

EMGLAB User's Guide

version 1.0

EMGlab Overview

EMGlab is an interactive program for viewing and decomposing EMG signals. EMG signals recorded by selective electrodes are made up of trains of discharges known as motor-unit action potentials (MUAPs). The goal of decomposition is to sort out the discharges that belong to each train.

The EMGlab program is able to read data files in a variety of different formats. It includes automatic decomposition algorithms as well as a graphical user interface for inspecting and editing the results. The waveforms and firing times of the identified MUAPs are saved in an "annotation" file.

The usual procedure for decomposing a signal is to load it into EMGlab, apply a high-pass filter, run an automatic decomposition algorithm on an initial portion of the signal (typically 5 or 10 seconds), manually inspect the results and correct any mistakes, and then move on to the next portion of the signal, and so on. The goal should be to achieve "full decomposition," i.e., to account for every discharge in the signal in terms of a set of physiologically realistic MUAP trains. This may require a substantial amount of manual effort, but it is the best way to ensure the accuracy of the results.

EMGlab is intended primarily for EMG signals recorded using needle or fine-wire electrodes during low or moderate contractions, i.e., signals in which individual motor-unit action potentials (MUAPs) are distinguishable. Signals from very strong contractions and signals from surface EMG electrodes are generally too complex, or have MUAPs that are too similar in shape, to be decomposed accurately.

Even signals from needle and fine-wire electrodes can vary considerably in quality and complexity depending on the location of the electrode, the strength and type of contraction, and other factors. Some signals can be decomposed with a high degree of confidence, others cannot. One should not accept the results of the automatic decomposition algorithm blindly, nor should one expect others to blindly accept results with gaps or unexplained activity. It is your responsibility to ensure that your results are accurate and scientifically valid.

This version of EMGlab is primarily intended for single-channel signals. It lets you load multi-channel signals, but you can really only work with one channel at a time.

The EMGlab package includes the open-source MTLEMG automatic decomposition program by Florestal and Mathieu, and the open source SON library for reading Spike 2 files by Lidieth.

Summary of New Features in Version 1.0

Moving through the signal. You can now scroll through signals by dragging the cursors or the panel contents, by using the arrow keys on the keyboard, and by using the mouse scroll wheel. You can also scroll continuously by holding down the < or > buttons. You can change the length of data shown in the Navigation Panel. You can move through long signals using a slider underneath the Navigation Panel.

Manual editing. You now see the templates as you drag them between panels and adjust them in the Close-up Panel. You can display templates as a cascade or shimmer plot, and you can display the residual in the Navigation Panel. You can copy firing patterns and templates between channels or between annotations. The program now supports ten levels of undo / redo.

Displaying auxiliary data. You can now display force or another auxiliary channel in addition to the EMG signal.

Working with large files. You can now work with very long signals without running out of memory.

Printing and plotting. You can now directly print the EMGlab window. You can also take a screen shot of any panel to save it as a figure.

Exporting data. You can now export templates and firing patterns to the Matlab workspace.

Comparing annotations. You can now compare different annotations of the same signal.

Installation

EMGlab is available as open-source Matlab code. It can be freely downloaded from <http://www.emglab.net>. It is distributed under the Aladdin Free Public License, which gives anyone the right to use, copy, and modify the code, but not to sell it.

Requirements. Matlab version 6.5 or higher. No Matlab toolboxes are required.

Installation Instructions.

1. Unzip the file *emglab1.XX.zip*. This zip file contains one folder named *emglab1.XX* which contains the EMGlab version 1.0 software. Note: if you are using WINZIP, you must type in the folder name in the WINZIP file browser, like this:
`c:\mydirectory\emglab1.XX`
2. Move the *emglab1.XX* folder to whichever directory you want it to reside in.
3. Run Matlab.
4. In Matlab, select the *emglab1.XX* folder as the current directory. You can do this using the Current Directory window, the `cd` command, or the File Open Menu.
5. At the Matlab command prompt, type:

```
>> install
```

The install script removes any previous versions of EMGlab from the Matlab path, permanently adds the current version of EMGlab to the Matlab path, and copies your old EMGlab preference file (if there is one) into the *emglab1.XX* folder.

Running EMGlab.

To run EMGlab, type "emglab" in the Matlab command window.

EMGLAB Screen

EMGlab runs in a single Matlab window, which is divided into five panels. At the top is the Signal Panel, which displays a segment of the EMG signal. Below that is the Template Panel, which displays the MUAP templates. Below that on the left is the Firing Panel, which displays the firing patterns of the identified MUs. To the right is the Close-up Panel, which displays a section of the EMG signal at an expanded scale. At the very bottom is the Navigation Panel which displays a thumbnail of the EMG signal.

Above the Signal Panel are pull-down menus for selecting the channel and high-pass filtering. The program also displays the names of the signal and annotation files.

The buttons on the edges of the panels are used to change the display characteristics.

+, -	zoom in or out vertically
<, >	scroll left or right
,	zoom in or out horizontally

Other buttons on the Firing and Close-up Panels are described below.

Signal Panel. The Signal Panel shows the EMG signal and the residual signal that remains after the templates of the identified MUAPs have been subtracted. The time scale shows the time in seconds from the beginning of the signal. The default time scale is 100 ms full screen. The unit numbers of the identified spikes are shown at the top of the panel. The expected firing times of the selected unit are indicated by red lines at the top. The green cursor indicates the interval shown in the Close-up Panel.

Template Panel. The Template Panel shows the templates of the identified MUAPs. The selected unit is shown in red. The Template Panel uses the same time scale as the Signal Panel. The "| |" and "|||" buttons control the number of templates displayed and the width of the displays. If all the templates do not fit on the screen at once, a vertical slider appears to the right of the panel. The slider or the < and > buttons can be used to scroll through the templates. The display can be changed to show individual MUAP discharges instead of averaged template using the "Cascade" and "Shimmer" options in View Menu.

Firing Panel. The Firing Panel shows the firing patterns of the identified MUAPs. The selected unit is shown in red. The green cursor indicates the interval shown in the Signal Panel. The display can be toggled to show instantaneous firing rates using the o button. For the instantaneous firing rates, the vertical scale is in Hz, and a popup appears to set the low-pass filter setting.

Close-up Panel. The Close-up Panel shows a segment of the EMG signal at an expanded scale. Interpolation is used to give higher temporal resolution. If the segment contains identified MUAPs, the templates and unit numbers are shown at the bottom, the reconstruction is overlaid over the signal, and the residual is shown in the middle.

Changes made in the Close-up Panel do not affect the other panels until the OK button is clicked.

Navigation Panel. This panel shows a thumbnail of the EMG signal and the times of all the identified discharges. There are three cursors. The short, wide cursor at the bottom shows the interval displayed in the Firing Panel. The middle cursor shows the interval displayed in the Signal Panel. The very small cursor at the top shows the interval displayed in the Close-up Panel. The panel can be zoomed out to show the entire signal (or to the length of the data buffer if it is shorter than the entire signal). The residual signal can be displayed instead of the actual signal using the View Menu.

Re-sizing the EMGlab window. In some versions of Matlab, the EMGlab window does not always redraw correctly when re-sized. If this happens try re-sizing again by an incremental amount, or select "Redraw window" from the File Menu.

Re-sizing the panels. You can change the size of all the panels except the Navigation Panel. When you move the pointer over one of the gaps between the panels, it changes to a cross-hairs. Click and drag the gap to change the panel sizes. You can hide a panel completely by dragging far enough. The gap remains, and you can unhide the panel by clicking in the gap and dragging it back.

Loading EMG Signals

EMGlab is able to read EMG data in a number of different formats. Binary files created by many acquisition programs can be read using the "Load EMG file" command. Signals that can be loaded into Matlab can be then imported into EMGlab. Files with proprietary formats can be read by creating a custom Matlab reader function.

Binary files. If your data is stored in a binary integer format, you can usually load it by selecting "Load EMG file" from the File Menu. The program will prompt you for:

number of channels.

byte order. Select PC (i.e., little-endian) if the file was created on a Windows computer or an Intel Mac. Select non-PC (i.e., big-endian) if it was created by a non-Intel Mac. If the signal doesn't look right, try selecting the other byte order.

sampling rate (per channel).

byte offset of first sample. This is the number of bytes to be skipped over at the beginning of the file.

The following settings are optional:

start time (in seconds): The time of the first sample. Default is 0.

gain, the number of ADC units per physical unit. The default is 1.

units: ADC units, millivolts, or microvolts. The default is ADC units.

You can also store this information in a WFDB header file associated with the data file. When you load a data file that has an associated header file, the program takes this information directly from the header. WFDB headers are described more fully in the Appendix.

Importing from Matlab. If you can load your data into Matlab, then you can import it from Matlab into EMGlab. In Matlab, put the data into a global variable named EMGSIGNAL:

```
>> global EMGSIGNAL  
>> EMGSIGNAL = my_data;
```

where "my_data" is the name of the variable in which the data is stored. The data can be in either double or int16 format. If the data is multi-channel, each channel should be in a separate column. Then in EMGlab select "Import EMG signal" from the File Menu. The program will prompt you for the sampling rate, start time, gain, and units.

Alternatively, you can put the data and the associated information into a structure:

```
>> global EMGSIGNAL  
>> EMGSIGNAL = struct ('data', my_data, 'rate', my_sampling_rate);
```

The fields "data" and "rate" are required. You can also optionally specify the fields "start," "gain," and "units."

You can either execute these commands directly from the Matlab command line before selecting "Import EMG signal," or you can put them, along with any other commands needed to read or pre-process the data, into a script named "my_emgsignal.m." The script should go into the EMGlab directory or another directory on the Matlab path. When you select "Import EMG signal" from the EMGlab File Menu, the program will execute this script before accessing EMGSIGNAL.

Custom reader. If your data file is in a proprietary format that cannot be described by a WFDB header, you can create a Matlab m-file to read in the data. Reader m-files are described more fully in the Appendix. EMGlab currently supplies readers for "smr" files (Spike 2, Cambridge Electronic Design Ltd), and "sc1" files (MUtools, G. Kamen). Please let us know if there are other widely used formats for which readers would be helpful.

What if you don't know the format? If you suspect that a data file is a binary file but you don't know its exact format, you can sometimes figure it out. First determine if there is a header. Try opening the file with number of channels = 1, byte order = PC, and header bytes = 0. If there is a header at the beginning of the file, it will usually appear like random noise with amplitude ranging up to 32000, perhaps alternating with segments of flat baseline. Try to estimate how many bytes long the header is and then reload the file, specifying this byte offset. Next, determine the byte ordering. If the signal ranges up to 32000, if it appears clipped, or if it has unlikely discontinuities, try opening with the other byte order and see if that looks better. Finally, determine the number of channels. If the signal zig-zags back and forth every other sample, try re-loading with a different number of channels until all the channels have valid looking EMG signals.

Long Data Files. If the program becomes sluggish or runs out of memory when working with long data files, you can limit the amount of data that the program loads into memory at any one time. Do this by changing "buffer length" in Preferences. The buffer length is normally set to 50 seconds. It limits the amount data that can be displayed on the screen or processed at any one time. Specifically, it limits the extent you can zoom out any of the panels. You can still scroll through the entire signal or jump anywhere in the signal using the slider under the Navigation Panel. To help you find a specific location, the time corresponding to the pointer location is displayed when the pointer is over the slider.

Auxiliary Signals

EMGlab allows you to display auxiliary signals, such as the force of contraction, along with the EMG signal. Auxiliary signals can either be in the same file as the EMG signal, or in a different file. They do not have to have the same sampling rate as the EMG signal.

Any channel of a multi-channel EMG file can be displayed as an auxiliary signal. You can also load one or more auxiliary signals from a separate file using the "Load auxiliary file" command from the File Menu, or by adding a line to the EMG file's WFDB header. You can also import one or more auxiliary signals from Matlab using the "Import auxiliary signal" command in the File Menu. Select the signal to be displayed using the "Auxiliary signal channel" command from the View Menu. Filter and scale the auxiliary signals using the "Auxiliary signal settings" command from the View Menu.

Moving Through the Signal

The Navigation Panel at the bottom of the EMGLab window shows a thumbnail of the EMG signal. By default the panel shows 10 seconds of the signal, but you can zoom in or out using the || and ||| buttons. You can zoom out to show the entire length of the signal (or to the length of the data buffer specified in Preferences, whichever is shorter). You can scroll through the signal by using the slider under the Navigation Panel. You can also jump to a point in the signal by clicking in the slider bar. To help you find a specific location, the program displays the time corresponding the pointer location whenever the pointer is over the slider.

You can scroll the Signal, Firing, and Navigation Panels by clicking on an empty part of the panel and dragging or (for the Signal and Firing Panels) by clicking the < or > buttons. Holding down the buttons causes continuous scrolling. You can also scroll by using the arrows or "<" and ">" keys on the keyboard. By themselves, these keys scroll the Signal Panel. When pressed with the shift key, they scroll the Firing Panel. When pressed with the CTRL key they scroll the Navigation Panel. You can scroll the Signal Panel using the mouse scroll wheel.

You can jump to a particular point in the signal by clicking in the Firing Panel or in the upper part of the Navigation Panel. You can jump to a particular point in the firing pattern by clicking in the lower third of the Navigation Panel. The green cursors in each panel indicate the intervals shown in the other panels. You can drag the cursors to scroll the corresponding panels.

If scrolling is too sluggish on your computer, you can try setting the drawing mode to "faster" in Preferences. (In Matlab versions 7.1 and higher on the Mac, "faster" mode causes the screen to update erratically. If this happens, switch back to "smoother" mode.) You can also turn scrolling off altogether. In this case, you will have to click within the panels or use the < and > buttons or arrow keys to move through the signal.

Playing the signal. You can play the signal in real time by clicking the P button to the right of the navigation slider. Click the P button again to stop playing. Click the << button to the left of the slider to rewind the signal back to the beginning. Playing uses the Matlab audioplayer function, which is not available in older versions of Matlab. Playing may not work well on slower computers.

Decomposition

Although it is possible to decompose signals entirely by hand, it is usually better to begin by having the program try to decompose the signal automatically. EMGLab provides two automatic decomposition algorithms. They both attempt to detect all the active MUs in the signal and identify as many of their discharges as possible.

Both automatic decomposition algorithms only operate over the interval displayed in the Firing Panel. This interval is 5 seconds by default, but you can make it shorter or longer using the Firing Panel zoom in and zoom out buttons.

Filtering. The first step in decomposition is to apply a high-pass filter to accentuate the MUAP spikes. For needle and fine-wire signals, a 1000 Hz filter is usually appropriate. If your signal is already high-pass filtered, you can decompose it without additional filtering, but you should change the "MUAP width" parameter in Preferences as described at the end of this section.

Auto Decomp. This is the default algorithm. It is intended to be used interactively on successive segments of the EMG signal. Typically, you run this algorithm on the initial segment of a signal to have the program create templates of the active MUAPs and identify as many of the discharges as possible. After checking the results and correcting any errors you move on to the next segment of the signal and run the algorithm again. The program identifies discharges of existing templates and checks to see if any new MUAPs have been recruited. For busy signals, the algorithm sometimes identifies additional discharges if you run it more than once on the same signal segment.

Determining the appropriate length of the analysis window involves a tradeoff between identification performance and tracking performance. A longer analysis window generally improves the algorithm's ability to identify all the active MUs, while a shorter window improves its ability to track changing MUAP shapes. A 5-second window provides a good compromise for many signals.

The default algorithm operates only on the signal in the currently selected channel.

Auto Decomp (MTL). This is a multi-channel version of the decomposition algorithm by Florestal, Mathieu, and Malanda. It is intended for analyzing the entire signal at a time, rather than in segments. Therefore you should zoom out the Firing Panel to include the entire length of signal you want to analyze before running this algorithm. Unlike the default algorithm, the MTL algorithm processes all the channels of the signal, not just the current one. The processing can take several minutes, depending on the number of channels, signal complexity, and processor speed.

For multi-channel signals, the MUs are numbered consistently across channels. MUs that have large templates in one channel may have flat templates in another channel. The timing of a particular discharge can be different in different channels because of propagation delay and jitter. When editing the results, remember that edits only take effect in the current channel. To make the changes in multiple channels, you have to edit each channel separately.

Spike width and MUAP width. EMGlab allows you to use a different template width when the signal is high-pass-filtered and when it is unfiltered. You can set these values via the "Spike width" and "MUAP width" parameters in Preferences.

The use of different widths is useful for analyzing signals recorded with wide amplifier bandwidths (e.g., 5 Hz - 5 kHz), such as clinical EMG signals. In such signals, the MUAP waveforms can be quite long, even though they are converted into narrow spikes by high-pass filtering. For these signals, decomposition is best accomplished by applying a high-pass filter to increase MUAP discrimination, and by using narrower templates to reduce processing time. However, in order to examine the MUAP waveforms in the unfiltered signal, wider templates are needed. By default, EMGlab uses template widths of 10 ms which high-pass filtering is applied, and of 60 ms for the unfiltered signal.

If your signals are already high-pass filtered you may not need to apply additional filtering in EMGlab. In this case you should decrease the value of "MUAP width" in Preferences to 10 ms (or whatever value is appropriate) to improve program efficiency.

Manual Editing

You can edit the results produced by the automatic decomposition algorithms, or decompose a signal from scratch using graphical commands. These commands generally involve clicking on an object or dragging an object to a different part of the screen. The effect of some commands can be modified by holding down a modifier key.

Normal clicks are used for the most common commands.

"Shift clicks" are used for special commands. They are made by holding down the shift key when you click.

"Option clicks" are used primarily for deleting. They are made by holding down the control key (or the option or command key in some Mac versions) or by right clicking (in Windows).

Unit Commands

Create a new unit. Click on a spike in the Signal Panel and drag it to the Template Panel.

Reorder the units. Click on a template and drag it to a different position in the Template Panel.

Merge two units. Shift-click on one of the templates and drag it over the other one. The two templates are shown in the Close-up Panel, and their merged firing pattern is shown in the Firing Panel. To accept the merger, click the "OK" button in the Close-up Panel. To cancel the merger, click the "X" button in the Close-up Panel.

Re-average a template. Option-click (or normal click in some Mac versions) on the template number in the Template Panel to open the context menu. Select "Re-average." You can also re-average all the units at once using the "Re-average all" command from the Analyze Menu.

Select a unit. Click on the unit's template or firing pattern. The template and firing pattern of the selected unit are shown in red. Its expected firing times are shown by red bars in the Signal Panel.

De-select a unit. Click on the unit's template or firing pattern, or click on a blank template or firing pattern.

Delete a unit. Option-click on the unit's template, or option click (normal click in some Mac versions) on the template number to bring up the context menu and select "Delete unit."

Discharge Commands

Identify a spike in the Signal Panel. Drag a template over the spike. The program aligns the template automatically. Or, select a unit and shift click on the spike. Or, de-select all the units and shift-click on the spike. In this case the program automatically chooses the best match.

Delete an identification. Option-click on a unit number in the Signal Panel or on a discharge in the Firing Panel.

Close-up Panel Commands

The Close-up Panel is used for resolving superimpositions. You can add and manipulate templates to try to fit a particular spike. Changes made in the Close-up Panel do not affect the other panels until you click the "OK" button.

Show a close-up. Click on a point in the Signal Panel, or drag a point on the signal to the Close-up Panel.

Add a template. Shift-click on the template or drag it to the Close-up Panel.

Realign a template. Click and drag it or its number.

Remove a template. Option-click it or its number.

"Nudge" into alignment. Click anywhere in the Close-up Panel.

Lock/unlock a template into position. Shift-click on the template or its number. This hides the template and prevents it from being moved.

Resolve. Click the "R" button or select "Auto Resolve (MTL)" from the Analyze Menu. This aligns the chosen templates (i.e., the ones currently in the Close-up Panel) to give the best fit to the signal. The "R" button always uses all the chosen templates. The "Auto Resolve (MTL)" command removes templates if that provides a better fit. Hidden templates are not affected.

Accept the results. Click the "OK" button. This transfers the contents of the Close-up Panel to the other panels.

Reject the results. Click the "X" button. This restores the contents of the Close-up Panel to match the other panels.

Saving Results

Following the usage in ECG analysis, we refer to the results of decomposition—the MUAP templates and firing patterns—as an "annotation" of the EMG signal. EMGlab provides two formats for saving annotations.

In order to open an existing annotation file, you must first load the EMG signal. You can compare different annotations of the same EMG signal using the "Compare annotation with ..." command from the Tools Menu, which is described in the "Comparing Annotations" section.

EMG Annotation Format (EAF). This is the recommended format for saving and exchanging annotations. It is a text-based XML format. It saves the templates and firing patterns for single- or multi-channel signals. The EAF format allows additional information to be saved as well, although EMGlab does not currently provide a way to do this.

Text format (ANN). You can also save the firing patterns as a simple text file in which each line corresponds to a MUAP discharge, listing the discharge time, in seconds from the beginning of the signal, and the unit number. Only the firing patterns in the current channel are saved. The templates are not saved, but they are re-calculated when you open an .ann file within EMGlab.

Menus

File Menu

About EMGlab. Displays the current EMGlab version number.

Load EMG file. Opens a dialog box to select an EMG data file. If the selected file has an unknown extension, the program checks to see if there is a WFDB header file with the same filename and the extension ".hea" in the same directory. If not, the program assumes the file is a binary file and opens a second dialog box to ask for the file format. Loading an EMG signal deletes the previous annotation. See "Loading EMG signals" for more information.

Import EMG signal. Imports an EMG signal from Matlab. The program first tries to execute the function `my_emgsignal`, and then it imports the data in the Matlab global variable `EMGSIGNAL`. `EMGSIGNAL` can be either a double or `int16` array, or a structure with fields "data" and "rate." If not supplied, the program opens a dialog box to ask for the sampling rate. See "Loading EMG signals" for more information.

Open Annotation. Opens a dialog box to read an annotation file. In order to open an annotation, an EMG signal must already be loaded. EMGlab can read .eaf and .ann files (see "Saving Results"). Opening a .eaf file overwrites existing annotations in all channels. Opening a .ann file only overwrites existing annotations in the current channel. EMGlab can also read .decomp files saved by previous versions of EMGlab and .tim files saved by the MUtools program.

Save/Save as ... Opens a dialog box to save the current results. You can select one of two formats: .eaf or .ann. A .eaf file saves the templates and firing patterns from all channels. A .ann file saves the firing patterns from the current channel in a simple text format. See "Saving Results" for more information. The older .decomp format is no longer supported for saving.

Load auxiliary file. Opens a dialog box to select an auxiliary data file. Auxiliary data, such as force of contraction, can be displayed but not further processed. The file format can be any of the formats that can be read by the "Load EMG signal" command. The auxiliary data can have a different sampling rate than the EMG signal. You can load more than one auxiliary file for the same EMG signal, and then choose the one for display using the View Menu. Loading a new EMG signal deletes all the existing auxiliary data.

Import auxiliary signal. Imports auxiliary data from the Matlab global variable `AUXSIGNAL`. The data can be either a double or `int16` array, or a structure with fields "data" and "rate." If not supplied, the program opens a dialog box to ask for the sampling rate. You can import more than one auxiliary signal for the same EMG signal, and then choose the one for display using the View Menu. Loading a new EMG signal deletes all the existing auxiliary data.

Preferences. Opens a dialog box for changing the EMGlab preferences. See the "Preferences" section for the list of the preferences.

Screen shot. Creates a copy of the selected panel as a separate Matlab figure. You can then use Matlab commands to edit it, or export it to a different program such as Adobe Illustrator.

Redraw window. Redraws the EMGlab window.

Print. Prints the EMGlab window using the Matlab print dialog.

Quit. Quits EMGlab and returns to Matlab. Note that you can return to Matlab without quitting EMGlab by clicking on the Matlab command window, or (in some Matlab versions) by pressing the return key.

Edit Menu

Undo/Redo. Undo or redo the last action. EMGlab supports 10 levels of undo/redo.

Copy/Paste. There is one copy command for templates and one for firing patterns. If a unit is currently selected, then only that unit's template or firing pattern will be copied. If no unit is selected, then all the templates or firing patterns will be copied. The copied item or items can be pasted to a different channel or a different annotation. See the "Copying and Pasting" section for more information.

Delete firings. Delete all the discharges shown in the Firing Panel, without deleting any templates.

Delete all. Delete all the units and firing patterns in all the channels.

View Menu

Grid. Turns the grid on a particular panel on or off.

Show auxiliary. Toggles whether or not to display the auxiliary signal in a particular panel.

Templates / Cascade / Shimmer. Selects the display format for the Template Panel. The normal mode is "Templates," which displays just the averaged MUAP templates. In "Cascade" mode, the waveforms of each discharge shown in the Firing Panel are displayed as a raster plot. The interference from the other identified MUAPs is subtracted out. In "Shimmer" mode, the waveforms of each discharge are displayed superimposed over one another. In both "Cascade" and "Shimmer" modes, you can click on a particular waveform to jump to it in the Signal Panel.

Navigation Signal / Residual. Selects whether the actual signal or the residual signal is displayed in the Navigation Panel.

Auxiliary Signal Channel. Selects which auxiliary signal to display. You can select any channel of the EMG signal or any additionally loaded or imported auxiliary signal.

Auxiliary Signal Settings. Opens a dialog box that allows you to change the name, high- and low-pass filtering, and scaling of the currently displayed auxiliary signal. In each panel, the auxiliary signal is normalized to the specified max and min values.

Analyze Menu

Auto Decomp. Invokes the default automatic decomposition algorithm. This algorithm is intended to be used interactively on a segment of the signal at a time. The analysis is performed only in the current channel and over the interval indicated in the Firing Panel. The algorithm tries to identify the discharges of all the active MUs, using any already existing templates and create new templates as necessary. It adds to, rather than overwriting, any existing annotations. Sometimes running the algorithm more than once on an interval will identify additional discharges. Note that it is usually desirable to select high-pass filtering before running automatic decomposition. See "Decomposition" for more information.

Re-average all. Re-averages all the templates. Use this command to improve the accuracy of manually created templates or to track slow changes in MUAP shape. The averaging is performed over all the discharges shown in the Firing Panel. You can zoom the Firing Panel in or out to accommodate templates that change more rapidly or more slowly. The averaging uses interference cancellation, so you can improve the quality of the averages by re-averaging more than once.

Center all. Shifts all the templates so that the largest peak (positive or negative) occurs in the center. This is useful if you paste firing patterns from one channel to another. See "More about Copying and Pasting" later in this section.

Show all. Un-hides all the units. See "More about Hiding Units" later in this section.

Auto Decomp (MTL). Invokes the Montreal automatic decomposition algorithm. This algorithm is intended for analyzing signals all at once, rather than a segment at a time. It operates over the interval indicated in the Firing Panel, so if you want to analyze the entire signal you must zoom out the Firing Panel. It can analyze single- or multi-channel signals. It overwrites any existing annotations. You can specify whether or not the algorithm should use the genetic algorithm to resolving superposition in Preferences. See "Decomposition" for more information. You can also call the MTL algorithm directly from Matlab. See the readme file in the mtlemg folder for more information.

Resolve Closeup (MTL). Resolves the superposition in the Close-up Panel using the genetic algorithm, as in the Montreal automatic decomposition. This command is essentially the same as clicking the "R" button in the Close-up Panel, except that it removes units that are not part of the superposition.

Tools Menu

Compare annotation with ... Opens a file dialog to select an annotation file to compare against the current annotation. See "Comparing Annotations" for more information.

Highlight differences. Toggles the way that annotations are displayed in Comparison mode. When selected, the templates and discharges that occur only in the current annotation are shown in the usual color, those that occur only in the comparison annotation are shown in the comparison color, and those that occur in both are dimmed. See "Comparing Annotations" for more information.

Swap annotations. Exchanges the current and comparison annotations. This allows you to edit the comparison annotation. See "Comparing Annotations" for more information.

Template and Firing Panel Context Menus

Option clicking (normal clicking in some Mac versions) on a unit number in the Template or Firing Panel opens a context menu. The template context menu has all these options, the firing-pattern context menu has only the "Export" option.

Re-average. Re-average this template over all the discharges shown in the Firing Panel.

Center. Center this template on its largest positive or negative peak.

Hide. Hide this unit. See the "Hiding Units" section for more information.

Export. Export data to Matlab. This command opens a dialog box that allows you to export the template, MUAP waveform, firing pattern, or instantaneous firing rate of this unit or all the units to a specified variable in the Matlab base (command-line) workspace. Templates are exported as an array of column vectors at a sampling rate of 10 times the sampling rate of the EMG signal. MUAP waveforms are exported as an array of column vectors at the same sampling rate as the EMG signal. Firing patterns are exported as a column vector or a cell array of column vectors, with each vector containing a list of discharge times, in seconds. Instantaneous firing rates are exported as an array of column vectors, in pulses per second, at a sampling rate of 100 Hz. The instantaneous firing rate is set to nan in intervals during which a unit is inactive for more than 0.5 s.

Delete unit. Delete this unit (template and firing patterns).

Preferences

The following preferences can be changed by selecting the "Preferences" command from the File Menu. The preferences are stored in the file EMGlab.prefs in the EMGlab directory.

Nyquist rate. This specifies the highest expected frequency in the signal. The default value is 5 kHz, which is appropriate for many needle or fine-wire EMG signals. In order to reduce memory and computation demands, the program downsamples EMG signals to the lowest sampling rate greater than 2 times the Nyquist rate. For example, if a signal has a sampling rate of 33 kHz and the Nyquist rate is equal to 5 kHz, then the program will downsample the signal to 11 kHz for processing. This only affects the way the signal is represented inside the program and does not affect the way the signal is displayed or the analysis results.

Spike width. This specifies the template width to use when high-pass filtering is selected. The default value is 10 ms.

MUAP width. This specifies the template width to use when no high-pass filtering is selected. The default value is 60 ms, which is appropriate for wide-band signals (e.g., 5 Hz - 5 kHz), such as clinical EMG signals. If your signals are already high-pass filtered you may not need to apply additional filtering in EMGlab. In this case you should decrease this value to 10 ms (or whatever value is appropriate) to improve program efficiency.

Color scheme. There are three choices: "scope" (black background), "paper" (white background), and "custom." There is currently no way to set the custom colors interactively, but you can change them by editing the function emgsettings.m.

Drawing mode. The default value is "smoother." In some versions of Matlab, you can set the drawing mode to "faster" to make the graphical interface respond more quickly. This does not work in all versions of Matlab, however. If the cursors behaves incorrectly in "faster" mode, switch back to "smoother" mode.

Dragging animation. The default value is "on." If scrolling and dragging are very sluggish because of slow computer speed, you can turn dragging animation to "off." Then you will only be able to use clicking, the < and > buttons, and the arrow keys to move around. You will still be able to drag templates, but you will not see them as you drag them.

You can also select "no auto scroll." In this case the panels only scroll once each time you press the < and > buttons. The other animation features are on. Use this option if the < and > buttons tend to stick so that scrolling continues even after you release the button.

Font size. This specifies the font size for the buttons and popups.

Buffer length. This specifies the maximum signal length (in seconds) that the program should load into memory at once. It also determines the maximum extent to which you can zoom out the Signal, Firing, and Navigation Panels. If your signals are short enough that you do not run into memory problems in EMGlab, you can set the buffer length to inf to force the program to always load the entire signal. If your signals are very long, you can set buffer length to a value shorter than the full length of the signal to improve program performance and prevent memory problems.

Use Genetic Algorithm. This specifies whether the MTL automatic decomposition algorithm should use the genetic algorithm to resolve superpositions. Using the genetic algorithm may improve the accuracy and completeness of the decomposition, but it will take much longer. The default value is 0 (off).

Copying and Pasting

You can copy firing patterns and templates between channels or between annotations. Select a unit and choose "Copy selected firing pattern" or "Copy selected template" from the Edit Menu. Or, deselect all units and choose "Copy all firing patterns" or "Copy all templates". Then go to a different channel, or open a different EMG file and select "Paste" from the Edit Menu. When you paste firing patterns, the program creates new units with these firing patterns and automatically averages the templates. When you paste templates, the program creates new units with these templates and no firing patterns.

You can use copying to perform spike-triggered averaging. Spike-triggered averaging is used to estimate the effect associated with each MUs in an auxiliary signal, such as force or surface EMG. Copy the firing patterns and then paste them into the channel containing the auxiliary signal. The program will automatically compute the spike-triggered averages. Select "re-average" from the Analyze Menu to improve the averages. Select "unfiltered" and zoom out the templates to see them in their entirety. (Note that if the auxiliary signal is not part of the original EMG file, or if it has a different sampling rate, you will have to load it as an EMG signal before you can paste the firing patterns.)

Another reason for copying firing patterns is to transfer them to a different channel of EMG data to avoid having to decompose it from scratch. Be aware that the templates might be offset because of conduction delays. If the recording sites are very far apart, the templates might be offset so much that they cannot be seen in the Template Panel. Select "Center all" from the Analyze Menu to shift them into view. If the conduction delay exhibits appreciable jitter, then the firing times in the second channel might not correspond exactly to those in the first, resulting in template misalignments. Some units from the first channel might not have detectable templates in the second channel, and not all the units in the second channel might have been detected in the first channel. (Note that these issues are handled automatically by the MTL decomposition algorithm.)

One reason for copying templates is to keep units ordered consistently in multiple contractions from the same muscle. Copy the templates from one signal to another, and then decompose that signal using those templates. Of course, this will only work if the MUAP shapes are sufficiently similar in the two signals.

Hiding Units

You can hide a unit by option clicking (normal clicking in some Mac versions) on the unit's number in the Template Panel and selecting "Hide" from the context menu. Hidden units are displayed in a dim color, and they are not subtracted from the residual signal in the Signal Panel. Unhide the unit by selecting "Hide" again from the context menu, or by selecting "Show all" from the Analyze Menu.

Hiding is useful if you copy firing patterns from one channel to a new channel, and some of them don't have detectable templates. The program does not try to adjust the positions of hidden units, or consider them for possible matches.

You can also use hiding to see the activity of just one or a few units with the activity of the other units subtracted out. In this case, hide the units you want to see. The residual signal in the Signal Panel will show the activity of the hidden units with the activity of the other units subtracted out.

Hiding works slightly differently in the Close-up Panel. In the Close-up Panel, hidden units are subtracted from both the signal and the residual. You may notice that units that discharge close to the edges of the panel are automatically hidden in the Close-up Panel. You can temporarily hide or unhide units in the Close-up Panel by shift-clicking on the template or unit number. Hidden units are locked in place and don't move if you manipulate the other templates or click the "R" button to resolve the superposition.

Comparing Annotations

EMGlab allows you to compare different annotations of the same signal. This is useful for comparing your annotation with a "gold standard" annotation, or for reconciling annotations from two different annotators.

Create the first annotation ("A") manually or load it using the Open Annotation command in the File Menu. Then open the second annotation ("B") using the "Compare annotation with ..." command in the Tools Menu. The program will match up the units in the two annotations and highlight the differences between them.

In comparison mode, the annotation name appears as "A vs B", with "A" in the normal color and "B" in the comparison color. The Template Panel shows all the "A" units in the normal color, plus any additional units that are in "B" but not "A" in the comparison color. In the Signal, Firing, and Navigation Panels, all the discharges that the two annotations agree on are dimmed, and all the discrepancies are highlighted. Discharges occurring in "A" but not "B" are shown in the normal color, and those occurring in "B" but not "A" are shown in the comparison color. The residual signal is based only on annotation "A". You can switch back to the normal display mode by toggling the "Highlight differences" command in the Tools Menu.

While in comparison mode you can edit annotation "A" as usual. You can't delete units or discharges that appear in the "B" color, because they are not part of annotation "A". You can, however, identify a spike as an instance of a "B" unit (e.g., by dragging the template to the Signal Panel). This will add the unit to annotation A.

If you want to edit annotation "B," you must select "Swap annotations" from the Tools Menu. This interchanges the roles of the two annotations. The annotation name says "B vs A". The units are re-ordered according to their order in annotation "B" and they are shown in the "B" color. Manual edits now affect annotation "B" rather than annotation "A," and selecting "Save" or "Save as..." from the File Menu saves the results in file "B".

Checking a practice annotation. After you have decomposed a practice signal, you can compare your annotation with the "correct" annotation. After completing your annotation, open the correct annotation using the "Compare annotation with ..." command in the Tools Menu. Any templates that you failed to identify and all the discharges you missed are shown in the comparison color. All the discharges you identified incorrectly (false positives) are highlighted in the normal color. You can click on the highlighted differences in the Firing Panel to display that part of the signal and see if you can find your mistake. As you correct your mistakes, the highlights are turned off. When all the discharges are dimmed, your annotation agrees with the correct one.

Comparing annotations from different annotations. You can compare annotations from two different people to see how well they agree. Load one annotation by selecting "Open annotation" from the File Menu and then load the other by selecting "Compare

annotations" from the Tools Menu. The highlighted discharges indicate the discrepancies between the two annotations. You can inspect each discrepancy to try to determine which annotation is correct. You can then edit one or the other annotation to reflect the correct result. (To edit annotation "B" you have to use "Swap annotations" from the Tools Menu.) Of course, it may not always be possible to determine which annotation is correct with complete certainty.

Decomposition Exercise 1

This section describes how to use EMGlab to decompose a simple signal. Run EMGLAB by typing "emglab" in the Matlab command window.

Select "Load EMG file" from the File Menu. Open the file R00108.heg in the "sample data" folder. The first 100 milliseconds of the signal are displayed in the Signal Panel and the entire 10 seconds are displayed in the Navigation Panel. This file has only one channel of data. The upper cursor in the Navigation Panel shows the interval displayed in the Signal Panel. The lower cursor shows the interval displayed in the Firing Panel.

Select the 1000 Hz high-pass filter. The filtered signal has sharp spikes and a flat baseline.

Select "Auto Decomp" from the Analysis Menu. The program analyzes the first 5 seconds of the signal (the interval shown in Firing Panel) and identifies 8 MUAP trains. The firing patterns are fairly regular, but some gaps are apparent. We will check the results and complete the decomposition of the first 5 seconds manually.

Looking at the Signal Panel, all the spikes in the first 100 ms have been identified. The residual signal is very small over the entire panel, indicating a good fit between the templates and the signal. Drag the cursor in the Firing Panel slowly to 2.45 s and inspect the residual. It remains small, confirming that the decomposition is correct.

At 2.456 s there is an appreciable unidentified spike in the residual signal. Looking at the Firing Panel, this time corresponds to a gap in the firing pattern of unit 8. Click on the peak of template 8 and drag it to the spike in the Signal Panel. The program automatically aligns it, and it gives a good fit.

There seems to be a gap in the firing pattern of unit 1 at about 3.9 s. Click in the middle of this gap in the Firing Panel. This selects unit 1 and advances the Signal Panel to that point. The red bar at the top of the Signal Panel shows the expected time of an occurrence of unit 1, but all the activity in the vicinity is already correctly identified. We conclude that unit 1 has an abnormally long inter-discharge interval.

There seems to be an extra discharge in the firing pattern of unit 4 at about 4.0 s. Click on it to select unit 4 and advance the Signal Panel. The program identified the spike at 4.0 s as an occurrence of unit 4, but the appreciable residual suggests this is not correct. Click on the spike in the Signal Panel to display it in the Close-up Panel. Option click on the number 4 in the Close-up Panel to remove this unit. Looking at the Firing Panel shows that the firing patterns of units 2 and 7 have gaps at this time, suggesting that they might be involved in this spike. Drag template 2 to the Close-up Panel and align it with the main spike. The residual looks like an occurrence of unit 7. Drag template 7 to the Close-up Panel. If you dragged the templates close enough to their correct locations, the program should have automatically aligned them to give a good fit to the waveform. If

not, click the R button. Finally, click the OK button to accept the new identifications. Note that the Signal and Firing Panels are not updated until you click the OK button.

There seems to be another gap in unit 1's firing pattern at 4.3 s. Click on the gap in the Firing Panel to advance there. Again there is no sign of a missed discharge. This must be another abnormally long inter-discharge interval of unit 1.

Finally, there is a gap in the firing pattern of unit 8 at 4.64 s. Click on it to advance there. There is clearly a missed occurrence of unit 8. Instead of dragging the template, you can also just shift click on the spike in the Signal Panel to identify it as an occurrence of the selected unit.

The firing patterns of the 8 units now seem complete for this section of the signal. Click on the "o" button in the Firing Panel to see the instantaneous firing rates. The instantaneous firing rates are all relatively smooth, and give no evidence of any errors. Click on template 1 to select it. Unit 1 has one of the lowest firing rates, and the long intervals do not stand out. Based on the small residual and the regularity of the firing patterns, we can be confident that this segment of the signal has been decomposed accurately.

Now we can proceed to the last half of the signal. Drag the lower cursor in the Navigation Panel to the last half of the signal (or just click in the lower part of the Navigation Panel at about 8 s.) Select "Auto Decomp" from the Analysis Menu. The program identifies many firings, but there are large gaps, especially for units 1 and 5. The program also finds another unit, but it is very small. Clicking on a few of its occurrences to inspect them in the Signal Panel reveals that it probably is a valid unit, but also probably too small to be identified reliably. Option click on the template to delete it. (Or option click on the template number and select "Delete unit" from the context menu. In some Mac versions of Matlab you get the context menu by normal on the number instead of option clicking.)

Try to decompose the rest of the signal by yourself. In order to check for missed discharges at the edges of the Firing Panel, click on an empty part of the Firing Panel and drag to the right or left to scroll the display.

To check your results against the correct results, select "Compare annotation with..." from the Tools Menu and open the file R00108.eaf from the sample data folder. All the templates and discharges you identified correctly are dimmed, and all your mistakes are highlighted. Discharges highlighted in the usual template color are false positives, and those highlighted in pink or brown are misses. If everything is dimmed, then your decomposition is correct. (At least it agrees with ours.)

Decomposition Exercise 2

This exercise discusses some more advanced topics. The signal to be analyzed was recorded using a selective electrode (cut end fine wire) with the intention of isolating the activity of a single motor unit. The signal also contains several smaller units which we will attempt to decompose as well. The signal is very long (20 minutes). We will not attempt to decompose it entirely, but instead will "spot-decompose" it at several points. Depending on the particular research question, this might provide sufficient information, or a more thorough analysis might be necessary. The file for this exercise is not included in the EMGlab package because of its size. You can download it from www.emglab.net.

Run EMGlab by typing "emglab" at the Matlab command-line prompt.

Select "Load EMG file" from the File Menu. Select file type ".smr" from the file dialog. Load the file R01001.smr. (This file was created by the Spike 2 program.) The first 10 seconds of the signal are shown in the Navigation Panel. The small size of the slider shows that this 10-second interval is just a small portion of the signal. You can use the ||| button to the left of the Navigation Panel to zoom out to see more of the signal, but you cannot zoom out to show more than 50 seconds (the value of the "buffer length" parameter, which can be changed in Preferences.)

This file contains two auxiliary signals: force and surface EMG. Select "Auxiliary Signal Channel" and the "Force" from the View Menu to display the force. You can also turn the display on or off in particular panels using the View Menu. The program normalizes the display based on the data in the first signal buffer. You can change this by selecting "Auxiliary Signal Settings" from the View Menu. Appropriate settings for this signal are High pass filter = 2 Hz, Display max= 25, and Display min = -0.5.

In this signal, motor units are recruited during the first few seconds. Recruitment is usually accompanied by changes in the MUAP shapes, which makes decomposition more difficult. For this exercise, we will move to a more stable part of the signal to start decomposition. Drag the lower green cursor in the Navigation Panel until it lies between about 15 to 20 seconds. Click at 15 seconds in the upper part of the Navigation Panel to advance the Signal Panel.

One large unit can be clearly seen in the Navigation panel, but the Signal Panel shows several other identifiable smaller units. Although the baseline is fairly flat, select the 1000 Hz filter to improve the spike signal-to-noise ratio. Select "Auto decomp" from the Analysis Menu. The program should identify 7 units and fill in most of their firing patterns. Use the manual editing commands to fill in the gaps and check for any decomposition errors.

Notice the spikes at 18.331 and 18.363 seconds. They appear to be discharges of unit 1. The program did not classify them because they don't match template 1 exactly and their inter-discharge intervals are much shorter than normal. They are, however, most

likely anomalous discharges of unit 1. Appreciable changes in spike shape are often seen after a very short inter-discharge interval. The short interval affects the conduction velocity of the second discharge, which can affect the relative latencies of the individual components that make up the spike. In this case, a large part of the shape differences is due to a change in the relative latency of the small initial component with respect to the latency of the main peak. A similar shape change is seen after the short interval at 17.331 seconds. There also appears to be an anomalous discharge of unit 2 at 16.824 seconds.

There is also an 8th identifiable unit in this signal. It is quite small and might not have been found by the program automatically, but you can see it in the residual signal, for example at 16.382, 16.462, and 16.537 seconds. Since this signal has a high signal-to-noise ratio you could attempt to decompose this unit manually. Most of its occurrences can be detected reliably. Consider the discharge near 15.93 seconds. The residual signal suggests it could occur at 15.9204 or at 15.9337 seconds. Since the residual at the former time can be attributed to the variability of unit 1, the latter time is more likely,

Usually we would keep decomposing the signal segment by segment to get complete firing patterns and track any gradual changes in MUAP shapes. Let's try instead to jump a little further ahead. Drag the slider to show the interval 25-30 seconds in the Navigation Panel. Click in the lower third of the Navigation Panel to advance the Firing Panel to 25-30 seconds.

Select "Auto decomp" from the Analyze Menu. The program is able to still recognize the shapes of the larger units. It creates a new unit that looks very much like unit 8. (Your results may be slightly different, depending on the exact starting time of the analysis segment. The program may also create some other small unreliable units. You can go ahead and delete them.)

When you inspect the signal, you will find that the program made a number of mistakes. It misidentified several occurrences of unit 6 because of shape variability. It incorrectly found different trains for units 8 and 9, when in fact they both correspond to the same unit. To correct this you can merge the two units together, but then you have to check each discharge to decide if it is correct or not. Note that unit 1 continues to show anomalous discharges. There is another possible anomalous discharge of unit 2 at 28.064 seconds. The interval from the previous instance of unit 2 is very short, however. Perhaps this is not really a discharge of unit 2, but rather a new unit. Create a template for this spike and compare it with the spike at 16.824 seconds. They match quite well. This suggests that this may indeed be a new unit. Hide this unit since we don't have a very good template for it yet.

Now skip ahead to 100–105 s. The units have changed in shape. Automatic decomposition identifies many discharges, but again considerable manual effort is needed to correct check and correct mistakes. Note that an additional unit is active here. It must have been recruited between the last segment analyzed and this one.

Now skip ahead to 200-205 s. The units have changed even more. Rather than use the current templates, which probably do not fit very well, hide all the current units. Then run automatic decomposition. The program ignores the existing units and decomposes the segment from scratch. Now show all the units, and merge each new unit with the corresponding old one based on shape. Even though the shapes have changed, it is clear which goes with which. Unit 8 was not identified and must be decomposed by hand. There is also another small new unit. The superposition at 203.121 s is a good example of destructive interference. The superposition at 203.964 s is challenging, but the program is able to resolve it.

As you skip further and further ahead in this signal, the units continue to change. Some units lose their sharp details, some become smaller. At the same time new units are recruited, and the baseline becomes noisier. It becomes more challenging to decompose the segments and to match the units. In order to be absolutely certain that you are tracking the units correctly, you might need to take shorter jumps.

This signal illustrates the tradeoffs between "single-unit" analysis and decomposition. The discharges of the largest unit could be detected reliably using an amplitude criterion to provide the firing pattern of one motor unit with little or no manual interaction and only a brief computation time. Decomposition provides the firing patterns of eight or more simultaneously active motor units, but at a much higher interactive and computational cost. The choice of single-unit analysis or decomposition depends on the scientific question.

WFDB headers

For simple binary EMG data files, you can create a text header file to specify the file format instead of having to enter it every time you load an EMG signal. WFDB refers to a software package originally developed by MIT for viewing, analyzing, and creating recordings of physiologic signals. It is described at www.physionet.org. EMGlab recognizes a subset of the WFDB header specifications.

There should be a header file for each "record." In EMGlab terms, a "record" refers to the results of a single contraction. This can be a single EMG channel, multiple EMG channels simultaneously from different electrodes, or a combination of EMG data and auxiliary signals, such as force. The different channels and signals can be in the same or in different data files. Multiple channels of EMG data must have the same sampling rate. Auxiliary signals can have different sampling rates than the EMG data, but if so they must be in a different file.

The header file is a text file that contains a record line, one signal line for each channel of data, and optional comment lines. A full description can be found under "header – WFDB header file format" in the "WFDB Programmers Guide," on the [physionet.org](http://www.physionet.org) web site (<http://www.physionet.org/physiotools/manuals.shtml>). Briefly:

The **record line** contains the following fields:

record name: a string of characters that identifies the record.

number of channels: an integer greater than zero.

sampling rate: an integer or floating-point number, interpreted as samples per second per channel.

Each **signal line** contains the following fields:

file name: the name of the file containing the signal. If multiple channels are interleaved in a single file, then there should be a separate signal line with the same file name for each channel. If multiple channels are stored in separate files, then there should be a signal line for each channel with the appropriate name for each channel. The file name should either contain an absolute path name or refer to a file in the same directory as the header file or in the Matlab path.

format: an integer that specifies the storage format:

16 means 16-bit little-endian integer (the format used on PCs and Intel Macs)

61 means 16-bit big-endian integer (the format used on non-Intel Macs)

samples per frame: [optional]. This field is only needed when you want to include auxiliary signals that have different sampling rates than the EMG signals. In this case, set the sampling rate in the record line to 1, and put the sampling rate of each signal

as an integer or floating point number preceded by an "x" immediately after the format field.

byte offset: [optional]. If present, this field must be preceded by a "+" sign and must follow immediately after the format or samples per frame field with no intervening spaces. This field is an integer that tells how many bytes to skip over from the start of the file to get to the first sample of this signal.

ADC gain: [optional]. This is a floating point number that specifies the number of A/D units that correspond to one physical unit. If absent, EMGlab will plot signal amplitude in A/D units.

units: [optional]. If present, this field must be preceded by a "/" sign and must follow immediately after the ADC gain field with no intervening spaces. This is a string that specifies the physical units of the signal.

Comment lines must start with the "#" character. For files that will be shared on the EMGlab website, we recommend that comments have the following form:

item: value

where "item" indicates the type of information and "value" specifies the information. We suggest that the following documentary information be included. Items taking default values (shown in square brackets) can be omitted.

source: [natural] or synthetic
species: [human] or (other species)
gender: m or f
age: (in years)
diagnosis: [normal] or (type of disorder)
muscle: (name of muscle)
protocol: (e.g., isometric ramp contraction)
level: (level of contraction, in %MVC)
electrode: (type of recording electrode)
high-pass filter: (analog high-pass filter)
low-pass filter: (analog anti-alias filter)
contributor: (name of contributor)
date: (date on which the signal was recorded)
identifier: (this uniquely identifies the record in the contributor's data base)
contact: email address for contacting the contributor
remarks:

Protecting the Identity of Human Subjects. Data to be exchanged or posted on the EMGlab website must not contain any information that would reveal the identity of a particular human subject, either in the header file, the data files, or the file names.

Custom Readers

If your data files are in a format that cannot be described by a WFDB header, you can create a Matlab function for reading them. Name the function "read_XXX.m", where XXX is the file extension of your data files, place it in the EMGlab readers folder. EMGlab will automatically use this reader function when you open data files of this type.

There are two types of readers. If your data files are short enough to read in all at once, you can write an "all-at-once" reader. If your data files are too long to be read in all at once, you should write a "random access" reader.

All-at-once Reader. An all-at-once reader read an entire data file at once. It should use the following function statement:

```
[signal_info, data, status] = read_XXX (option, filename);
```

where:

- option (char) will always be 'open'
- filename (char) is the full filename of the file to be read
- signal_info (structure) provides information about sampling rate and scaling, as described more fully below
- data (int16 or double) is the data from the file, with one column per channel of data
- status (double) should be set to 1 if the reading was successful, or to 0 otherwise.

Random Access Reader. Random access readers read only a specified range of data at a time. They should use this skeleton:

```
[varargout] = read_XXX (option, varargin);
switch option;
case 'open'
    filename = varargin{1};
    ... % open the requested file
    varargout = {signal_info, private_info, status};
case 'load'
    [private_info, channel_range, start, count] = deal (varargin{:});
    ... % read the requested data samples
    varargout = {data};
end;
```

When called with option = 'open,' the function should open the requested file to make sure that it exists and is valid, ascertain the sampling rate and scaling information, extract any information (e.g., from the file header) that will be needed to access the file in subsequent calls, and then close the file. The file name and access information should be saved in the structure "private_info". EMGlab does not use this information, but it saves

it and provides in on future function calls so you will not have to re-calculate it. The following variables are relevant:

filename (char) is the full filename of the requested file
 signal_info (structure) provides information about the sampling rate and scaling, as described more fully below
 private_info (structure) is the private information
 status should be set to 2 (indicated that the function is capable of random access) if the opening is successful, or to 0 otherwise.

When called with option = 'load' the function should re-open the file, read the specified data points, and then close the file. The following variables are relevant:

private_info (structure) is the information to help you access the file
 channel_range (integer) is a two-element array containing the indices of the first and last channel for which data is requested (with 1 being the first channel)
 start (integer) is the index of the first sample (not byte) to be read, with 0 being the first sample in the file
 count (integer) is the number of samples to read per channel
 data (int16 or double) is the data read from the file, with one column per channel. The length of each column should equal count.

Signal_info. When called with option = 'open', the reader function must supply a struct array with one entry for each data channel. Each entry should contain the following fields:

name	name of the channel (char, optional, default = '')
rate	sampling rate in Hz (double, required)
start	starting time of signal in seconds (double, optional, default = 0)
gain	ADC units per physical unit (double, optional, default = 1)
units	name of physical units (char, optional, default = 'ADC')
length	length of data in samples (integer, required)

Data. The reader function should return unscaled data. The scaling will be done by EMGlab, using the information in signal_info. The returned data should be in int16 format if possible to reduce storage requirements.