# STRFPAK: A Spatio-temporal Receptive Field Estimation Software

Junli Zhang, Patrick Gill, Michael Wu
Ryan Prenger, Frederic Theunissen and Jack Gallant
Helen Wills Neuroscience Institute/Department of Psychology
University of California at Berkeley

Version 4.1 - July 31, 2006

# Contents

# List of Figures

# Chapter 1

# Introduction

STRFPAK is a Matlab toolbox for estimating the linear/nonlinear stimulus-response transfer function of a sensory neuron. The resulting spatio-temporal receptive field (STRF) provides a quantitative description of neural filtering properties that can be used in subsequent computational modeling studies. The estimation techniques implemented by STRFPAK are quite general. Several algorithms are provided for estimating both linear and nonlinear STRFs from responses to either simple or complex stimuli, including natural signals. This documentation will describe the motivation and philosophy behind its design and provide details on how to use it.

This manual documents version 4.1 of STRFPAK, the Spatio-Temporal Receptive Field Estimation Software. All of the methods have their own functions and a graphical user interface (GUI) for easy access to the data and results. The STRFPAK toolbox can import data in various formats, and it is also able to cope with non-supported formats with very little user-intervention. There is also a choice of preprocessing methods and post-processing routines.

The toolbox was developed on Linux and Window platforms and should work on all platforms with Matlab version 6.0 or later. Due to some differences in the way Matlab operates between the Unix and MS Windows flavors, some fonts in the GUI windows look bigger. Resizing the windows solves this problem.

In addition to this manual, there are also a few other pieces of help documentation for this toolbox. **"A Quick Guide to STRFPAK"** has been written to show the first-time user how to download, install and run STRFPAK as quickly as possible. In addition, a STRFPAK tutorial with an example can be found on strfpak's website.

## 1.1 Explanation of the Conventions

Within this documentation, we have tried to improve the readability of the text by using certain styles and fonts. Any reference to Matlab code appears in monospace. Commands written at the Matlab commandline will have the Matlab prompt included on the left-hand side, for example,

```
>> help strfFirstGUI
```

Any reference to a particular $\boxed{\text{button}}$ in the STRFPAK GUI will have a box around it, while **Window Titles** and **Subwindow Menus** will be shown in bold.

## 1.2 Online Support and Downloading STRFPAK

The STRFPAK software has its own homepage containing software and documentation. It is updated regularly.

`http://strfpak.berkeley.edu`

From the website, you can download the Unix/Linux version and Windows version compressed tar files that contain all the Matlab files and sample data sets.

The toolbox also has a user mailing list for discussion, problems and announcements.

`strfpak_users@ist.berkeley.edu`

Since the mailing list is maintained manually, you need send an email to **theunissen@berkeley.edu** or **gallant@socrates.berkeley.edu** to join the list.

## 1.3 Contact Information

Besides the STRFPAK mailing list,

`strfpak_users@ist.berkeley.edu`

we can also be contacted by email directly:


```
Frederic Theunissen (PI) theunissen@berkeley.edu
Jack Gallant (PI) gallant@socrates.berkeley.edu
Patrick Gill patgill@berkeley.edu
Ryan Prenger prenger@socrates.berkeley.edu
Junli Zhang (Programmer)
```

# Chapter 2

# STRFPAK's Design Philosophy

Graphical visualization is a standard technique for facilitating human comprehension of complex phenomena and large volumes of data. The study of neuron behavior is extremely complex, involving measuring and characterizing how stimulus attributes, such as light or sound intensity, are represented, and how neurons respond to a wide variety of stimuli. It is natural to use visualization techniques to gain insights into what input data look like, and how good we can fit and predict neuron responses from our given input data. The main window of STRFPAK shown in Figure 3.2 clearly pictures overall flow from getting input to doing estimation, prediction and validation.

The three principal goals in designing STRFPAK are ease of understanding, ease of use and portability.

## 2.1 Ease of Understanding

STRFPAK is designed as a software tool with a graphical user interface for ease of understanding and use. Since the purpose of a graphic interface is to facilitate human understanding, it is imperative that the visual displays provided be as intuitively meaningful as possible. From the main window of STRFPAK, the basic functionality of STRFPAK is easily understood: getting input data, displaying input data, estimating the STRF, displaying intermediate results such as stimulus statistics and the estimated STRF, predicting the neuron response on a different data set using the estimated STRF, and validating goodness of the STRFs prediction. The **help** button and self-comment field are provided for every window of STRFPAK. They convey detailed information on each figure, chart or diagram shown in the window.

## 2.2 Ease of Use

One of the main purposes of software tools is to relieve tedium, not to promote it. Through the use of color, pop-up menus, editable text fields, and a mouse- and menu-oriented user interface, STRFPAK is designed to be easy to learn. It also provides an interactive tracing window for computationally intensive steps so that you can easily gain insight into how much time each calculation procedure takes. The dialog windows are also implemented for showing detailed warning, error, and confirmation messages.

## 2.3 Portability

STRFPAK is implemented using Matlab programming language. It can run on any operating systems that support Matlab and the X Window System. Although STRFPAK is effective in color, it also works on monochrome and grayscale monitors. For some future version of STRFPAK, we propose to develop a standalone executable program that will be independed of the Matlab environment.

# Chapter 3

# User Manual

This chapter deals with everyday usage of the toolbox. In a step-by-step manner we show how data is loaded, the preprocessing options available, what type of analysis we have, how the intermediate and final results can be saved, and how these results can be viewed. More information on each of the stages, such as a description of the available data formats and the different modeling algorithms, can be found in the later chapters.

## 3.1 Installing STRFPAK

STRFPAK is supported by grant from NIMH and developed by the Theunissen Lab and the Gallant Lab at the University of California, Berkeley. It is free software. You can go to the STRFPAK website

`http://strfpak.berkeley.edu`

to get the source codes.

After downloading the Unix/Linux version STRFPAK to the directory of your choice, on the unix/linux command prompt type

```
gzip -dc STRFPAK-4.1.tar.gz | tar -x
```

to uncompress and then to untar it to the directory STRFPAK. For installing the Window Version STRFPAK, use **WinZip** software to unzip it and then install it directly to the directory. For any problems or questions about downloading and installing STRFPAK, please refer to our STRFPAK FAQ page in the STRFPAK web site.

## 3.2 Getting Help

An overview of all the functions in the STRFPAK toolbox can be obtained through the standard Matlab help function. Clicking Help button on each STRFPAK GUI window gives you more information about detailed explanation of that window. For some new term/parameter, STRFPAK also provides a brief explanation by clicking ? mark right after the parameter name.

## 3.3 Starting STRFPAK

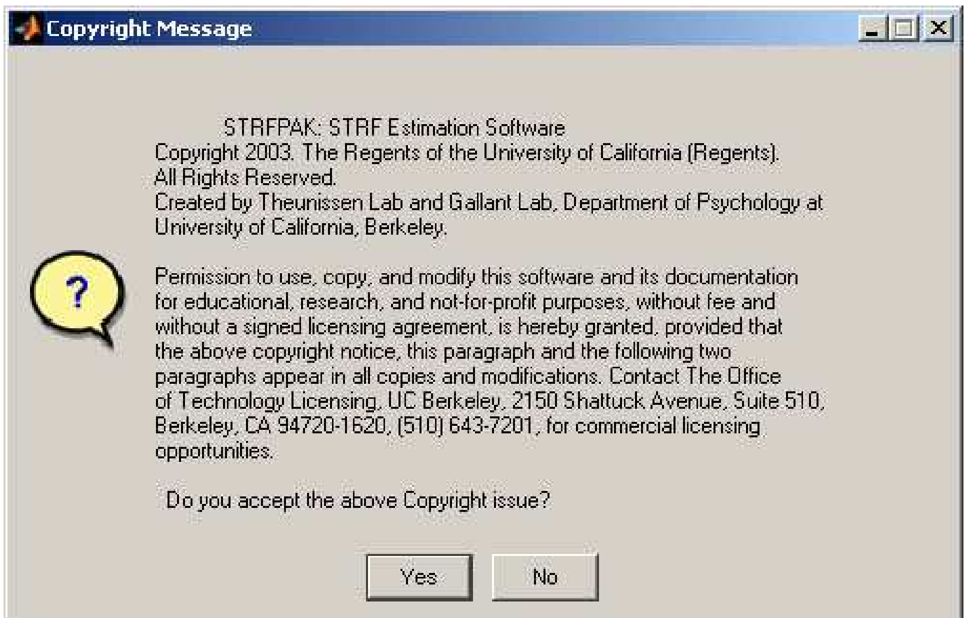


Figure 3.1: STRFPAK copyright agreement

STRFPAK can be run either through its graphical interface or through the Matlab commandline. For the first time users, the GUI version is far easier to use. After running the GUI one time, one Matlab script file will be generated containing all the parameters, input data and functions called in the GUI process. This gives the advanced user great flexibility and the ability to add and modify the code for the batch processing of large jobs. We also provide a template batch-processing Matlab script file. The detailed description is given in the later sections.

To start the graphical interface, on Matlab commandline type:

```
>> strfpak
```

If you get an error saying "undefined function or variable: strfpak", it is because you have not added the path to the STRFPAK directory in Matlab. To fix this, you can "cd" to the STRFPAK directory and simply type:

```
>> addpath(pwd)
```

Another way you can write your own startup.m file under your Matlab directory and include the STRFPAK directory in this file.

After you type "strfpak" on Matlab command line, STRFPAK copyright agreement in figure 3.1 pops up. If you agree with the copyright agreement, next you will be asked whether you want to start disk caching. If you choose **Start Cache**, **STRFPAK: Caching Option** window in figure 3.31 shows up. The reason for caching option is that you might run the same stimulus sets over and over again in your work. With turning on disk caching option, you didn't have to re-compute them every time you run a cell through STRFPAK can save the output of long computations in a central directory. So every time you need them you just have to look up the answers, not recompute them. If you are interested in how STRFPAK handles caching, you can type "help Cashing_Readme" or click $\boxed{\text{Help}}$ from **STRFPAK: Caching Option** window. You can also choose not using cache option now.

## 3.4 STRFPAK's Main Window

STRFPAK's main window, **STRFPAK: STRF Estimation Software**, that pops-up after committing copyright agreement is shown in Figure 3.2. In the window, the main functionalities of the STRFPAK toolbox are clearly shown by its panels' labels: **Load Input**, **Preprocess**, **Estimate**, **Validate** and **Predict**. You can go through these five panels sequentially to explore all the features. For real case, it may not be necessary. For example, if you just want to load your input and display them graphically, you can stop in **Load Input** panel. But if you want to use your data to predict the neuron response, you need go through the whole procedure from loading input to prediction. STRFPAK's main window's left bottom panel provides brief explanation after you click any button in the window. It serves as a reference. For example, it shows you which stage you are after you click one button. The right bottom shows information, e.g. **Set Cache Option** and **Clear Cache**. **Load Prev Results** A detailed description of each button on each panel is given in the following sections.

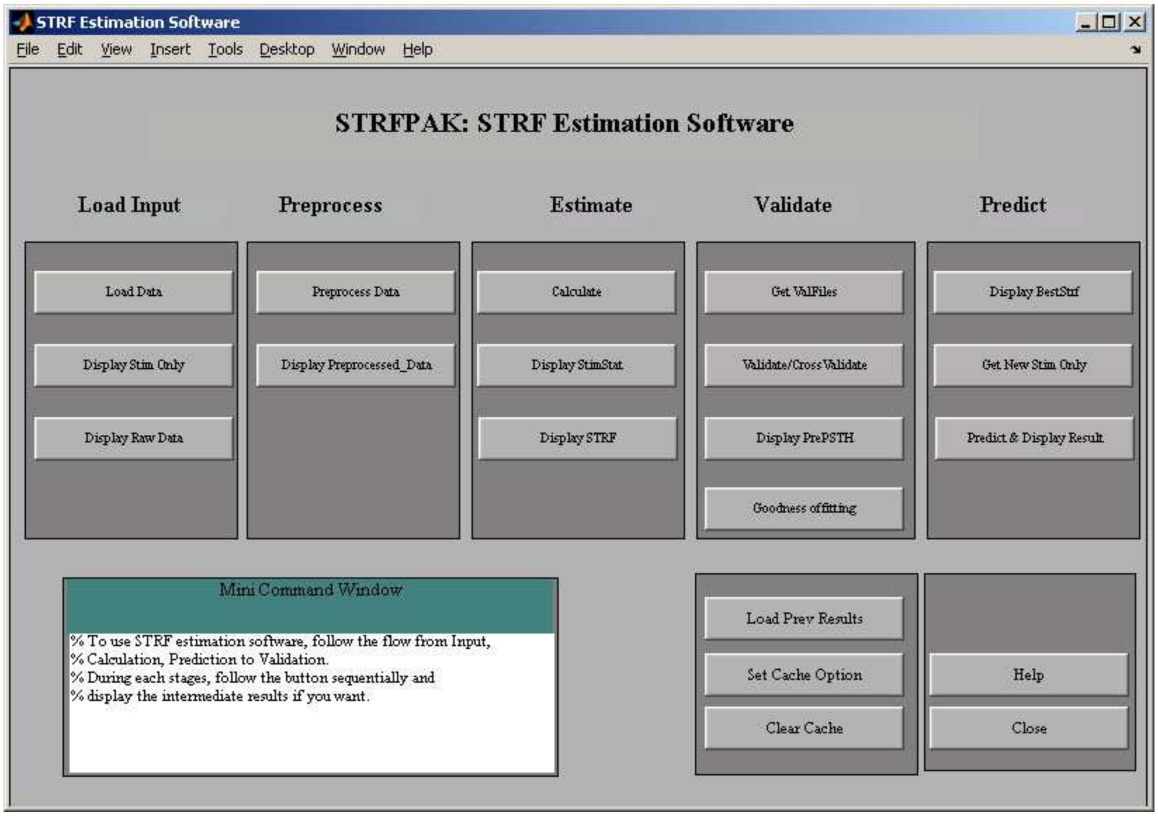


Figure 3.2: The Main window of STRFPAK

## 3.5 Load Input

**Load Input** deals with loading new data and collecting any STRF-related information, e.g. data type (1-D or 2-D). This panel includes loading data, displaying raw stimulus and display raw stimulus and neuron response data within one window.

### 3.5.1 Load Data

This window in Figure 3.3 lets you to load input files and specify the type of input files.

- **Choose Stimulus:** Top panel shows the directory of stimulus or you can type the new directory. Middle panel lists all the files of the current directory. Bottom panel is the button. When it is clicked, you can browser the directory.

- **Choose Response:** Top panel shows the directory of response or you can type the new directory. Middle panel lists all the files of the current directory. Bottom panel is the button. When it is clicked, you can browser

- **Select/Remove:** To select the data set, you need click the file names from the middle panels of CHOOSE STIMULUS and CHOOSE RESPONSE and then click SELECT button. Similarly for removing. For multiple selection, please click Ctrl key and the file names at the same time. Again it is similar for removing.

- **Show Selected Data Sets:** The panel shows the stimulus name and its corresponding response.

- **Data Type Options:** Raw Data: It means the stimulus and the response are in raw format. For example: raw auditory stimulus is song wave file with the file ext as .wav and the response data is in spike arrival time or in text format with the number of spikes at the particular time location. For raw vision stimulus, it could be in movie format, X x Y x T. Here X and Y are pixel space position and T is for time. Preprocessed Data: It means the stimulus is in X x T and the response is in Trial x T.

- **Stim Dimension:** This list option asks you to specify what spatial dimension size is. For non-spatial data, you need specify 0-D data; for song type data, the dimension size is 1-D; and for movie type data, the size is 2-D.

- **HELP** click help button to get this help window.

- **Cancel** - clear all variables and close the window.

- **DONE** - assign the global variables with selected data files and the type of the data sets and close the window.

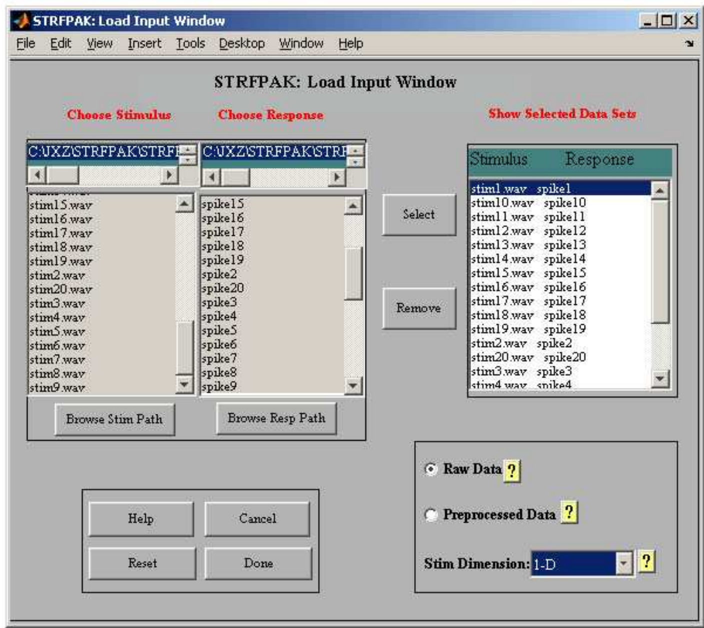


Figure 3.3: The Load data window in STRFPAK

### 3.5.2 Display Stim Only

After clicking this button, **STRFPAK: Display raw stimulus window** in Figure 3.4 and 3.5 show up. Figure 3.4 is one auditory example and figure 3.5 is an example for vision data. In the left panel, the top two text boxes shows the following plot's starting position and ending position. They do not have unit. You can also select data by clicking Select Data and then choosing the starting position and ending position using your mouse on **data selector** yellow area. If the stimulus is song-type data, plot of stimulus is amplitude of the song wave file. **Frame Set** is only for 2-D data. If the data is movie-type data, plot of stimulus is 6 frames each time. You can choose next frame set by typing in the number of frame set you want to display or clicking the frame slider. In the left bottom panel, there are two buttons:

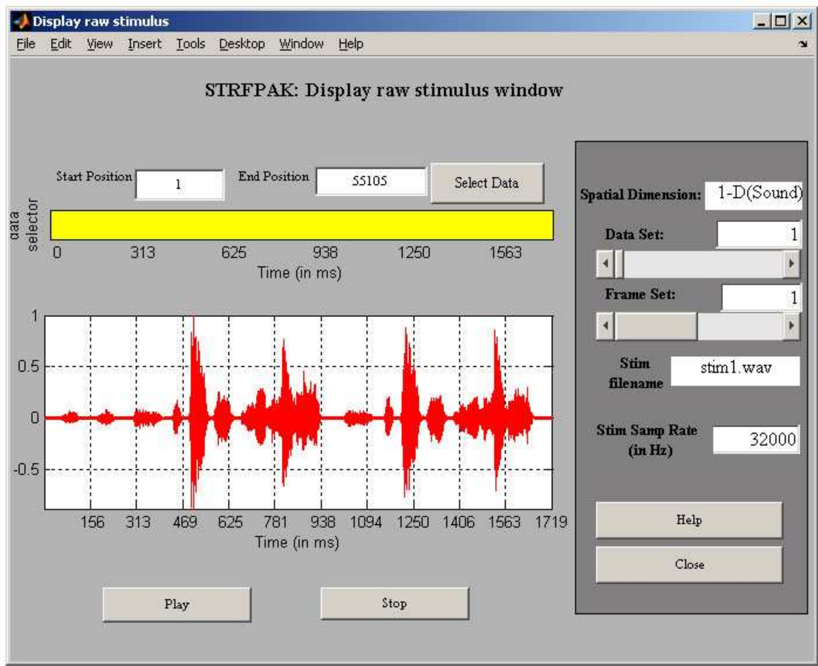


Figure 3.4: STRFPAK: Display raw stimulus window

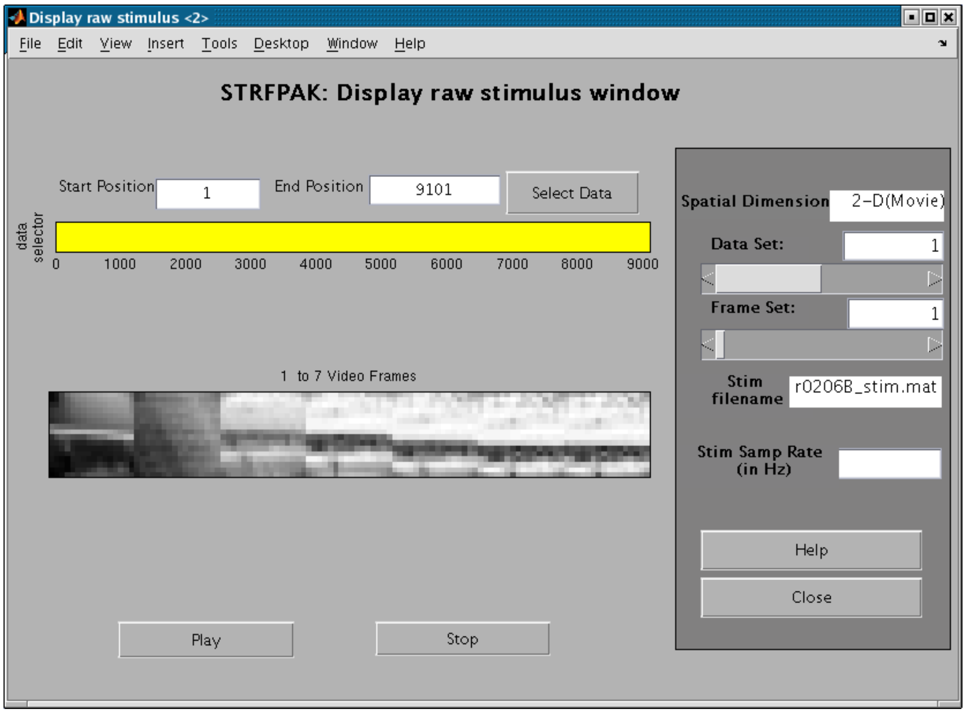


Figure 3.5: STRFPAK: Display raw vision stimulus window

Play and Stop, for playing and stopping song or movie. The right panel in dark gray background is information panel. Each item should be obvious based on its label name.

### 3.5.3 Display Raw Data

This option displays stimulus and response data together in one window shown in figure 3.6. The most part of this window is very similar to **Display Stim Only**. The new feature of this window comparing with **Display raw stimulus** is to display spike trains and smoothed/unsmoothed PSTH. The smoothing window is 21 msec by default. You can change the window size by typing new values in smoothing textbox field. If you do not want to smooth PSTH, you type 0 to turn off smoothing in the smoothing text field. Right below Play and Stop buttons, STRFPAK displays at most 10 trials of spike trains each time. When your response data have more than 10 trials, you can display the rest by clicking Next_10Trials or Prev_10Trials.

## 3.6 Data Formats and Dimension Layout Requirement

Currently the STRFPAK toolbox can support the following data formats, shown in Table 3.1.

Table 3.1: Data formats supported by STRFPAK

| File Formats | Extension | Explanation |
|---|---|---|
| Matlab binary | .mat | binary data in Matlabs format |
| ASCII files | .txt, .dat | any human-readable format |
| Raw wave files | .wav | wave sound data and sample rate |
| Raw Spike files | .txt, .dat | single or multiple trial spike arrival times |

It is sometimes desirable to preprocess the stimulus before executing the STRF. For example, most auditory STRF estimation is done using a time-frequency representation of the sound instead of the raw wave file. STRFPAK comes with a variety of built-in preprocessing options, or you can perform their own transformation on the raw file and save it in a STRFPAK-readable format. For your already-preprocessed data, the STRFPAK also requires

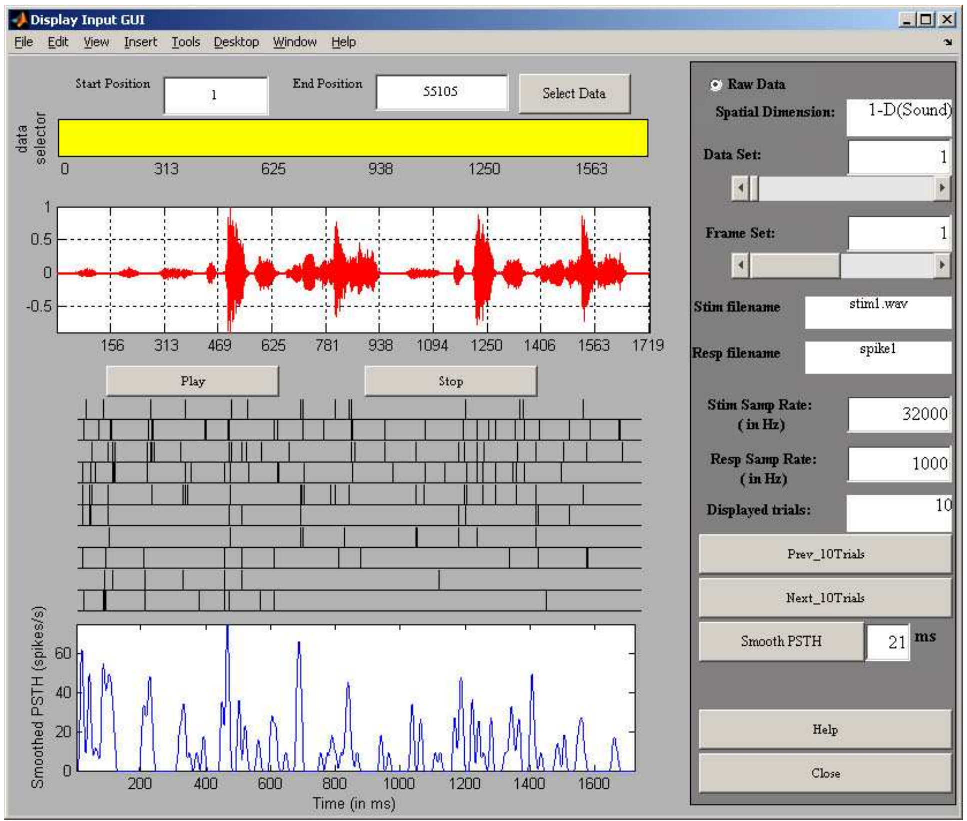


Figure 3.6: STRFPAK: Display raw stimulus and response

that a stimulus matrix to have $non - Time$ X $Time$ format (*i.e.* have $non - Time$ rows and $Time$ columns). Here $non - Time$ refers any domains except time, e.g. the frequency domain for sound, the wrapped spatial domain for video, or other stimulus properties such as orientation or luminance. STRFPAK takes only square matrices as input; if the $non - Time$ part of the stimulus naturally has more than one dimension, it needs to be wrapped into one dimension. Here I give a simple Matlab script to do that.

```
%  Assume stim is 3-D which is X x Y x T. Here X and Y
%  are a pixel's position and T is time.

>>X = size(stim, 1);
>>Y = size(stim, 2);
>>T = size(stim, 3);
>>desired_stim = reshape(stim, ...
         X*Y, T);
```

## 3.7 Preprocess

**Preprocess** stag includes processing raw data files using STRFPAK's preprocessing methods and then preparing the right data formats for the later analysis.

### 3.7.1 Preprocess Data

Once Preprocess Data is clicked, **STRFPAK: Input Preprocessing Menu** window shown in Figure 3.7 pops up. STRFPAK provides very flexible options to preprocess time-series data. It offers **Pre-specified Options** and **User-specified Options**. **Pre-specified Options** include preprocessing options for 1-D(sound) and 2-D (movie).

#### Pre-specified Options

For 1-D option:

- 1. Short-time Fourier Transform: This option is to extract signal's frequency information at time location. The method is to create a bank of band-passed filters whose impulse response consists of a carrier

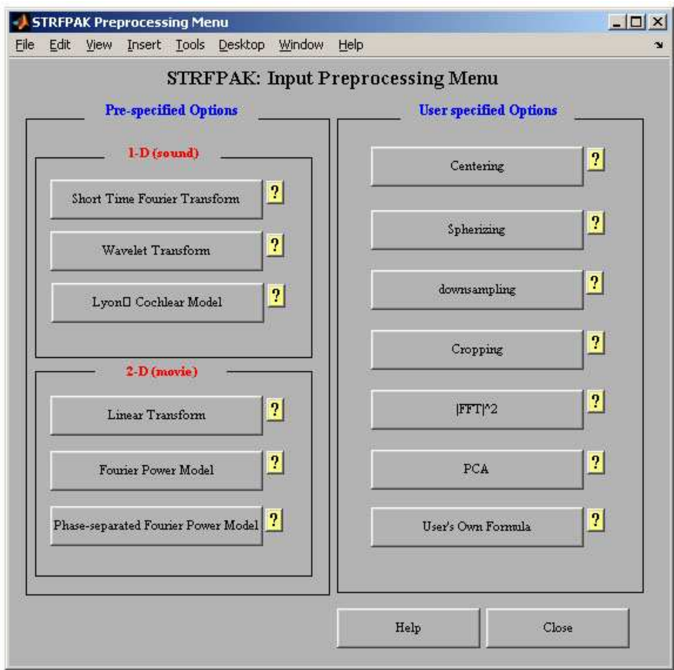


Figure 3.7: The Preprocessing Menu in STRFPAK

frequency (the center frequency of the filter) modulated by a short-time window. Then take Fourier transform of this time window to get the frequency gain curve of the band-passed filter.

- 2. Wavelet Transform: This transform uses the Morelet wavelet filter banks to get approximately spaced logarithmically in Frequency domain.
- 3. Lyon Model: Lyon model includes not only the approximate logarithmic spacing of filter center frequencies (log at high frequencies and more linear at low frequencies), but also the rectification performed by inner hair cells and, optionally, adaptive gain control.

For 2-D options, STRFPAK has the following choices:

- 1. Linear Transform: This option is actually to reshape movie frames and then line up with its corresponding response data. For example, the vision data is 16 X 16 X 1000. After this transform, the preprocessed vision data becomes 256 X 1000.
- 2. Fourier Power Transform: It removes spatial phase but preserves information about stimulus orientation and spatial frequency. For more detailed explanation, please refer to the following David and Gallant's paper.
- 3. Phase-separated Fourier Power Transform: This option is similar to **Fourier Power Transform** but keeps spatial phase information of stimulus.

#### User-specified Options

This option includes **Centering**, **Spherizing**, **Downsampling**, **Cropping**, $|FFT|^2$, **PCA** and **User Own Formula**. This list is still a proposed list. They are under construction.

#### Short Time Fourier Transform

In auditory system STRF estimation, it is common to represent sound in a time-frequency representation rather than a raw waveform. The short time Fourier transform (STFT) is a simple example of such a representation,

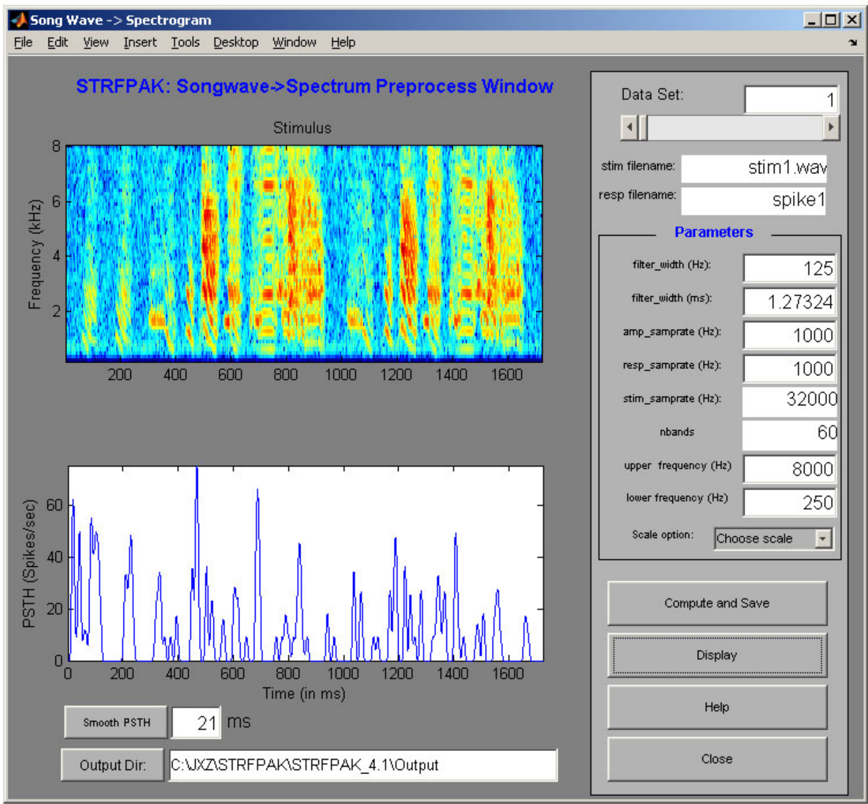


Figure 3.8: The Songwave → Spectrogram window in STRFPAK

and is included in STRFPAK. The STFT representation of a sound can be thought of as the amplitude envelope of the different frequency components of a sound. After clicking the Short Time Fourier Transform button, the **Songwave → spectrogram** window shown in Figure 3.8 appears. The right panel of that window displays options for and properties of the STFT. The left panel is used for displaying your output when you click the Display button.

- **Data Set**: This includes text field and slider. It is used for showing all the files loaded. If you load more than one data sets, you can use the slider to choose the data set. But this is just for display. So you have to do computation first.

- **Parameters**: All the parameters needed for the STFT on the sound wave stimulus files are specified here. For detailed description about this method, please refer to our preprocessing paper [**?**].

  - **filter_width (Hz)**: the width of the filter in Hz. It defines window length of filter (here we use a Gaussian filter).
  - **filter_width (ms)**: the width of the filter in ms, inversely proportional to the filter width in Hz.
  - **amp_samprate (Hz)**: the sampling rate that we want for the amplitude envelope. By default, it is 10 times that of **filter_width**, but you can change it.
  - **resp_samprate (Hz)**: the sampling rate of the spike data. This must be set to the sampling frequency in Hz when you collect your data. For example, with 1 ms bins, the resp_samprate would be 1000 Hz.
  - **stim_samprate (Hz)**: the sampling rate of stimuli in Hz. e.g. the demo songwave file has a sampling frequency of 32000 Hz, and many music files have a sampling rate of 44100 Hz. It is automatically calculated for you based on your sound wave file.
  - **nbands**: the numbers of frequency bands covered for the calculation. You don't need to specify since the code will it figure out based on your **filter_width (Hz)**.

- **Scale option**: the choice of whether linear scale or logarithmic scale is used for the amplitude envelope. For many sensory systems a neurons response correlates better to the log of a stimulus intensity better than to the stimulus intensity itself.

- Compute and Save: Computes the spectrogram of the signal and saves the result into the directory which you are asked to specify. The computing status bar also shows up so that you can know the progress of the computation.

- Display: Graphically displays the spectrogram of the stimulus and the psth smoothed with **psth_smoothconst** window size. If more than one data set are chosen, Next and Prev buttons show up so that you can click to see the next data set.
  - **Stimulus**: Image scaling plot of the stimulus. The X axis is time domain and the Y axis is frequency domain.
  - **PSTH**: Smoothed or unsmoothed PSTH plot. The X axis is time domain and the Y axis is PSTH (in spikes/sec).
  - Smooth PSTH: the window length for smoothing your psth. For validation purposes, it is common to smooth the PSTH before calculating a correlation coefficient.
  - OutputDir: This button and text field show output directory.

- Help: Causes a help window on this window to appear.

- Close: Closes this window and saves all the parameters and all the results.

### Wavelet Transforms

After clicking the Wavelet Transform button, the **Songwave → scalogram** window shown in Figure 3.9 appears. The right panel of that window displays options for and properties of the wavelet transform. The left panel is used for displaying your output when you click the Display button.

- Load files: is used to load raw data, e.g. sound wave files and their associated response data files into STRFPAK. The **Load file window**

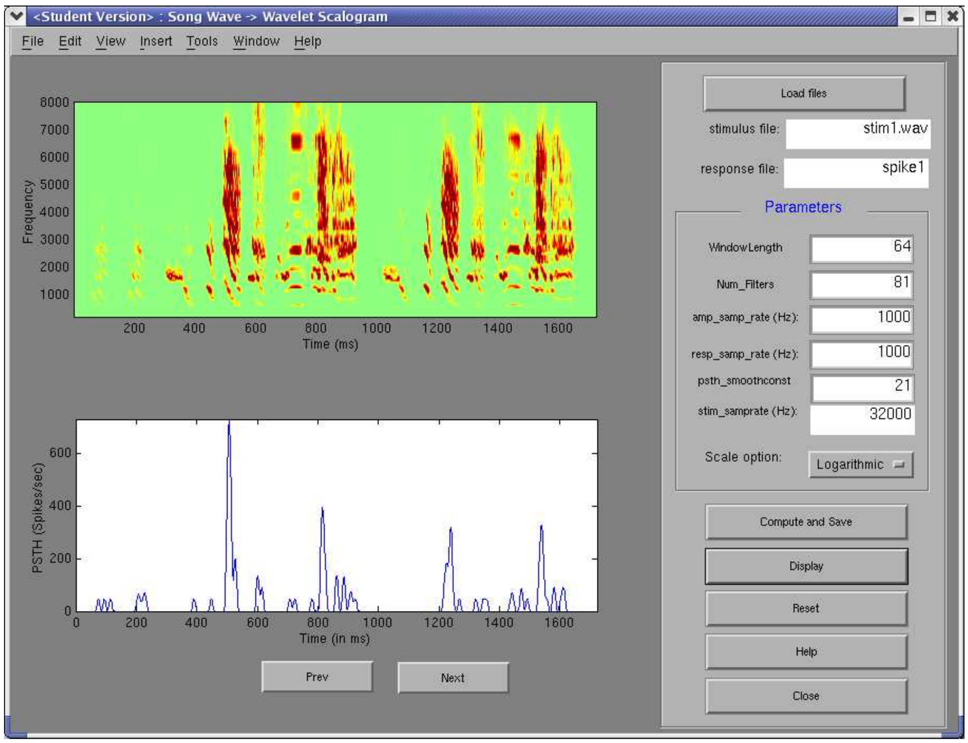


Figure 3.9: The Songwave → Scalogram window in STRFPAK

shown in Figure 3.3 pops-up after Load files is clicked. For a detailed description about this **Load file window**, please see above section.

- **Parameters**: All the parameters needed for the wavelet transform of the sound stimulus files are specified here. For a detailed description about this method, please refer to our preprocessing paper [**?**].

  - **WindowLength**: the window length of the Morelet analyzing wavelet at its coarsest scale.
  - **Num_filters**: the total number of the wavelet filters you want to use. The larger the number of filters, the smoother of the scalogram, but the longer it takes to compute.
  - **amp_samprate (Hz)**: the sampling rate that we want for the amplitude envelope. By default, it is 10 times of **filter_width**. But you can change.
  - **resp_samprate (Hz)**: the sampling rate of the spike data. This must be set to the sampling frequency in Hz when you collect your data. For example, with 1 ms bins, the resp_samprate would be 1000 Hz.
  - **psth_smoothconst**: the window length for smoothing your psth. For validation purposes, it is common to smooth the PSTH before calculating a correlation coefficient. psth_smoothconst is the width in ms of this smoothing window.
  - **stim_samprate (Hz)**: the sampling rate of stimuli in Hz. e.g. the demo songwave file has a sampling frequency of 32000 Hz, and many music files have a sampling rate of 44100 Hz. It is automatically calculated for you based on your sound wave file.
  - **Scale option**: the choice of whether linear scale or logarithmic scale is used for the amplitude envelope. For many sensory systems a neurons response correlates better to the log of a stimulus intensity better than to the stimulus intensity itself.

- Compute and Save: Computes the spectrogram of the signal and saves the result into the directory which you are asked to specify. The computing status bar also shows up so that you can know the progress of the computation.

- $\boxed{\text{Display}}$: Graphically displays the scalogram of the stimulus and the psth smoothed with **psth_smoothconst** window size. If more than one data set are chosen, $\boxed{\text{Next}}$ and $\boxed{\text{Prev}}$ buttons show up so that you can click to see the next data set.

- $\boxed{\text{Help}}$: Causes a help window on this window to appear.

- $\boxed{\text{Close}}$: Closes this window and saves all the parameters and all the results.

### Linear Transform

Figure 3.10 is STRFPAK's linear transform window. The layout is very similar to STRFPAK's STFT window. The left panel shows plots of preprocessed stimulus and PSTH. The right panel displays all related information about the plots.

### Fourier Power Model

The layout of this window in Figure 3.11 is the same as other preprocessing option. The only difference is **Parameter** panel. We will go over parameter lists here.

- **new start frame** - new selected starting frame.

- **new stop frame** -new selected stopping frame. If they are all zero, that means keep all.

- **Hanning flag** - apply hanning window before fft (default 0, off)

- **Power Scale** - take Fourier amplitude to this power (default 1)

- **DC value** - subtract this value from movie before window, then add back before Fourier transform... unless **Temp Replacement** =0, in which case negative component gets stuck in an weak sf channel (def 0)

- **Temp Replacement** - experimental code for replacing high sf channels with nl-transformed temporal information (default 0)

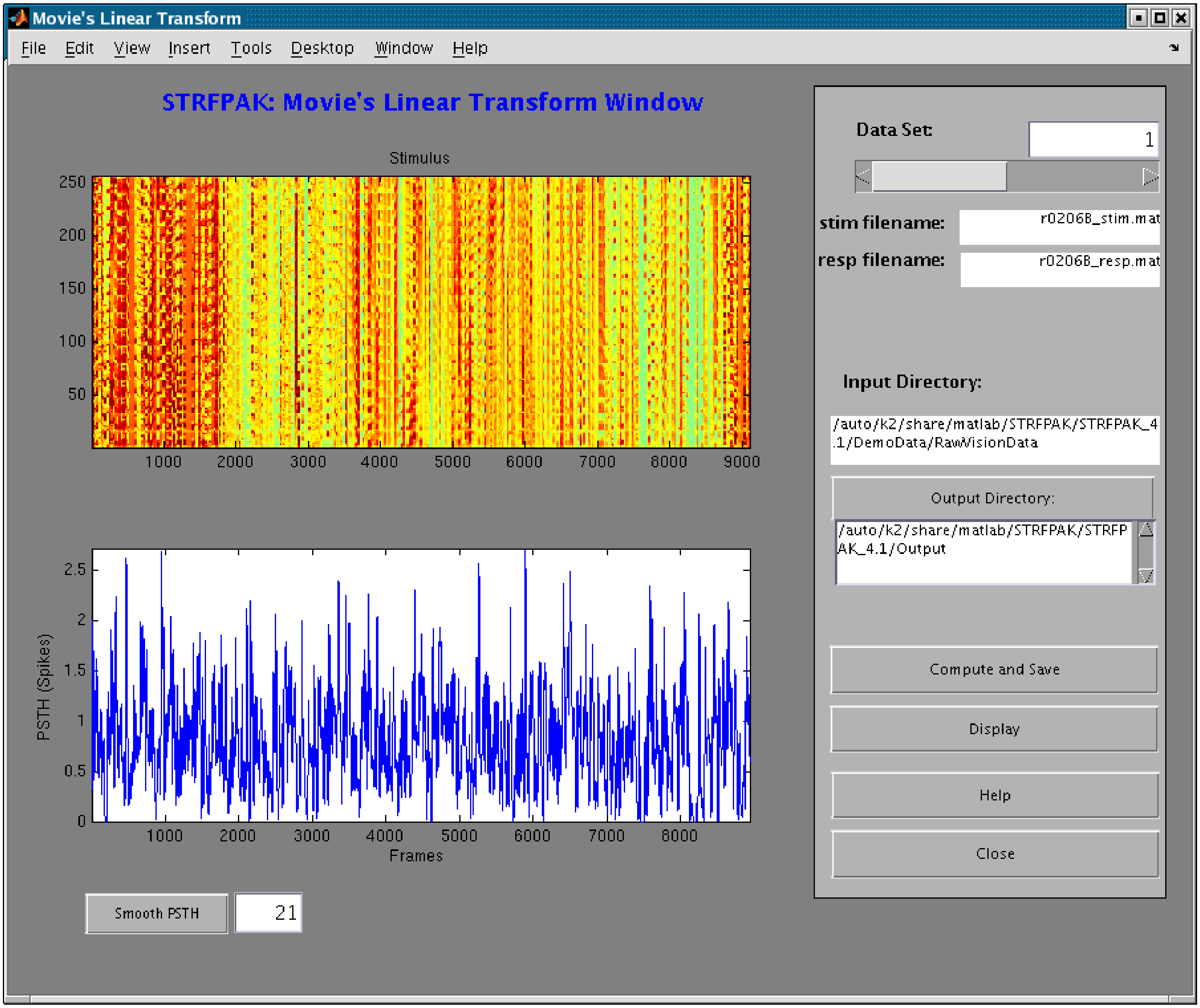


Figure 3.10: STRFPAK: Movie Linear Transform Window

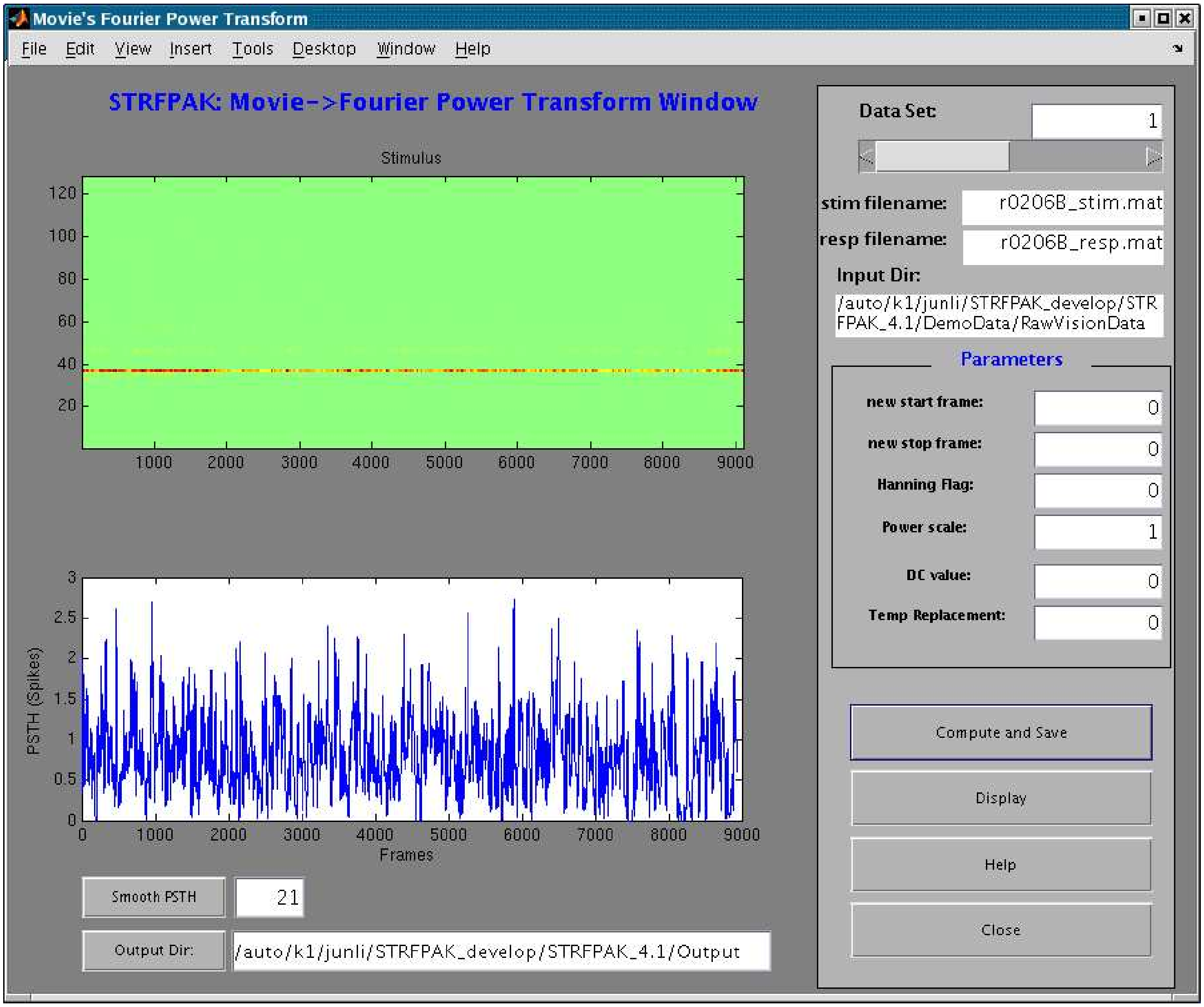


Figure 3.11: STRFPAK: Fourier Power Model Window

**References for STRFPAK's preprocessing**

- Patrick Gill, Junli Zhang, Sarah M. N. Woolley, Thane Fremouw, Frederic E. Theunissen (2005). Analysis of Signal Preprocessing Methods in Spectro-temporal Receptive Field Estimation, Journal of Comp.

- David SV, Vinje WE and Gallant JL (2004). Natural sitmulus statistics alter the receptive field structure of V1 neurons Journal of Neuroscience, 24, 6991-7006.

- David, SV, Gallant JL (2005). Predicting neural responses during natural vision, Network: Comp. in Neural Systems, 16, 239-260.

### 3.7.2 Display Preprocessed Data

This option is for displaying preprocessed data. If you use STRFPAK preprocessing option to preprocess your data, this option is just for your reference. If you load already preprocessed data from Load Data, you can choose this option to give your visual display. The following figure 3.12 displays preprocessed auditory data. New features of this option is also displaying PSTH, displaying PSTH with removing time-varying mean rate and displaying PSTH with removing overall mean rate.

## 3.8 Estimate

In this stage, the second order statistics of the stimulus are calculated and the STRF is estimated in Figure 3.13. The stimulus auto-correlation matrix, the stimulus-response cross correlation and modulation spectrum can be displayed by clicking the Display StimStat button. The estimated STRF and its related analysis can be displayed by clicking the Display STRFs button.

### 3.8.1 Calculate

In **STRFPAK: Calculation GUI** window, Parameters include all the calculation parameters used for STRF estimation. Those parameter buttons can be clicked to show a brief explanation These parameters are:

- **Time-lag** – is the time lag used for calculating the spike triggered average and the stimulus auto-correlation. Time-lag should be long

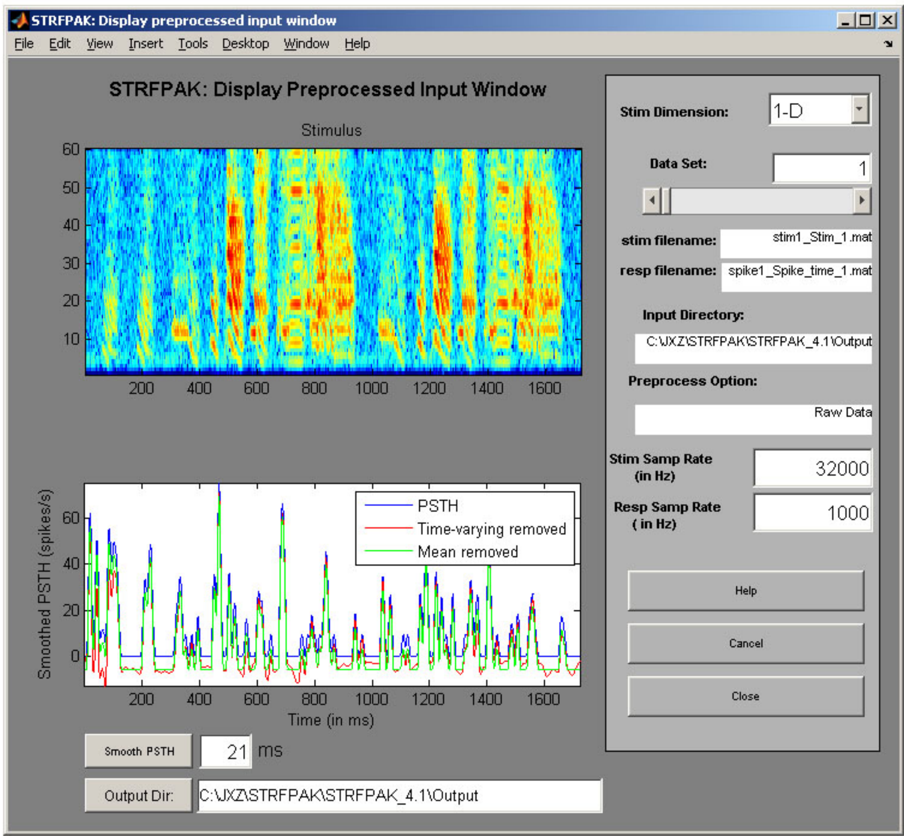


Figure 3.12: STRFPAK: Display Preprocessed Input Window

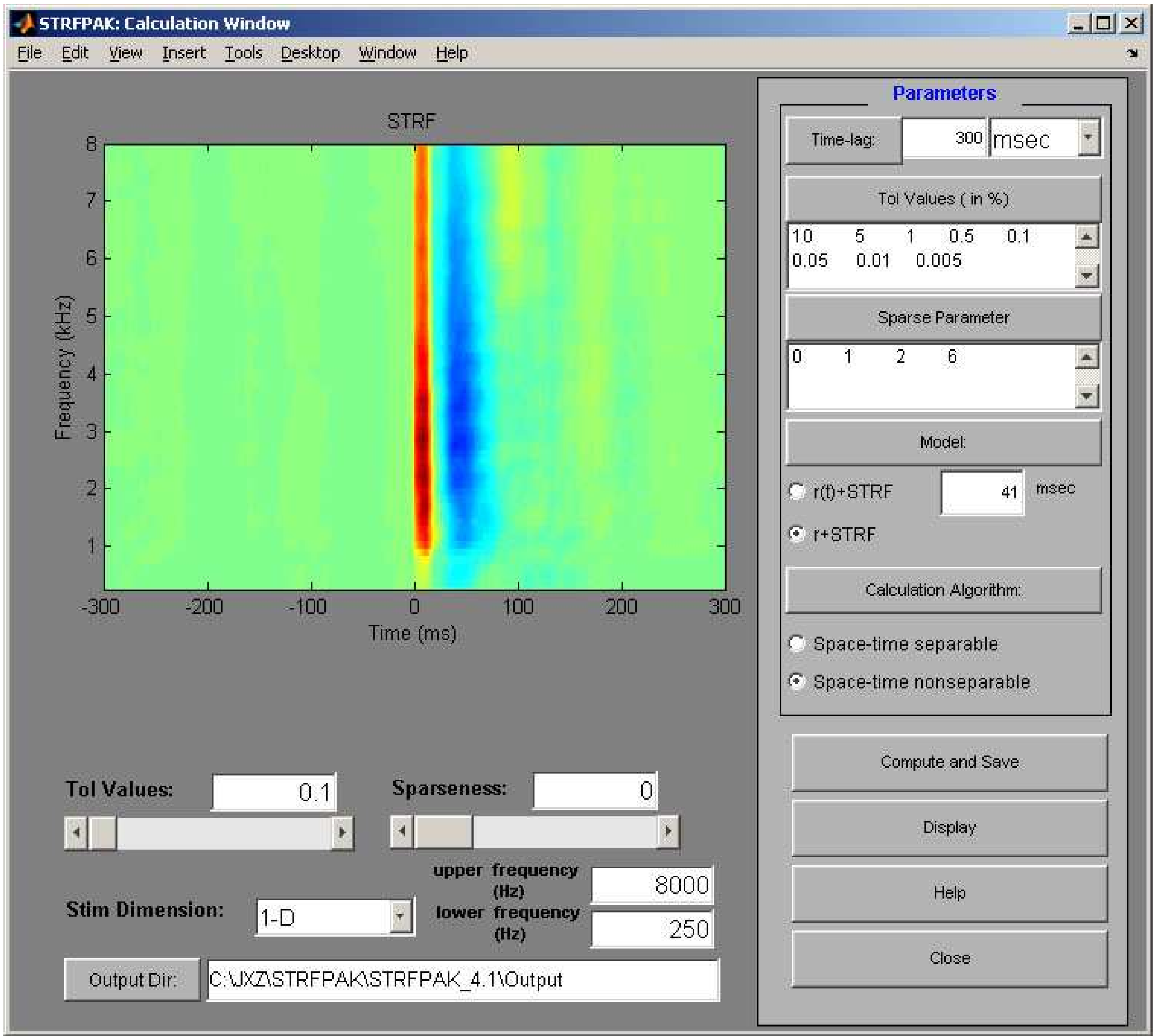


Figure 3.13: STRFPAK: Calculation Window

enough to capture the memory of the neuron and the correlation time of the stimulus.

- **Unit** – the unit of **Time-lag** could have two choices: in msec or frames. If STRFPAK or you provide stimulus sampling rate, it can be set msec or frames. Otherwise, you have to choose frames (here also means points).

- **Tol Value** – a regularization hyperparameter used to estimate the STRF. Larger tol values yield smooth STRFs. Lower tol values remove more of the stimulus correlation but can amplify noise. Use a range of them to find the best STRF for your specific stimulus and neuron. You can separate them by ",", "space" or "tab". For example: Tol_Val = [10 5 0.01 0.05].

- **Sparse parameter** – a second hyperparameter (also called Std_Val) in the STRF estimation that enforces sparseness. High Std values limits the number of significant pixels in the STRF. Use a range of them to find the best STRF for your specific stimulus and neuron. You can separate them by ",", "space" or "tab". For example: Std_Val = [0 0.1 2 4].

- **Model** – Model #1, r+STRF, means that overall mean firing rate is removed from neuronal response when estimating STRF. Model #2, r(t)+STRF, means that time-varying mean firing rate is removed from neuron response. When Model #2 is selected the smoothing window for the time-varying mean rate needs to be specified. The smoothing window is in ms. This model is useful when short stimuli are used and the neural response to all stimuli has a song an onset component. The predictions of the models to new stimuli will include both the mean rate (constant or time-varying) and the STRF prediction.

- Calculation Algorithm – In the toolbox, the two algorithms are implemented: one is for the space-time nonseparable case and the other is for the space-time separable case. If there is reason to believe the real STRF is well approximated by the outer product of a function of time and another in space, the separable algorithm may be preferred because it is faster and uses fewer free parameters. Otherwise, the more general non-separable algorithm may be favorable. All the functions used by the **Estimate** section of STRFPAK have the prefix *cal_* in their filenames. The rest of the filenames describes their specific function. For example, *cal_AutoCorr.m* program calculates the auto-correlation of the time-series signals.

- $\boxed{\text{Compute and Save}}$ Compute the spectrogram of the signal and save the results into the directory where you will be asked to input. The computing status bar also shows up so that you can know progress.
- $\boxed{\text{Display}}$ Graphically display the spectrogram of the stimulus and the smoothed psth with Smooth_PSTH window size. The smoothing width for psth is shown here. You can modify it by typing different number.
- $\boxed{\text{Close}}$: Close window and save all the parameters and all the results.

STRFPAK implements the generalized reverse correlation method described in Theunissen et al 2001 **??** to estimate the STRFs of sensory neurons from their responses to complex stimulus ensembles. Since a complete description of the second-order statistics of the stimulus ensemble is required for estimation, STRFPAK first computes the stimulus auto-correlation matrix and the stimulus-response cross correlation vector. The auto-correlation matrix of the stimulus $C_{ss}$ and the stimulus-response cross-correlation vector $C_{sr}$ are described as follows (see **??** for more details).

$$C_{ss} = \begin{pmatrix} c_{0,0} & \cdots & c_{0,M-1} \\ \vdots & \ddots & \vdots \\ c_{M-1,0} & \cdots & c_{M-1,M-1} \end{pmatrix}$$

and

$$C_{sr} =< sr >= \begin{pmatrix} < s[t-0]r[t] > \\ \vdots \\ < s[t-NM+1]r[t] > \end{pmatrix}$$

where $c_{i,j}$ denotes the correlations between spatial dimensions $i$ and $j$ for all the relevant time delays. $N$ is the length of time dimension and $M$ is the number of spatial parameters.

The estimated STRF from the linear mean-square estimation $< (\hat{r}-r)^2 >$ is given as follows:

$$h = C_{ss}^{-1} C_{sr}$$

where $h$ is the estimated STRF. STRFPAK then normalizes the cross-covariance matrix between the stimulus and the response by auto-covariance matrix of the stimulus to get the estimated STRF. For error estimation analysis, the Jackknifed STRFs are also calculated if multiple data sets are selected. For the Jackknifed error estimation techniques, please refer to [8].

As STRF estimation can be computationally intensive, STRFPAK-4.1 provides a progress status bars. When the calculation are done, a small **Done Estimation** window appears and the estimated STRF is shown in the left plot panel.

### 3.8.2 Display StimStat

As mentioned above, the second-order statistics of the stimulus ensembles is computed in the **Calculate** stage. This window visually displays the results. In the **Display Stim Statistics** window, there is a pop-up menu which gives display options:

- **Display Stimulus Spike Cross Correlation**: If this option is chosen, the window shows two image plots of the cross-correlation: the top one is the plot of original stimulus-response cross correlation, also called the **Spike-triggered Average** (STA); the bottom one is a plot of the smoothed STA. Here the smoothed version of the STA is smoothed by a Hanning Window.

- **Display Stimulus Auto-correlation in Separate Window**:

  - **For the non-separable algorithm**: The auto-correlation matrix of the stimulus is a large matrix since each entry in the matrix corresponds to the temporal cross-correlation of the stimulus intensity at two different spatial locations. Thus a question dialog box shows up to ask how large a matrix you want to display, then draws it in a separable window. Since the $i, j$ entry of the auto-correlation matrix is the same as the $j, i$ entry with time reversed, the plot we draw here is upper triangle. For the auditory case, it is organized with the lowest center frequency at the top left corner and the highest frequency at the bottom right.

  - **For the separable algorithm**: For the separable case, the auto-correlation of the stimulus is calculated in the spatial domain and in temporal domain separately. We display them in one figure: the top one is the stimulus' spatial second-order correlation and the bottom one is the stimulus' temporal correlation.

- **Display Stimulus-response Cross-correlation in Separate window (for 2-D only)**: This option is for plotting the 2-D STA. The top plot shows time-varying frames of STA. The frame duration is based on the parameter **amp_samprate**. For example, if **amp_samprate** is 72 Hz, then the frame duration is about 14 ms.

- **Display Modulation Spectrum**:

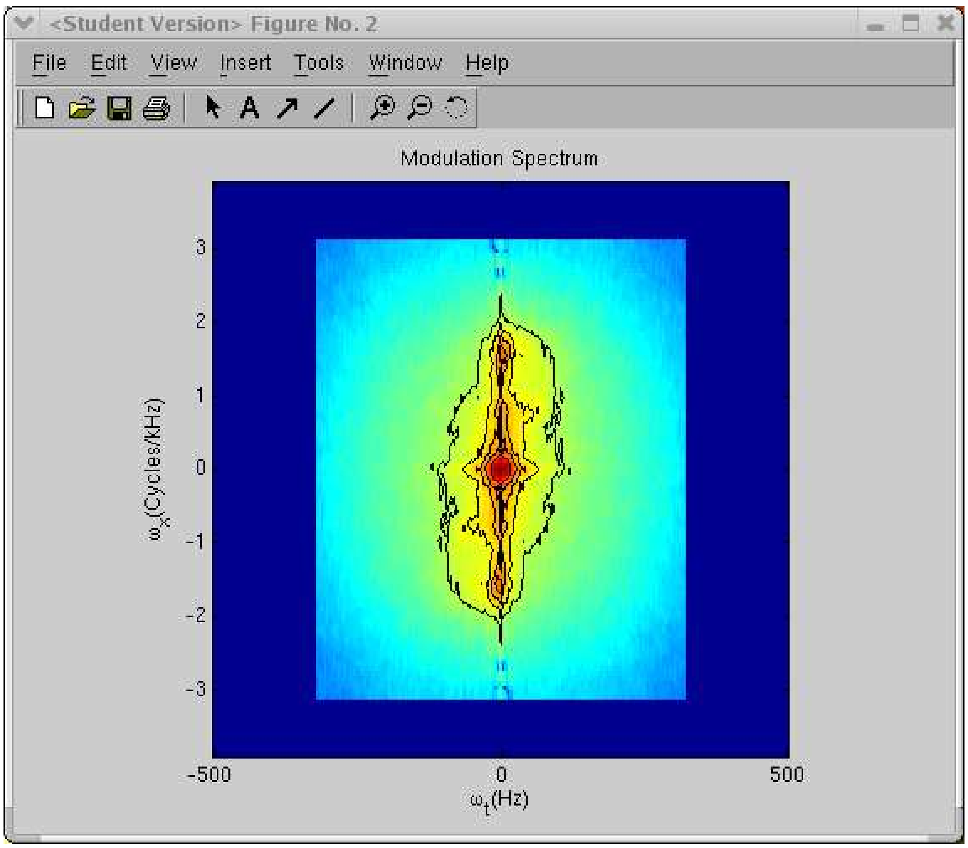


Figure 3.14: Modulation Spectrum of a sound

The modulation spectrum of an ensemble of sounds is the two-dimensional power spectrum of the spectrogram of that sound. Figure 3.14 shows the modulation spectrum of an ensemble of zebra finch songs. The $x - axis$ refers to their temporal modulations in units $\omega_t$ (in Hz) while the $y - axis$ refers to their spectral modulations, $\omega_f$ (in cycles/Hz).

### 3.8.3 Display STRF

In the toolbox, there are six display/analysis options once the $\boxed{\text{Display STRF}}$ button is clicked. The options when you click the pop-up menu on the window include:

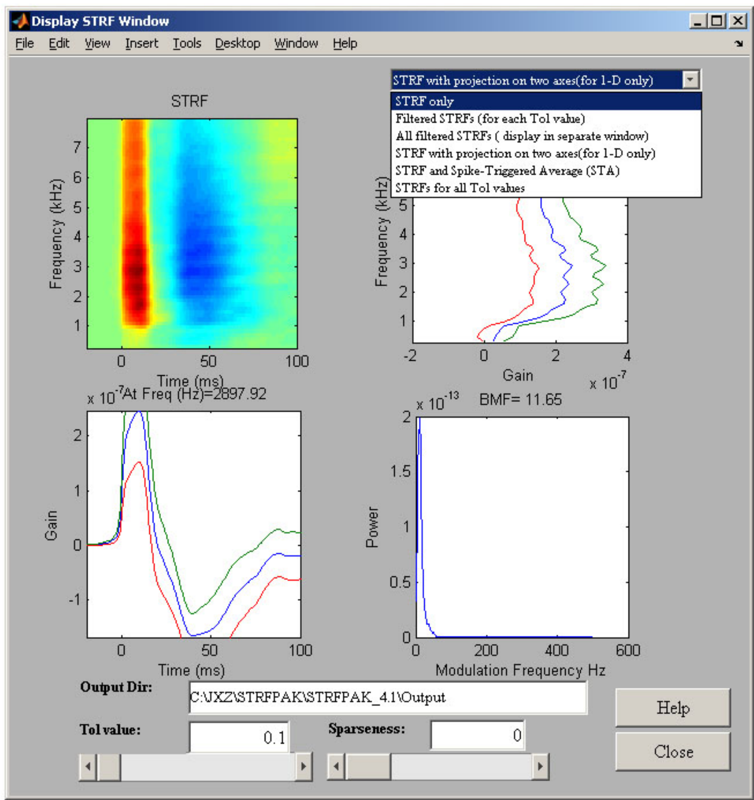


Figure 3.15: Display STRF: Auditory STRF only

- **STRF Only**: This option is for displaying only the STRF. The main plot is the estimated STRF. The right panel on the window shows the tolerance value (see the Parameters section for details) used for this

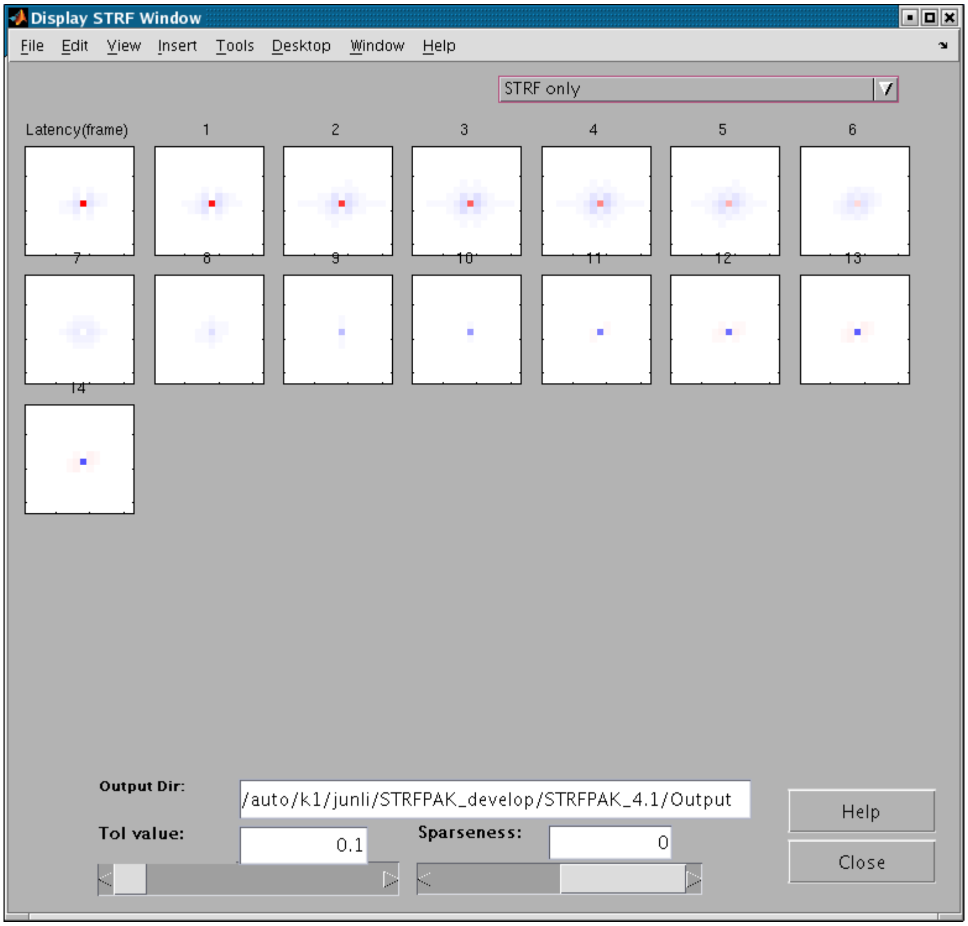


Figure 3.16: Display STRF: Vision STRF only

STRF. $\boxed{\text{Prev}}$ and $\boxed{\text{Next}}$ buttons display the STRF calculated using different tolerance values. Figure 3.15 and 3.16 are examples of the output of **STRF Only** in the **Display STRF** window.

- **Filtered STRFs (for each Tol value)**: Filtered STRFs for each Tol value mean the estimated STRF smoothed by using the second regularization parameter, **Sparseness**.

- **All filtered STRFs (display in separate window)**: For 1-D case, the filtered STRFs are displayed in another window. The row refers to $Tol\_Val$ and the column refers to $Sparseness$. But for 2-D case, there are going to have a number of figures showing the filtered STRFs. The number depends on how many **Sparseness** parameters are used.

- **STRF projection on two axes**: This option helps to obtain traditional response parameters from the STRF, e.g. best frequency, tuning bandwidth, and the neurons excitatory latency and inhibitory latency. Figure 3.16 is an example of this option. The top left figure is the plot of the estimated STRF. The top right figure (called **Spectral Profile** ) is a vertical slice of the STRF that crosses its peak excitatory response. The bottom left figure, called **Temporal Profile**, is a horizontal slice of the STRF that crosses the same

- **STRF and Spike-Triggered Average (STA)**: This option helps to compare the estimated STRF to the spike triggered average (STA). $\boxed{\text{Prev}}$ and $\boxed{\text{Next}}$ buttons help go to different STRF tolerance values. Figure 3.17 is an example of this option.

- **STRFs for all Tol values**: This option plots all the estimated STRFs with all corresponding $Tol\_Valu$.

### 3.8.4 References for STRFPAK's Calculation

For a better understanding of the theory and practice of STRF estimation, please consult the following papers.


- Theunissen, F. E., David, S. V., Singh, N. C., Hsu, A., Vinje, W. and Gallant, J. L. (2001) "Estimating spatio-temporal receptive fields of auditory and visual neurons from their responses to natural stimuli", Network: Comp. Neural Syst. 12, 1-28.

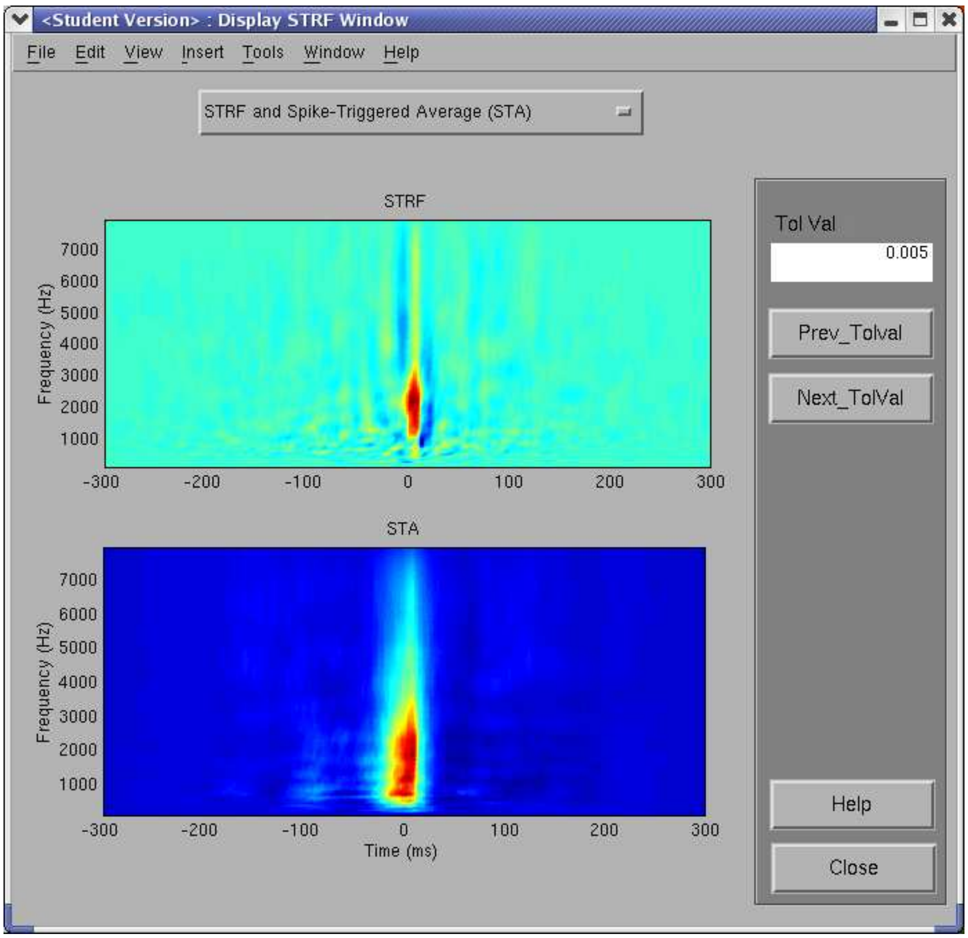


Figure 3.17: Display STRF: STRF and STA

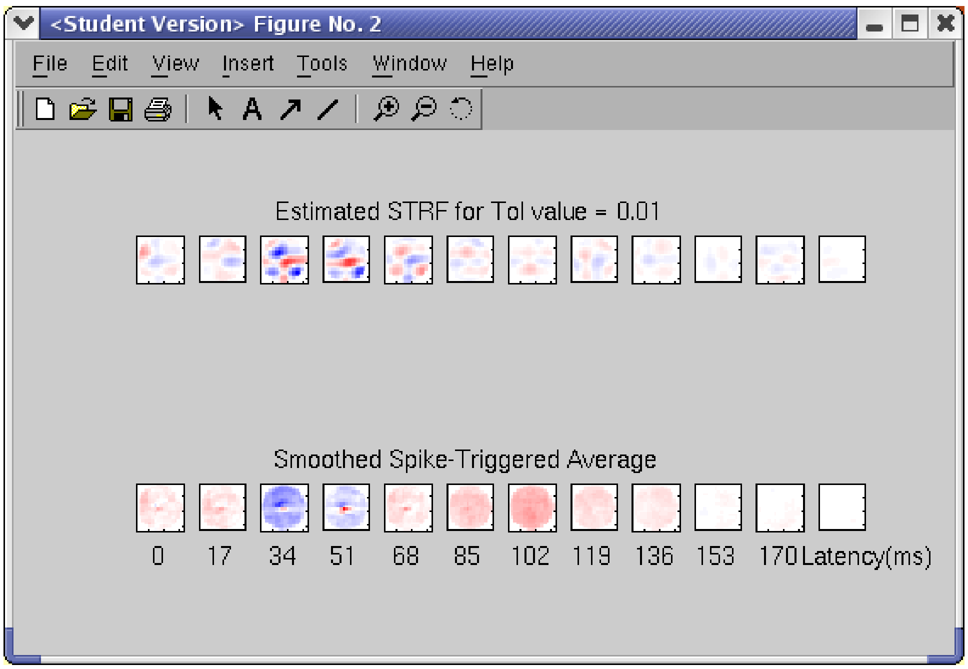


Figure 3.18: Display STRF: STRF and STA (for 2-D only)

- David, S. V.,Vinje, W. and Gallant, J. L., "Natural Stimulus Statistics Alter the Receptive Field Structure of V1 Neurons", (in progress).
- D.J. Thomson and A.D. Chave, "Jackknifed error estimates for spectra, coherence and transfer functions", Advances in Spectrum Analysis and Array Processing vol 1, ed S Haykin (Upper Saddle River, NJ:Prentice-Hall).


### 3.8.5 Reference for STRFPAK's Modulation Spectrum

To learn more about modulation spectra, please consult the following paper.


- Singh NC and Theunissen FE, "Modulation spectra of natural sounds and ethological theories of auditory processing", J Acoust Soc Am 2003 Dec, 114(6 Pt 1): 3394-411.


## 3.9 Validate

In this toolbox, the **Validate** stage helps to see how good the estimated STRF is. The current version provides two options: cross-validation or validation. For cross-validation option, your original estimation data sets have to be more than one data sets. Here one data set means one stimulus file and one response file. In this stage, you can also display the predicted PSTH with the original PSTH in one figure. STRFPAK also calculate different measures to quantitatively compare the goodness of fit. For detailed description, see the following sections.

### 3.9.1 Get ValFiles

STRFPAK applied Jackknifed ideas to do cross-validation if you choose using **Cross-Validation** option from the window shown in figure 3.19. If your estimation data sets have more than one, STRFPAK estimates the STRF using the whole data sets and also estimates the Jackknifed STRF by using the data sets removing one data sets. For example, if you provides 5 data sets, STRFPAK finally estimated 6 STRFs for you. The Jackknifed STRFs are used for cross-validation and estimating the error bars of the STRF. You can also validate the goodness of fit by providing the new validation data sets. The current limitation of STRFPAK is your validation data sets have


Figure 3.19: A questioning window for choosing cross-validation or validation

to be preprocessed using the same preprocessed methods used in preparing the estimation data sets.

The STRFPAK uses a parameter, $MATCH\_FLAG$, to keep track of your choice about cross-validation or validation. The value of the $MATCH\_FLAG$ is set based on your selection. If you click $\boxed{\text{Yes}}$ button, the STRFPAK applies the cross-validation to do validation. Otherwise, the STRFPAK predicts the neuronal response to a new stimulus you provide. If you click $\boxed{\text{No}}$ button, the **STRFPAK: Load Input Window** pops out. This window's layout and buttons are exactly the same as the **Load Data** window. For any questions or problems, please refer to the previous **Load Data** section.

### 3.9.2 Validate/Cross Validate

STRFPAK predicts neuronal responses for validating STRFs after you provide the new stimulus from the previous section. Whether the estimated Jackknife-version STRF or estimated STRF is used are based on the value of $MATCH\_FLAG$. The equation needed to generate new predictions given a stimulus and a STRF is as follows:

$$\hat{r}[t] = \sum_{i=0}^{MN-1} h[i]s_t[i]$$

where $h[i] = [h_0, ..., h_{NM-1}]^T$ is the STRF and $s_t[i]$ is the prediction stimulus.

When all calculations are done, small "Done Validation" window appears.

### 3.9.3 Displaying PrePSTH

After STRFPAK's cross-validation or validation, a predicted PSTH and an original PSTH from the actual neuron response are saved to the output directory. **Display Predicted PSTHs** window shows graphical display of these results.

Figure 3.20 shows predicted results for the auditory example at with a Tol Value of 0.05. The predicted results for the visual example are shown in Figure 3.21. From the figures, we can see that the top left panel in the window shows the input stimulus used for prediction. If the spatial domain of the input stimulus is $1D$, the $x$ axis is time in $msec$ and the $y$ axis the spatial domain (e.g. frequency in $Hz$ for auditory spectrograms). If the spatial domain of the input stimulus is $2D$, it only shows first 10 $2D$ video frames in the current version of STRFPAK. The bottom left panel shows the predicted PSTHs together with the raw neuronal responses. You can easily see the goodness of fit from this plot. The right panel of the window displays information and options. The first text box shows the tolerance value used for the left plots. You can see the plots of STRFs at other tolerance values and different sparseness parameters by clicking the $Tol\_Val$ slider or $Sparseness$ slider. The associated file names of stimulus and response used for the plots in the left panel are shown in the **pred stim file** field and the **pred resp file** field. If more than one stimulus-response file is used for the prediction, the results for different files can be displayed by clicking the *DataSet* slider. *DataSet* text field shows the number of the data sets which are displaying. Help and Close buttons are provided in the bottom right position of the window. The **Smooth_window** field shows the value used for smoothing the raw neuronal psth, which can be changed at any time.

### 3.9.4 Goodness of fitting

**Display Predicted PSTH** shows a visual comparison of the predicted and actual responses using validation datasets. Goodness of fitting will give you a quantitative comparison.

To quantify the goodness of fit of the estimated STRF, STRFPAK implements two measures: the coherence and the correlation coefficient. The

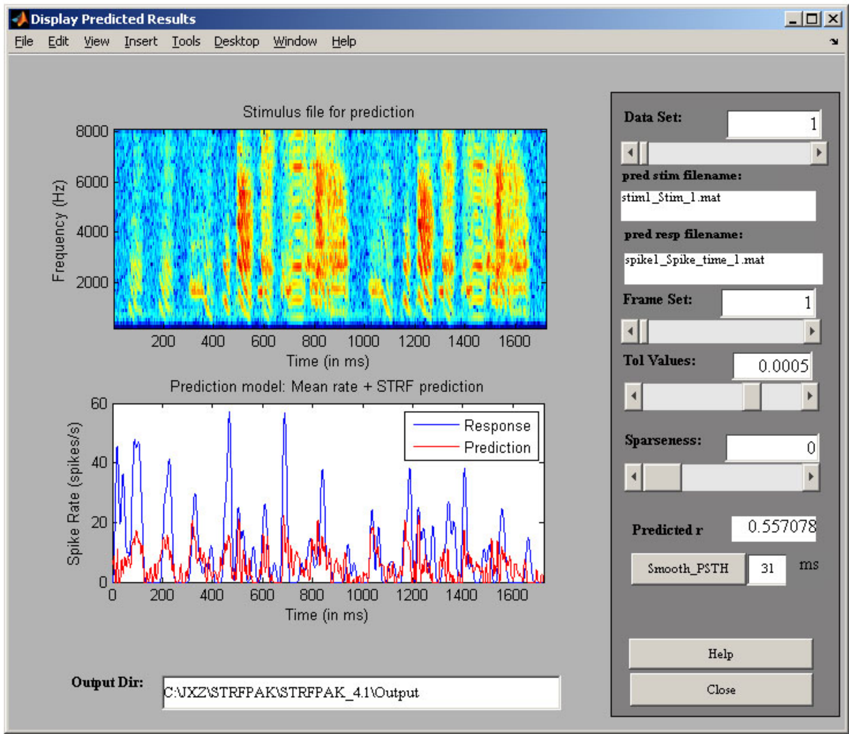


Figure 3.20: Display predicted results from validating STRFs

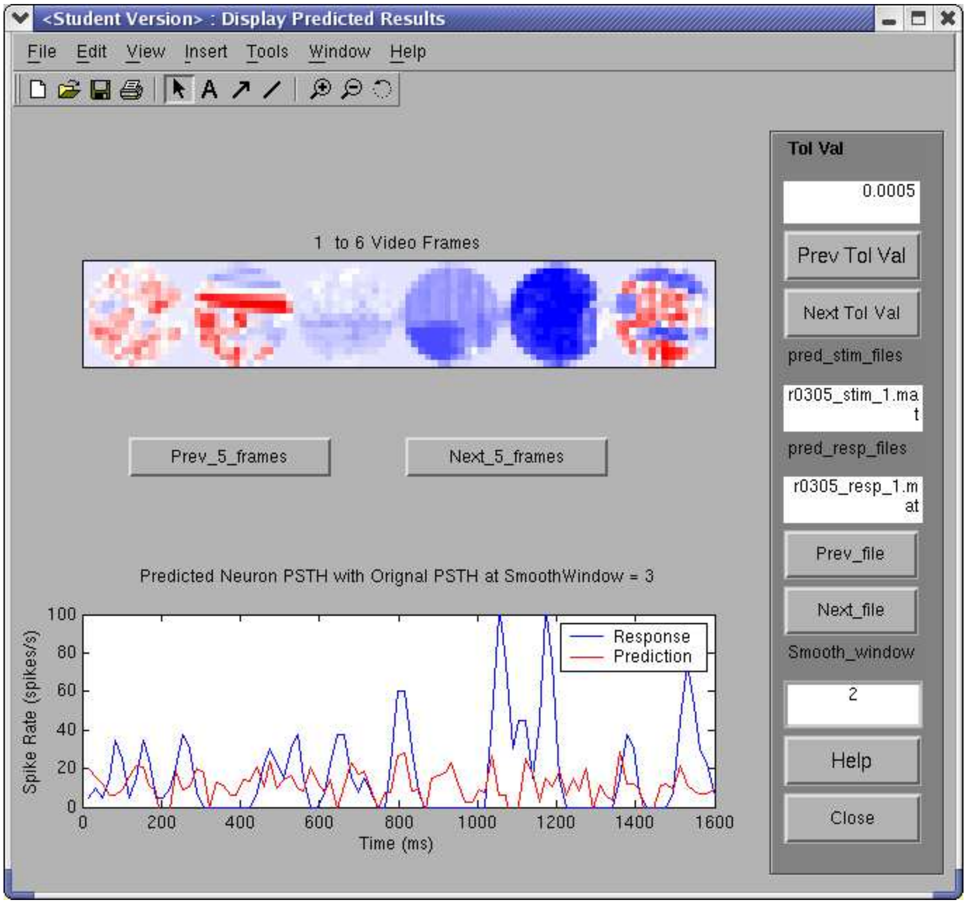


Figure 3.21: Display prediction results for the visual example

coherence is a function of frequency and is given by:

$$\gamma^2(\omega) = \frac{< R(\omega)\hat{R}(\omega)^* >< R(\omega)^*\hat{R}(\omega) >}{< R(\omega)R(\omega)^* >< \hat{R}(\omega)\hat{R}(\omega)^* >}$$

Here $R(\omega)$ and $\hat{R}(\omega)$ are actual and predicted neuron responses at each temporal frequency, $\omega$. An overall goodness-of-fit estimate, $I$, is obtained by integrating the coherence function. $I$ can be thought of as the mutual information between the prediction and the response, and is often called info for short. The lower bound of $I$ is obtained if the noise follows a Gaussian distribution and the upper bound is obtained if the neuron response is a Gaussian.

The correlation coefficient($cc$) between $r(t)$ and $\hat{r}(t)$ is calculated based on:

$$cc = \frac{< (r(t) - r\bar{(t)})(\hat{r}(t) - \hat{r}\bar{(t)}) >}{\sqrt{< (r(t) - r\bar{(t)})^2 >< (\hat{r}(t) - \hat{r}\bar{(t)})^2 >}}$$

Here $r(t)$ and $\hat{r}(t)$ are actual and predicted neuron responses. Since $cc$ depends on the time bin that is used to obtain $r(t)$ from the PSTH, STRFPAK only generates a $cc$ between similar time windows. When all calculations are done, a small "Done Prediction" window appears.

The details of how to measure the goodness of fit are beyond the scope of this manual. STRFPAK uses an unbiased measure of how well the prediction would match the true PSTH given infinite data size. To do this, STRFPAK compares how well one spike train is expected to fit the extrapolated infinite data PSTH versus how well one spike train is expected to fit the prediction. The first of these is the "r" displayed in the cc window, the second is the "r predicted". The ratio of "r predicted" to "r" is called the cc ratio. A perfectly linear neuron characterized perfectly by a STRF should have a cc ratio of 1.

The cc ratio will be a function of the smoothing window size. Depending on the researcher's perspective, it may be most appropriate to look at the cc ratio of the cells at the smoothing window giving the maximum cc ratio, or it may be better to impose a fixed smoothing window length so that comparisons across many cells are more fair. STRFPAK calls the first $max\_CC\_ratio$ and the second $const\_CC\_ratio$. The built-in constant smoothing width used bt STRFPAK is 21 ms.

### 3.9.5 STRFPAK's smoothing window information


Figure 3.22: STRFPAK: Hanning window range for correlation coefficient

Reference **??** gives more detailed explanation about why we have to smooth raw spike data while validating the goodness of fit using correlation coefficient. For flexibility, STRFPAK provides an option for you to choose different smoothing window sizes in **STRFPAK: Hanning window range for correlation coefficient** shown in figure 3.22. That window is pretty self-document. It includes **Smallest smoothing window width (in ms)**, **Smoothing window width step (in ms)**, **Largest smoothing window width (in ms)**, and **Fixed smoothing window width (in ms)**. The default values are 5, 6, 100 and 21. Then you have 16 smoothing windows from 5 to 95. To set the right range of smoothing window, you need check **Display CC**. If you got the maximum *cc_ratio* around window boundary, you need either low the smallest window or enlarge the largest window.

### 3.9.6 STRFPAK: Computing goodness of fit and display window

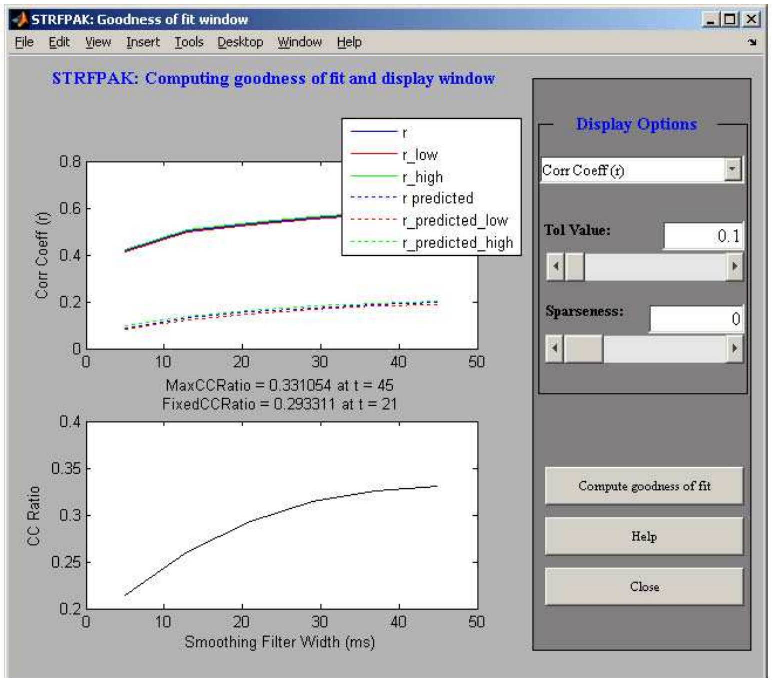


Figure 3.23: Predicted cc and cc for the auditory example

Once the Goodness of fitting button is clicked, the **STRFPAK: Computing goodness of fit and display window** shown in figure ?? pops up. The right panel includes **Display Options**, Compute goodness of fit, Help and Close. If you have not computed goodness of fit yet, you will get error message when you choose any values in display option panel. The layout of the window is the same as before. The left panel shows the graphical display and the right panel shows all the related information about the figures on the left. The current version provides three display options. They are:

- **Corr Coef (r)**: Figure 3.23 is an example of this option. In the left panel, the top figure is a plot of the original and predicted correlation coefficient as a function of smoothing window width. The original correlation coefficient an estimate of the correlation expected between one

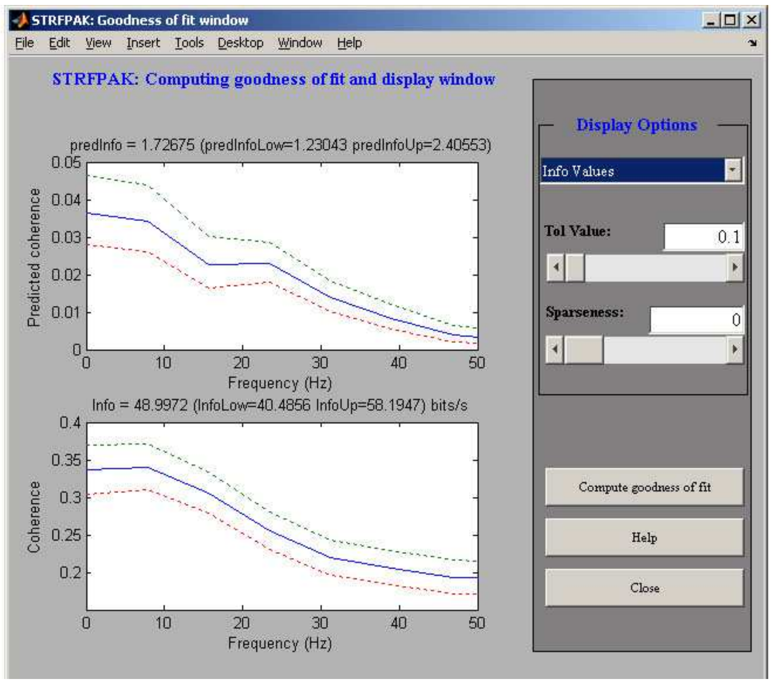


Figure 3.24: Predicted coherence and coherence for the auditory example

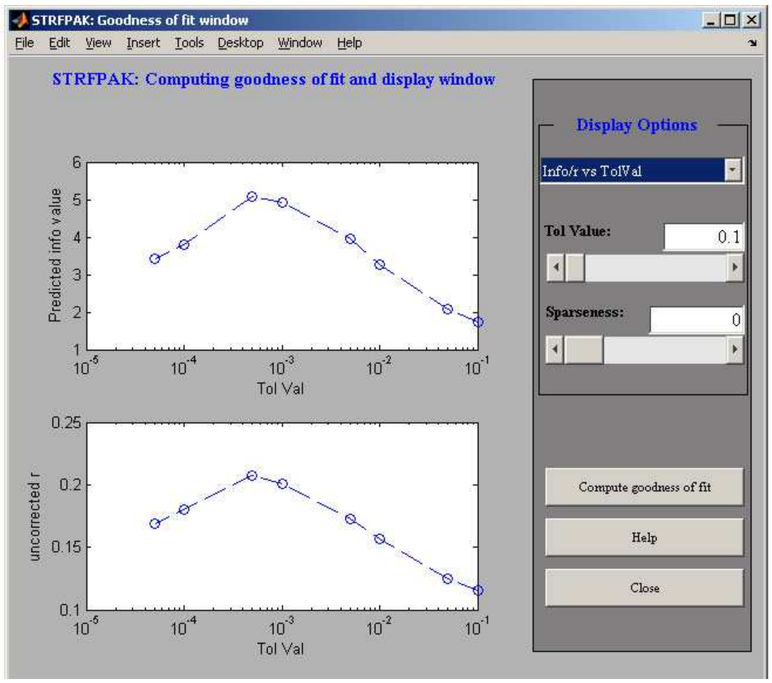


Figure 3.25: Predicted info value vs Tol values and predicted r vs Tol values

spike train and the PSTH given infinite data. The bottom figure is the **CC_Ratio**, which is the ratio of the predicted $r$ and the original $r$. The right panel shows the $Tol\ Val$ of the current STRF used for prediction. The slider can help you to display the results generated from particular $Tol\ Val$. $Sparseness$ parameter is a hyperparameter for smoothing the estimated STRF using Jackknifed STRF's Std value. Again the text field shows $Sparseness$ values and the slider can help to choose different values.

- **Info Values**: Figure 3.24 is an example of this option. In the left panel, the top figure is a plot of the predicted coherence as a function of frequency. **Info** values are obtained by integrating coherence over the whole frequency range. The bottom figure is the plot of raw coherence as a function of frequency. The red and green lines in the plots refer to the upper and lower bounds of the estimated coherence.

- **Info/r vs TolVal**: Figure 3.25 is an example of this option. This option gives you some ideas about how these two validation methods differ. The plots are in log scale.

### 3.9.7 Validation Reference in STRFPAK

Please see the following for a more complete explanation of the goodness of fit measures used in STRFPAK.

- Hsu A, Borst A and Theunissen F E (2004), "Quantifying variability in neural responses and its application for the validation of model predictions", Network: Comp. In Neural Systems 15, 1-19.

## 3.10 Predict

In this stage, STRFPAK first help to choose the best estimated STRF. The best STRF has the highest predicted information value. Then later you can use this best STRF to predict on any new stimulus. The current limitation is the new stimulus that has been already preprocessed and used the same preprocessed methods as used for the estimation data sets.

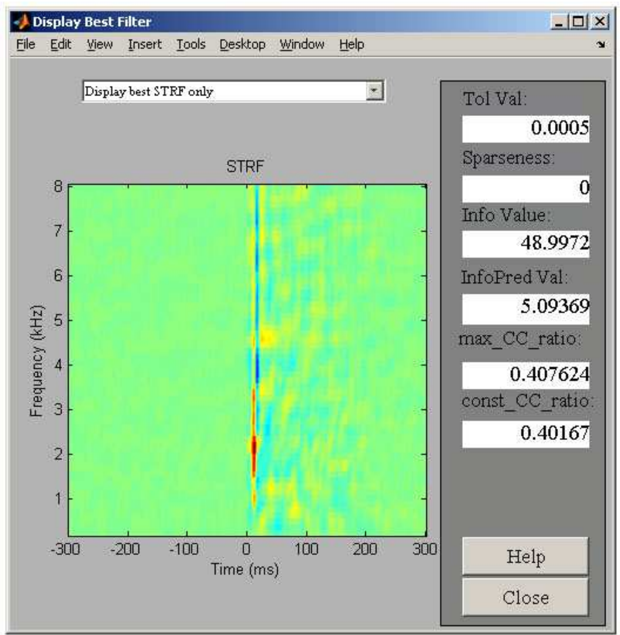


Figure 3.26: Best estimated STRF for the auditory example

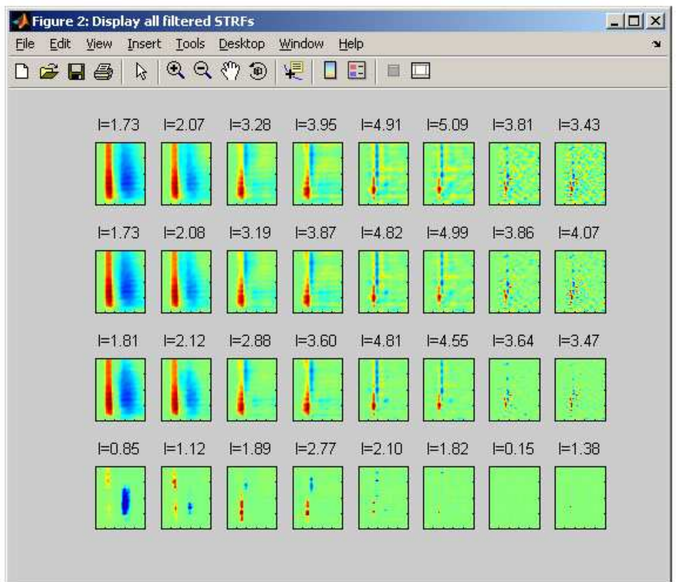


Figure 3.27: Display all estimated STRFs with their predicted information values

### 3.10.1 Display BestStrf

Once the Display BestStrf button in the main window is clicked, STRFPAK does the following three things: sorting the estimated STRFs based on the predicted information values; choosing the best STRF with the largest predicted information values; and displaying the best STRF in the **Display Best Filter** window. The current version of STRFPAK can also display all the estimation STRFs or filters with their predicted information values. Figures 3.26 and 3.27 show the best estimated STRF and all the estimated STRFs with their predicted info values. The right panel in the window shown in figure 3.26 gives the tolerance value used for this STRF, the information value from the actual data, the predicted information value of the best estimated STRF, the *max_CC_ratio* and the *const_CC_ratio* (see above, under the Validation section). If the spatial domain of the stimulus file is $2D$, the best filter is shown as a list of $2D$ video frames. If Help button is clicked, a help window appears.

### 3.10.2 Get New Stim Only

The reason why STRFPAK offers these last two buttons is because you may want to use the estimated STRF to do prediction on totally new stimulus, for example, for the neural prediction challenge project. For more information about that project, please go to its website *http://neuralprediction.berkeley.edu*. If you click this button, STRFPAK then ask you to load new stimulus. Here only stimulus is needed. Right now STRFPAK only accepts already preprocessed stimulus and the preprocessed stimulus has to be used in the same preprocessing method used for preparing the estimation data sets. The layout of this window in figure 3.28 is very similar to Load Data. If you have any questions, please refer to Load Data section.

### 3.10.3 Predict and Display Result

This option helps you to simulate the neuron prediction. After you load a new stimulus, you can get new response using the estimated STRFs. STRFPAK also provides you two choices: using the best STRF only or using all the estimated STRFs. After your preference, the **Predict and Display Result** window shows up. If you choose the best STRF, the right figure in the right shows the best STRF. After you click Predict, the prediction progress bar

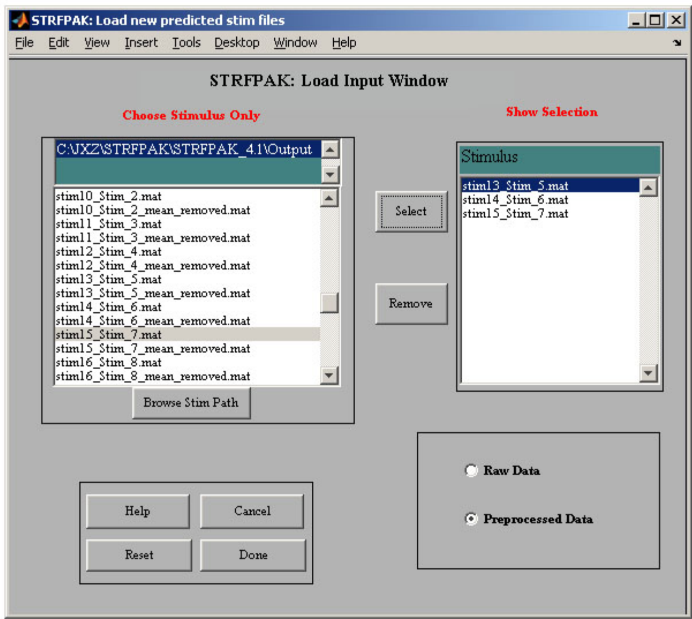


Figure 3.28: STRFPAK: Load new stimulus only for prediction

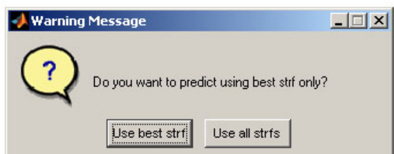


Figure 3.29: Using the best STRF or using all to predict?

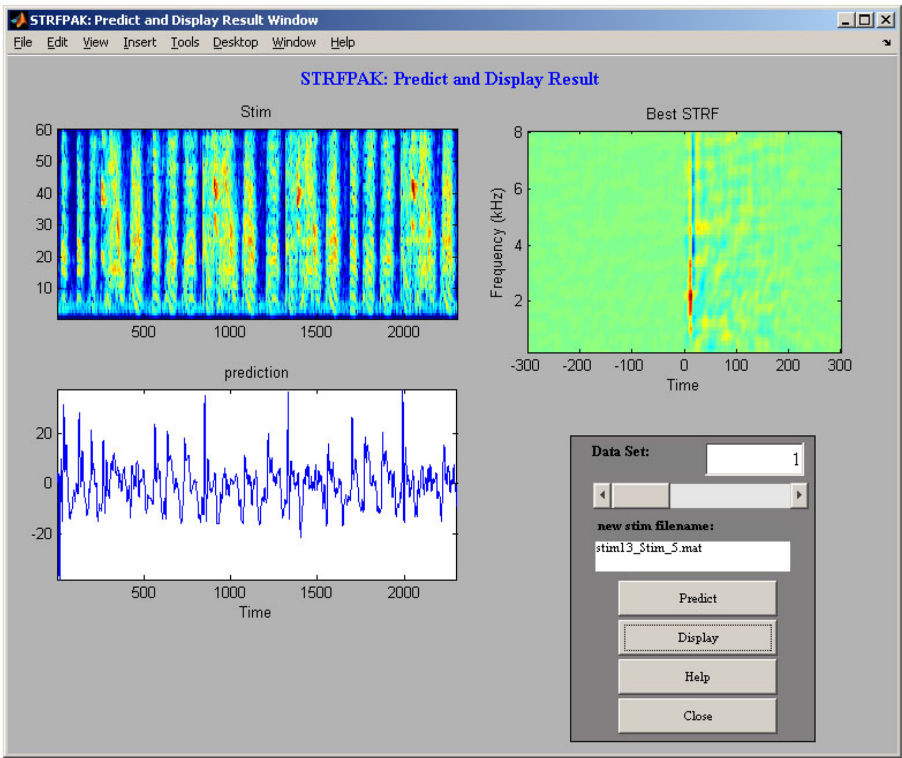


Figure 3.30: Predicted neuron response with its stimulus and its STRF

pops up. After prediction is done, you can click Display to show the stimulus and STRFPAK's prediction in the left. If you choose using all the STRFs, after you click Display, the left bottom also shows $Tol\_val$ and $Sparseness$ parameters. You can use sliders to choose different STRFs and then need click Predict and Display buttons again to update your prediction and display.

## 3.11 Load Prev Result

From the main window, there is a button called, located in the right bottom corner. This button loads and displays results for data sets on which STRFPAK has already run. Load Prev Result will ask you to choose a directory in which results are stored, and load as much information about a previous STRFPAK session as possible. You can then resume displaying previous results.

## 3.12 STRFPAK: Caching Option

This window asks you to input **Disk Cache Directory**, e.g. $C : Temp\_Cache$, and **Disk Cache Size (GB)**, e.g. 10. For the reason why introducing cache option and how it works, please click Help button from the window. The bigger disk cache size is, the better the performance will be. However, if you set it too big, it may also slow the computation. The best size is 1 or 5 GB. Once you provide all the cache information (directory and size), STRFPAK records them into a file, **dir_of_cache.mat**. So you will not need to input them again when you start to run STRFPAK next time unless you click Clear Cache from the main window.

## 3.13 Clear Cache

Clear Cache removes everything saved under **Disk Cache Directory** and delete a cache file **dir_of_cache.mat**.


Figure 3.31: STRFPAK caching option

# Chapter 4

# Batch-mode/Nongraphical processing option

For a large sets of data, you may want to run STRFPAK-4.1 in a batch-mode. The easiest way to do this is as follows: First, run one cell from your data set with the graphical version of STRFPAK. As well as the result files like **strfResult.mat** containing the strfs and **info_r_result.mat** containing their goodness of fit, a file called **STRFPAK_script.m** will be saved in the output directory, containing a script of all the actions performed by the graphical version of STRFPAK. Edit it (Note: you may need add preprocessing routines) and embed it into a loop as you would any other Matlab script file for a fast batch mode. Remember that the output directory for each STRFPAK run should be unique to avoid having STRFPAK overwrite previous results.

Also provided are template batch-mode codes, *STRFPAK_batch_template.m* and *STRFPAK_core.m*. All the required parameters need specified in *STRFPAK_batch_template.m*. If you have different data layout from demo data, you also need modify *STRFPAK_core.m*. Feel free to email Junli (junli@socrates.berkeley.edu) if you run into any problems.

# Chapter 5

# Summary and Things to do

We have developed the second version of the spatio-temporal receptive field estimation software package, STRFPAK. This software is now available on-line at http://strfpak.berkeley.edu. It estimates stimulus-response transfer function of a sensory neuron. The resulting spatio-temporal receptive field provides a quantitative description of the transformation between a time varying spatial stimulus and the neural response, which can be used in subsequent computational modeling studies. In the current version, we have implemented a generalized reverse correlation technique and the Jackknifed error estimation algorithm to calculate the linear spatio-temporal receptive field. Two different measures, coherence and correlation coefficients, that quantify the estimated STRF's goodness of fit are also included in this version. We have also developed a graphic user interface with tutorial examples and help documents. STRFPAK is implemented using Matlab programming language and organized as a Matlab tool box. It has been tested on Unix, Linux and Windows.

For future work, there are a number of areas proposed here:

- **More raw data format support**: To allow more experimental data formats.

- **More preprocessing options**: There are many more preprocessing options (routines which convert a raw stimulus format, such as a .wav file, into a format better for obtaining a STRF) that could go in here, such as the Fourier Power method, the Complex Wavelet method and Lyons cochlea model. Most of these we already have implemented, but we have not yet have time to embed them into the toolbox.

- **Nonlinear estimation models to be added**: Nonlinear estimation techniques such as a multi layer Neural network, Automatic relevance determination (ARD) and a thresholding algorithm are being developed and tested. They also will be embedded into the toolbox.

- **More post-precessing options**: There are many interesting possibilities for analysis of the results. We want to expand post-processing algorithms. If you have any interesting ideas, please let us know.

# Appendix A

# Glossary

- **STRFPAK**: STRFPAK is the nickname of our Spatial Temporal Receptive Fields Estimation Software Package.
- **Preprocessing**: A transformation of the stimulus to put it into a form more amenable to computing a STRF. For example, calculating a spectrogram from a .wav file is a way of preprocessing data into a format more amenable to STRF calculations.
- **Post-processing**: Any calculation that comes after the estimation of the STRF, the prediction of neural response and the validation of the STRF. It usually consists of some form of analysis of the results sets, such as clustering of all the result parameters.
- **Prediction**: One application of the estimated STRF is to predict the neural response with different stimuli data provided.
- **Validation**: The stage after prediction, which measures how good the estimated STRF is.
- **Experiment**: A term used to describe the entire set from which results are derived. An experiment can consist of several trials over different days, each of which may have several runs.
- **Frame**: The 2D spatial data available at each time-point.

# Appendix B

# A list of Main Functions

The following lists the main functions and script programs available in STRFPAK-4.1.

```
Main files
----------------
1. Core Calculation Functions:
----------------
cal_AVG.m          : Calculates the average of the stimuli, the
                     response, and the PSTH of the cell.
cal_AutoCorr.m     : Calculates the autocorrelation of the stimuli.
cal_AutoCorrSep.m  : Calculates the autocorrelation of the stimuli
                     using an assuming space and time are separable.
cal_CrossCorr.m    : Calculates the cross-correlation of stimuli
                     and responses.
cal_Strf.m         : Calculates the STRF.
cal_StrfSep.m      : Calculates the STRF using a separable algorithm.
calStrf_script.m   : Calcualtes the STRF and normalizes.
calStrfSep_script.m: Calcualtes a separable STRF and normalizes.
cal_PredStrf.m     : Predicts a PSTH using the estimated STRF and
                     new stimulus.
cal_Validate.m     : Validates predictions.

2. Display Functions:
-----------------
displayinput_GUI.m      : Displays preprocessed data.
```

```
displaystimstat2d_GUI.m : Displays the stimulus-response
                          statistical results.
displaystrf_GUI.m       : Displays all the estimated STRFs and
                          associated post-processed results
displaypredstrf_GUI.m   : Displays the predicted PSTH with its
                          associated stimulus.
displayinfocc_GUI.m     : Displays cc values and mutual information
                          between predicted and actual PSTH.
displaybeststrf_GUI.m   : Extracts the best STRF and displays it
                          with its associated info value.

3. Preprocessing Functions:
-----------------
songwave_spectrum.m   : Calculates a spectrogram using a Short-time
                        Fourier Transformation
wavelet1d_scalogram.m : Calculates a scalogram using a Morlet Wavelet
                        Transformation.

4. Batch-mode Functions:
-----------------
STRFPAK_cluster_template.m: Sets up all the parameters and data paths
                            for core computing.
STRFPAK_core.m            : Script performing estimation, prediction and
                            validation.  Intermediate results also saved.

Main intermediate output file list
-----------------
Stim_autocorr.mat           : stimuli autocorrelation matrix
StimResp_crosscorr.mat      : stimuli-response crosscorrelation vector
SR_crosscorrJN.mat          : Jackknifed crosscorrelation
strfResult.mat              : the STRF matrices
predResult-EstSpike_Tolx.mat: predicted psth for tolrence value x
predResult-avgSpike_Tolx.mat: original psth used for prediciton at tol x
info_r_result.mat           : validation results
```

# Appendix C

# Quick Guide to Run STRFPAK

```
Quick Installation about STRF package
---------------------------------
1. Open Matlab first.

2. On the MATLAB command prompt line, type:
   >> strfpak

3. Click each button in order on the main window
   of STRFPAK to get input data, estimate the STRF,
   predict a PSTH using the STRF and validate the
   goodness of fit.

4. All the intermediate results are saved in the output
   directory you specified under the "Calculate" button
   on the main window.
```

# Bibliography


[1] Theunissen FE,David SV, Singh NC, Hsu A, Vinje W and Gallant JL, "Estimating spatio-temporal receptive fields of auditory and visual neurons from their responses to natural stimuli", Network: Computation in Neural Systems 12 (2001) 289-316.

[2] Hsu A, Borst A and Theunissen F E (2004), "Quantifying variability in neural responses and its application for the validation of model predictions", Network: Comp. In Neural Systems 15, 1-19.

[3] Jones JP, Stepnoski A, Palmer LA (1987), "The two-dimensional spectral structure of simple receptive fields in cat striate cortex", J Neurophysiol, 58:1212-1232.

[4] Ringach DL, Sapiro G, Shapley R (1997), "A subspace reverse-correlation techniques for the study of visual neurons", Vision Research 17:2455-2464.

[5] Klein DJ, Depireux DA, Simon JZ, Shamma SA (2000), "Robust spectro-temporal reverse correlation for the auditory system: optimizing stimulus design", J Comput Neurosci 9:85-111.

[6] F.E. Theunissen, K. Sen and A.J. Doupe, *Spectral-temporal receptive fields of nonlinear auditory neurons obtained using natural sounds*, J. Neurosci., 20 (2000) 2315-31.

[7] Singh NC and Theunissen FE, "Modulation spectra of natural sounds and ethological theories of auditory processing", J Acoust Soc Am 2003 Dec, 114(6 Pt 1): 3394-411.

[8] D.J. Thomson and A.D. Chave, *Jackknifed error estimates for spectra, coherence and transfer functions*, Advances in Spectrum Analysis and

Array Processing vol 1, ed S Haykin (Upper Saddle River, NJ:Prentice-Hall).

[9] Willmore B, Smyth D (2003), "Methods for first-order kernel estimation: simple-cell receptive fields from responses to natural scenes", Network: Comput Neural Syst 14:533-577.

[10] Prenger R, Wu MCK, David SV, Gallant JL (2004), "Nonlinear V1 Responses to Natural Scenes Revealed By Neural Network Analysis", Neural Networks (In Press).

[11] David, S. V.,Vinje, W. and Gallant, J. L., "Natural Stimulus Statistics Alter the Receptive Field Structure of V1 Neurons", (in progress).

[12] Gill P, Zhang J, Theunissen FE, "Physically driven time-frequency representation of natural sounds in the study of spatio-temporal receptive fields", (in progress).

[13] P. Dayan and L.F. Abbott, *Theoretical Neurosciense: Computational and Mathematical Modeling of Neural Systems*, The MIT Press, Cambridge, Massachusetts, 2001.

[14] P Marmeralis and V. Marmeralis, *Analysis of Physiological Systems: The White Noise Approach*, New York: Plenum, 1978.

[15] Lars Kai Hansen, Finn rup Nielsen, Peter Toft, Matthew Liptrot, Cyril Goutte, Stephen C. Strother, Nick Lange, Anders Gade, David A. Rottenburg, Olaf B. Paulson, "Lyngby" - A modeler's toolbox for spatio-temporal analysis of functional neuroimages, NeuroImage 9 (6 part 2):S241, 1999.

[16] F. Auger, P. Flandrin, P. Goncales and O. Lamoine, *Time-Frequency Toolbox*, http://www-syntim.inria.fr/fractales/Spftware/TFTB.