Analysis of time-resolved scattering data at CoSAXS

Summary

Because of the nature of TR-XSS experiments there is not a well defined software. Instead analysis and manipulation of data collected within the time-resolved setup at CoSAXS is performed by a collection of MATLAB functions. These functions aim to provide the user with a relative simple means of manipulating and visualizing the data in a useful manner. The user may use the analysis scripts simplePlot.m and rrAnalysis.m as a starting point to set up their own, personalized analysis code. The following sections will describe the various parts of the TR-XSS analysis tools.

Calibration

Use for example PyQtFAI to set up PONI and mask files for the SAXS data processing. Radial integration of SAXS data is performed using the online pipeline.

WAXS data is recorded using the Mythen2 1K detector. This is an array detector, which means that no further radial integration is needed. To translate the pixel indices on the detector the matlab function prepareWAXScalib can be used.

prepareWAXScalib

This function is run from the MATLAB command window. Assuming that the current naming convention for images is the same as what is recognized by prepareWAXScalib you simply call the function as:

>> prepareWAXScalib('path/to/data',<scannumber>)

In case the current name is not recognized you can instead call the function as

>> prepareWAXScalib('full/path/to/data/including/filename',[])

prepareWAXScalib will load the data and present it as a plot (see below). The function is interactive and will ask you for the number of peaks, their position and corresponding q-values.

Finally, the function will also ask you to name the calibration file. At this step the function will ask if you want to overwrite or append to an existing file. The q-vs-index calibration will then be plotted

and a straight line fitted to the entered data. This allows you to judge if there is a linear dependence q-vs-index or not.

You may repeat the process several times, each time loading an image with a different calibrant, entering indexes and q-values and compile all of these into one WAXS calibration file.

The calibration file will later be used when data is loaded and in order to calculate the WAXS q-range.

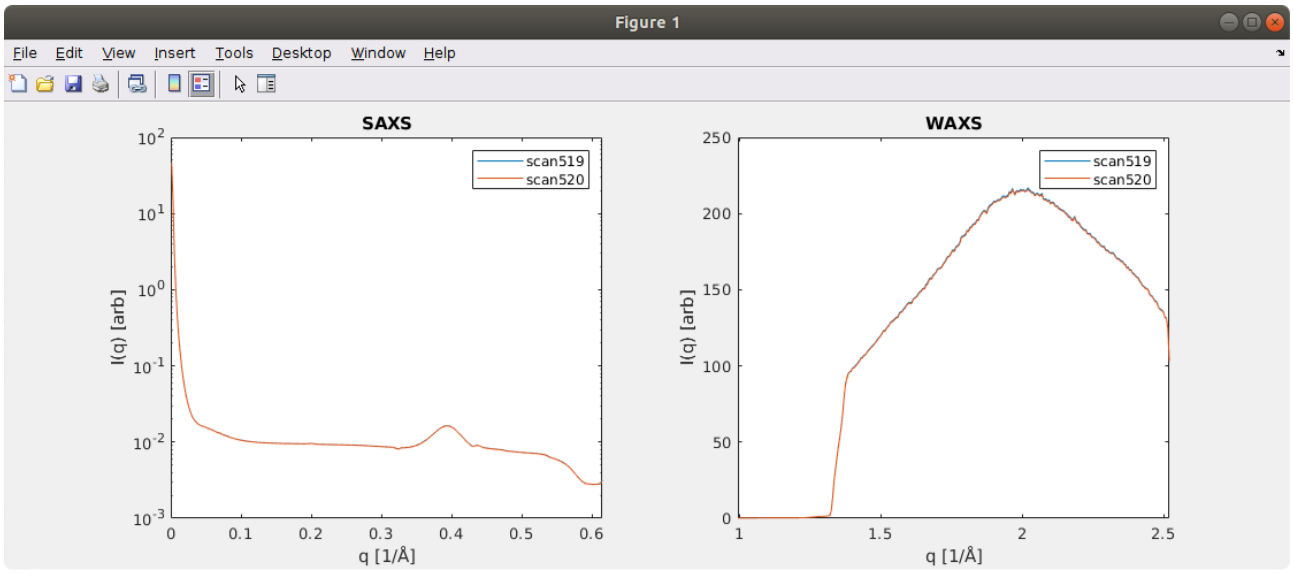
Data analysis

simplePlot

simplePlot is a very simple plotting script. It loads data from scans and averages all scattering curves within one scan, after which it plots them on top of each other. It may serve as a template starting from which more elaborate plotting can be designed. For example to load, average, and save scattering data that are to be used or further analyzed later.

simplePlot.m

```
% A simple load data and plot script. This doesn't perform any outlier
% rejection, normalization or calculation of difference scattering.
% It simply loads and avererages data on a per scan basis.
close all
clear
%%
dataFolder.SAXS = './sampleData/SAXSdata'; % SAXS data are found at...
dataFolder.WAXS = './sampleData/WAXSdata'; % WAXS data are found at...
WAXScalib = 'waxscalib.txt'; % Mythen calibration file

dq=0.005; % The desired q-spacing
loadSAXS = true; % Load SAXS data (or not)
loadWAXS = true; % Load WAXS data (or not)
scanNumbers = [519 520]; % which scans to load
```



If, for some reason, the file naming convention changes, you will need to update lines 34 and 64 accordingly.

```
32 -    if loadWAXS
33 -        WAXSfile = [dataFolder.WAXS filesep sprintf('mythen_scan_%u_data.hdf5',scannumber)];
34 -        tmpInfo = h5info(WAXSfile);
```

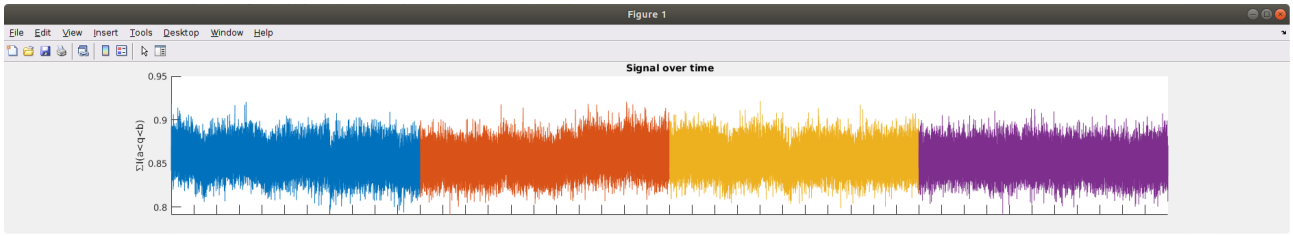
```
63 -    if loadSAXS
64 -        SAXSfile = [dataFolder.SAXS filesep sprintf('eiger_scan_%u_data.hdf5',scannumber)];
65 -        tmpInfo = h5info(SAXSfile);
```

rrAnalyze

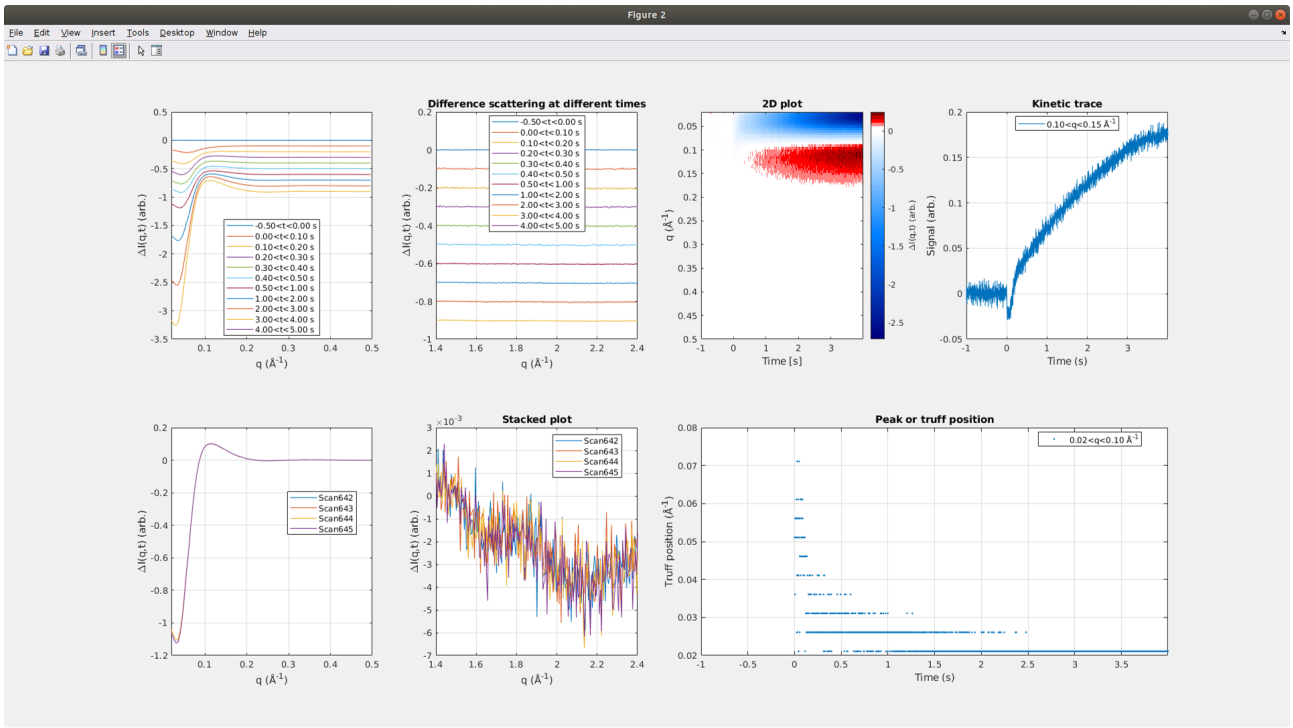
rrAnalyze is the main data analysis script. It loads the data via rrLoad (which will be described below) and shows how the data may be plotted to reveal various features. It uses a set of plotting functions, aimed to make it easy to set up a common set of plots (stackPlot, qt2DPlot, kintracePlot, peakPosPlot). You can also subtract the solvent heat contrbution from the data (subtractHeat) and perform SVD analysis (trxssSVD). Use rrAnalyze as inspiration and a starting point to generate an analysis script that suits your experiment.

rrAnalyze.m

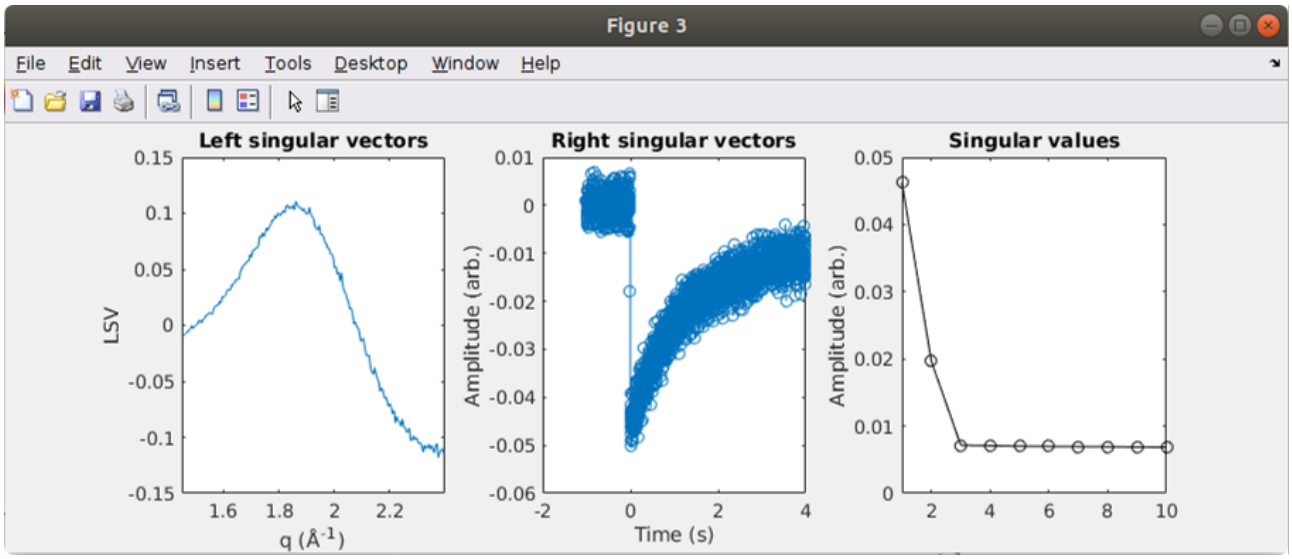
```
% This script shows an example on how the data analysis could be set up.
%%
close all
clear
scrsz = get(0,'screensize'); % get info on the screensize - might be useful
%%
% Where to find the data
dataFolder.SAXS = './sampleData/SAXSdata'; % Wgere to find SAXS data
dataFolder.WAXS =  './sampleData/WAXSdata'; % Where to find WAXS data

% Info on the setup
setupInfo.WAXScalib = 'waxscalib.txt'; % Mythen calibration file
setupInfo.detector_readoutrate = 500; % Hz - to get the time-vector correct
setupInfo.nStepsPerCycle = 2500; % the number of steps per cycle.
setupInfo.dq = 0.005; % the q-step. For rebinning the q-vector.

%%
scanNumbers = [715:718]; % Which scans (images) to load

timeShift = 1; % The detector-laser offset, s
imPerCycle = setupInfo.nStepsPerCycle; % the number of images per cycle
replot = false; % just replot the data, not reloading
timeSlices = [-0.5, 0, 0.1, 0.2, 0.3 0.4, 0.5, 1, 2, 3, 4, 5]; % how to div
qSlices = [1.6 2.4];% different q-ranges to look at, for kinetic tracing

plot_dI = true; % whether to plot I or dI
qPower = 0; % plot data as q^qPower*dS
qRange_SAXS = [0.02,0.5]; % show the data in this q-range (1/Å)
qRange_WAXS = [1.4 2.4];
smoothspan = 1; % smoothing
qScale = 'linear'; %'linear' or 'log'
offset = 0.1; % The curves are offset with this value

% SVD
nComp_SVD =1; % number of SVD components to plot
qRange_SVD = [0.03 0.5]; % over which q-range do you want to perform the SV
qPower_SVD = 1; % what qPower would you like to use for your SVD analysis

monRange = [0.01 0.1]; % over which q-range to monitor drifts in data.

% Outliers and normalization
outlierRange = [2.02 2.12]; % look for outliers in this q-range
outlierLevel = '0.2prcnt'; % 'Xprcnt', 'Xsigma' or 'Xunits'
normRange = [1.45, 1.55]; % normalize to the scattering in this range

% Heat subtraction
do_subtractHeat = false; % false = do not subtract heat, true = do subtract
heatData = 'buffer_heating.mat'; % the file with the heat data. The heat da
qRange_heat = [1.45 2.4]; % q-range in which to scale heat data to regular

```



You can monitor how the signal changes over time in a certain q-range. This is defined by monRange and might give indications about drifts during the experiment.



This figure shows some plots that may be useful. Please refer to the tutorial to get more in depth information on what the various plots might tell you.



SVD analysis may help to figure out how many events are present in the data.

rrLoad

The function that loads data and calculates difference scattering (along with outlier rejection and normalization).

If the fileformat or naming is changed additional changes may have to be done in the code of rrLoad.

```
42      % Define and get some info from the datafiles
43 -    WAXSfile = [dataFolder.WAXS filesep sprintf('mythen_scan_%u_data.hdf5',scanNumber)];
44 -    SAXSfile = [dataFolder.SAXS filesep sprintf('eiger_scan_%u_data.hdf5',scanNumber)];
68 -      frameInd = h5read([dataFolder.RAW filesep sprintf('eiger_scan_%u_data.hdf5',scanNumber)],'/entry/data/meta');
```

Auxillary analysis scripts

stackPlot.m

This function helps set up some of the most common and useful plots of your data.

qt2DPlot.m

This function sets up a 2D plot where the data is plotted against q and time.

kintracePlot.m

This function setus up a plot where the change in, either summed or average, scatteirng is monitored over time.

peakPosPlot.m

This function sets up a plot which attempts to track peak or truff q-position over time.

trxssSVD.m

This function performs an SVD analysis of the data and plots the result.

subtractHeat.m

This function scales and subtracts data corresponding to heated solvent.

qAver.m

This function calculates the average scattering over a certain q-range.

qSum.m

This function calculates the summed scattering over a certain q-range.

qCut.m

This function cuts the data over a certain q-range.

normalize.m

This function performs normalization of data.

filter1D.m

This function analyzes the data for outliers.

Tutorial

This tutorial will give an example of how the different scripts and functions are used, and what the output means. Note that it will only give info on what you might provide as input and what output will come of it. For further details you are referred to the scripts themselves (in particular rrAnalyze).

Sample data is found in ./sampleData/SAXSdata and ./sampleData/WAXSdata. The scan numbers are as follows:

For calibration of Mythen: scan440 (Lupolen), scan442 (LaB_6)

Water at different temperatures: scan519-520 (20 ^oC), scan523-524 (30 ^oC)

T-jump, solvent contribution: scan715-718 HEPES T-jump starting form 20C

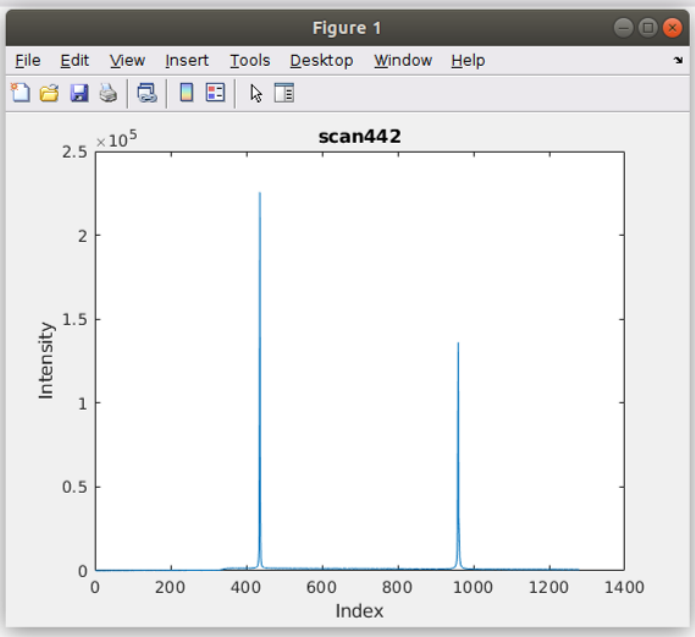
Protein sample: scan642-645 Lysozyme, T-jump starting from 20C

Calibration

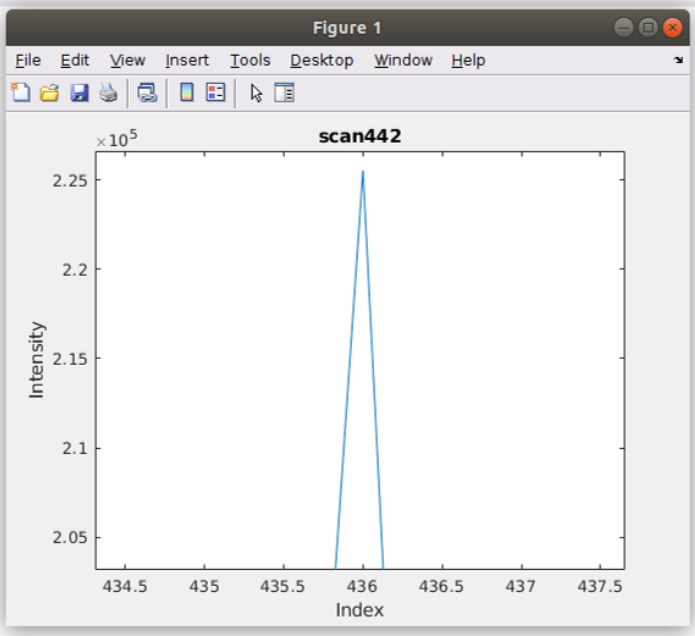
Prepare a calibration file for the Mythen detector using LaB_6 (scan442) and perhaps Lupolen (scan440). These compounds have peaks within the typical Mythen range at 1.512 $Å^{-1}$ and 2.138 $Å^{-1}$ (LaB_6) and 1.53 $Å^{-1}$, 1.69 $Å^{-1}$ and 2.12 $Å^{-1}$ (Lupolen).

```
>> prepareWAXScalib('./sampleData/WAXSdata',442)
```

```
Command Window
>> clear
>> prepareWAXScalib('./sampleData/WAXSdata',442)
Enter number of peaks:
```

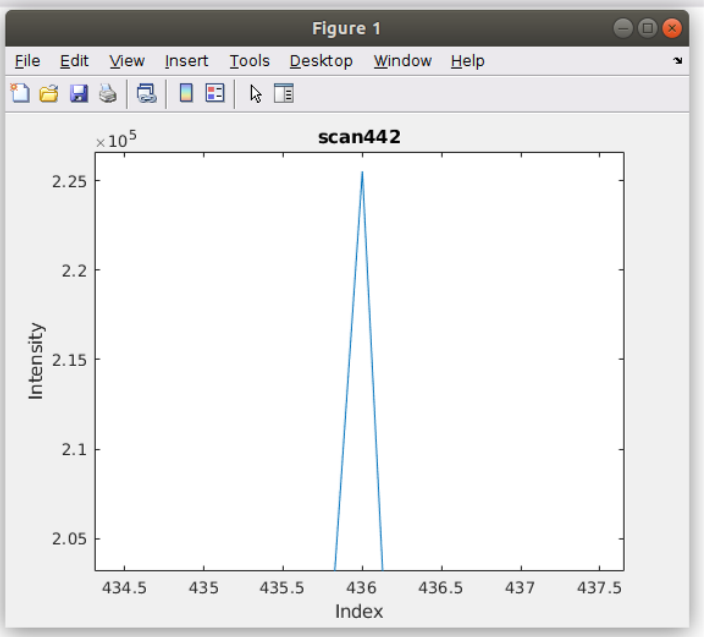



```
Command Window
>> clear
>> prepareWAXScalib('./sampleData/WAXSdata',442)
Enter number of peaks: 2
Enter index for position of peak 1: 436
```



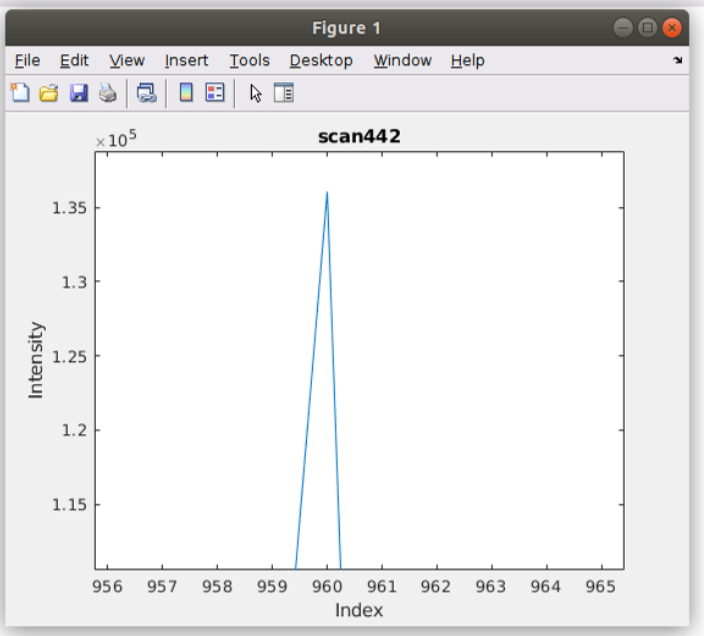
Command Window

```
>> clear
>> prepareWAXScalib('./sampleData/WAXSdata',442)
Enter number of peaks: 2
Enter index for position of peak 1: 436
Enter q-value for peak 1: 1.512
```



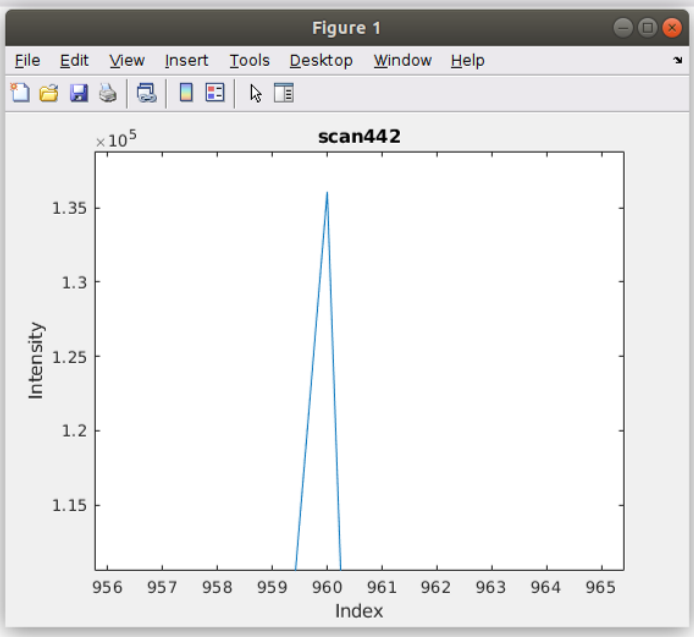
Command Window

```
>> clear
>> prepareWAXScalib('./sampleData/WAXSdata',442)
Enter number of peaks: 2
Enter index for position of peak 1: 436
Enter q-value for peak 1: 1.512
Enter index for position of peak 2: 960
```



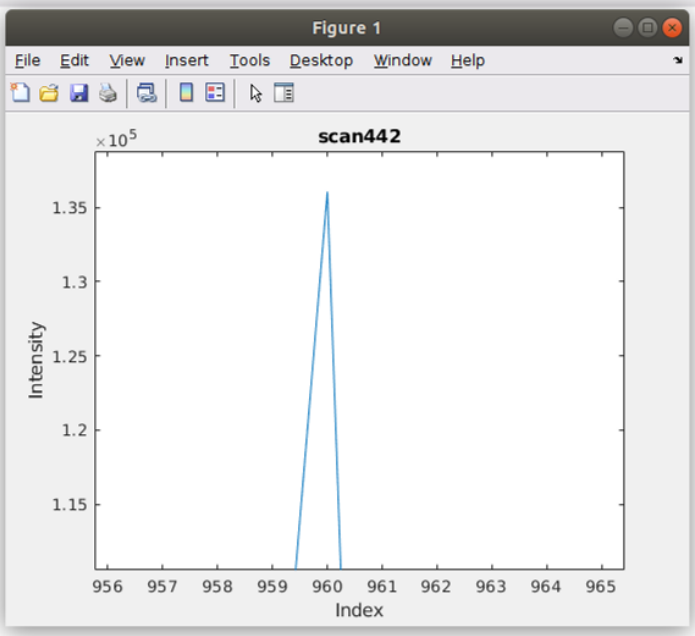
Command Window

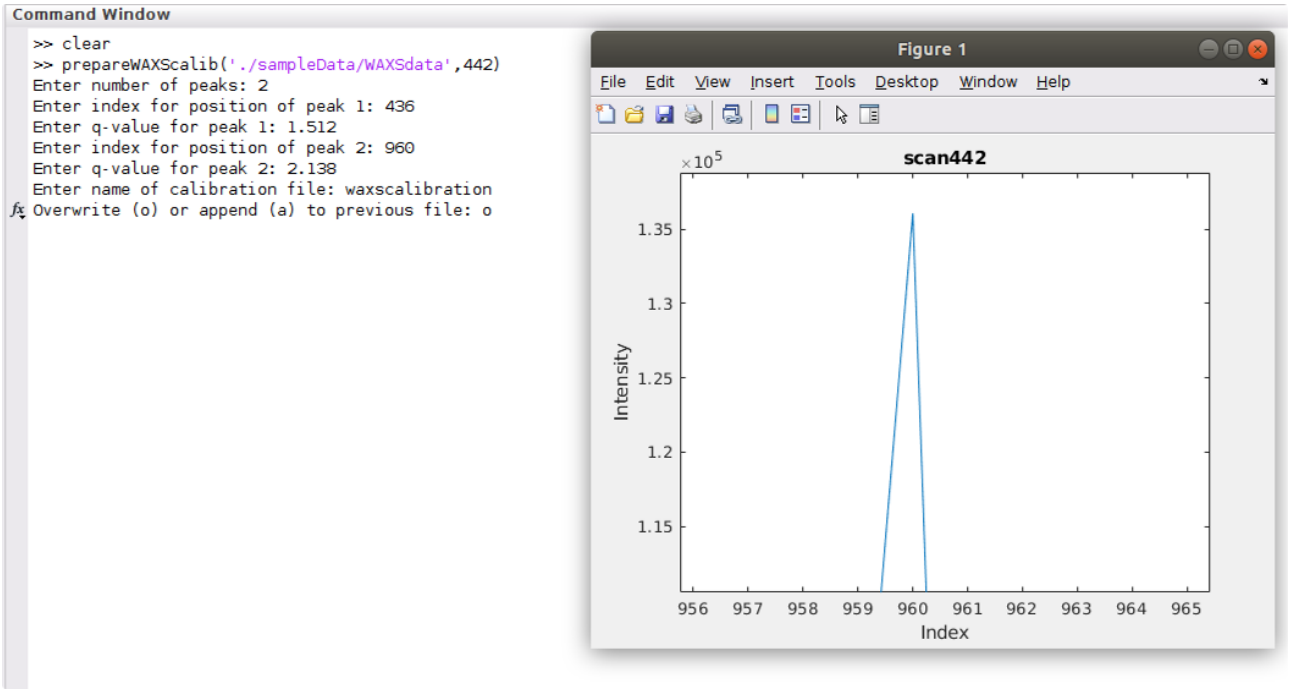
```
>> clear
>> prepareWAXScalib('./sampleData/WAXSdata',442)
Enter number of peaks: 2
Enter index for position of peak 1: 436
Enter q-value for peak 1: 1.512
Enter index for position of peak 2: 960
Enter q-value for peak 2: 2.138
```

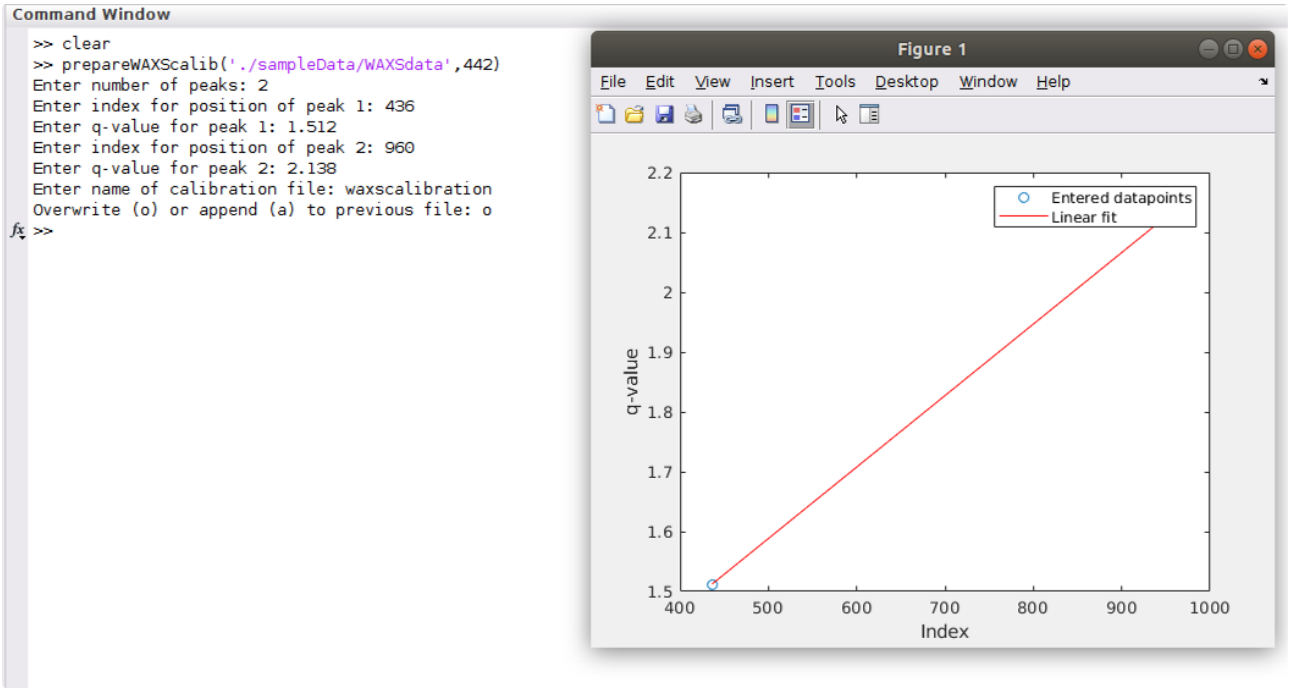


Command Window

```
>> clear
>> prepareWAXScalib('./sampleData/WAXSdata',442)
Enter number of peaks: 2
Enter index for position of peak 1: 436
Enter q-value for peak 1: 1.512
Enter index for position of peak 2: 960
Enter q-value for peak 2: 2.138
Enter name of calibration file: waxscalibration
Overwrite (o) or append (a) to previous file:
```







After this you can repeat the process with another calibrant.

Data analysis

Open, or make a copy, of rrAnalyze.m. Start by adding the paths to SAXS and WAXS data, the number of points per step, and the detector readout frequency. Also set the path to the Mythen

calibration file you just prepared. These settings will likely remain the same for much of the experiment.

```
% Where to find the data
dataFolder.SAXS = './sampleData/SAXSdata'; % Wgere to find SAXS data
dataFolder.WAXS =  './sampleData/WAXSdata'; % Where to find WAXS data

% Info on the setup
setupInfo.WAXScalib = 'waxscalib.txt'; % Mythen calibration file
setupInfo.detector_readoutrate = 500; % Hz - to get the time-vector correct
setupInfo.nStepsPerCycle = 2500; % the number of steps per cycle.
setupInfo.dq = 0.005; % the q-step. For rebinning the q-vector.
setupInfo.scaleWAXS = 2.86E-4; % approximately puts SAXS and WAXS data on t
```

The scanNumbers defines which data you want to look at, timeShit is how many seconds from the start of detector readout the laser pulse is triggered.

```
scanNumbers = [715:718]; % Which scans (images) to load

timeShift = 1; % The detector-laser offset, s
imPerCycle = setupInfo.nStepsPerCycle; % the number of images per cycle
replot = false; % just replot the data, not reloading
```

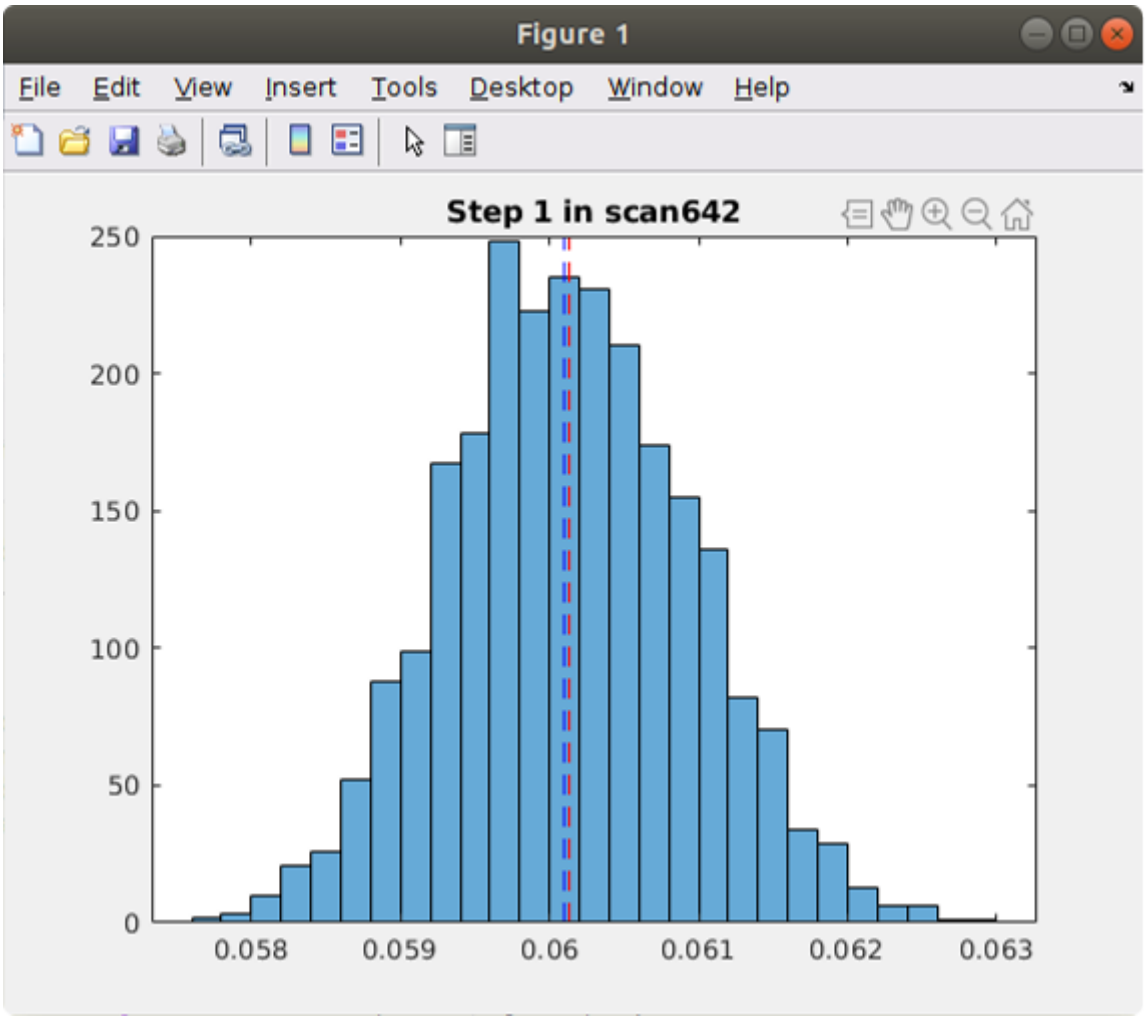
Next are various settings to get the plot the way you prefer. These are likely to change throughout the experiment, depending on what within the data you want to focus on. In particular, timeSlices reduces the (in this case) 2500 datapoints to 11 points. Using this you would probably average together data during times when not much is going on. Using qSlices (where each row is one range) you can monitor the change in scattering over a certain q. It is common in TR-XSS analysis to weight the data by q (to plot qΔI vs q). This is to make features at higher q more visible.

```
timeSlices = [-0.5, 0, 0.1, 0.2, 0.3 0.4, 0.5, 1, 2, 3, 4, 5]; % how to div
qSlices = [1.6 2.4];% different q-ranges to look at, for kinetic tracing

plot_dI = true; % whether to plot I or dI
qPower = 0; % plot data as q^qPower*dS
qRange_SAXS = [0.02,0.5]; % show the data in this q-range (1/Å)
qRange_WAXS = [1.4 2.4];
smoothspan = 1; % smoothing
qScale = 'linear'; %'linear' or 'log'
offset = 0.1; % The curves are offset with this value
```

You can also perform various types of analysis. You can define the range and q-power you want to use for an SVD analysis and the number of components you want to plot. To monitor how scattering changes form one scan to another you define in monRange. This can give you information on drifts in the data.

```
34      % SVD
35 -    nComp_SVD =1; % number of SVD components to plot
36 -    qRange_SVD = [0.03 0.5]; % over which q-range do you want to perform the SV
37 -    qPower_SVD = 1; % what qPower would you like to use for your SVD analysis
38
39 -    monRange = [0.01 0.1]; % over which q-range to monitor drifts in data.
40
41      % Outliers and normalization
42 -    outlierRange = [2.02 2.12]; % look for outliers in this q-range
43 -    outlierLevel = '0.2prcnt'; % 'Xprcnt', 'Xsigma' or 'Xunits'
44 -    normRange = [1.45, 1.55]; % normalize to the scattering in this range
45
46      % Heat subtraction
47 -    do_subtractHeat = false; % false = do not subtract heat, true = do subtract
48 -    heatData = 'buffer_heating.mat'; % the file with the heat data. The heat da
49 -    qRange_heat = [1.45 2.4]; % q-range in which to scale heat data to regular
```

It is common to normalize the data around 1.5 $Å^{-1}$ (because for water heating, this is an isosbestic point) and to reject outliers based on changes in the water scattering. If you are doing a T-jump the water scattering will change quite a lot, so it may be advisable to focus on a q-range where the scattering does not change quite as much (1.5 $Å^{-1}$ or 2.07$Å^{-1}$). You will have to try different outlierlevels to figure out what works best for your data. The current outlier rejection is based on comparing the mean and median of a step within a scan. This is roughly expected to follow a normal distribution (see image below). In the case of something happening that has a pronounced impact on the data, this is likely to change, for example to become bi-modal. In this case, the histogram will be plotted within rrLoad, and some feedback will be provided in the command window.

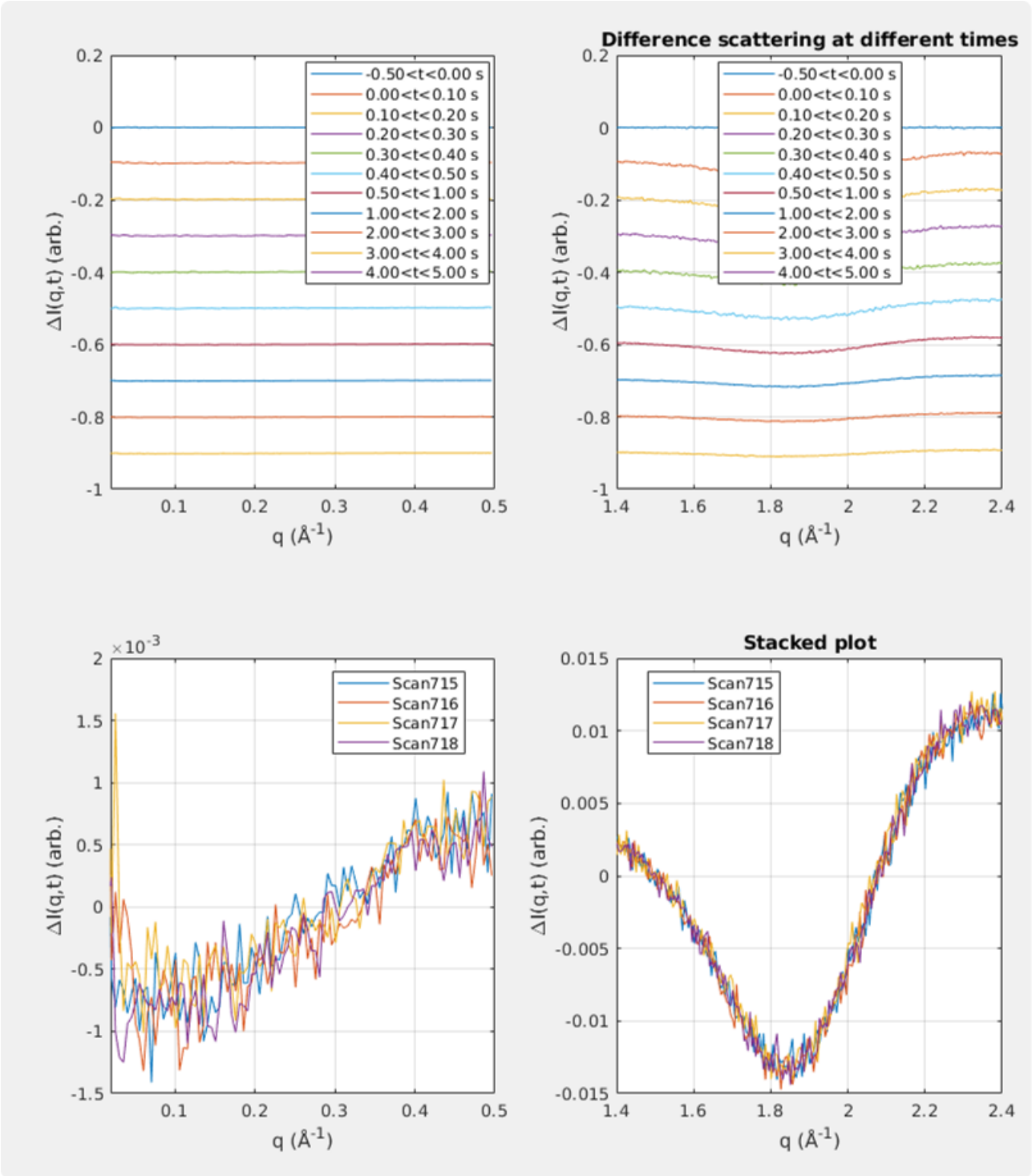


If you have recorded data of the solvent thermal response, this can also be scaled to and subtracted from the data.

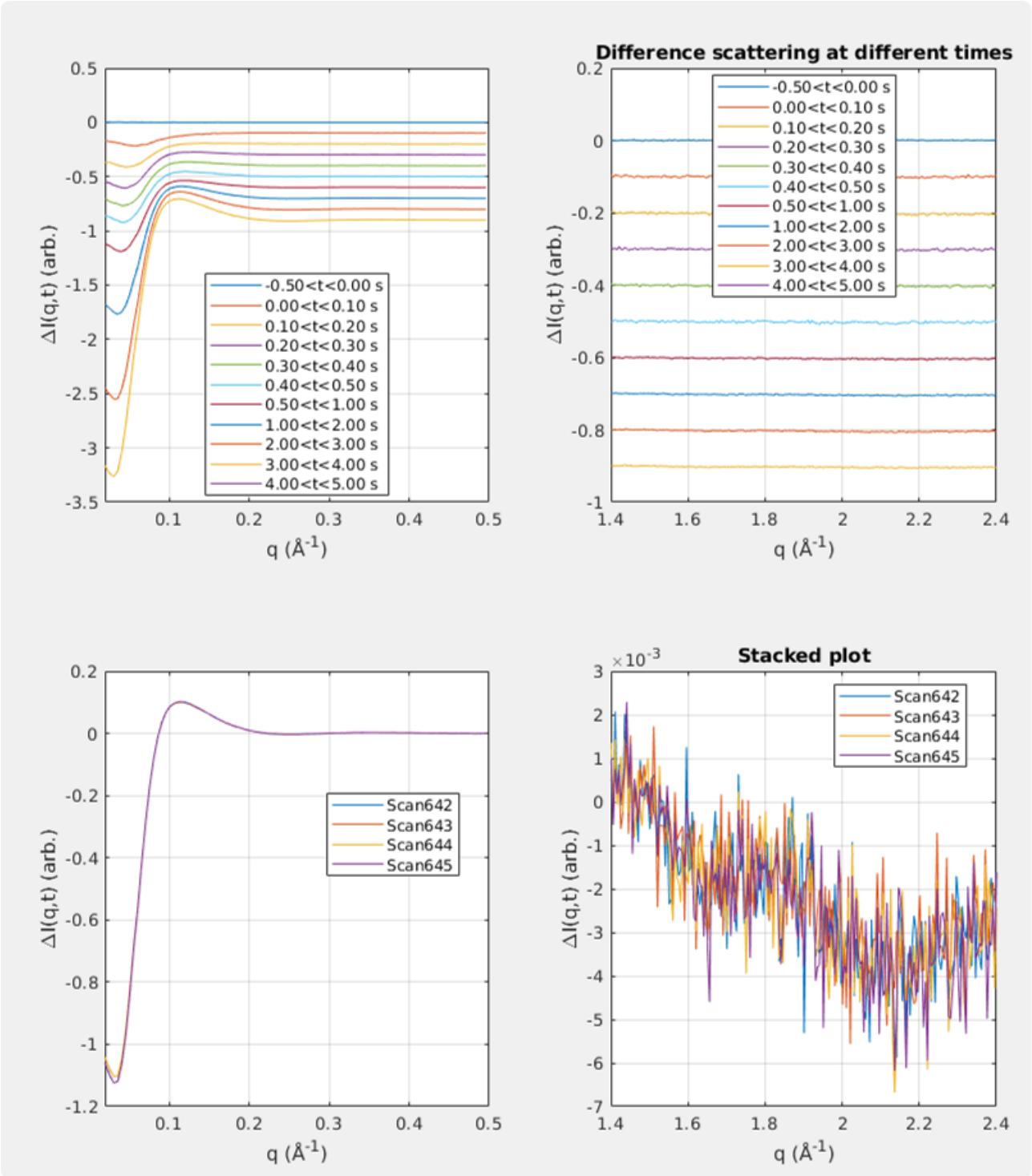
Below is the backbone plot of the data analysis, a stacked ΔI vs q plot. The upper row shows the data averaged according to timeSlices, whereas the lower row shows the per-scan averaged data.

The upper plot allows you to evaluate the temporal evolution of the signal, and the lower allows you to judge scan-to-scan variation.

The particular dataset shown here is a ca. 10-20 °C T-jump of an aqueous buffer. As you can see, it is primarily the scattering at higher q which is affected by this and the lower right plot shows the fingerprint of a temperature increase in water.

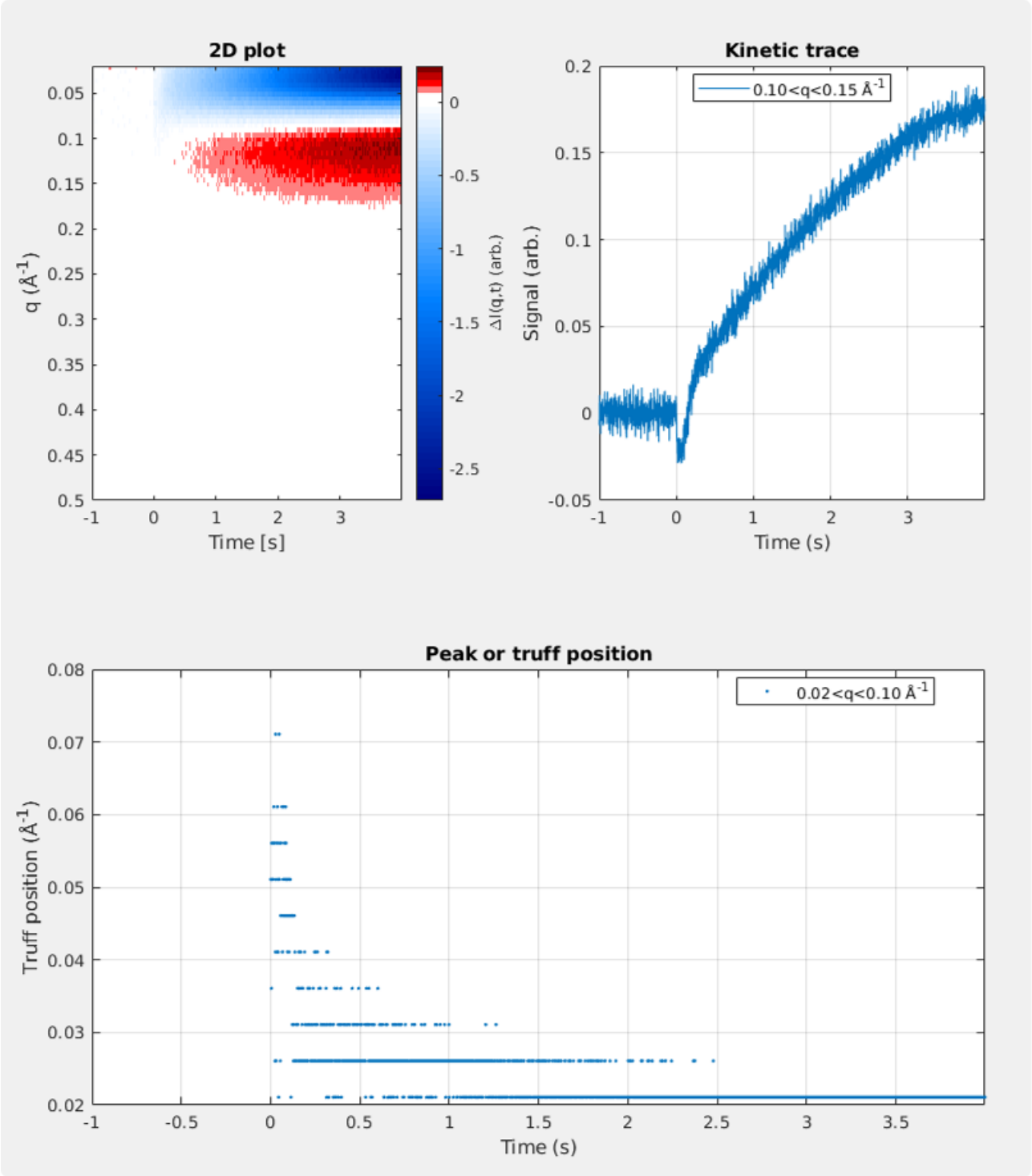


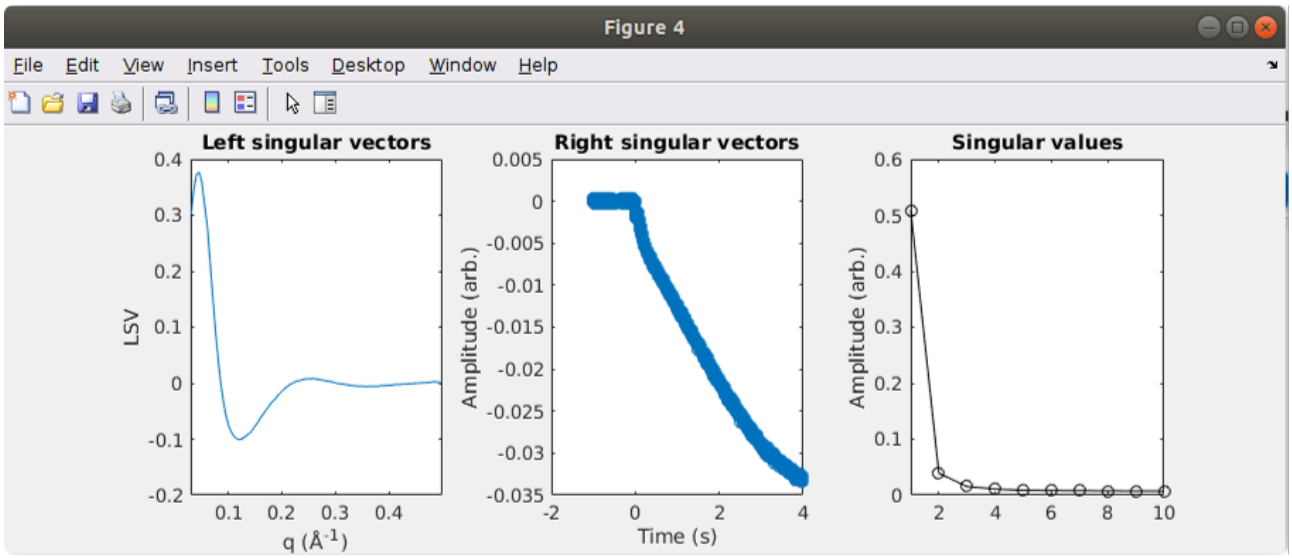
You can also select to subtract the contribution of solvent thermal expansion, as can be seen in the plots for lysozyme below.



The plots below shows a 2D plot (upper left), the kinetic trace for the scattering at a certain q (upper right) and a plot showing how the position of the truff shifts shifts towards lower q for longer times (lower plot). All of these plots are created by specific MATLAB functions for each type

of plot, and those functions can with relative ease be assembled into your own, custom made, analysis script.





Finally, this plot shows the result of an SVD analysis. Judging by the singular values, essentially only the first component is relevant, and this is also the only component displayed (although, this is manually set at the top of the script).