# STRFPAK: A Spatio-temporal Receptive Field Estimation Software

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

Version 2.0 - May 12, 2004

# 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 2.0 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 SunOS 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 strfpak
```

Any reference to a particular [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 following website, you can download the Unix/Linux version and Windows version compressed tar files that contain all the Matlab files and sample data sets.

`http://strfpak.berkeley.edu/download.html`

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

`strfusers@socrates.berkeley.edu`

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

## 1.3 Contact Information

Besides the STRFPAK mailing list,

`strfusers@socrates.berkeley.edu`

we can also be contacted by email directly:


```
Junli Zhang (Programmer) junli@socrates.berkeley.edu
Frederic Theunissen (PI) fet@socrates.berkeley.edu
Jack Gallant (PI) gallant@socrates.berkeley.edu
```

# 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 3.1 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 the user 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. The user can go to the STRFPAK website

`http://strfpak.berkeley.edu`

to get the source code.

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

```
gzip -dc STRFPAK-2.0.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 Starting STRFPAK

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.

## 3.3 Getting Help

An overview of all the functions in the STRFPAK toolbox can be obtained through the standard Matlab help function:

```
>> help strfpak
```

You can also get help for the parameters or specific data formats by clicking the Help button on each STRFPAK GUI window.

## 3.4 STRFPAK's Main Window

STRFPAK's main window, **STRF Estimation Software**, that pops-up after clicking agreement to the STRFPAK copyright window is shown in Figure 3.1. In the window, the main functionalities of the STRFPAK toolbox are clearly shown by its panels' labels: **Get Input**, **Estimate**, **Predict**, and **Validate**. For the first time usage, you need follow these four panels sequentially to go through the complete procedure, which may not be necessary. For example, if you just want to load your data and display them graphically, you can stop after the **Get Input** panel. But if you want to see how good your data can predict the transfer function of the neurons, you need go through the whole procedure from getting input to validating. A detailed description of each button on every panel is given in the following sections.

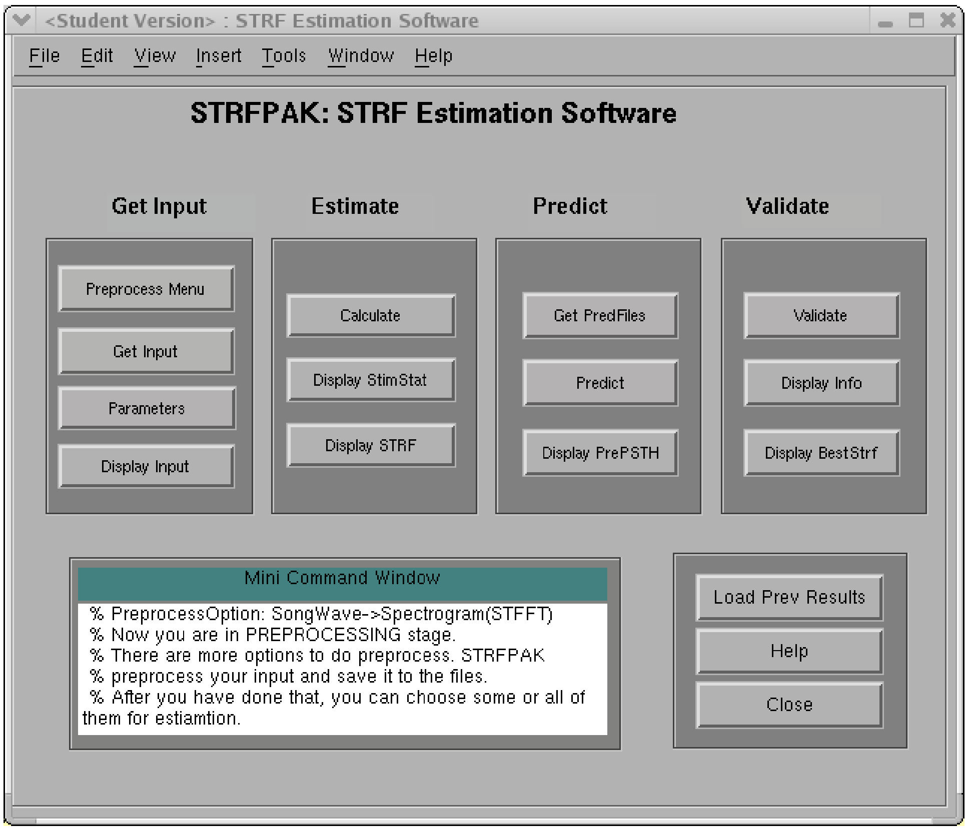


Figure 3.1: The Main Screen of STRFPAK

## 3.5 Get Input

The **Get Input** panel deals with the issues of the data formats and preparation before the estimation stage. It includes loading the data, preprocessing options, showing the preprocessed input data and specifying the calculation parameters.

### 3.5.1 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 |

It is sometimes desirable to preprocess the stimulus before executiong 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 the user 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 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 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.

       size_x_stim = size(stim, 1);
       size_y_stim = size(stim, 2);
```

```
length_stim = size(stim, 3);
desired_stim = reshape(stim, ...
      size_x_stim*size_y_stim, length_stim);
```

### 3.5.2 The Preprocessing Menu

The preprocessing step includes processing raw data files using STRFPAK's preprocessing methods and then preparing the right data formats for the later analysis. Once the Preprocessing Menu button is clicked, the **STRFPAK Preprocessing Menu** window shown in Figure 3.2 pops up. There are two panels in that window. One is for **1-D option** and another is for **2-D option**. Here the dimension refers to a spatial domain. For example, for auditory systems, the 1-D non-time dimension could represent frequency; for visual systems, the 2-D dimensions could represent a 2-D spatial domain. STRFPAK-2.0 provides only two transforms of sound wave files: Short Time Fourier Transform (STFT) and Wavelet Transform. The other transformations proposed on the preprocessing menu are currently under development.

### 3.5.3 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, 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 3.4 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.

- Load files: It is used to load raw data, e.g. sound wave files and their associated response data files into the STRFPAK structure. The **Load file window** shown in Figure 3.3 pops-up after Load files is clicked.
  - **Load stimulus files**: This panel is used for showing all the files under the directory which is shown in the dark green area. It

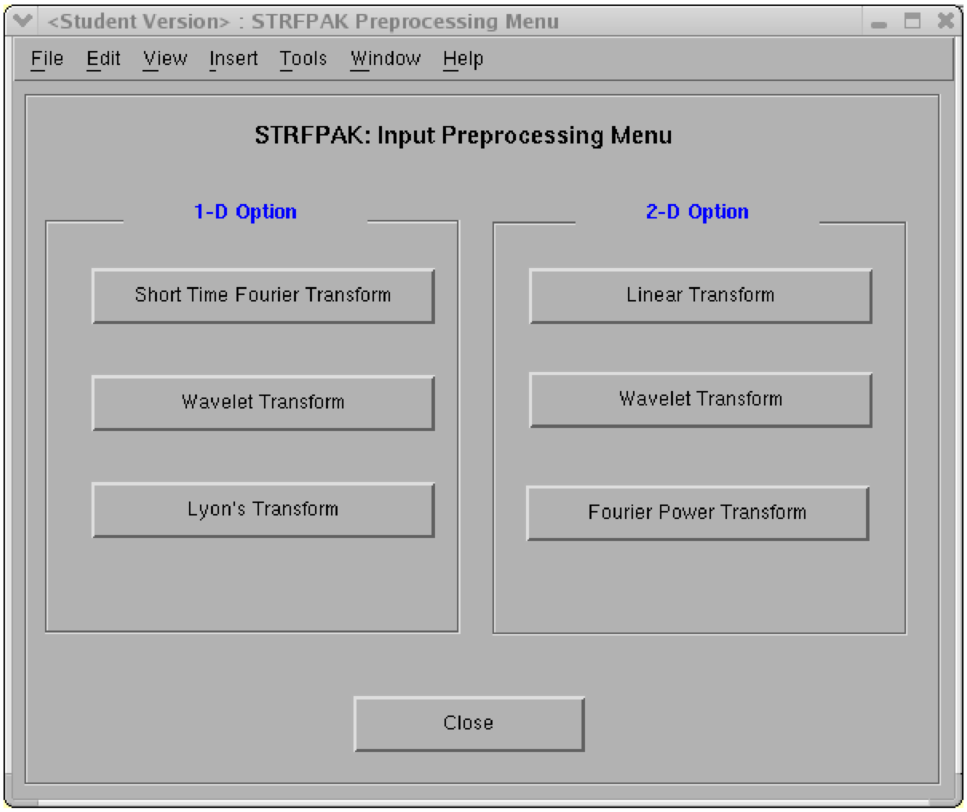


Figure 3.2: The Preprocessing Menu in STRFPAK

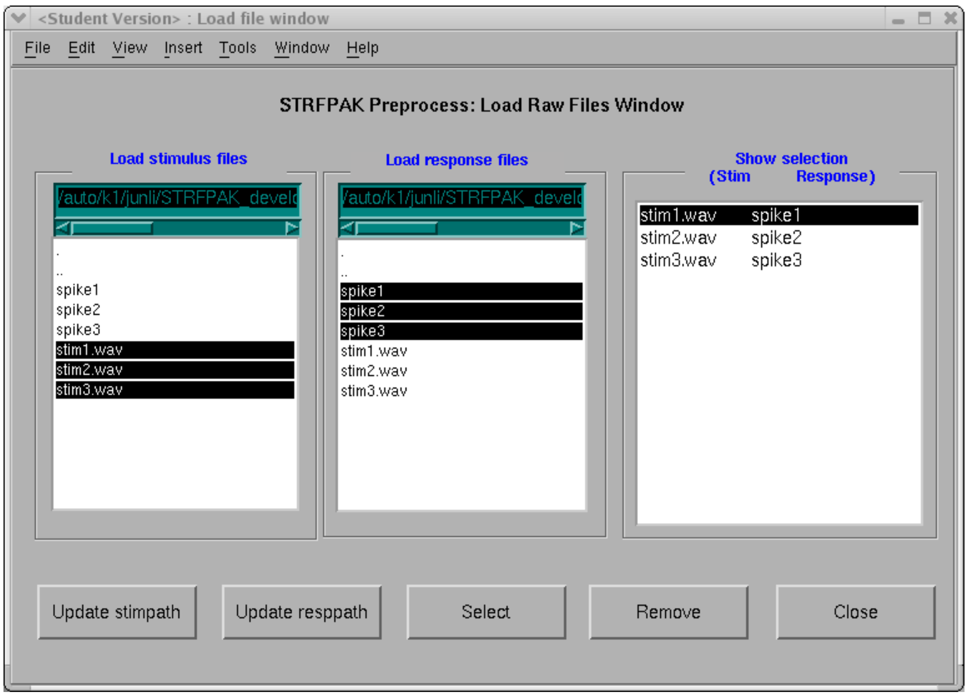


Figure 3.3: The Load file window in STRFPAK

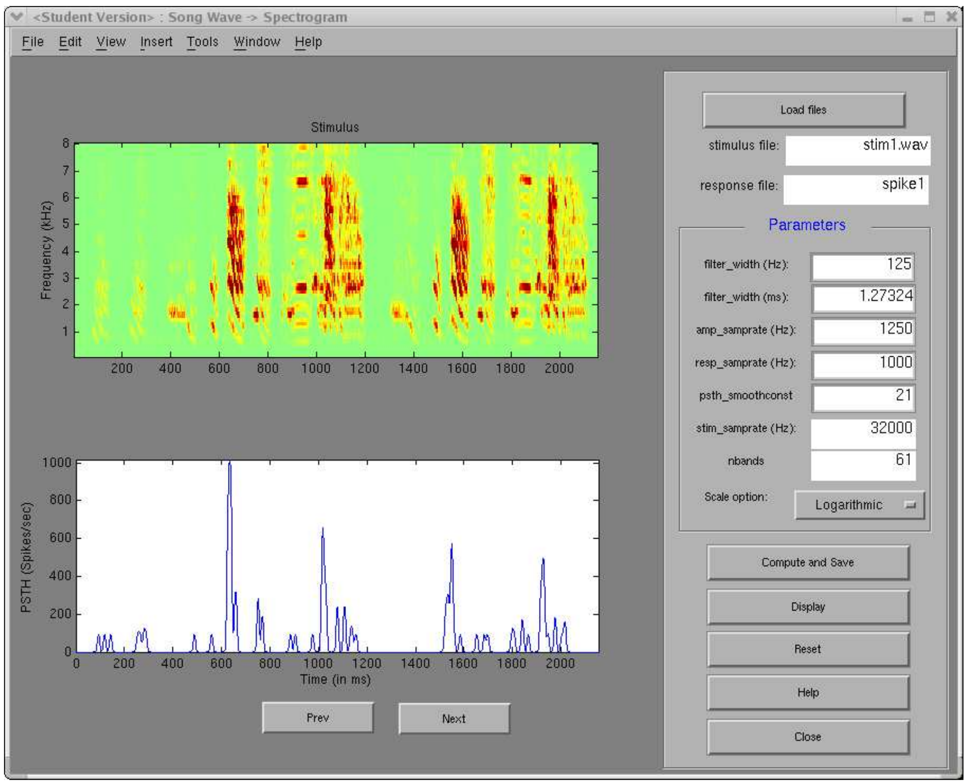


Figure 3.4: The Songwave → Spectrogram window in STRFPAK

may contain the stimulus and response files under that directory. However, you need to select only the stimulus files from this panel. To select multiple stimulus data, hold down the **CTRL** key while clicking your selection.

- **Load response files**: This panel is used for showing all the files under the directory which is shown in the dark green area. It may contain the stimulus and response files under that directory. However, you need to select only the response files from this panel. To select multiple response data, hold down the **CTRL** key while clicking selection.
- **Show selection**: This panel shows what you just selected. It helps you double check your selection. Note the first file in the data pair needs to be the stimulus and the second needs to be its associated response. **The order is very important**.
- Update stimpath: This button helps you to change the directory of the **Load stimulus files** panel.
- Update resppath: This button helps you to change the directory of the **Load response files** panel.
- Select: After you highlight what you want from **Load stimulus files** and **Load response files**, you need click this button to load them into the toolbox and show them in the **Show selection** panel.
- Remove: You can remove any one or more data sets by clicking Remove button from the **Show selection** area.

- **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.
  - **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.
  - **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.

- [Reset]: Resets all the parameters and the data sets chosen.

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

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

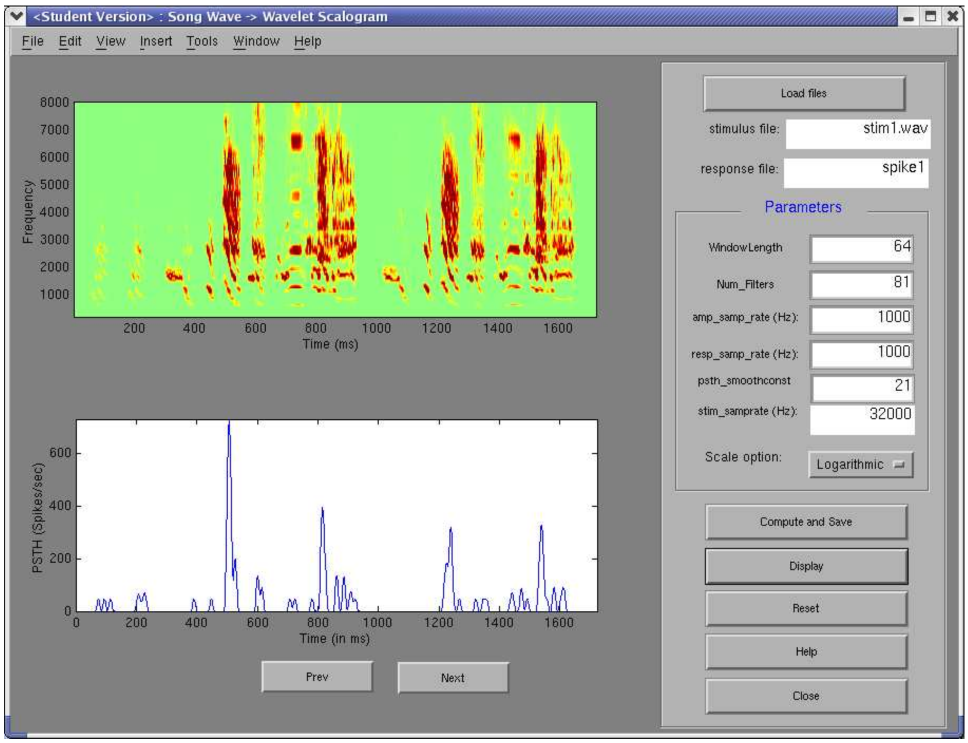


Figure 3.5: The Songwave → Scalogram window in STRFPAK

### 3.5.4 Wavelet Transforms

After clicking the Wavelet Transform button, the **Songwave → scalogram** window shown in Figure 3.5 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** 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 Morlet 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.

- 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, Next and Prev buttons show up so that you can click to see the next data set.

- Reset: Resets all the parameters and the data sets chosen.

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

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

### 3.5.5 Getting Input

If you use STRFPAK's **Preprocessing Menu** options to preprocess your raw experimental data, you will not need to go to this stage. However, if you choose to preprocess your data with a routine outside STRFPAK, you will need to use this window to load up your preprocessed data. These data must be in either ASCII or Matlab binary format with extensions *.txt*, *.dat* or *.mat*. Stimulus data files must contains a spatio-temporal description of a stimulus, $s(x,t)$. Response data files contain multiple trials of spike trains, $r(trialnumber, t)$. For example, in the auditory system, $s(x,t)$ represents the time-varying amplitude of sound in a frequency band centered at $x$. In the visual system, $s(x,t)$ represents the light intensity as a function of position and time. Trials should ideally be much longer than the expected duration of the STRF for best results.

Figures 3.6 and 3.7 show the **Get Datafiles** Windows for the auditory example and the visual example respectively. In this window, the user can choose stimulus and response input from the top left panel of the window.

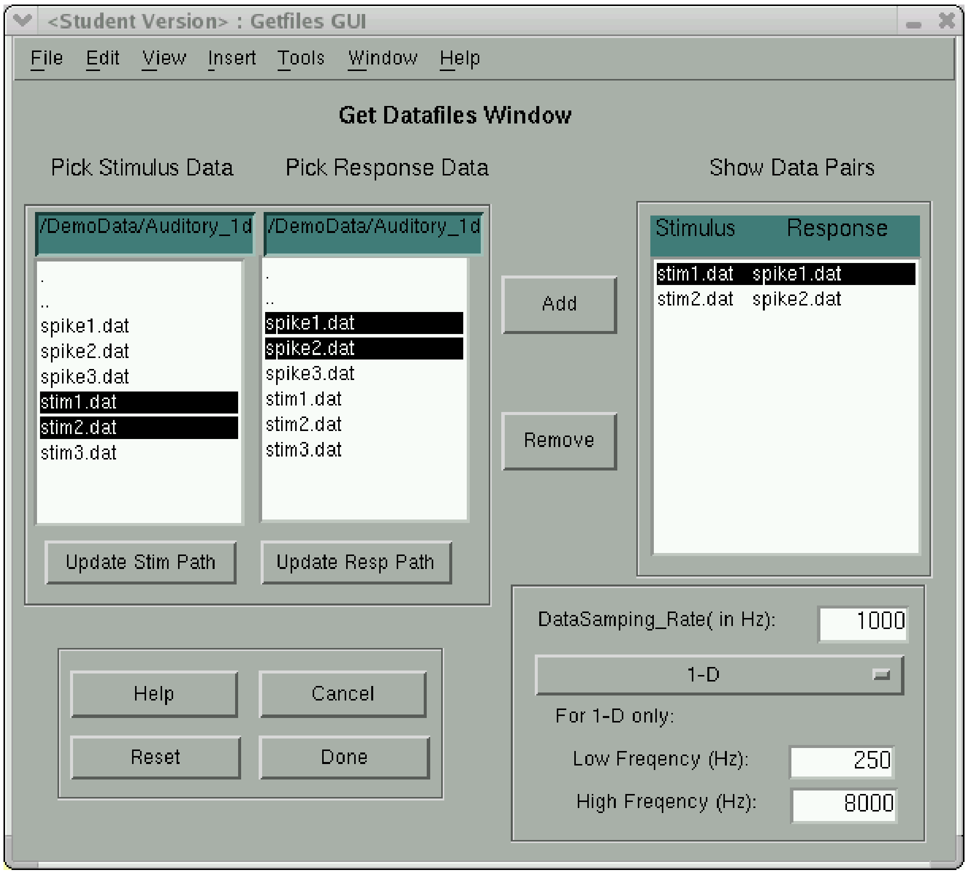


Figure 3.6: Get datafiles window for the auditory example

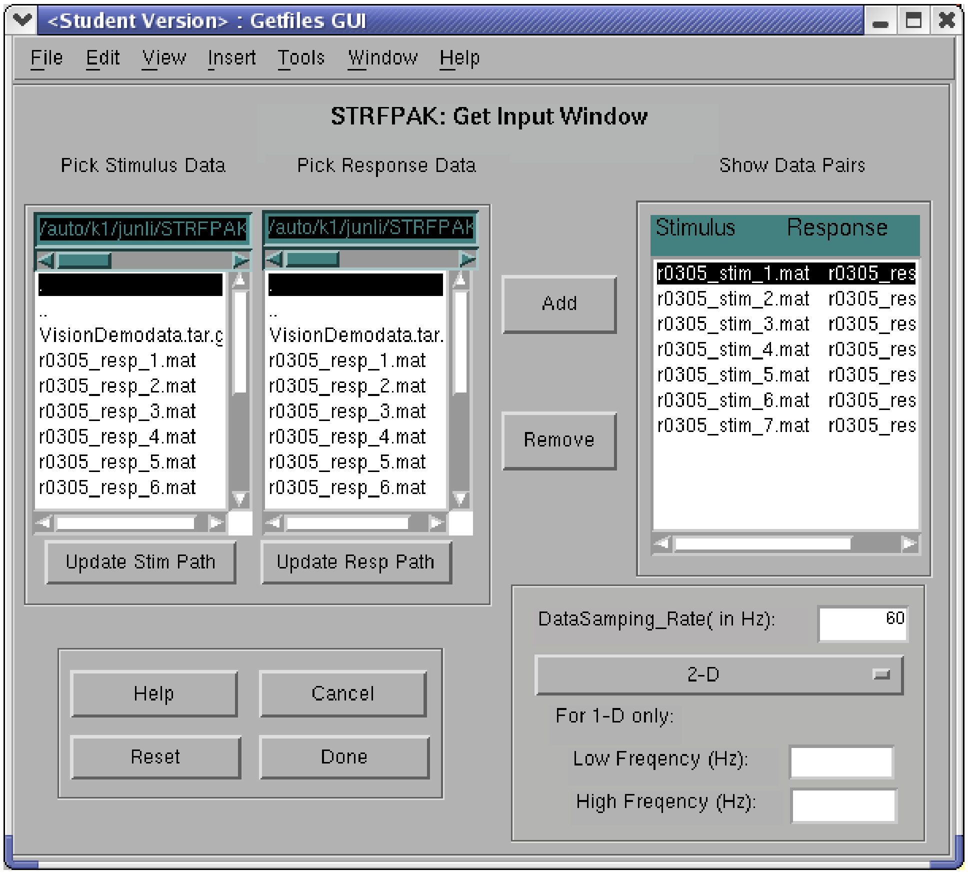


Figure 3.7: Get datafiles window for the visual example

In this panel, there are two list boxes, for stimulus files and response files. The list boxes display the files under current directories of the stimulus and the response. The user can interactively update the stimulus and response directories by clicking the $\boxed{\text{Update Stim Path}}$ or $\boxed{\text{Update Resp Path}}$ buttons. When the user clicks on a list item in the stimulus or the response list box, one of the following happens:

- If the item is a file, the file is selected.
- If the item is a directory, the GUI reads the contents of that directory into the list box.
- If the item is a single dot (.), the GUI updates the display of the current directory.
- If the item is a double dot (..), the GUI changes to the directory up one level and populates the list box with the contents of that directory.

To select multiple data sets, hold down the CTRL or Shift key while clicking selections.

After selecting stimulus files from the stimulus list box and the corresponding response files from the response list box, the user can click the $\boxed{\text{Add}}$ button to check if the file type is allowed. They will show on the **Show Data Pairs** list box if their data format is OK. The user can remove the selected data files by pressing $\boxed{\text{Remove}}$ button.

After the selection has been done, the user needs to specify data parameters in the right bottom panel of the window by clicking the pop-up menu **Please Choose Spatial Domain** and by filling in the editable text field. These parameters include data sampling rate (in Hz), dimensionality of spatial domain, and low and high frequency (in Hz) for auditory STRFs. For the auditory example, the data sampling rate is 1000 Hz, the spatial domain is a frequency band so the dimensionality of the spatial domain is set as $1D$. The low frequency is 250 Hz and the high frequency is 8000 Hz since these are the lower and upper frequencies in the sample spectrograms. For the visual example, the data sampling rate is 72 Hz. This spatial domain is $2D$ since the stimulus is a time-varying two dimensional image.

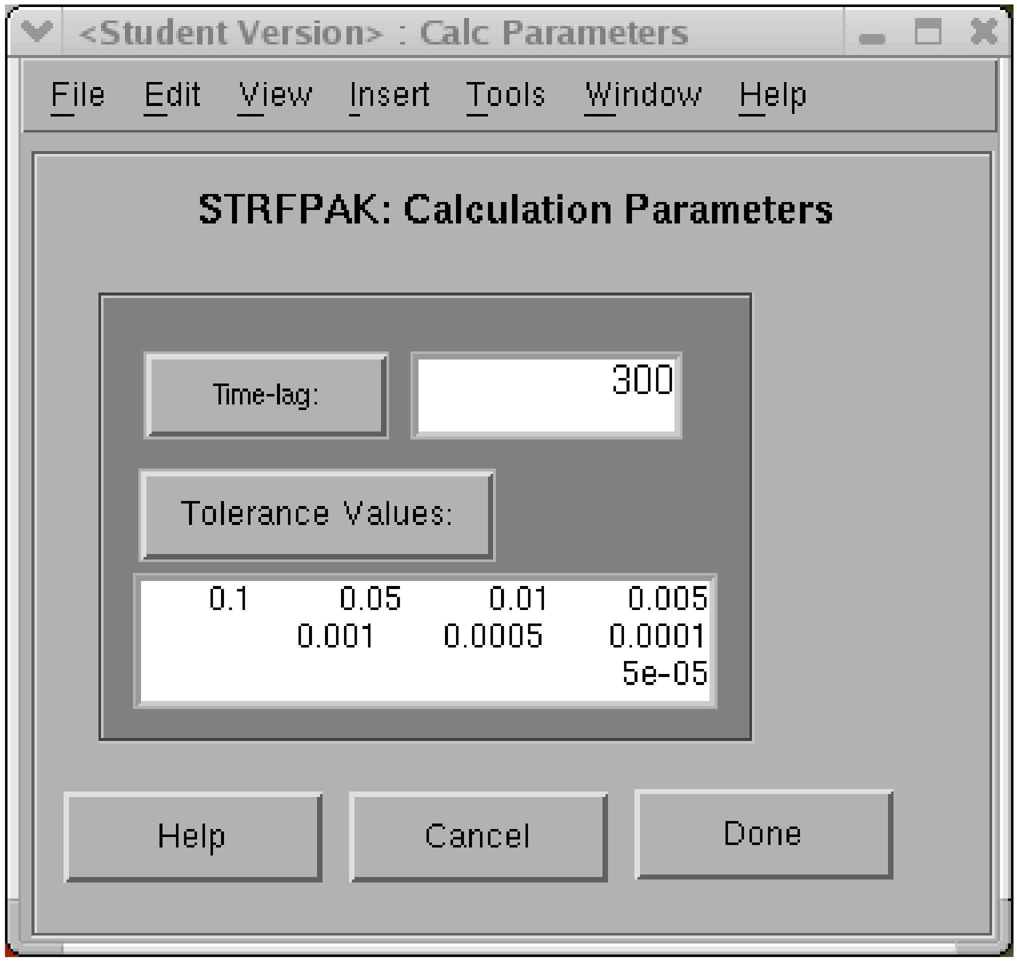


Figure 3.8: Calculation parameter window for the auditory example

### 3.5.6 Parameters

$\boxed{\text{Parameters}}$ in STRFPAK mean the calculation parameters used for the estimation of the STRF. The **Calc Parameters** window is shown in the following figure 3.8. There are two types of calculation parameters required by STRFPAK-2.0: $\boxed{\text{TimeLag}}$ and $\boxed{\text{Tolerance value}}$. Use the help button to reveal the meaning and the format needed for specifying these parameters.

The **TimeLag** parameter controls the magnitude of the maximum temporal duration of the STRF. STRFPAK assumes values of the STRF more than TimeLag bin sizes away from simultaneity are 0. For example, since the bin size for the above auditory data is 1 ms, the time course covers a period of ±300 ms if we set **TimeLag** as 300. For the visual example, the bin size is 14 ms, the time course covers a period of ±70 ms if we set **TimeLag** as 5. If a larger range is needed, **TimeLag** can be increased.

STRFs with large TimeLags or large-dimensional stimuli will have so many free parameters that a form of regularization is desired. STRFPAKs regularization works by projecting the STRF onto a subspace spanned by the strongest few eigenvectors of the stimulus autocorrelation. **Tolerance value** controls how many eigenvectors are included in this subspace. All eigenvectors with a corresponding eigenvalue less than a given tolerance value are set to zero in the estimated STRF. Since it is difficult to guess what the tolerance value should be for any cell *a priori*, and since estimating and validating a STRF at several tolerance values is computationally easy, a list of trial tolerance values are typically used. In the above examples, we have set trial **Tolerance value**s to 0.1 0.05 0.001 0.0005.

### 3.5.7 Displaying Input

For convenience, STRFPAK provides graphical display of the input data in the **Display Input** window. You can check whether the selected input data will work for later estimation and analysis. There are two panels in the window: the left one shows the stimulus and response and the right one contains an information menu. Figure 3.9 shows the spectrographic representation of a stimulus, 10 trials of neuron response and its post-stimulus time histogram (PSTH) for the auditory example included. Figure 3.10 shows the first 12 frames of a natural scene used in the above visual example, one single trial (since we have only one trial here) and its PSTH.

In the figures, the left panel of the window is a graphical display of the

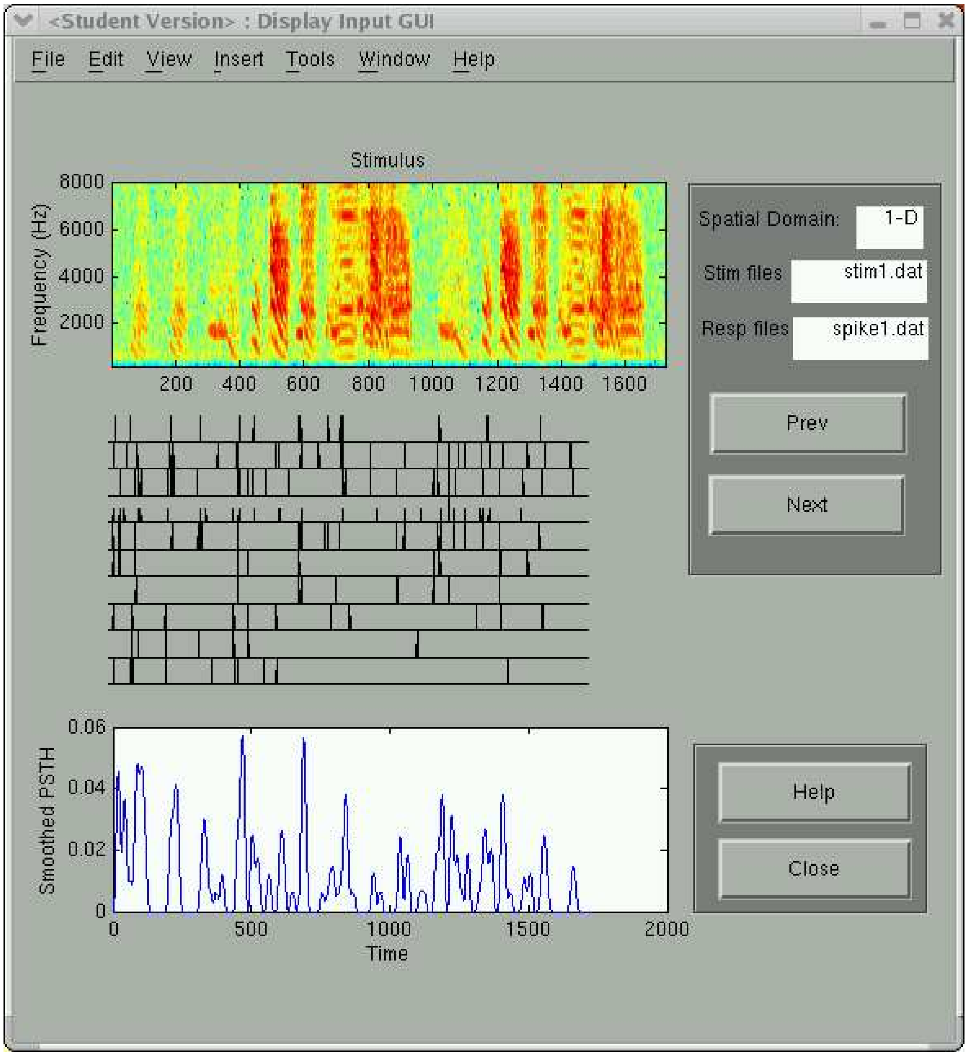


Figure 3.9: Sample of an auditory stimulus-response pair

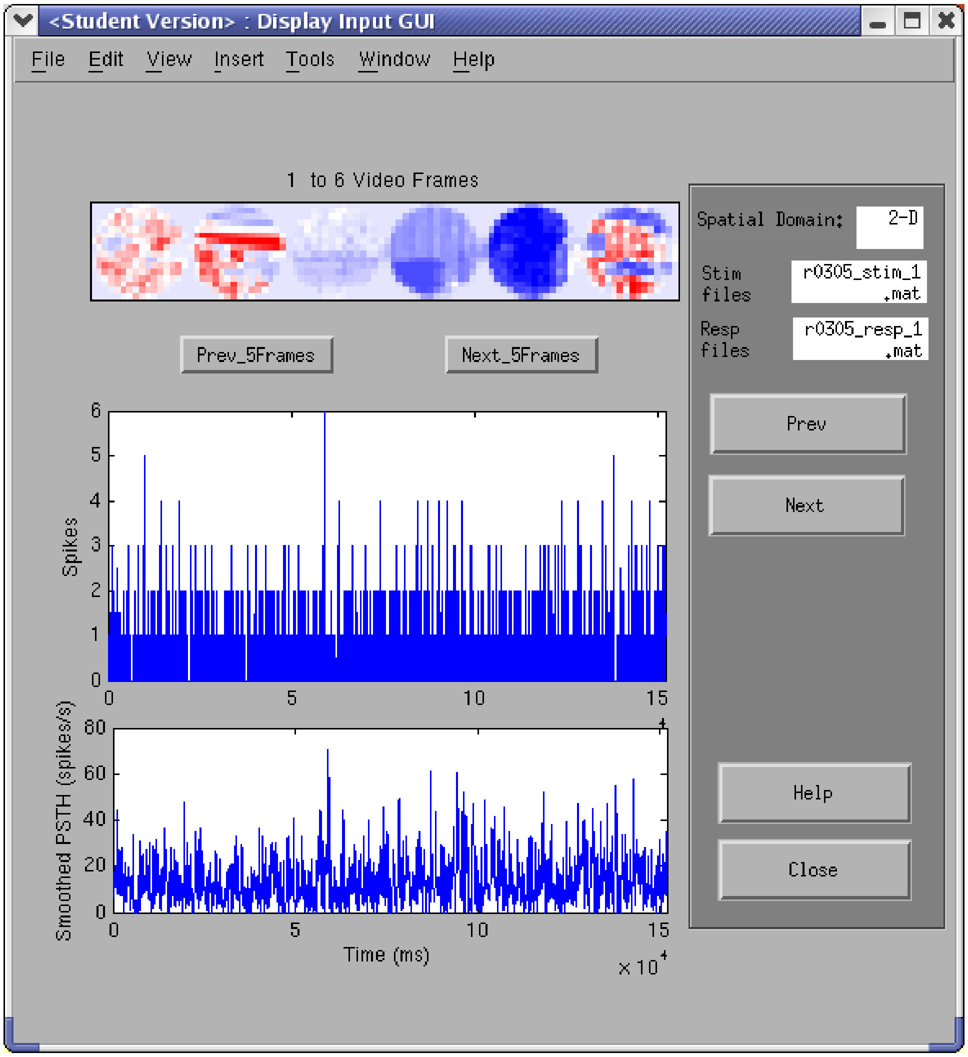


Figure 3.10: Sample of a visual stimulus-response pair

stimulus, 10 trials of spike trains and the smoothed PSTH. Here are some more details on this panel.

- Plot of the stimulus file:
  If the dimensionality of the spatial domain is $1D$, the x-axis is time (in milliseconds) and the y-axis is the spatial domain (e.g. frequency in Hz for auditory spectrograms). If the dimensionality of the spatial domain is $2D$, the first 12 video frames show up. For the current version of STRFPAK, we only show the first 12 frames.

- Raster plot of spike trains (in the order they were recorded):
  Note: We only display the first 10 trials in the current version of STRFPAK.

- Plot of the smoothed psth (in the time domain):
  Note: If we have single trail spike train, we take its psth as itself. Otherwise, we compute the average of multiple trials as their psth. The smoothed psth is computed as a convolution of the raw psth and a Hanning window.

The right panel shows information of the figures displayed in the left panel. You can click [Prev] or [Next] button to display previous/next data sets if multiple data pairs are selected. The stimulus and response text fields show the input data pair files currently displayed.

## 3.6 Estimating the STRF

In this stage, the second order statistics of the stimulus are calculated and the STRF is estimated by clicking the [Calculate] button. The stimulus autocorrelation 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.

### 3.6.1 Calculating the STRF

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].

### 3.6.2 STRFPAK's Calculation Implementation

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.

As STRF estimation can be computationally intensive, STRFPAK-2.0 provides a progress status bars. When the calculationa are done, the small **Done Estimation** window appears.

### 3.6.3 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.
- 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).

### 3.6.4 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 popup 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 $i, j$ antry of the autocorrelation 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 autocorrelation 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**:

  The modulation spectrum of an ensemble of sounds is the two-dimensional power spectrum of the spectrogram of that sound. Figure 3.11 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).

### 3.6.5 Display STRF

In the toolbox, there are four display/analysis options once the Display STRF button is clicked. The STRFPAK team is planning to add more options. The options available when you click the popup menu on the window include:

- **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 STRF. Prev and Next buttons display the STRF calculated using

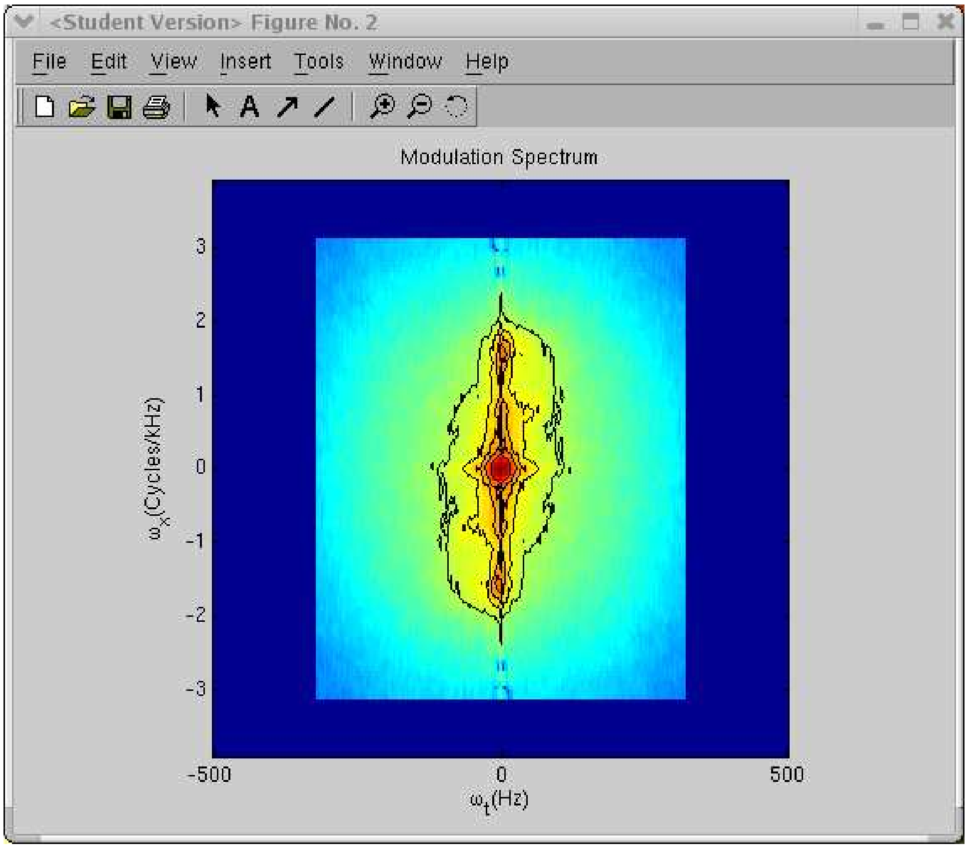


Figure 3.11: Modulation Spectrum of a sound

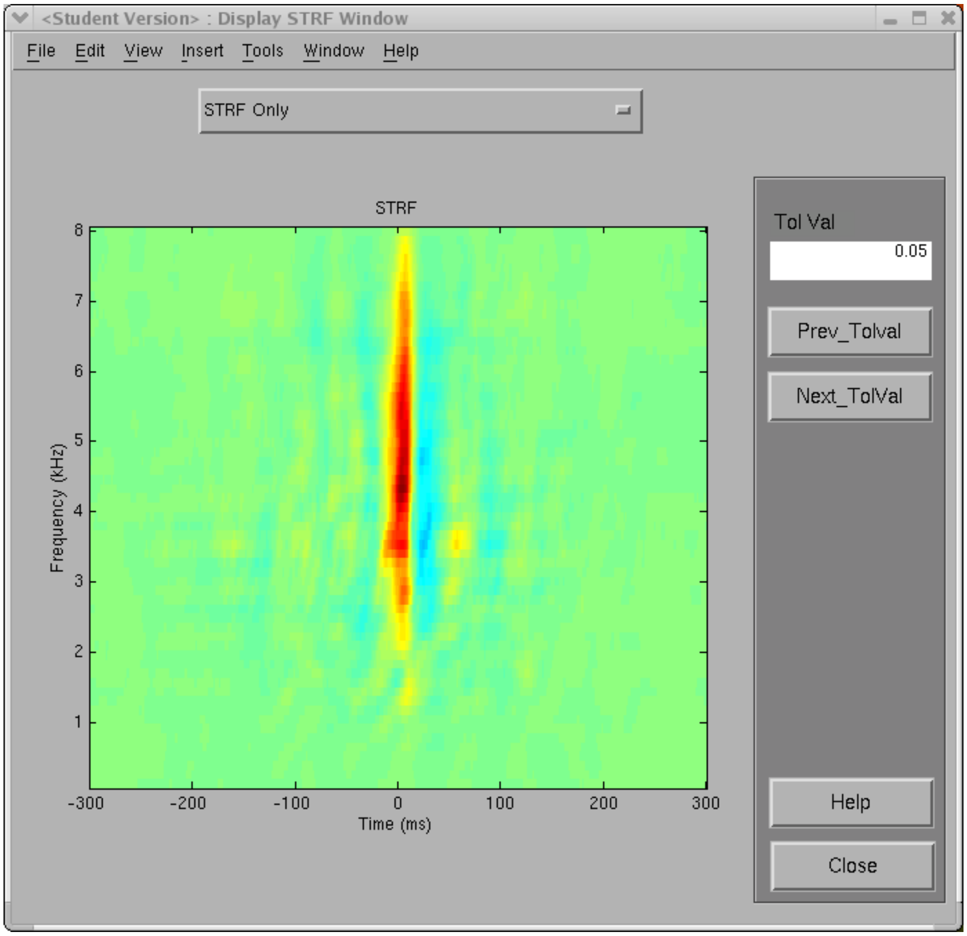


Figure 3.12: Display STRF: STRF only

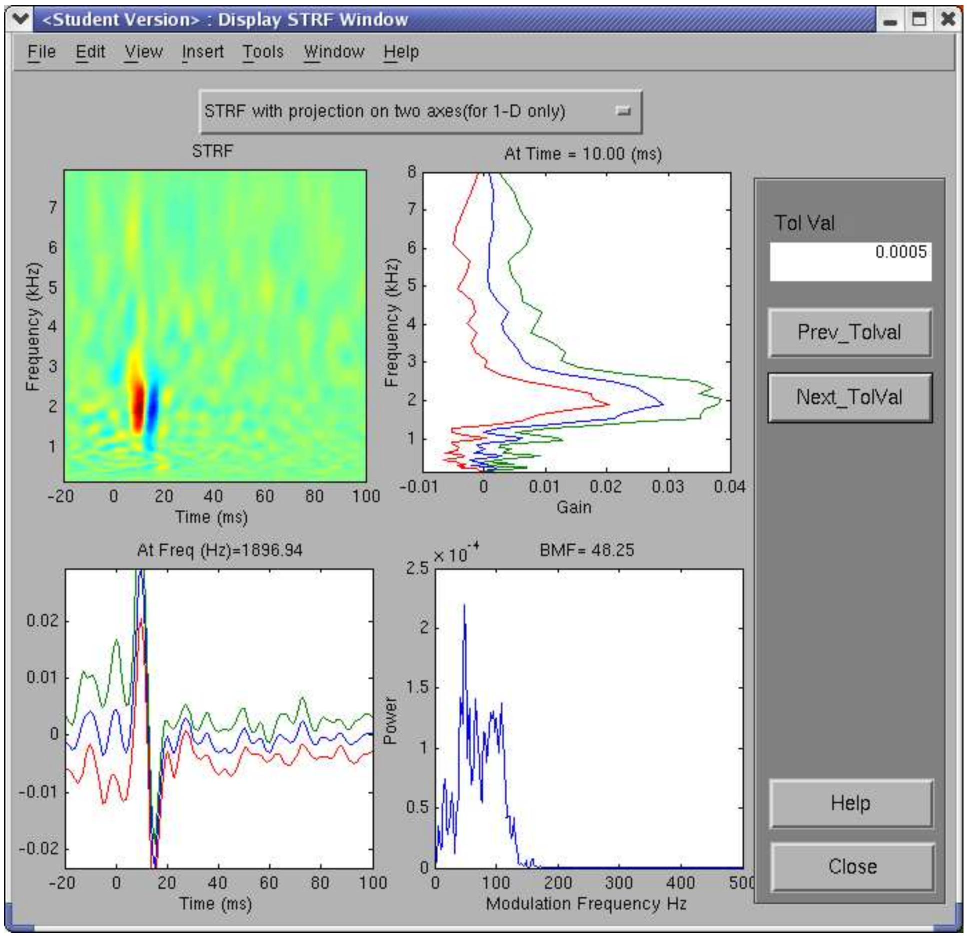


Figure 3.13: Display STRF: STRF projection on two axes

different tolerance values. Figure 3.12 is an example of the output of **STRF Only** in the **Display STRF** window.

- **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 3.13 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

  peaks excitatory response.

- **STRF and Spike-Triggered Average (STA)**: This option helps to compare the estimated STRF to the spike triggered average (STA). Prev and Next buttons help go to different STRF tolerance values. Figure 3.14 is an example of this option.

- **STRF and STA (for 2-D display)**: This option is for 2-D only. Figure 3.15 gives an example of this option. ¿From this figure, we can see the STRF is displayed as a time-varying sequence of the 2-D frames. In order to see how the STRF is different from the STA, this option draws them together.

### 3.6.6 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.

## 3.7 Prediction

In this toolbox, the **Predict** stage helps to see how well the estimated STRF generalizes to new data. It uses the estimated STRF to predict new responses if new stimuli are provided. Results will be saved for the validation procedure later. The current **Predict** stage includes three subsections:

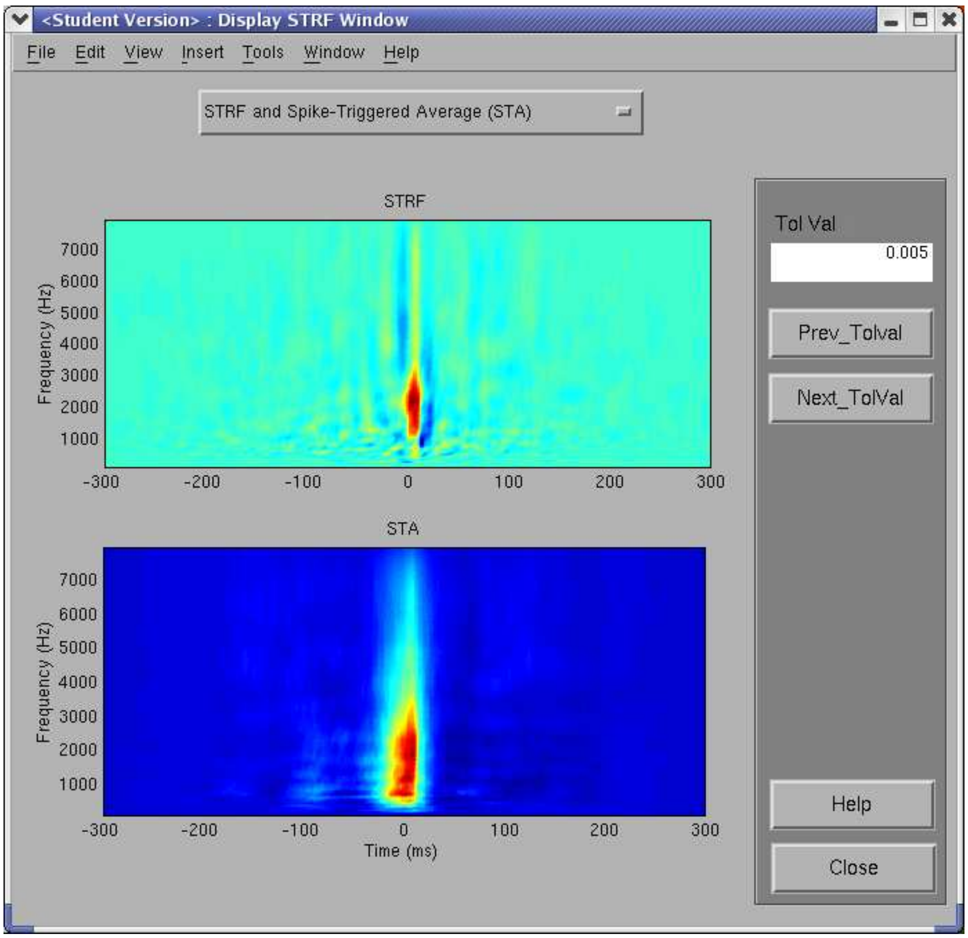


Figure 3.14: Display STRF: STRF and STA

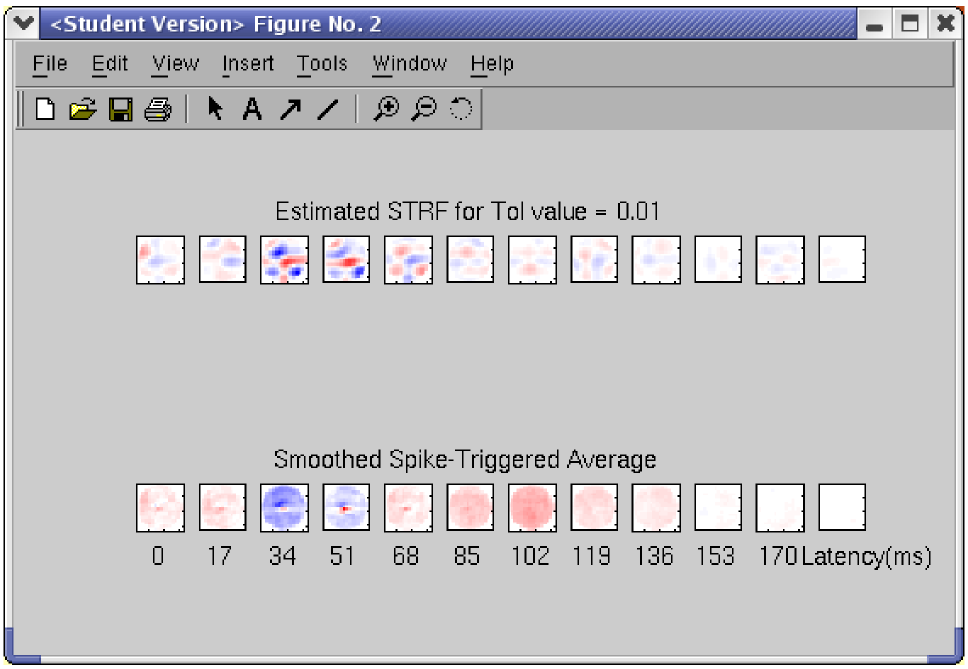


Figure 3.15: Display STRF: STRF and STA (for 2-D only)

- Get PredFiles: In order to avoid overfitting to noise, we provide this option so that you can choose new data sets to do predication. Or if you want to use the original data sets, the Jackknifed STRF will be used instead.

- Predict: After you provide new stimulus data, clicking Predict begins to compute the neuronal response.

- Display PrePSTH: This option plots the prediction input and prediction result on one figure. It helps you clearly see how good the prediction is.

### 3.7.1 Get PredFiles

The STRFPAK uses a parameter, $MATCH\_FLAG$, to keep track of the users choice regarding Jackknifing. After clicking the Get PredFiles button

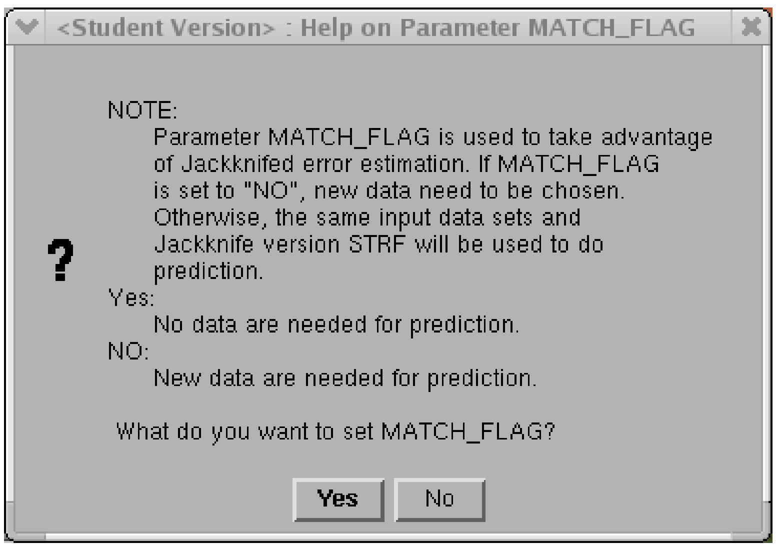


Figure 3.16: A questioning window for implementing cross-validation

in the main window, a window shown in Figure 3.16 appears waiting for your selection. The value of the $MATCH\_FLAG$ is set based on your selection.

If you click Yes button, the STRFPAK applies the cross-validation to do prediction. Otherwise, the STRFPAK predicts the neuronal response to a new stimulus you provide. If you click No button, the **STRFPAK: Get**

**Prediction Input** window pops out. This window's layout and buttons are exactly the same as the **Get Files** window. For any questions or problems, please refer to the previous **Get Files** section.

### 3.7.2 Prediction

STRFPAK predicts neuronal responses 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 Prediction" window appears.

### 3.7.3 Displaying PrePSTH (the predicted PSTH)

After the prediction stage, we have a predicted PSTH and an original PSTH from the actual neuron response. The **Display Predicted PSTHs** window provides a graphical display of these results.

Figure 3.17 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 3.18. ¿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 $ms$ 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 12 $2D$ 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 plots based of STRFs at other tolerance values by clicking the $\boxed{\text{Next Tol Val}}$ and $\boxed{\text{Prev Tol Val}}$ buttons. 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, different files can be displayed by pressing the $\boxed{\text{Next file}}$ and $\boxed{\text{Prev file}}$ buttons. $\boxed{\text{Help}}$ and $\boxed{\text{Close}}$ 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.

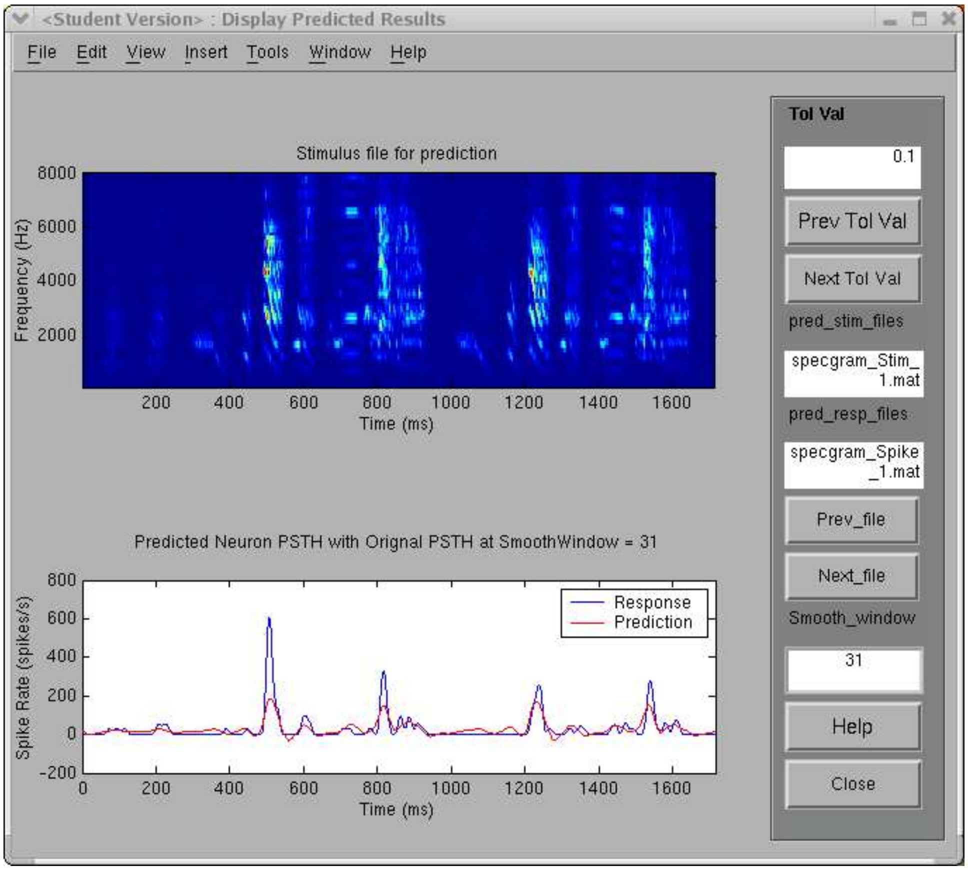


Figure 3.17: Display prediction results for the auditory example

## 3.8 Validate

The **Display Predicted PSTH** window shows a visual comparison of the predicted and actual responses. The **Validate** stage includes validation and displaying results.

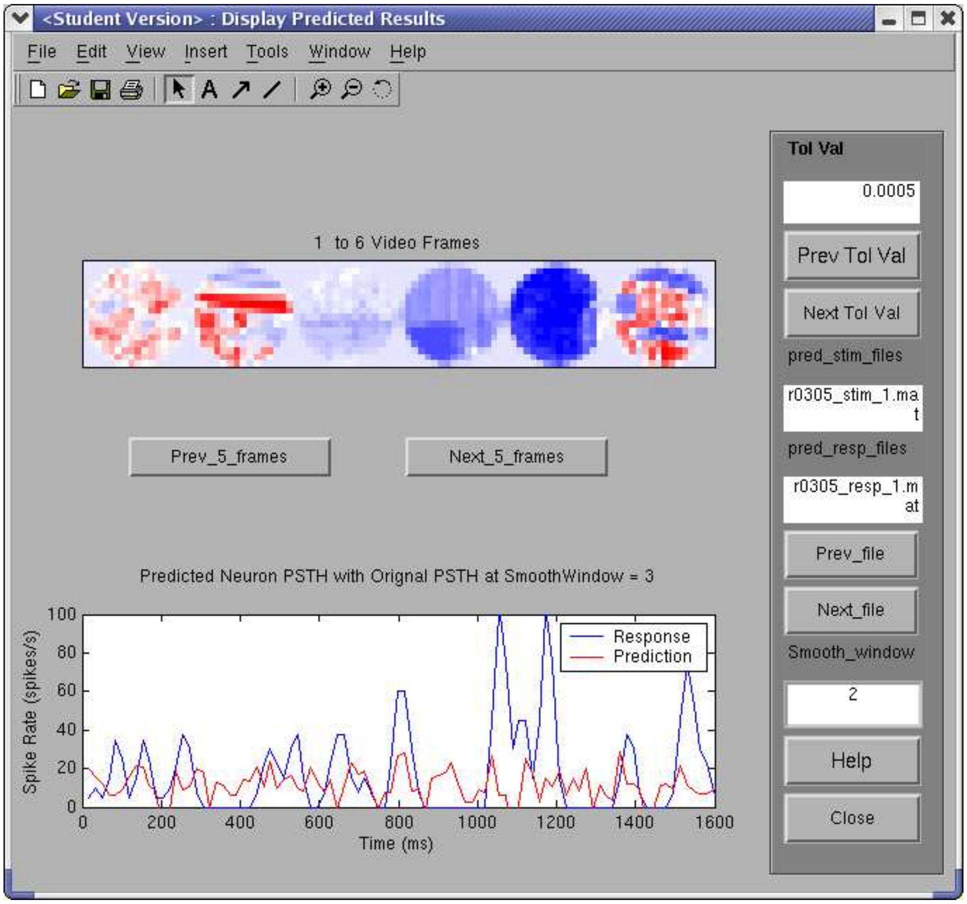


Figure 3.18: Display prediction results for the visual example

### 3.8.1 Validate

To quantify the goodness of fit of the estimated STRF, STRFPAK implements two measures: the coherence and the correlation coefficient. The 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 researchers 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 calles the first *max_CC_ratio* and the second *const_CC_ratio*. The built-in smoothing width used bt STRFPAK is 21 ms.

### 3.8.2 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.

### 3.8.3 Display Info

Once the [Display Info] button is clicked, the **Display CorrCoef/Info** window shows up. This option displays both goodness of fit measures used by STRFPAK. The layout of the window is the same as before. The left panel shows the graphical display and the right panel shows all the information about the results on the left. In the right panel, there are three display options. They are:

- **Corr Coef (r)**: This option is the default option. Figure 3.19 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 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. [Next] and [Prev] buttons take you to the results generated from each $Tol\ Val$.

- **Info Values**: Figure 3.20 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. The [Next] and [Prev] buttons take you to the results generated from each $Tol\ Val$.

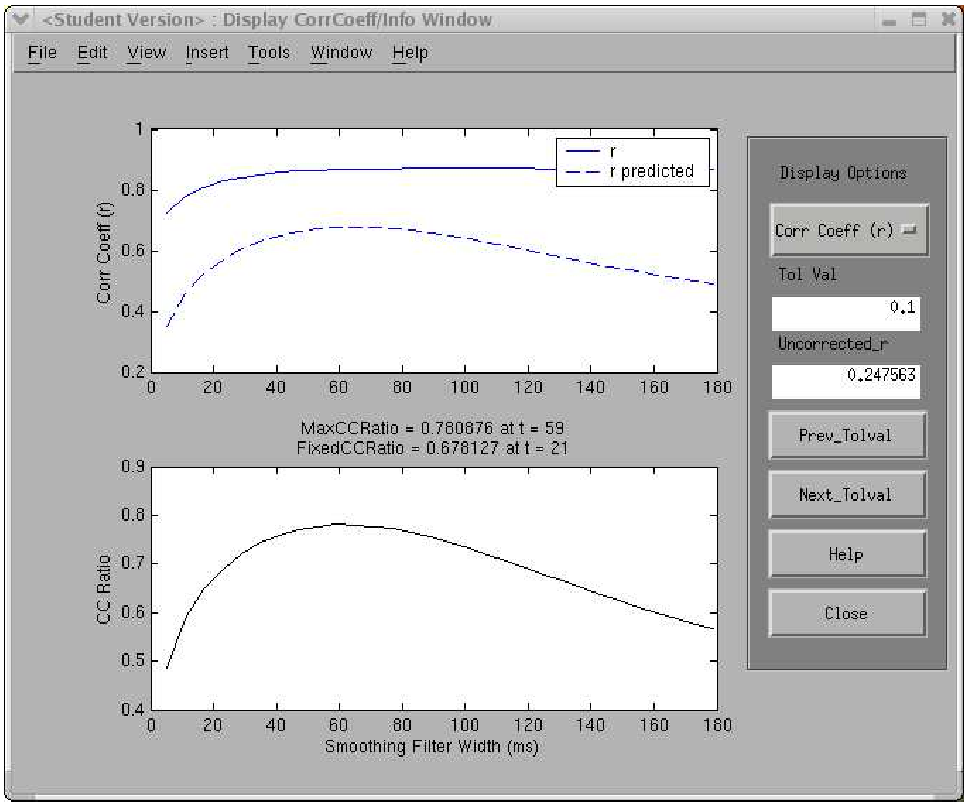


Figure 3.19: Predicted cc and cc for the auditory example

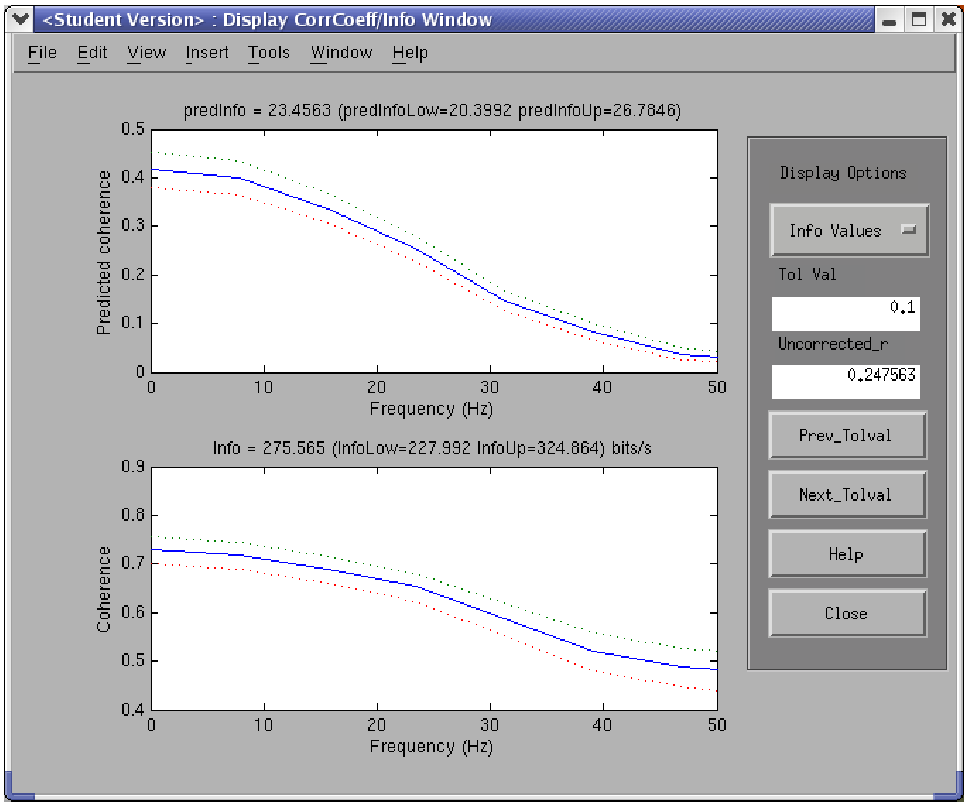


Figure 3.20: Predicted coherence and coherence for the auditory example

- **Info/r vs Tol Val**: This option plots the information value as a function of $Tol\ Val$ in the left top panel and the correlation coefficient as a function of $Tol\ Val$ in the left bottom panel.

### 3.8.4 Display BestStrf

Once the Display BestStrf button in the main window is clicked, STRFPAK does the following three things. It sorts the estimated STRFs based on the predicted information values, chooses the best STRF (based on info values) and displays the best STRF in the **Display Best Filter** window.

Figures 3.21 and 3.22 show the best estimated STRFs for the auditory and visual example. The right panel in the window 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 the Help button is clicked, the intermediate help window appears.

## 3.9 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 for a directory in which results are stored, and load as much information about a previous STRFPAK session as possible. The user can then resume examining previous work.

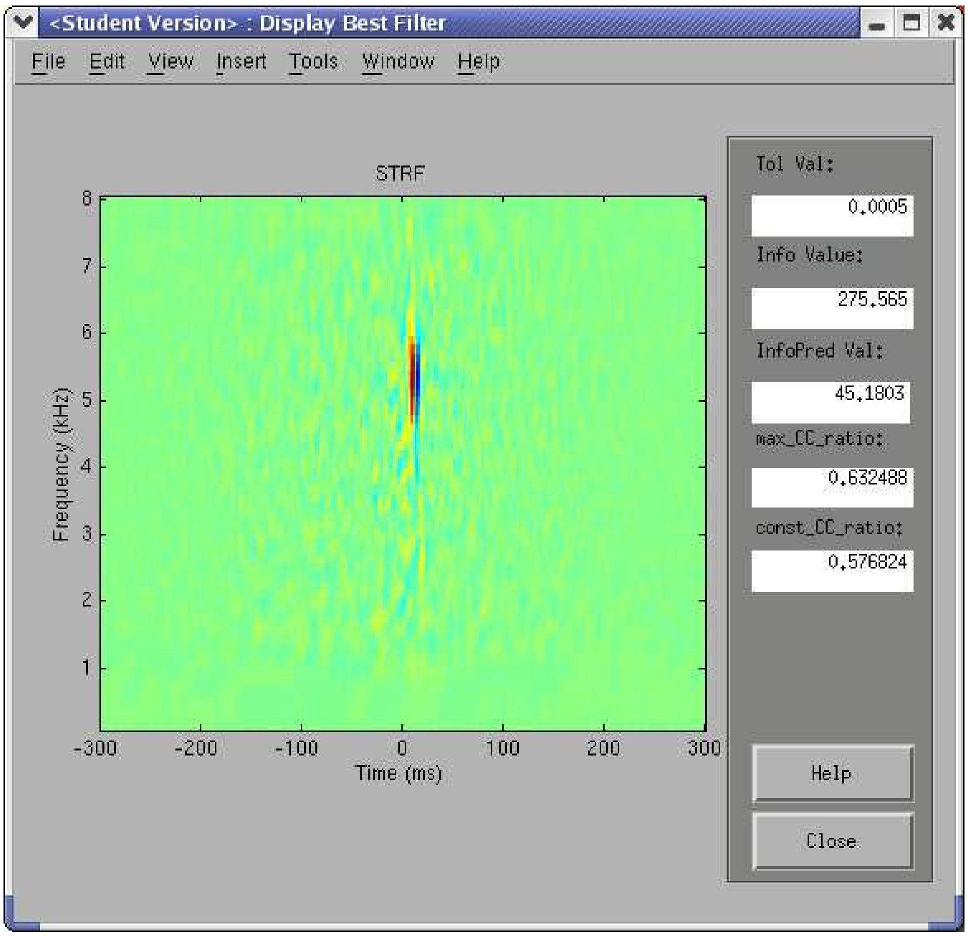


Figure 3.21: Best estimated STRF for the auditory example

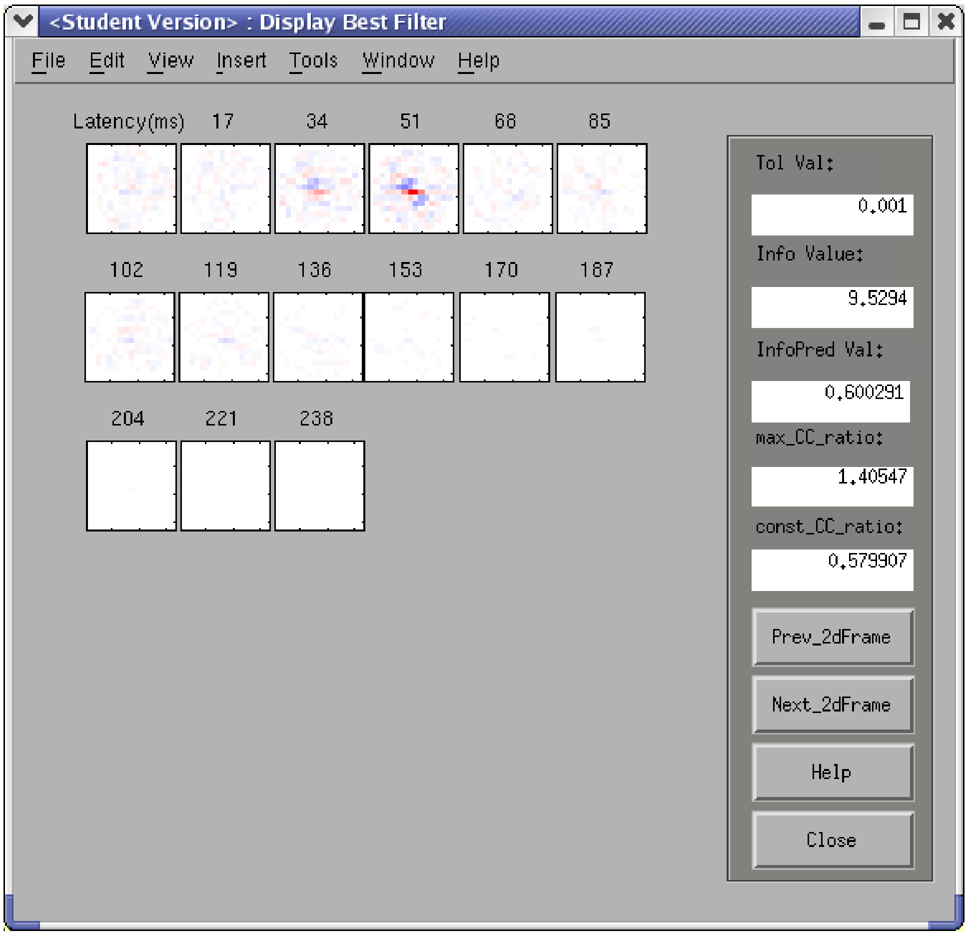


Figure 3.22: Best estimated STRF for the visual example

# Chapter 4

# Batch-mode/Nongraphical processing option

For a large sets of data, you may want to run STRFPAK-2.0 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 cochlear 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-2.0.

```
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.