions

The noHops field of the fitting parameters structure specifies forbidden transitions. For example, setting noHops = $[1\ 2; 3\ 4]$ means $a_{12} = a_{34} = 0$. Since states are numbered in order of increasing mean, in this example we would forbid transitions from the state with the lowest mean (state 1) to the state with the second-lowest mean (state 2).

If some states have identical emissions distributions as other states (see 2.3.1) and tryPerms is set to true, the noHops field will be permuted to respect the ordering of non-distinct states.

For example, if we are fitting a three-state model with only two distinct FRET levels, possibly two different lifetimes corresponding to one of these FRET levels, and no interconversion between the two hidden states that have the same FRET level, we should set discStates = [1 2], tryPerms = true, and noHops = $[2\ 3; 3\ 2]$. Since tryPerms is set to true, we will attempt to fit discStates = [2 1] as well, at which point noHops will be set to $[1\ 2; 2\ 1]$, since now states 1 and 2 are have the same (lower) emissions mean. If tryPerms is set to false, we will fit only the model in which the higher emissions mean corresponds to two hidden states, without trying the model in which the lower emissions mean corresponds to two hidden states.

Note that setting imposeDetailedBalance to true may not be compatible with noHops. For example, in fitting a three-state model, setting noHops = [1 3] ($a_{13} = 0$, but not $a_{31} = 0$), will result in an inferred model that in general has nonzero a_{13} and thus does not respect the noHops setting, but satisfies detailed balance.

The number of free parameters in the model is reduced by the number of forbidden transitions to calculate the BIC.

2.4 Working with the fit output

The output structure produced by calling output = TrainPostDec_(params); contains the fields in table 2.5.

2.4.1 Displaying likelihood ratio confidence bounds

To display figures showing the 2-D or 1-D likelihood ratio confidence bounds in the output enter (demofit.m line 64)

```
>> ShowErrorBounds(output, 'auto');
>> ShowErrorBounds(output, 'manual');
```

An output produced by demofit.m is shown in figures 2.2 and 2.1.

2.4.2 Converting probabilities to transition rates

As discussed in the methods, converting inferred transition probabilities (unitless) to rates (Hz) is approximately multiplying by the sampling rate so long as transition probabilities are small. To do the exact conversion enter (demofit.m line 68)

```
>> A_cont = DiscToContA(output.A, f_sample);
```

where f_sample is the sampling frequency with the same units as the transition rates (so if f_sample is in units of Hz, so are the transition rates). A_cont$(i, j) = k_{ij}$, the transition rate from state i to state j.

2.5 Clustering traces from the command line

This section walks through the democluster.m example script provided in the SMART HMM folder. This script demonstrates simulation of a traces from a mixture of

output field	description
A	inferred transition probability matrix (A(i,j)= a_{ij})
E	inferred emissions distributions (format given in section 2.2)
fitChannelType	cell array that lists channel types (emissions distributions). Same value as for fit parameters
auto_confInt	fraction between 0 and 1. Same value as for fit parameters
logPx	log likelihood of data given the inferred model
BIC	Bayesian Information Criterion (BIC) value given the inferred model
freeParams	number of free parameters in inferred model used in computing BIC
SNRMat	Matrix of signal to noise ratio (SNR) values between every pair of states. SNRMat(i,j) gives the SNR between the emissions distributions in state i and j
discStates	List of integers that adds up to nStates. Specifies which states have distinct emissions model (see 2.3.1). This may be reordered from its value in fit parameters if tryPerms is set to true in fit parameters
noHops	List of forbidden transitions (see 2.3.2). This may be reordered from its value in fit parameters if tryPerms is set to true in fit parameters
postFit	K by N matrix such that postFit(i,j) = P(hidden state at time $j=i\|$ all observations)
flags	contains error messages if crash occurred in fitting model parameters or SNR between a pair of states is below the threshold SNRwarnthresh in fit parameters
imposeDetailedBalance	true or false to guarantee that output satisfies detailed balance or not. Same value as for fit parameters
errorBoundsAuto	contains likelihood ratio confidence bounds for parameters specified in paramsErrorToBoundAuto in fit parameters. Use ShowErrorBounds.m script to display (see 2.4.1)
errorBoundsManual	contains likelihood ratio confidence bounds for parameters specified in paramsErrorToBoundManual in fit parameters. Use ShowErrorBounds.m script to display (see 2.4.1)

Table 2.5: Fields in output structure determined by fitting a single trace

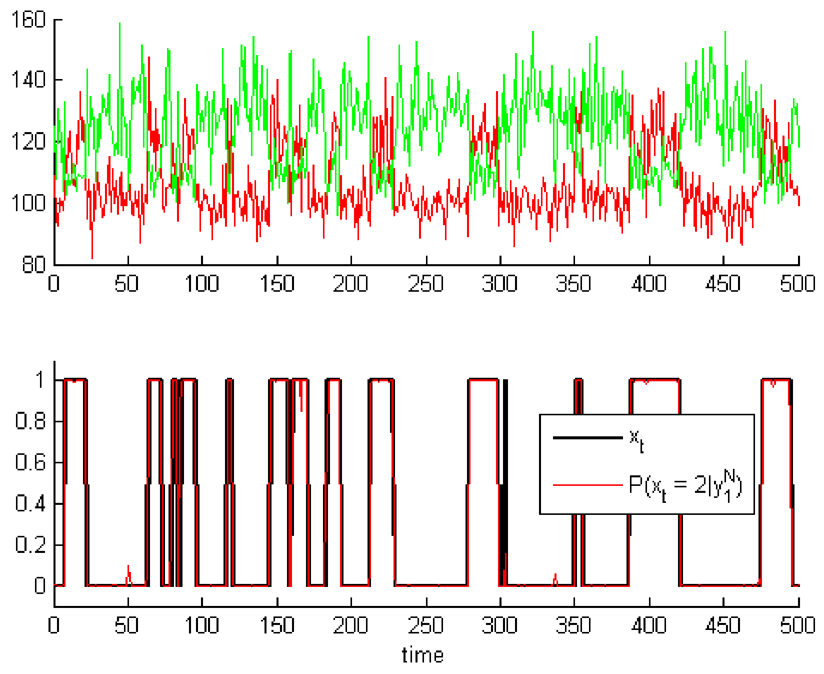


Figure 2.1: HMM fit output. The red and green traces (top subplot) show the noisy observations. The black line (bottom subplot) shows the true state and the red line shows the inferred probability of being in state 2 given all of the observations and the inferred model.

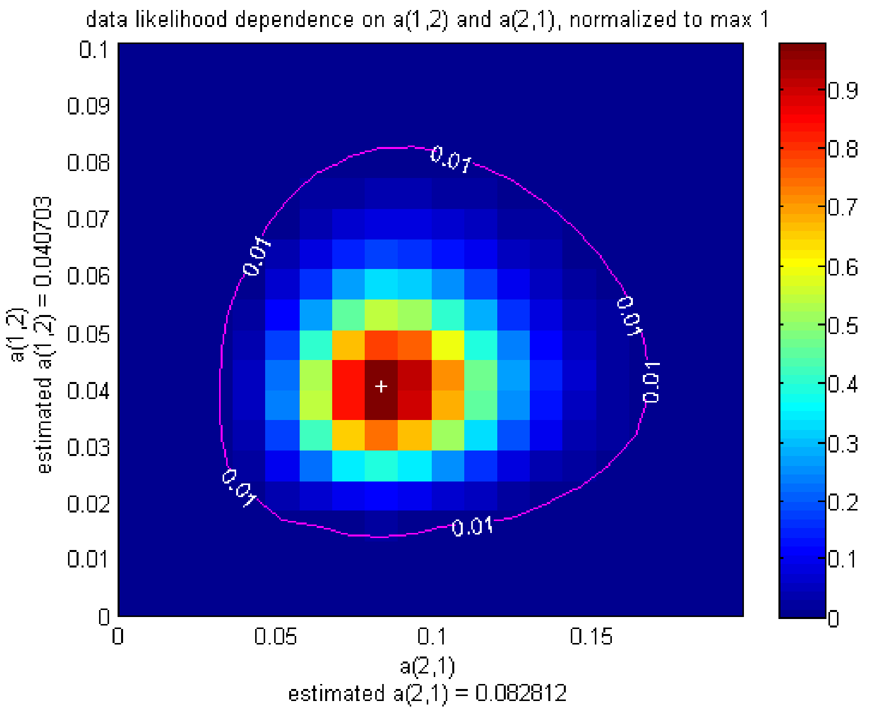


Figure 2.2: HMM fit output. Colored blocks show data likelihood evaluated at the corresponding value of the transition probabilities a_{21} and a_{12}. The purple border indicates the points at which the data likelihood is 0.01 of its maximum value. The white cross indicates the inferred a_{21} and a_{12} that locally maximizes the data likelihood.

models, setting up fit parameters, performing the fits, clustering the outputs, and viewing the results. To run the script, go to the HMM folder and enter

```
>> democluster
```

into the command window of MATLAB.

2.5.1 Clustering setup

To simulate fits from a mixture of models, we simulate 30 traces, each equally likely to arise from one of three two-state models: $a_{12} = a_{21} = 0.1$, $a_{12} = 0.1, a_{21} = 0.2$, $a_{12} = 0.2, a_{21} = 0.1$. All three models have Poisson-distributed channels with means 100 and 200, corresponding to SNR of about 9 (democluster.m lines 4-8). clustdemo.m lines 12-58 set up fitting parameters (see section 2.3). clustdemo.m lines 63-81 simulate traces from this mixture of models and fit these traces to a two-state HMM as specified in the fit parameters structure params. Trace lengths are sampled uniformly from 500 to 1000.

The command line clustering scripts require a cell array outputs, such that outputs$\{i\}$ = output of HMM fit to the i-th trace, and a cell array traces, such that traces$\{i\}$ = i-th trace (N by C matrix, where N is the number of time points and C is the number of channels.

2.5.2 Approximating data likelihood near the MLE

As discussed in Methods, in order to avoid computatins that scale like the total length of all traces, we approximate the data likelihood by a normal distribution centered on the MLE and with a yet-to-be-determined covariance matrix.

We jointly cluster up to 3 of the model's parameters (rates and emissions distribution parameters), so we must estimate this covariance for subsets of up to 3 of the model's parameters. The sets of parameters to jointly cluster are specified by the string cov_mats_string, which has the same format as the string paramsErrorToBoundAuto (see section 2.3). For example, democluster.m line 90 sets

```
>> cov_mats_string = '(a(2,1),a(1,2)),a(1,2),(a(1,2),a(2,1),e(1,1,1))';
```

Meaning cluster parameters a_{21} and a_{12} jointly (2-D), cluster a_{12} alone (1-D), and cluster a_{12},a_{21}, and $e(1,1,1)$ jointly (3-D) (where $e(1,1,1) = E\{1,1\}(1)$ is the state 1, channel 1, parameter 1, the mean). To estimate the 3 by 3, 1 by 1, and 2 by 2 covariance matrices corresponding to these parameters for the i-th trace and i-th HMM fit output using a numerical solver in MATLAB, enter (democluster.m line 95)

```
>> outputs{i} = AppendCovMatsToHMMFitOutput(outputs{i}, traces{i}, cov_mats_string);
```

2.5.3 Fitting clusters

Now that the estimates for the covariance matrices have been appended to the outputs we can cluster the outputs. We must specify how many clusters to fit to our set of outputs in a row vector of positive integers. democluster.m sets (democluster.m line 102) numClusterList_2D= [2 3 4 5]; to fit 2 through 5 clusters. To perform the fit we enter (democluster.m line 103)

```
>>  clustFitOutputs_2D = GetClustsRatesMult_v2(...
outputs,'(a(2,1),a(1,2))',numClusterList_2D);
```

Note that the second argument to GetClustsRatesMult_v2, the string '(a(2,1),a(1,2))', must match match a string from cov_mats_string (see section 2.5.2), meaning covariance matrices for this set of outputs have already been computed. Otherwise, outputs that have not had the necessary covariance matrices appended to them are excluded from clustering (as in democluster.m lines 108-111).

The clustering fit output clustFitOutputs_2D is a cell array with one entry per number of clusters to fit. Each entry is a structure with the fields listed in table 2.6

2.5.4 Displaying clustering outputs

To display the clustering outputs enter (democluster.m line 117)

```
>>  ShowClustFitOutputs_v2(...
clustFitOutputs_2D,traces,numClusterList_2D, 0.1, false,false,true);
```

The arguments of the clustering output plotting function clustFitOutputs_2D are given in table 2.7. An output produced by democluster.m is shown in figure 2.3

output field	description
clustPos	K by d array of inferred positions of K clusters for d clustered parameters
clustFrac	K by 1 array of inferred fraction of all traces to have come from each of K clusters, e.g. clustFrac(i)=P(trace from cluster i)
clustMembershipP	N by K array of probabilities such that clustMembershipP(i,j)=P(trace i from cluster j), N is the number of traces and K is the number of clusters
minTracesInClust	K by 1 vector such that minTracesInClust(i)= number of traces assigned to cluster i. The assignment is done by choosing the single likeliest cluster for each trace to be from (maximizing clustMembershipP(i,j) over j)
logPx	log likelihood of all traces given the inferred clustering model (see Methods for definition)
logPxTracesList	N by 1 array such that logPxTracesList(i)= log likelihood of trace i given the inferred clustering model. The logPx clustering output is the sum of the entries of this vector
BIC	Bayesian Information Criterion (BIC) value given the inferred clustering model (see Methods for definition)
invalidTraces	vector of trace indices that were not included in clustering because the necessary covariance matrix was not appended to their fit outputs
numClusters	number of clusters in inferred model. This is determined by input to clustering script (GetClustsRatesMult_v2)
coords	N by d matrix of HMM fit outputs such that coords(i,j)=inferred value of parameter j for trace i, where N is the total number of traces and d is the number of parameters clustered. Handy for plotting clustering outputs
numIter	number of iterations before clustering algorithm converged
initialIndices	1 by K vector of trace numbers which were used to seed the positions of the K clusters in the clustering algorithm. These are chosen uniformly at random without replacement from the N traces 10 different times and the best clustering output (maximizes logPx above) is chosen

Table 2.6: Fields in one entry of output structure for clustering

argument number	argument description	suggested default value
1	clustering fit output (returned by GetClustsRatesMult_v2, see section 2.5.3)	
2	traces structure such that traces{i}= i-th trace. Same input as for GetClustsRatesMult_v2 (see section 2.5.3)	
3	row vector of positive integers specifying number of clusters in fit. Same input as for GetClustsRatesMult_v2 (see section 2.5.3)	
4	true or false to show trace numbers next to points in plot. Can be handy for finding outliers	false
5	true or false to plot on log log scale	false
6	true to color clustered points in color of nearest cluster, false to take mean color weighted by probability to be in each cluster	true

Table 2.7: inputs to clustering output plotting function ShowClustFitOutputs_v2

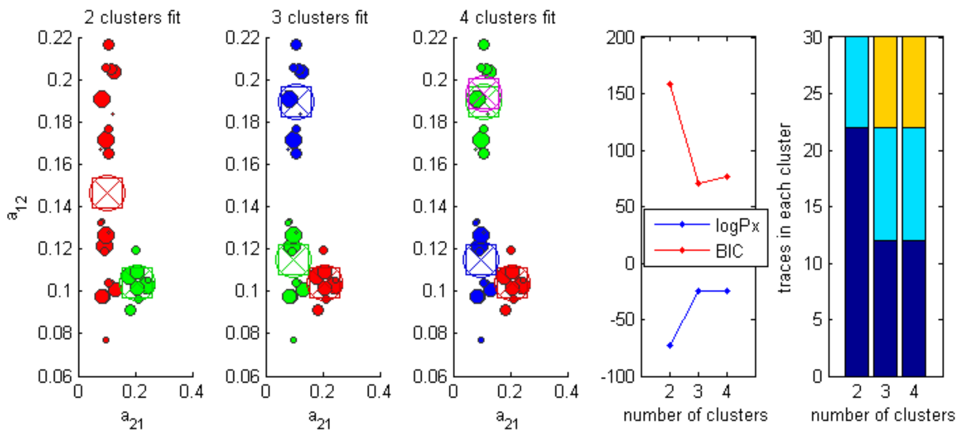


Figure 2.3: Clustering fit output. Points are colored according to their nearest (highest likelihood to produce them) cluster. Point sizes are proportional to trace length. The 4-cluster model assigns 0 traces to cluster 4. Cluster centroids are shown as squares, circles, crosses.