Documentation of Nirs Analysis Scripts

L.Beichert(at)stud.uni-heidelberg.de

July 22, 2014

This file serves as a brief documentation of the Matlab-scripts found on `https://github.com/lbeichert/nirs-scripts`. The purpose of these scripts is to help with the analysis of multimodal nirs and systemic data.

1 Requirements

The scripts work with nirs/systemic(/phosphorus) data structs as they are created by the nirs data synchronisation scripts (`https://github.com/aot/ucl-nirs-analysis`). The data structs are expected to be save into files called 'pigData.m'. The sub-folder `/tools` must be included in the Matlab-path in order to be able to run the other scripts.

2 overviewPlots

Used like:

```
>> overviewPlots(pigDataNTB)
```

Creates an overview plot of every subject in the data-struct 'pigDataNTB' (use loadData to load) and saves it as .pdf and .fig file under `overviewPlots/output/`.
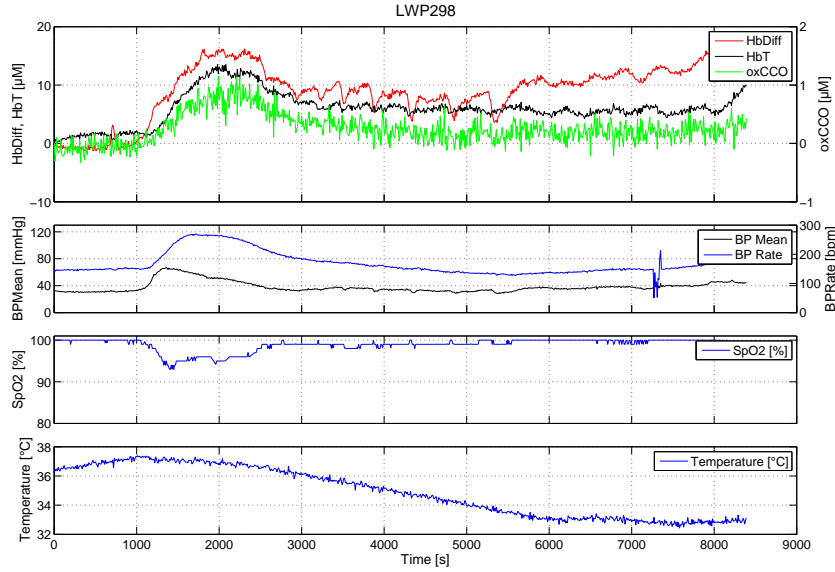
Figure 1: Sample overview plot created with overviewPlots

3   diffsBeforeAfter

Used like:

$$>> results = diffsBeforeAfter(pigData, signalName, tempStart,$$
$$tempTarget, plots)$$

where signalName = 'HbDiff'/'CtOx'/... , tempStart and tempTarget something like 37.0 and 33.5, plots = (true/false) plots wanted or not.
Alternatively use:

```
>> results = diffsBeforeAfterAll(pigData)
```

which calls diffsBeforeAfter for a number of important signals. tempStart/tempTarget are to be changed in the script.
The function takes 20 data points of a signal from around the time where temperature=tempStart and temperature=tempTarget and calculates the difference between the averages of the two sets of data points. It also performs a t-test or wilcoxon test in order to check if the calculated difference is statistically significant. The results are saved into two excel-sheets, one with the raw numerical data and one that is more readable.
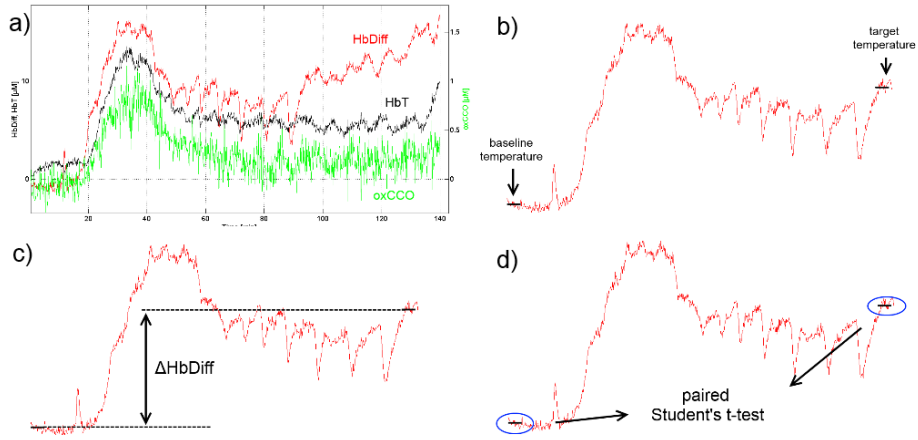
Figure 2: Sketch of data analysis performed by diffsBeforeAfter

4 recoveryFraction

Used like:

```
>> results = recoveryFractionAll(pigData, signalName, halfWidth)
```

where signalName='HbDiff'/'CtOx'/..., halfWidth=half width of averaging window.
The script provides a graphical user interface to determine the recovery fraction.

How to use:

- Start the script. A plot is displayed.

- Click into the plot where the baseline level of the signal is supposed to be taken. A black line symbolises the selected data points and the average value.

- If you are happy with the selection click the 'baseline' button. The average value and standard deviation is displayed on the button.

- Repeat for insult level and recovery level.

- Click the 'Done'-Button. A plot of the next individual appears.

- Once you are done with all the data set, an array recoveryData is returned each of whose line consists of: ID, value and std of baseline, insult and recovery, recovery fraction

- The recovery fraction is calculated like

$$recoveryFraction = \frac{recovery - insult}{baseline - insult} \cdot 100\%$$
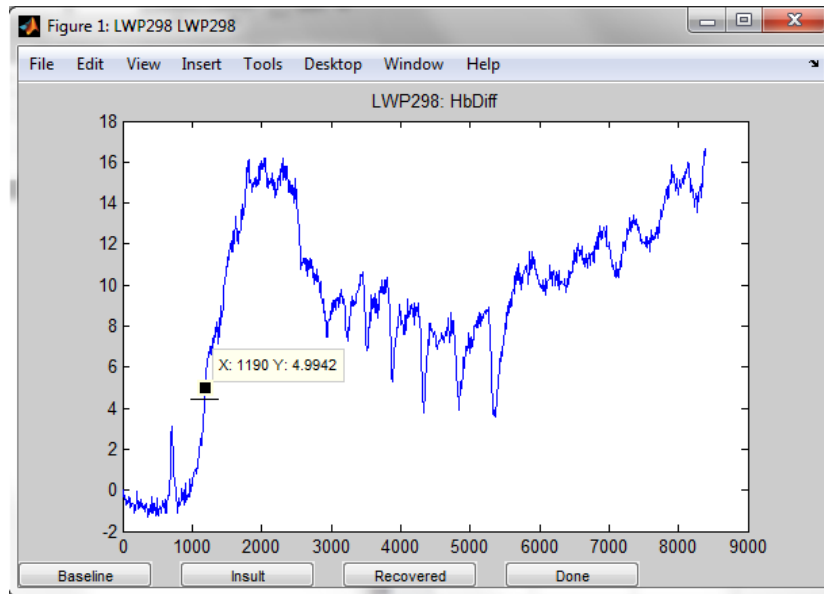
3

Figure 3: GUI of recoveryFraction (showing non-insult data)

5 tools

5.1 loadData(suffix)

Used like:

```
>> loadData('31p')
```

Loads pigData-structs of type 'suffix'(31p, 1h, cooling, ...) into workspace variable 'pigDataNTB'.

Expects the structs to be saved as files called 'pigData.m' under the location `pigletdatadir/argon-dex/output/suffix/`. The path to the data directory must be saved as MATLAB-appdata 'pigletdatadir'. This can be done by copying the 'startup.m' file from `/tools` to the home directory of MATLAB and editing it according to the local path or by integrating the lines into a preexisting startup-file.