Introduction to the NIRS AnalyzIR toolbox

Theodore Huppert PhD
Associate Professor
Dept of Radiology
University of Pittsburgh

Background

Started in 2014 by Jeff Barker, PhD (BioE grad student)
Currently 2,018 unique downloads
Average 37dn/day
Huppert Lab: +NIRS-toolbox

www.bitbucket.org/huppertt/nirs-toolbox

Software Specs
• Matlab based
  • 2014b or newer
  • Statistics and signal processing toolboxes
• Open-source
• Command-line based (few GUIs added)
• Mercurial ("CVS") control
• Wiki manual (in progress)
• Issue reporting
• Demo code and data
www.bitbucket.org/huppertt/nirs-toolbox

Software Specs
• Matlab based
  • 2014b or newer
  • Statistics and signal processing toolboxes
• Open-source
• Command-line based (few GUIs added)
• Mercurial (“CVS”) control
• Wiki manual (in progress)
• Issue reporting
• Demo code and data

Updating the toolbox via TortoiseHG GUI
https://tortoisehg.bitbucket.io

1) Install Windows or MacOS version
2) From GUI, “Clone” new repository
3) set source to https://bitbucket.org/huppertt/nirs-toolbox
Updating the toolbox via TortoiseHG GUI

https://tortoisehg.bitbucket.io

1) Install Windows or MacOS version
2) From GUI, “Clone” new repository
3) set source to http://bitbucket.org/huppertt/nirs-toolbox
Updating the toolbox via TortoiseHG GUI

https://tortoisehg.bitbucket.io

Two-stage update system

<table>
<thead>
<tr>
<th>Public Web Version</th>
<th>Push</th>
<th>Pull</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Change History</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Commit</td>
<td>Update</td>
<td></td>
</tr>
<tr>
<td>Working Version</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Updating the toolbox via Command Line (MacOS/Linux)

https://www.mercurial-scm.org/

Two-stage update system

<table>
<thead>
<tr>
<th>Public Web Version</th>
<th>Push</th>
<th>Pull</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Change History</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Commit</td>
<td>Update</td>
<td></td>
</tr>
<tr>
<td>Working Version</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Adding to Matlab Path

Note folders that start with "+" (e.g. /+nirs) denote Matlab namespaces and cannot be added directly to the path. You must add the parent folder containing this namespace.

1) Add folder
<root>/nirs-toolbox

2) Add with Subfolders
<root>/nirs-toolbox/external
<root>/nirs-toolbox/demos

+nirs/
NI RS analysis toolbox
+eeg/
EEG & MEG analysis toolbox
+dtseries/
Dense time series (CIFTI) formatted data (e.g. MRI [surface based] or MEEG/NIRS image reconstructed data
+advanced/
repository for work in progress, depreciated code, and features for non-general use.
<< use with caution >>
demos/
Example code for NIRS processing
external
Dependencies, and redistributed 3rd party code and GUIs called by toolboxes. This code is not intended to be called directly
Matlab Namespaces
- Defines context specific functions
- Called in form “nirs.io.loadDirectory” (+nirs/+io/loadDirectory.m)
- Overwritten methods (e.g. nirs.core.Data and eeg.core.Data)

Matlab Class Definitions
- custom classes (e.g. nirs.core.Data)
- contain common methods (“draw”, “table”, etc) that act on the local object.

Abstract classes
- parent object classes defining inheritance to children
- not intended to be called directly

Dependent variables
- “read-only” object fields that are computed on demand.
- E.g. Stats.p which depends on the beta, (noise covariance) and DFE fields
• **nirs.core.Data**
  main class for holding time-series data (also eeg.core.Data, dtseries.core.Data and nirs.core.GenericData).
  Encodes data, stimulus information, demographics, and probe
  >> raw.draw(<channels>)  plots the data

• **nirs.core.Probe**
  holds probe geometry, data labels (“link” table) and registration information (including head model)
  >> probe.draw()  plots the probe
  >> probe.defaultdrawfcn = “10-20”  sets the behavior (for registered probes)

• **nirs.core.ChannelStats**
  holds first and second-level statistical models
  >> stats.draw(‘tstat’,[-5 5],’q<0.05’)  draws the statistical map
  >> stats.ttest(‘A-B’)  does a t-test of two conditions

• **Dictionary**
  similar to use in Python
  holds variables based on keys
  MyDictionary(‘age’) = ?

• **nirs.design.StimulusEvents**
  holds discrete task events
  onset - onset time of events in seconds
  duration - duration of each event
  amp - amplitude of each event (used in parametric models)
  metadata - unused at moment (to encode reaction time etc)

• **nirs.design.StimulusVector**
  holds continuous task events (e.g. short-separation src-set data)
  amp - the regressor time-series data
  time - the sample times of data (interp’d to match nirs data)