Matlab Software for Pulse Oximetry

Thomas Jensen

October 22, 2009

Contents

1 Introduction

This document describes five Matlab functions for estimating the optical ratio R in pulse oximetry. It is presumed that basic understanding of pulse oximetry and algorithms for saturations estimation is known, see [1] and [2]. The functions uses either Independent Component Analysis (ICA) algorithms from the ICA:DTU Toolbox[1] or FastICA[2] or an implementation of the Masimo Discrete Saturation Transform (DST) algorithm. The ICA packages must be downloaded separately from their respective web pages. Scripts for demonstrating how the functions can be used and results evaluated are described as well. The software is online available[3] together with a small dataset for demonstration.

[1]`http://isp.imm.dtu.dk/toolbox/ica/`
[2]`http://www.cis.hut.fi/projects/ica/fastica/`
[3]`http://isp.imm.dtu.dk/pulseoximetry/`

2 Description of programs

2.1 Overview of files

Each of the five functions for estimating R divides data into windows of a given length and applies the same preprocessing by normalization and bandpass filtering. R is then estimated for each window. The names of the functions implies which algorithm they use:

- `icaMLsat.m` - function for estimating R using the Maximum Likelihood ICA algorithm from the ICA:DTU Toolbox.
- `icaMSsat.m` - function for estimating R using the Molgedey and Schuster decorrelation ICA algorithm from the ICA:DTU Toolbox.
- `icaMFsat.m` - function for estimating R using the Mean Field ICA algorithm from the ICA:DTU Toolbox.
- `fastICAsat.m` - function for estimating R using the FastICA algorithm.
- `masimoDSTsat.m` - function for estimating R using the Masimo DST algorithm.
  - `masimoDST.m` - subroutine with an implementation of the DST algorithm used by `masimoDSTsat.m`.

Make sure the path to the respective ICA-package are correct according to the location of the ICA-packages. This can be set in the first line of code in each of the functions depending on ICA algorithms.

Beside the above listed functions, the following scripts and subroutines can be used for demonstration and evaluation:

- `evalEstimates.m` - script for evaluation of R estimates against reference S_pO_2 measurements using a training- and test-set topology.
- `loocv.m` subroutine used by `evalEstimates.m` to fit a linear model as calibration curve using the Leave-One-Out method.
- `demo_sat.m` is intended for demonstrating how data can be loaded and R estimated and evaluated.

In figure 1 it is illustrated how the scripts and subroutines are connected.

2.2 Functions for estimating R

The usage of the five functions for estimating R is similar except `masimoDSTsat.m` that takes one more input argument. The functions are called by e.g.:

```
R = icaMLsat(X, fs, winLength, settlTime, overLap, filterType, order, fl, fh)
```

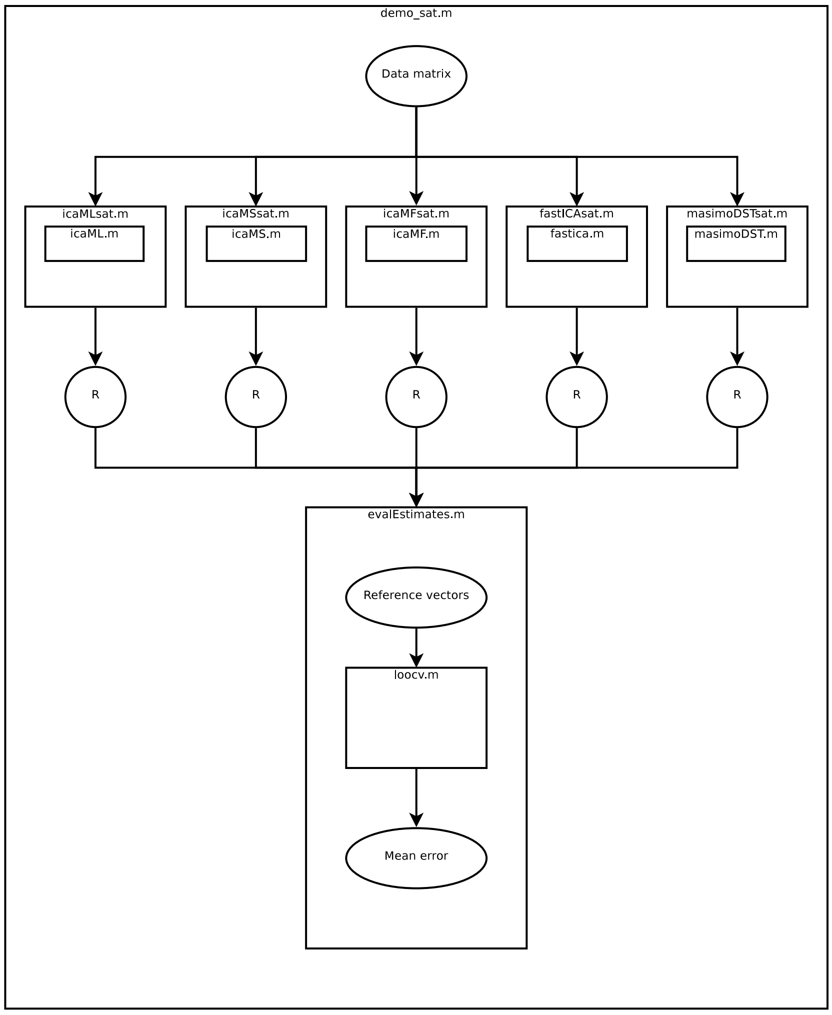

Figure 1: Schematic illustration of scripts and functions. Squares indicates programs and ellipses indicates data.

The input arguments are explained in the following:

X	is a matrix of data from recordings. Data from the red channel is stored in the first column and data from the infrared channel is stored in the second column.
fs	is the sampling frequency. All other time dependent parameters are relative to the sampling frequency.
winLength	determines the length of windows in seconds used to estimate R.
settlTime	can be used to extend the window length in seconds to compensate for filter settling distortion. The extension is then removed again after filtration.
overLap	can be used to make the windows overlap and is specified in seconds. Set to `0` (zero) for no overlap.
filterType	can be set to either `1` or `2` to choose between FIR or IIR filters respectively.
order	specifies the order of the filter.
fl	specifies the lower cut-frequency of the filter.
fh	specifies the higher cut-frequency of the filter.

The usage of `masimoDSTsat.m` is similar:

```
R = masimoDSTsat(X, fs, M, winLength, settlTime, overLap, filterType, order, fl, fh)
```

with one additional parameter:

M	is the order of the adaptive filter used by the DST algorithm.

2.3 ICA algorithms

All ICA algorithms are well documented in their respective references and will not be covered in this report. icaML and icaMS does not require any configuration as icaMF does. Configuration of icaMF has to be done in `icaMFsat.m`. Type `>> help icaMF` in Matlab for details on the parameters. (Make sure you are in the right directory or icaMF is added to the path). An example of how it can be configured is shown below:

```
% Setup ICAMF, see 'help icaMF' for info.
PAR.Sprior = 'combi';           % Use different prior pdf. for each source signal.
PAR.sources = 1;                % Number of source signals.
PAR.S(1).prior = 'bigauss';     % Prior pdf. of source 1.
PAR.S(2).prior = 'mog';         % Prior pdf. of source 2.
PAR.S(3).prior = 'mog';         % Prior pdf. of source 3...
PAR.Sigmaprior = 'diagonal';    % Individual noise on each sensor (wave-length).
PAR.solver = 'ec';              % Seems to be fastest.
PAR.optimizer = 'em';           % Seems to be fastest.
PAR.draw = 0;                    % Do not draw!!!
PAR.Aprior = 'positive'; % Constrain mixing matrix to be positive.
% PAR.A_init = rand(2,PAR.sources); % Positive initial mixing matrix.
```

The most important parameters to set is the number of source signals and prior probability distribution of each source. The mixing matrix **A** in the ICA model can be constrained to be positive as shown in the code above. Using more than one source signal, the mixing matrix could alternatively be initialized by small positive numbers two allow the signs of the e.g. motion artifact signal to take negative values.

The FastICA algorithm can be configures in several ways as well. Type `>> help fastica` in Matlab for details on the parameters. (Make sure you are in the right directory or FastICA is added to the path). Below it is shown how FastICA is called in `fastICAsat.m`:

```
[SS AA WW] = fastica(Xt', 'g', 'skew','maxNumIterations',10e4);
```

2.4 Masimo DST

The Masimo DST algorithm is implemented in the function `masimoDST`. The function call is:

```
[r ppg noise] = masimoDST(X, M, R, stepSize)
```

where the input arguments are:

X	is a matrix of data from recordings. Data from the red channel is stored in the first column and data from the infrared channel is stored in the second column.
M	is the order of the adaptive filter used by the DST algorithm.
R	is a vector specifying the interval to search for r_a the arterial saturation.
stepSize	specifies the step size that increases r in each iteration.

If only X is set, default values for M, R and stepSize are used.

The output arguments are:

r	is the arterial saturation.
ppg	is the separated PPG signal.
noise	is the separated noise signal.

2.5 Evaluation of Estimates

To evaluate the estimates of R, the script `evalEstimates.m` can be used. This script loads the matrices with references S_pO_2 measurements and sets up three combinations of training- and test-sets. Notice that this script can only be used with the provided data-set and for 30s window length. The subroutine `loocv` is called to fit a linear model as calibration curve between estimates of R and reference S_pO_2 measurements using the Leave-One-Out method. The script displays the mean Euclidean error.

2.6 Demonstrations of Programs

To demonstrate how the functions can be used, the script `demo_sat.m` can run by typing `>> demo_sat`. This scripts loads data set, estimates R and evaluates the estimates against reference S_pO_2 measurements by running the `evalEstimates.m` script. Plot of estimates and calibration curve are displayed. The different functions for estimating R can be chosen by uncommenting/commenting the function calls in the file. To load and save other data sets, see the documentation for the Matlab functions `load` and `save` by typing `>> help load` and `>> help save` respectively. Make sure that the data is organized correctly according to the function usage.

References

[1] Thomas Jensen, "Signal Processing of Nano Sensor Data", *Academic dissertation (M.Sc.), IMM-M.Sc.-2009-13*, 2009.

[2] Thomas Jensen, Sune Duun, Jan Larsen, Rasmus G. Haahr, Mette H. Toft, Bo Belhage, and Erik V. Thomsen, "Independent Component Analysis Applied to Pulse Oximetry in the Estimation of the Arterial Oxygen Saturation (S_pO_2) - a Comparative Study", *Engineering in Medicine and Biology Society*, 2009. EMBS 2009. 31th Annual International Conference of the IEEE.