LIISim 3

User guide

Version 3.0.6

April 2018

(03.04.2018)

Authors: **Raphael Mansmann**
Philip Schmidt
Tobias Terheiden

Website: www.liisim.com

www.uni-due.de/ivg/rf

Source code: www.github.com/LIISim/LIISim3

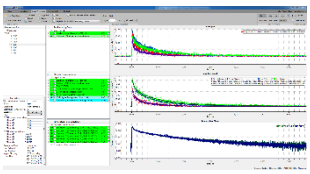
LIISim is a project of:

Table of Contents

1 Introduction

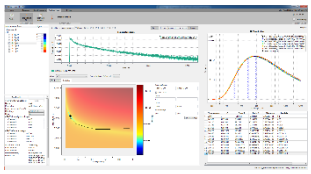
Welcome to LIISim3, a modular signal processing toolbox for time-resolved laser-induced incandescence measurements written in C++ with the Qt-Framework. This user guide describes the main functionalities of the software and how to use it. Please feel free to contact the authors if you have questions, find errors or have any other feedback.

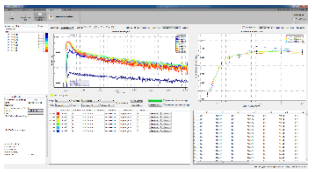
Some screenshots of the software:



SignalProcessing
The modular signal processing toolbox allows processing of raw signals, absolute signals and temperature traces. Processing steps can be individually set and arranged. Intermediate processing results can be visualized and analyzed with various plot tools.

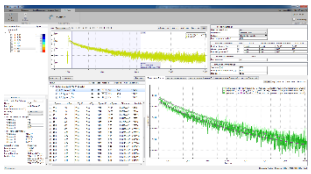
AnalysisTool: Temperature Fit
Visualization of spectral temperature fitting using Planck's law. Temperature traces that are calculated in the SignalProcessing module can be analyzed and all fitting iterations can be visualized.





AnalysisTool: Parameter Analysis
Experimental data can be systematically compared for various parameters (i.e., laser fluence, LII peak temperature, PMT gain voltage, …).

FitCreator
Select from a variety of heat transfer models and databases for simulation of LII signal traces and comparison with experimental data.



1.1 Laser-induced incandescence

Laser-induced incandescence (LII) is a non-intrusive method of measuring soot particle volume fraction and primary particle sizes in flames. Not only restricted to flames, this technique is recently used for characterization of car engine exhaust, atmospheric black carbon and synthetic nanoparticles.

These new applications require the same basic signal processing, but at the same time individual databases for material (soot or non-soot) and gas compositions as well as individual heat transfer models for the determination of particle sizes.

LIISim is a framework for basic signal processing and analysis experimental LII data to help researches across the world to compare their experimental data.

1.2 How to cite

If you use our software for your research, we would be grateful if you could cite the following paper:

```
R. Mansmann, T. Terheiden, P. Schmidt, J. Menser, T. Dreier, T.
Endres and C. Schulz: "LIISim: a modular signal processing
toolbox for laser-induced incandescence measurements" Appl.
Phys. B, DOI 10.1007/s00340-018-6934-9 (2018)
```

View at publisher:
https://doi.org/10.1007/s00340-018-6934-9

1.3 Motivation

LIISim is designed to provide transparent and flexible tools, which allow individual setting of processing parameters, visualization of intermediate processing steps, and comparison of multiple experimental data sets.

Main features:

- Modular choice of user-defined material properties (soot, silicon, germanium, …)
- Comparison of experimental data with different pre-implemented heat-transfer models
- Various analysis tools help visualizing dependencies between experimental data sets (plots, parameter comparison, statistical information)
- Easy-to-use import (TXT/CSV) and export (TXT/CSV/MATLAB) functionalities
- Copy/paste of signals and analysis results into spreadsheet software (MATLAB, Excel, Origin, …)
- Software architecture allows processing and comparison of large data sets (> 4 GB)

1.4 History

LIISim was so far known as **console application (LIISim 1.5)** and **web interface (www.liisim.com)**, developed by Max Hofmann between 2001 and 2007. The console application was written in C and the web interface was based on Perl. Modeling settings and file names for experimental data could be defined in DAT files and after execution of the console application, DAT output files containing the modeling results are created.

Later the console application was extended by a C++ based graphical user interface (GUI) by Tobias Terheiden and Martin Leschowski, which allowed the visualization of the modeling results from the DAT files (**LIISim Desktop 1.02** and **LIISim Console 2.14**). While the console application was distributed among the community, the desktop version was not published and was only internally used.

In 2013 the development of **LIISim 3** a new framework with object-oriented structure based on C++ was started by Raphael Mansmann designed for the application on individual material systems (soot and non-soot) with a modular implementation of signal processing steps and choice of heat-transfer models.

1.5 License

LIISim is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License (http://www.gnu.org/licenses/) for more details.

2 Get started

2.1 Downloading the latest LIISim release

The latest LIISim release can be obtained at https://github.com/LIISim/LIISim3/releases.

2.2 System requirements

LIISim is available and tested for Windows 7/8/10 (64bit or 32bit). We recommend using the 64bit version to overcome the 32bit memory limit of 4 GB. All drivers necessary for running LIISim are included in the package.

If you see an error like “MSVCR120.dll was not found” please try to download and install the Visual C++ Redistributable Packages from the Microsoft website. It should fix the issue:
https://www.microsoft.com/en-us/download/details.aspx?id=40784

2.3 Starting LIISim

Installation is not required. Extract the .zip file to the desired location and execute **LIISim3.exe** (no admin privileges necessary). Please see the “Quick Start” chapter for a small tutorial in using LIISim.

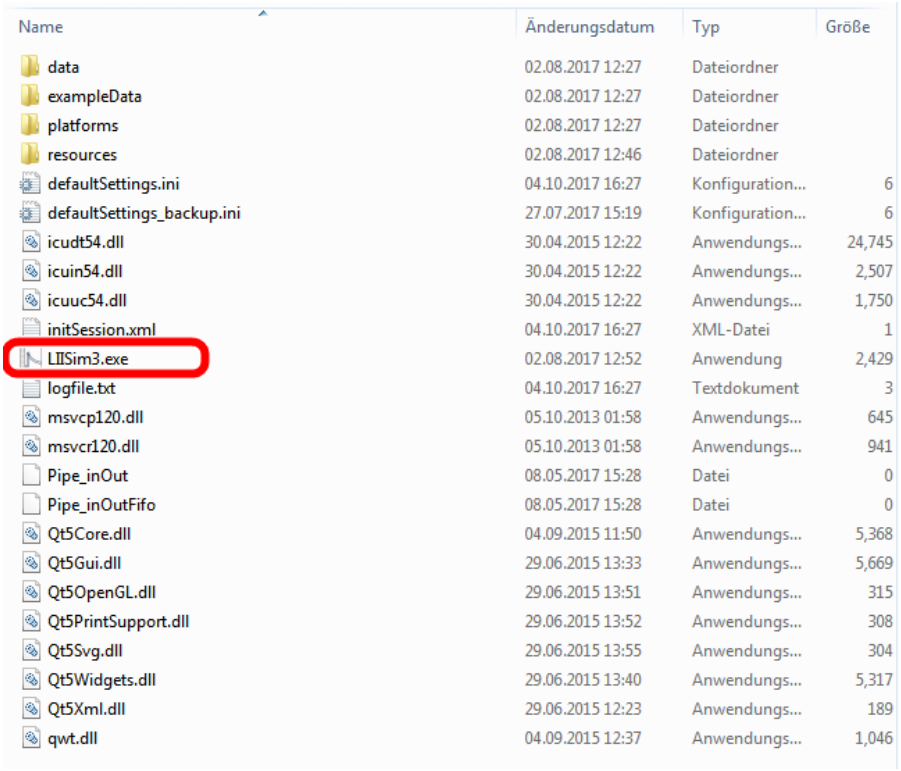


Figure 2-1 LIISim root folder

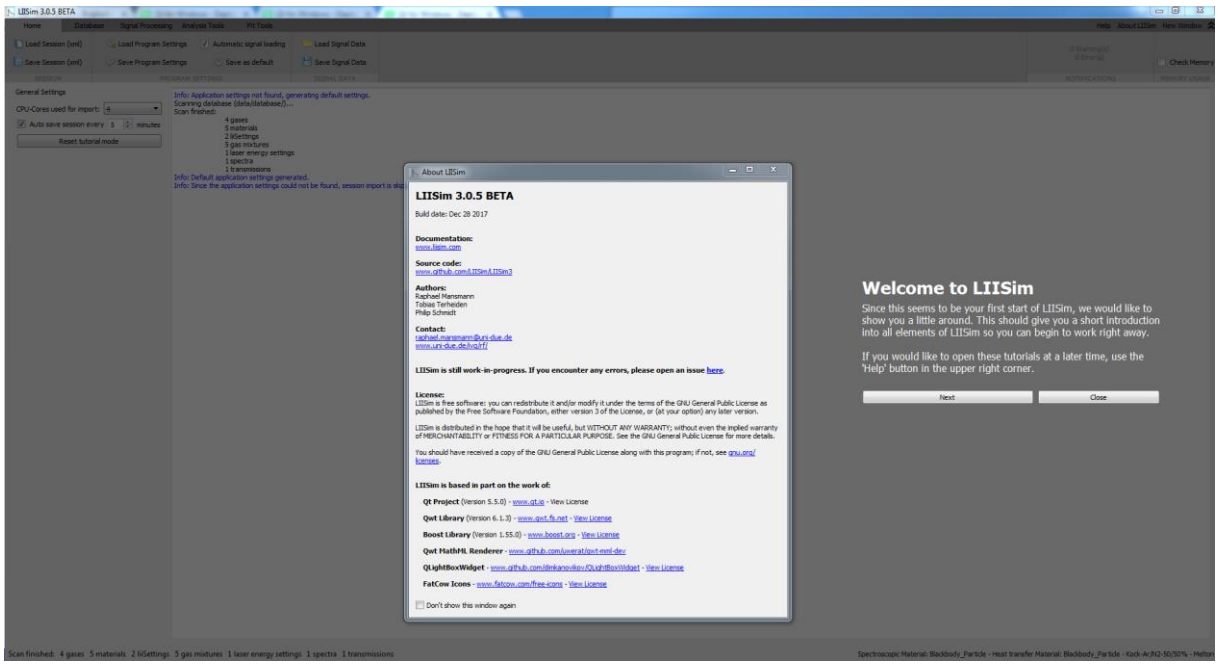
2.4 Example files

Example experimental data can be found in the “exampleData/” directory. The “Quick start” guide, uses these data to demonstrate the main LIISim functionalities.

2.5 Extend / compile LIISim yourself

There is a developer guide available at www.liisim.com with further information about the source code e.g. needed libraries, directory and class structure etc. if you like to add additional functionality or like to contribute.

3 Quick start



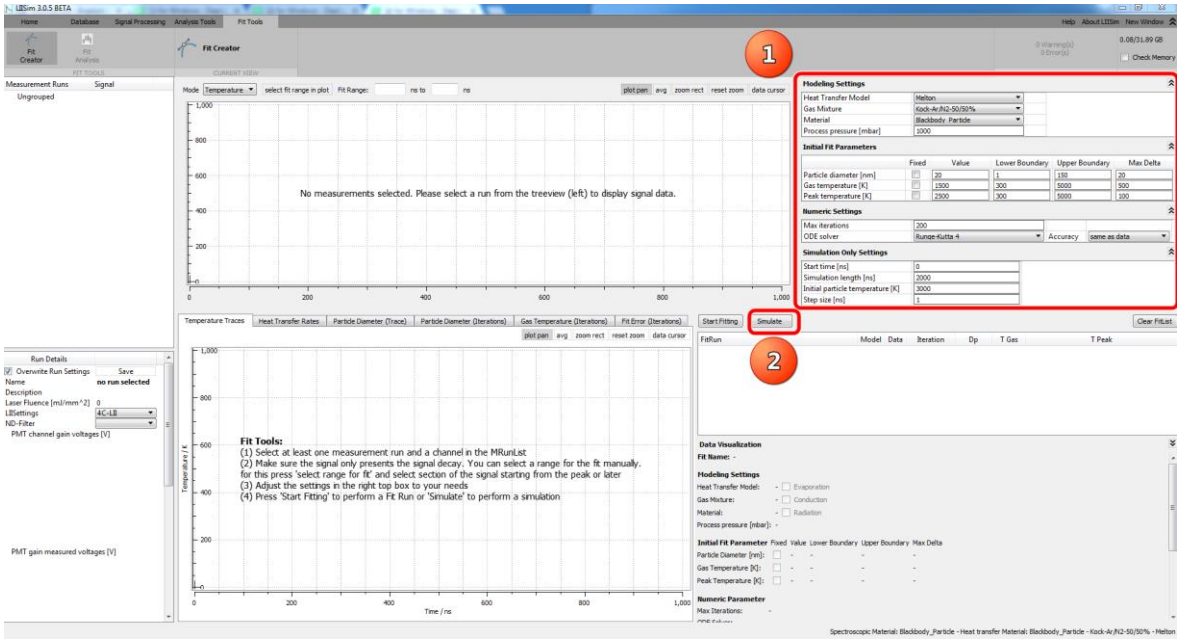
During the first start of LIISim all modules are explained by a **tutorial mode**.

This tutorial mode explains the most important functionalities and structure of the software. It is highly recommended to go through all tutorials before you start using LIISim.

You can restart each tutorial by pressing the “Help” button in the top right corner.

3.1 Example: Simulation of LII signals

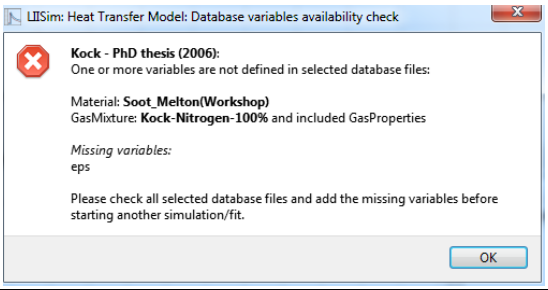
For simulation of LII temperature traces, open the **Fit Tools** module:



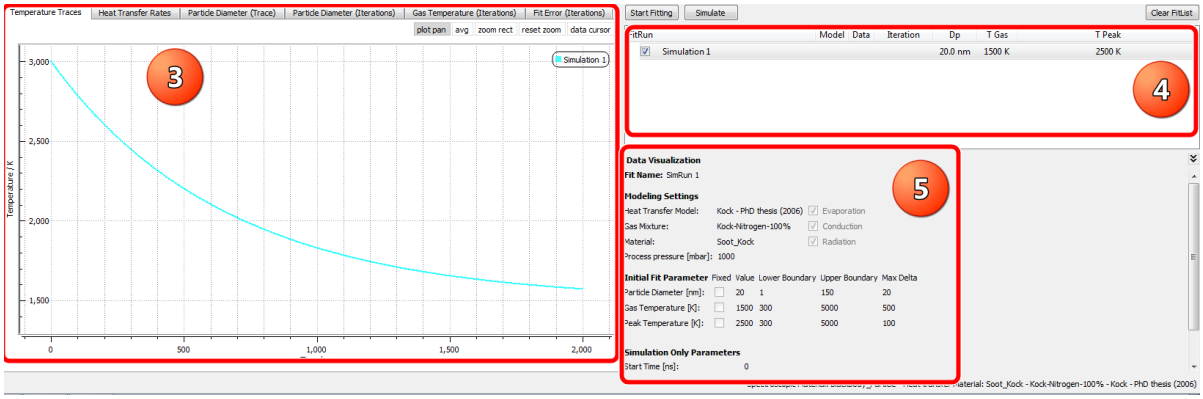
(1) You can modify modelling settings, fit parameters, numeric and simulation settings

Make sure to use compatible databases and models, otherwise a warning message appears.

Example: The "Soot_Melton(Workshop)" database is not compatible with the "Kock"-model because of the missing "eps" variable. Switch the Material database to Soot_Kock to solve this issue.



(2) After pressing "**Simulate**", the settings are used to simulate a temperature trace which is visualized. You can modify the settings and add another simulation, which appears in **(4)**.

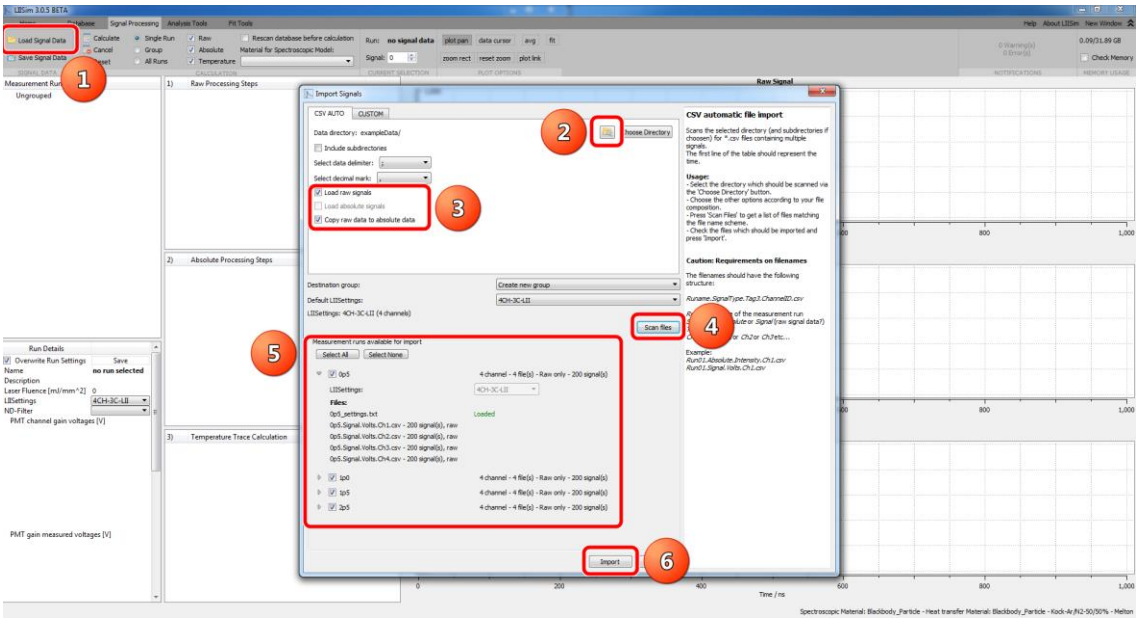


You can then:
(3) Analyze the temperature trace, heat-transfer rates and compare to other simulated traces
(4) Switch between simulations with different parameters
(5) Show specific simulation settings for the selected simulation in **(4)**

3.2 Example: Signal processing and data evaluation

This is a step-by-step guide for importing the example data set, setup the signal processing and use data evaluation tools. First, you need to switch to the **“Signal Processing”** module.

3.2.1 Import of example data set



(1) Use the “**Load signal Data**” button to open the import dialog

(2) Open the file explorer and make sure the folder exists and contains data:

Name	Änderungsdatum	Typ	Größe
0p5.Signal.Volts.Ch1.csv	08.12.2017 15:05	Microsoft Exc...	24,502 KB
0p5.Signal.Volts.Ch2.csv	08.12.2017 15:05	Microsoft Exc...	24,490 KB
0p5.Signal.Volts.Ch3.csv	08.12.2017 15:05	Microsoft Exc...	24,441 KB
0p5.Signal.Volts.Ch4.csv	08.12.2017 15:05	Microsoft Exc...	24,360 KB
0p5_settings.txt	28.12.2017 15:58	Textdokument	1 KB
1p0.Signal.Volts.Ch1.csv	08.12.2017 16:28	Microsoft Exc...	24,512 KB
1p0.Signal.Volts.Ch2.csv	08.12.2017 16:28	Microsoft Exc...	24,507 KB
1p0.Signal.Volts.Ch3.csv	08.12.2017 16:28	Microsoft Exc...	24,526 KB
1p0.Signal.Volts.Ch4.csv	08.12.2017 16:29	Microsoft Exc...	24,457 KB
1p0_settings.txt	28.12.2017 16:00	Textdokument	1 KB
1p5.Signal.Volts.Ch1.csv	08.12.2017 16:34	Microsoft Exc...	24,514 KB
1p5.Signal.Volts.Ch2.csv	08.12.2017 16:35	Microsoft Exc...	24,508 KB
1p5.Signal.Volts.Ch3.csv	08.12.2017 16:35	Microsoft Exc...	24,523 KB
1p5.Signal.Volts.Ch4.csv	08.12.2017 16:35	Microsoft Exc...	24,473 KB
1p5_settings.txt	28.12.2017 16:00	Textdokument	1 KB
2p5.Signal.Volts.Ch1.csv	08.12.2017 16:43	Microsoft Exc...	24,520 KB
2p5.Signal.Volts.Ch2.csv	08.12.2017 16:43	Microsoft Exc...	24,515 KB
2p5.Signal.Volts.Ch3.csv	08.12.2017 16:43	Microsoft Exc...	24,509 KB
2p5.Signal.Volts.Ch4.csv	08.12.2017 16:43	Microsoft Exc...	24,487 KB
2p5_settings.txt	28.12.2017 16:01	Textdokument	1 KB

For this data set, LII decay curves were detected at four different wavelengths for various laser fluences. 200 signals are stored in each .csv file per channel. Usually raw signals are processed in LIISim and can later be exported as calibrated “absolute” signal files. The two separate processing chains simplify comparison between the raw and calibrated signals. This allows import/export of both, raw and calibrated curves.

(3) Here, only raw signals (“Signals.Volts”) are available, so we **duplicate the raw signals to the absolute processing chain** by activating the checkbox “Copy raw data to absolute data”

(4) Using the default settings you should be able to perform a directory scan using “**Scan files**”

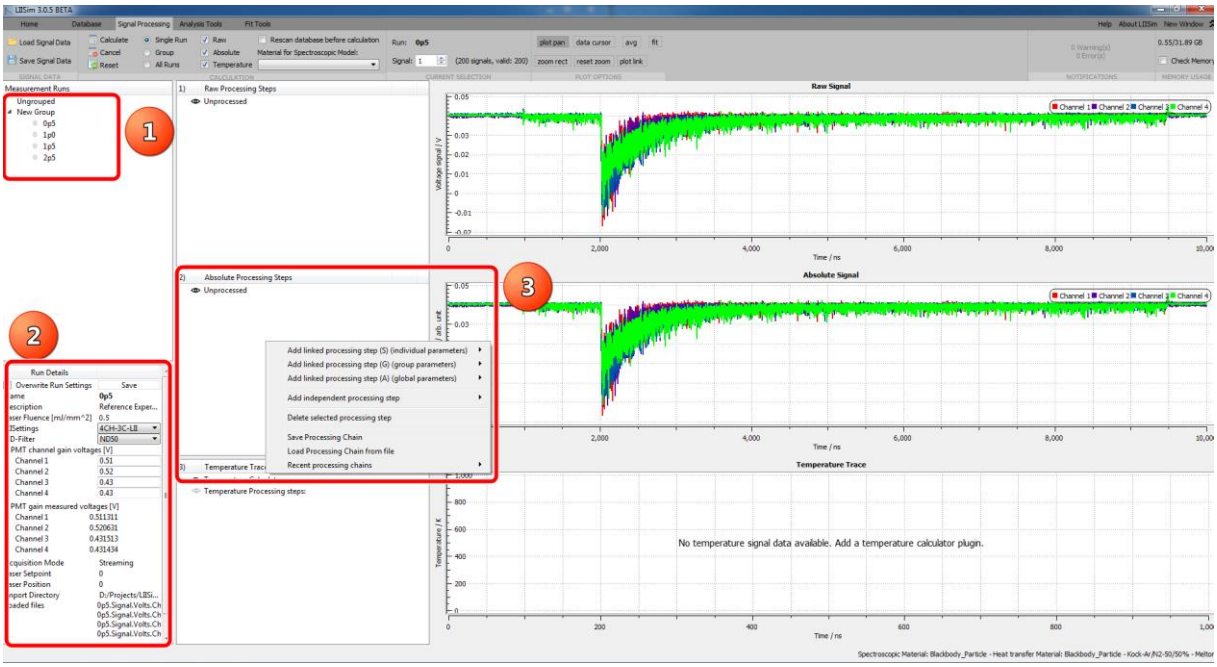
(The LIISettings-file for this experimental setup (4CH-3C-LII) is already included in LIISim and contains information about detector channel center wavelengths, calibration factors and neutral density filter transmission, see Database → LIISettings for further information)

(5) Files matching the file pattern and settings are listed and can be **selected** for import

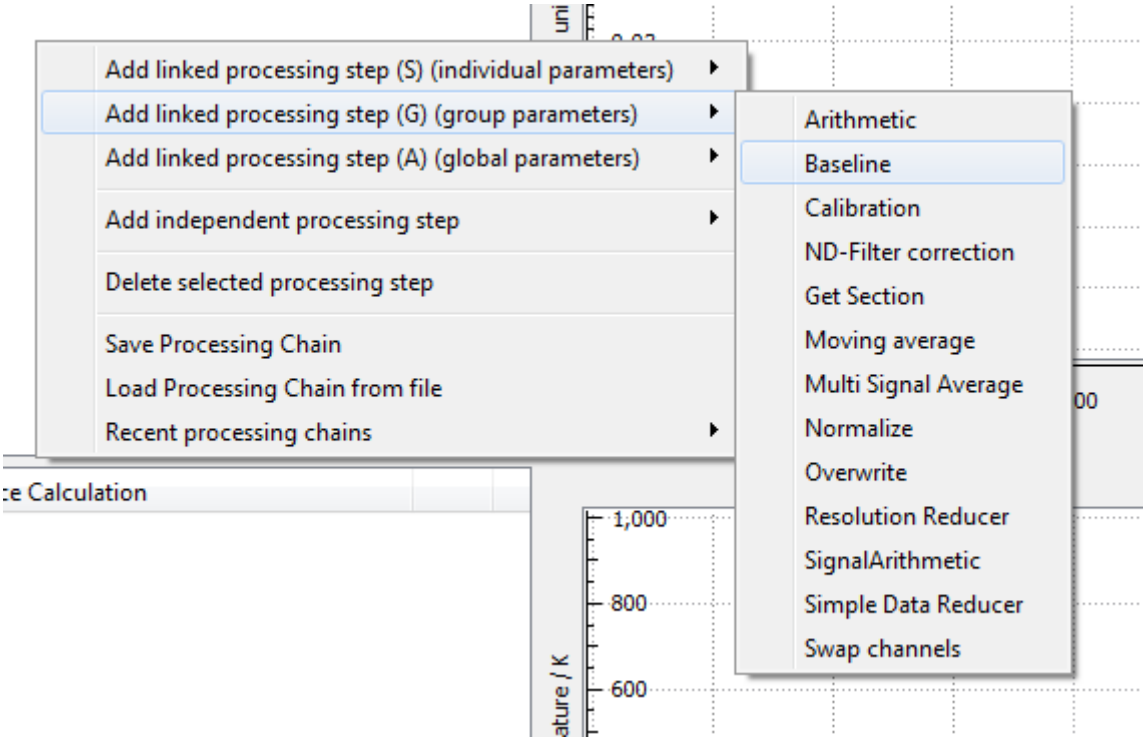
(6) Press “**Import**” to load the selected files

3.2.2 Basic signal processing

(1) The imported group of measurement data is shown in the left box. By selecting each measurement run you can see detailed information in **(2).**



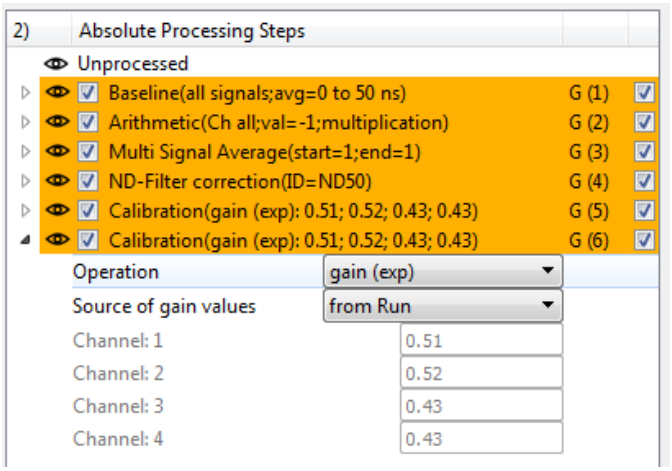
(3) Right-Click somewhere within the processing chain box opens the context menu for adding processing steps:



In this case, linked processing steps are used, because we want to have the same parameters for the four measurement runs in this group.

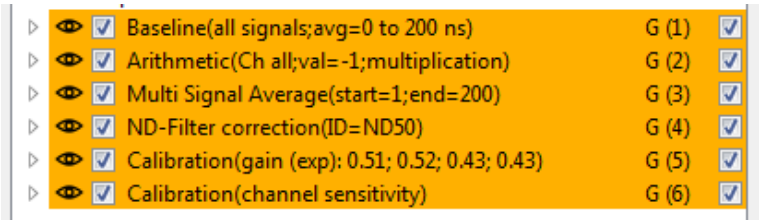
The following processing steps need to be added:

- **Baseline**
- **Arithmetic**
- **Multi-Signal Average**
- **ND-Filter correction**
- **Calibration (sensitivity)**
- **Calibration (gain (exp))**



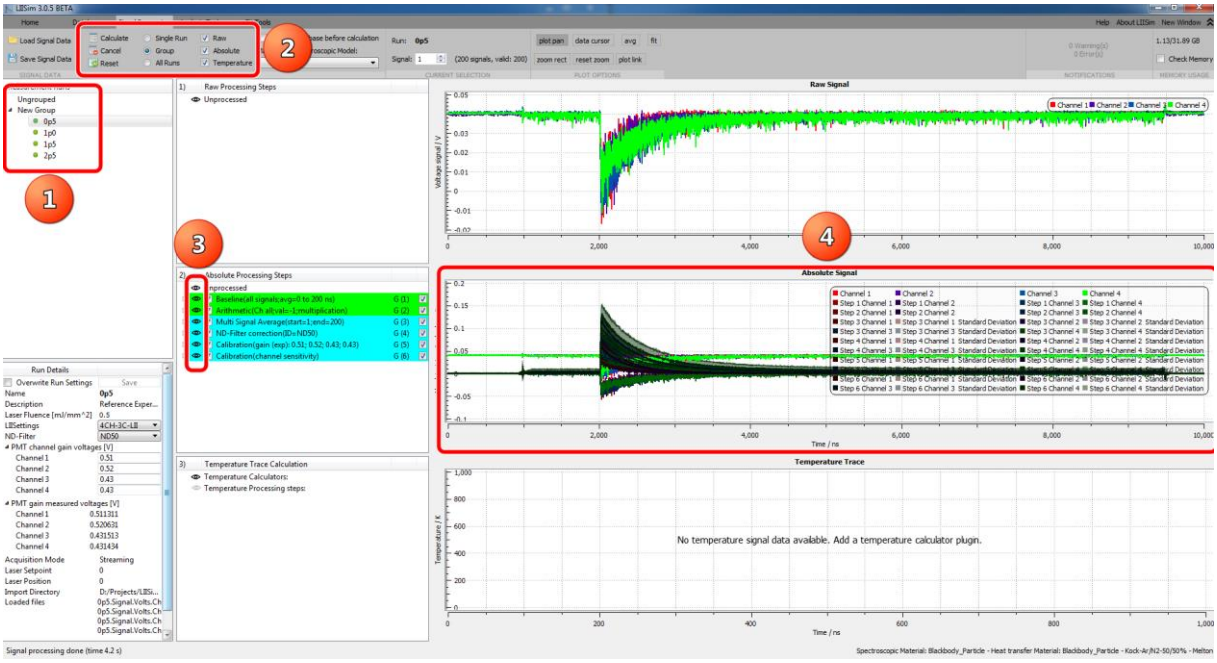
The small arrow can be used to show the parameters and change settings of each processing step. Some parameters are automatically retrieved from the settings file and the input is then disabled (for example ND-Filter correction and Calibration).

Make sure the steps have the following parameters:

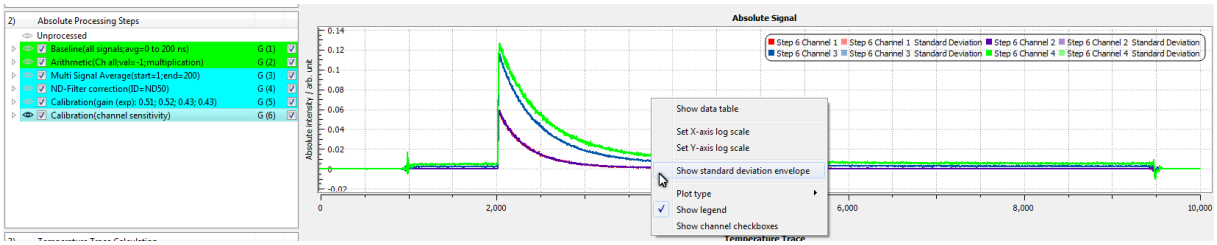


For processing of the new chain, you need to select one of the measurement runs **(1)**, then set the processing mode **(2)** to “Group”, to process all measurements in this group and the processing chain should turn green/blue if the data are processed correctly. The blue color in the processing chain indicates that an average is calculated from all signals and the averaged signal is further processed.

Now the plot **(4)** looks a bit crowded because all steps are visualized. Use the “eye”-icons **(3)** to toggle visibility of the signal traces for each processing step.



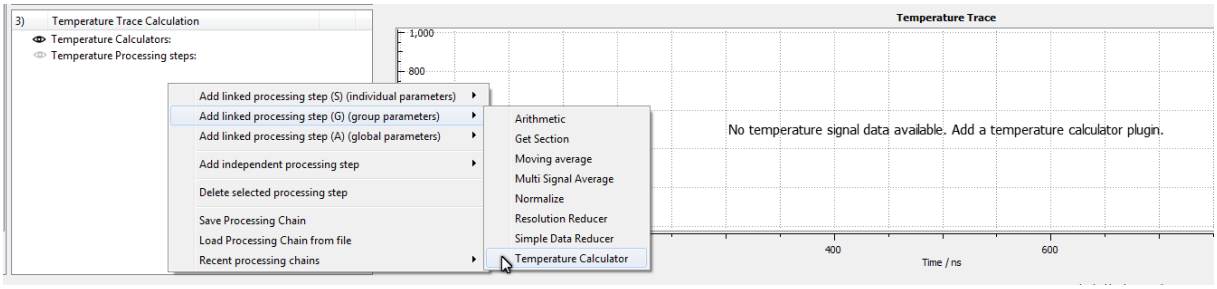
The standard deviation curve can be toggled using the **right-click context menu on the plot**:



The result from the last processing step (Calibration (channel sensitivity) can now be used by the “Temperature Trace Calculation”-chain for further processing.

3.2.3 Temperature trace calculation (Two-Color)

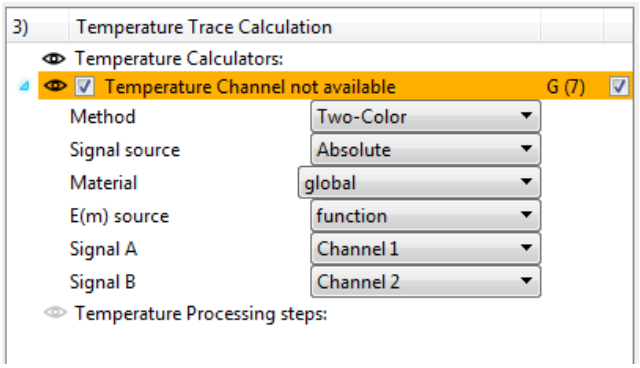
With the calibrated absolute signals, temperature traces can be calculated. Right-click on the temperature processing chain box and add a linked processing step: “**Temperature Calculator**”



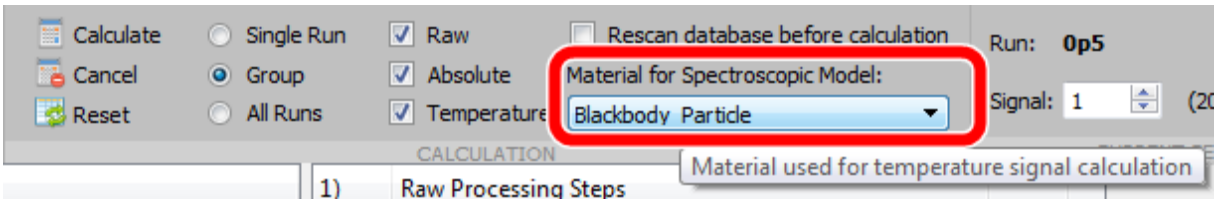
You can add multiple TemperatureCalculators with different settings and compare the traces.

By default, the "**Two-Color**" method is selected:

Make sure to change the channels of Signal A/B to two different wavelengths (Channel 1 and 2 have the same bandpass filter wavelength (500 nm) for this data set)

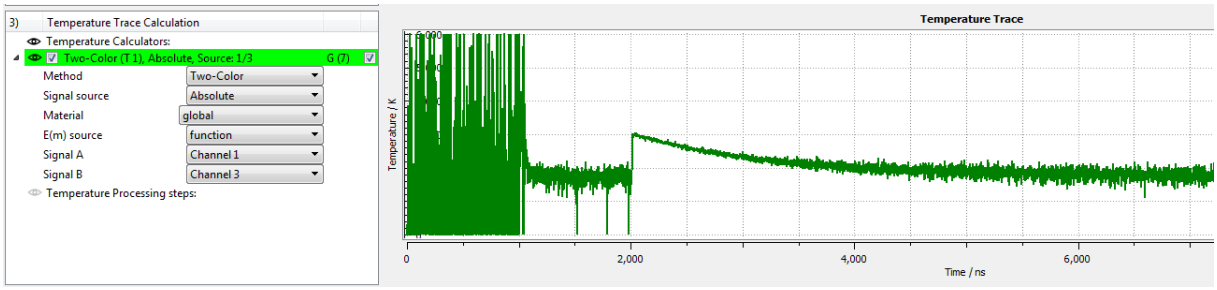


Before calculation the **spectroscopic model** needs to be chosen (here: constant E(m) values: "Black-body_Particle"). This spectroscopic model is used if Material: "global" is selected.



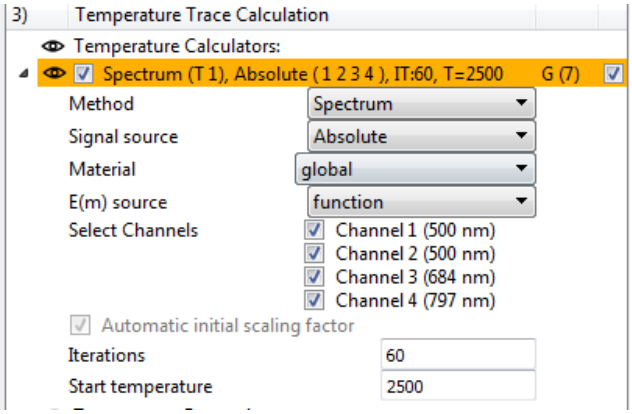
E(*m*) values are provided by the Material-database file as single values ("values"), function of wavelength ("function") or calculated from Drude model ("Drude"). Here, "Blackbody_Particle" provides *E*(*m*) as "function".

Pressing **"Calculate"** now processes also the temperature processing chain, here with two-color pyrometry applied on the result from the "Absolute"-processing chain using Channel 1 and 3:

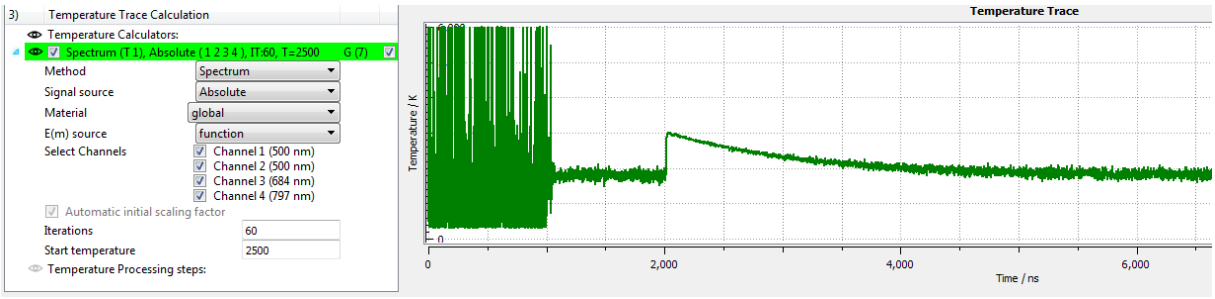


3.2.4 Temperature trace calculation (Spectrum)

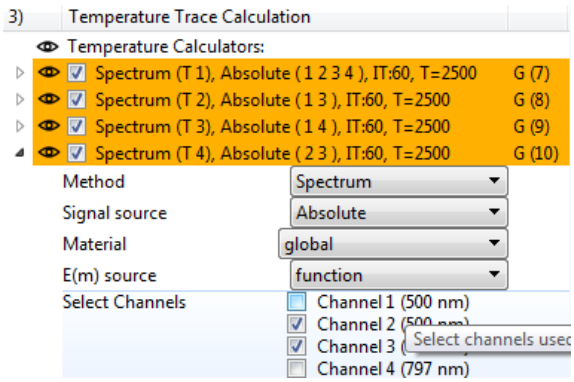
Proceed exactly as with “Two-Color” method, but change the method to “Spectrum”:



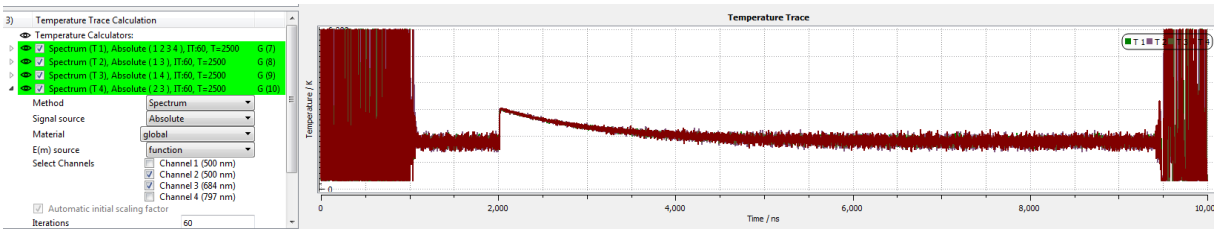
You can now **select channels** that can be used for spectral fitting:



Multiple temperature traces can now be calculated using different settings. In this example we added a spectral fit using all four channels (T1) and three times a spectral fit only with two channels (T2-T4):



Processing now takes a bit longer, because every data point is fitted in several iterations for every Temperature-Calculator.

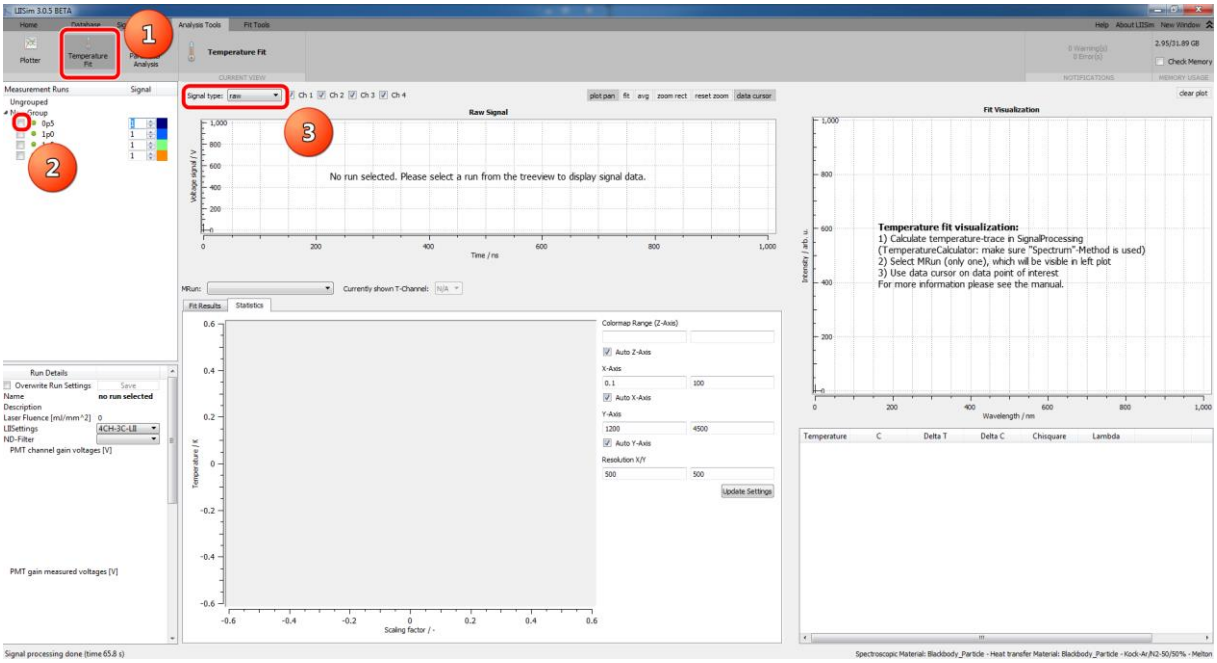


We have now four temperature traces that can be analyzed using the AnalysisTools.

3.2.5 AnalysisTools: Temperature Fit

Switch to the AnalysisTools module and select **Temperature Fit (1)**. Then select the measurement run **(2)** to be analyzed (only one MRun can be analyzed at the same time). Switch signal type to be displayed **(3)** to **“absolute”** or **“temperature”.**

This tool only works with temperature traces calculated with the “Spectrum” method. “Two-Color” traces cannot be visualized.



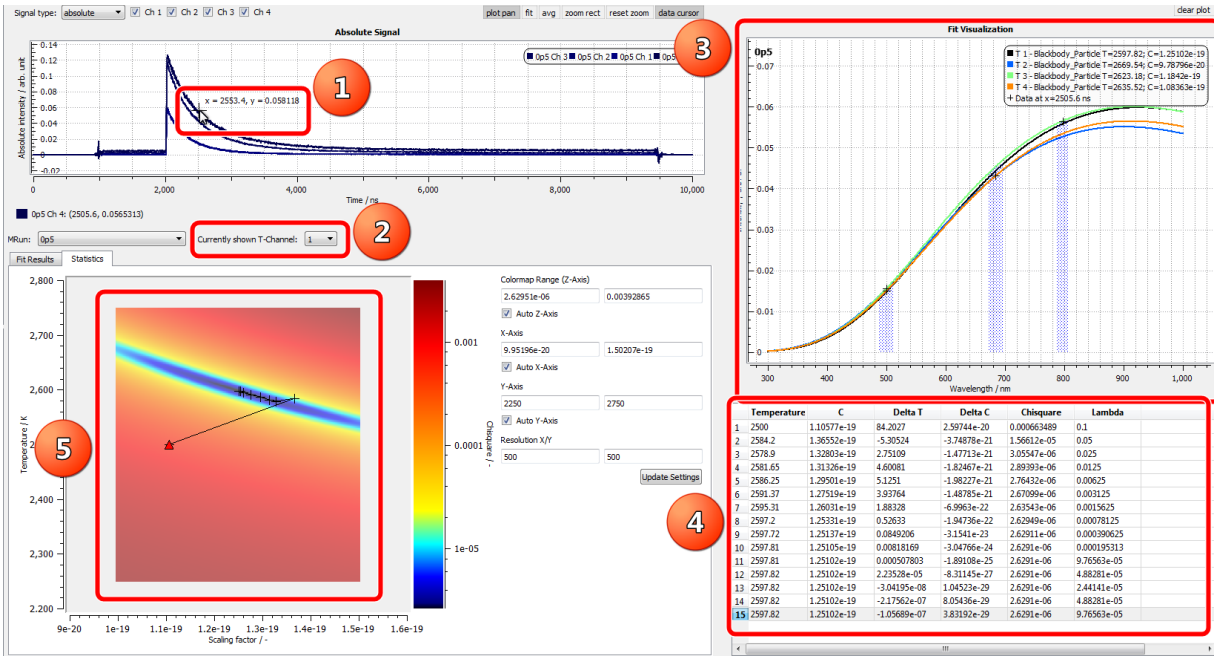
(1) Select now in the top left plot a data point you want to analyze. The temperature trace selected in **(2)** is then visualized.

(3) Compares the selected spectral fit to the data points (crosses) and to other available “Spectrum” temperature fits. The legend entries contain the parameters for the curves (Temperature and Scaling factor).

(4) Shows the spectral fitting iterations for the selected data point and **(4)** the chi-square map of the two parameters used during fitting (Temperature and Scaling factor).

The crosses and lines in **(5)** indicate the progress of iterations of the fitting procedure and the red triangle the initial conditions.

- Click on the map **(5)** to show the spectral fit in **(3)**.
- Click on any iteration in **(4)** to show the corresponding spectral fit in **(3)**



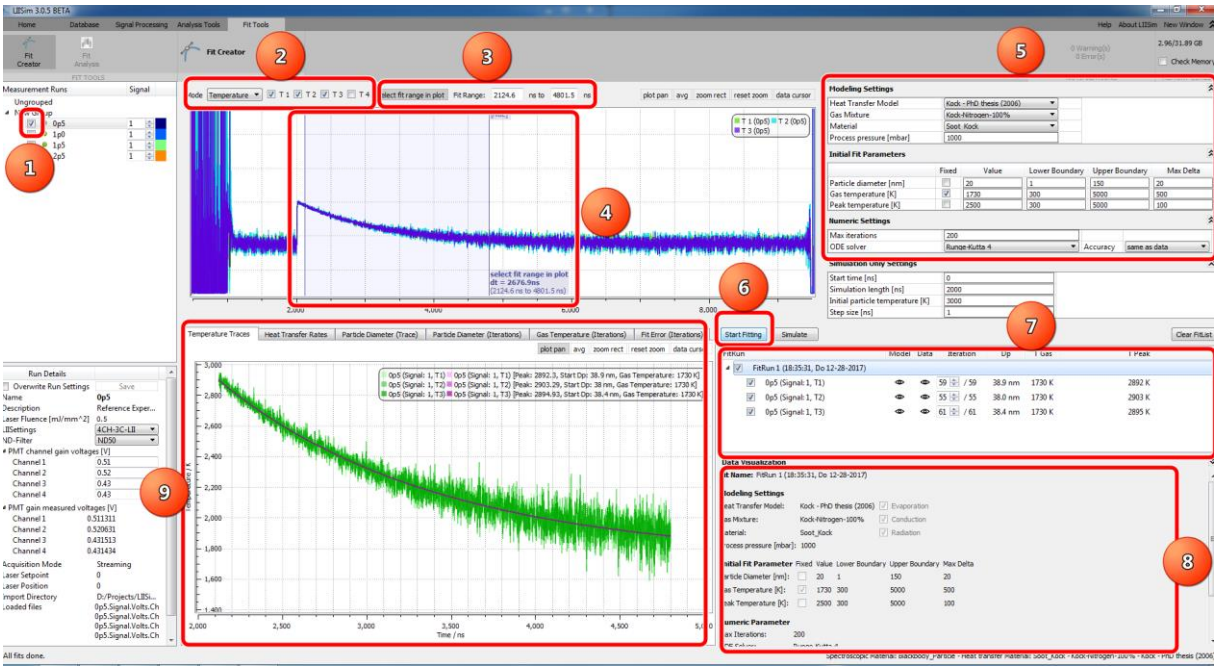
3.2.6 Fit Tools

The previously calculated temperature traces can be used for comparison with modeled temperature traces using various heat-transfer models and databases.

First, you need to select at least one measurement run **(1)** and the temperature traces to be used for the fitting **(2).**

The temperature traces should now appear in the plot and you can select a data range using the button **(3)** and then select the range in the plot **(4)** or you can enter the range manually.

After adjusting the settings **(5)** for the fitting you can press “Start Fitting” **(6)** to perform a fit on all selected measurement runs and temperature traces.



The result from this fitting is called a “FitRun” which is listed in the FitRun-List **(7).** You can select the temperature traces and look at the detailed information in **(8)** or at various plot tools in **(9).**

4 Data structure

The following table lists frequently used terms within the software:

Term	Abbreviations	Description
Measurement run	MRun, run	A complete set of (multiple) signals
LII settings	LIISettings	Definition of detection related settings (channels, center wavelengths, calibration, ND-filter transmissions)
Measurement run settings	MRunSettings	Additional file for every run (<*>_settings.txt), contains name, description, LIISettings-filename etc.
Measurement point	MPoint	A single measurement, containing multiple channels, but only one signal per channel
Raw signal	RAW, raw	Raw data signal
Absolute signal	ABS, abs	Scaled (calibrated) data signal
Temperature	TEMPERATURE, temp	Calculated temperature trace

The data structure used within the LIISim software is described in Figure 4-1. All elements are explained in detail in the following sections.

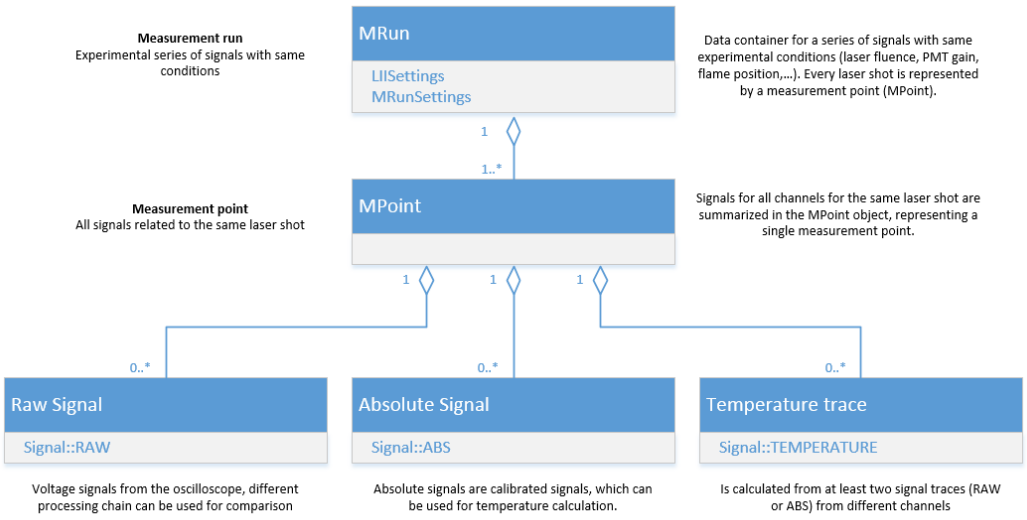


Figure 4-1 Data structure used within LIISim

4.1 LIISettings

LIISettings summarizes information of the LII device used for the measurements. There can be an infinite number of different LIISettings that can be selected within LIISim. They contain infor-

mation about laser wavelength, bandpass filter width, sensitivity, reference voltage, gain calibration, signal offset, available ND-filters and filter transmission. LIISettings can easily be visualized in the DatabaseEditor (see 7.4).

Example files can be found at *data/database/liisettings/*.

4.1.1 Example file structure (4-C-LII.txt)

The following lines represent a sample LIISettings file for a LII device with four channels. The variables and values are separated by a semicolon (;).

4C-LII.txt:

```
1  4C-LII;LIISettings;0.1;LII detector with 4 channels;
2  laser_wavelength;1064;
3  channel;390;18;1.4;0.4;7.988;0;0;
4  channel;500;24;1.5;0.4;8.126;0;0;
5  channel;684;24;1;0.5;7.54554;0;0;
6  channel;800;12;1.3;0.399;8.60;0;0;
7  filter;ND0.1;0.02811;0.0431;0.09869;0.1484;
8  filter;ND1;0.46;0.73;1.45;1.95;
9  filter;ND10;8.256;8.73;11.0;11.9;
10 filter;ND50;43.8;45.5;49.8;52.4;
11 filter;ND79;76.6334;79.168;81.5745;82.539;
```

Basic LIISettings information (Line 1):

```
4C-LII;LIISettings;0.1;LII detector with 4 channels;
```

Parameter breakdown:

Variable	Example	Allowed Value Type	Description
Name	4C-LII	String	Name of LIISettings, will be displayed within LIISim
Type of .ini	LIISettings	-	This string determines that this file is a LIISettings file. Should always be 'LIISettings'.
Version	0.1	Floating point	Version of this LIISettings, can be incremented for tracking and documenting changes by the user
Description	LII detector with 4 channels	String	Description for this LIISettings file.

Laser wavelength (Line 2):

```
laser_wavelength;1064;
```

Parameter breakdown:

Variable	Example	Allowed Value Type	Description
Laser wavelength	laser_wavelength;1064;	Floating point	LII device laser wavelength (currently not used)

PMT channel properties structure (Line 3 to 6):

```
channel;390;18;1.4;0.4;7.988;0;0;
```

Parameter breakdown:

channel;	Identifier. Every following 'channel' identifier will increase the channel number, so the channel number must not be appended
390	Channel center wavelength [nm]
18	Bandpass filter width [nm]
1.4	Calibration (sensitivity) (see 6.1.3)
0.4	Gain (reference voltage) [V] (see 6.1.3)
7.988	Gain (calibration) (A) (see 6.1.3)
0	Gain (calibration) (B) (see 6.1.3)
0	Offset, which can be used by "Baseline"

ND-filter information structure (Line 7 to 11):

```
filter;ND10;8.256;8.73;11.0;11.9;
```

Parameter breakdown:

filter	Identifier. Every line defines a single ND-filter.
ND10	Filter identifier. This value is displayed and selectable inside LIISim. This value should be used to connect the filter setting in LIISettings and MRunSettings (see 4.2).
8.256	Transmission for channel 1 in percent (8.256 %)
8.73	Transmission for channel 2 in percent (8.73 %)
11.0	Transmission for channel 3 in percent (11 %)
11.9	Transmission for channel 4 in percent (11.9 %) (the number of transmission values needs to match the channel number)

4.2 Measurement run (MRun, run)

A measurement run is a set of one or multiple MPoints and associated MRun settings. Signals within a measurement run share the same detection settings like PMT gain, ND-filter, laser fluence,…

4.2.1 MRun settings

The MRun settings contain meta information for each MRun like name, description, LIISettings file name, filter, laser fluence etc. The filename should be the same as the signal files, with the '_settings' appended. So, if you have a signal file named

'Run_2016-09-26_15-08-08.Signal.Volts.Ch1.csv',

your settings file should be named

'Run_2016-09-26_15-08-08_settings.txt'.

Example files can be found in the 'exampleData' folder. The file structure is a basic .ini file. This means, every parameter value pair is denoted as a single line of text, the parameter followed by an equals sign and the parameter value,

e.g. for the 'name'-parameter:

```
name=RunABC
```

Example for a MRunSettings file:

```
[MRunSettings]
name=2018-12-04_11-22-33
description=LDF - 42 mm HAB
liisettings=liisettings/4C-LII.txt
filter=1
laser_fluence=1.5
pmt_gain_voltage_channel_1=0.4
pmt_gain_voltage_channel_2=0.36
pmt_gain_voltage_channel_3=0.31
pmt_gain_voltage_channel_4=0.32
pmt_measured_gain_voltage_channel_1=0.40156
pmt_measured_gain_voltage_channel_2=0.360573
pmt_measured_gain_voltage_channel_3=0.311273
pmt_measured_gain_voltage_channel_4=0.32133
udp_count=2
udp_1_identifier=Pressure
udp_1_value=50
udp_2_identifier=Operator
udp_2_value=JohnDoe
```

The following table lists all available parameters:

Line	Parameter	Allowed Value Type	Description
1	[MRunSettings]	-	Identifier for LIISim
2	name	String	Name of the associated run, can differ from the filename. This name will be used as identifier in LIISim and can be changed through the MRun-Details box.
3	description	String	Text, shows up in the run details.
4	liisettings	String	Path and filename of the associated LIISettings.
5	filter	String	Neutral density-filter identifier defined in LIISettings (see 4.1.1)
6	laser_fluence	Floating point	Laser fluence [mJ/mm²]
7-10	pmt_gain_voltage_channel_<#>	Floating point	Photomultiplier gain voltage (set) The <#> number corresponds to the channel number, ranging from 1 to x, depending on LIISettings.
11-14	pmt_measured_gain_voltage_channel_<#>	Floating point	Photomultiplier gain voltage (measured) The <#> number corresponds to the channel number, ranging from 1 to x, depending on LIISettings
15	udp_count	Integer	Current number of user defined parameters.
16	udp_<#>_identifier	String	Name/identifier of the user defined parameter with the number <#>.
17	udp_<#>_value	String	Value of the user defined parameter with the number <#>. There should always be pairs of identifier and value.

5 Databases

Database files can contain physical properties of materials, gases and gas mixtures. The first line of a database file defines the data structure and gives additional information about the file (name, description and file version):

```
<name>;<type>;<version>;<description>;
```

<name>	Name of the database file, which will be used within LIISim
<type>	String defining the type of the file. Can be [Material|Gas|GasMixture]
<version>	Version of the file defined by the user. Changes should be tracked by the user manually.
<description>	Description of the file, including general source of properties

All following lines describes properties of materials, gases, and gas mixtures and follow the same scheme:

```
   <varia-
ble>;<type>;<0>;<1>;<2>;<2>;<3>;<4>;<5>;<6>;<7>;<8>;<source>
```

<variable>	Name of the variable, which will be used in heat transfer models
<type>	Type defines if variable is constant or a function of temperature or wavelength. (please see the following tables for detailed description of property types)
<0> to <7>	Parameters a_0 to a_7 will be used dependent on <type>
<source>	A literature reference can be added for tracing and visualization within LIISim

Example:

```
   C_p_mol;poly2;22.5566;0.0013;0;0;0;-1.8195e+06;0;0;0;Boris
Kock – PhD thesis (2006);
```

This line describes the molar heat capacity (Boris Kock – PhD thesis (2006):

$$C_p(T) = 22.5566 + 0.0013\,T - 1.8195 \cdot 10^6\,T^{-2}$$

Type	Function	Equation (temperature T, parameter $a_0 \dots a_8$)
const	Constant value	$f(T) = a_0$
case	Piecewise-defined	$f(T) = \begin{cases} a_1, & T \le a_0 \\ a_2, & T > a_0 \end{cases}$
poly	Polynomial	$f(T) = a_0 + a_1T + a_2T^2 + a_3T^3 + a_4T^4 + a_5T^5 + +a_6T^6 + a_7T^7 + a_8T^8$
poly2	Polynomial	$f(T) = a_0 + a_1T + a_2T^2 + a_3T^3 + a_4T^{-1} + a_5T^{-2}$
polycase	Piecewise-defined polynomial	$f(T) = \begin{cases} a_1 + a_2T + a_3T^2 + a_4T^3, & T \le a_0 \\ a_5 + a_6T + a_7T^2 + a_8T^3, & T > a_0 \end{cases}$
exp	Exponential with Euler number as base	$f(T) = a_0 + a_1\exp(a_2 + a_3T^{-1} + a_4T)$
exppoly	Exponential with polynomial exponent	$f(T) = a_0 + a_1\exp(a_2 + a_3T + a_4T^2 + a_5T^3 + a_6T^4 + a_7T^5)$
powx	Exponential with Parameter as base	$f(T) = a_0 + a_1a_2^{(a_3 + a_4T^{-1} + a_5T)}$
optics_temp	Wavelength dependent polynomial provided for each wavelength	$f(\lambda, T) = f(a_0, T) = a_1 + a_2T + a_3T^2 + a_4T^3$
optics_case	Wavelength dependent piecewise-defined polynomial	$f(\lambda, T) = f(a_0, T) = \begin{cases} a_2, & T \le a_1 \\ a_3, & T > a_1 \end{cases}$
optics_lambda	Wavelength dependent polynomial	$f(\lambda) = a_0 + a_1\lambda + a_2\lambda^2 + a_3\lambda^3 + a_4\lambda^4 + a_5\lambda^5 + a_6\lambda^6 + a_7\lambda^7 + a_8\lambda^8$
optics_exp	Exponential wavelength dependent	$f(\lambda) = a_0\lambda^{(1-a_1)}$

5.1 Materials

Example:

```
1  Soot_Kock;Material;1.0;Michelsen et al.- Appl. Phys. B 87,
   503-521 (2007).;
2  alpha_T_eff;const;0.23;0;0;0;0;0;0;0;0;no source;
3  eps;const;1;0;0;0;0;0;0;0;0;no source;
4  rho_p;const;1860;0;0;0;0;0;0;0;0;no source;
5  theta_e;const;1;0;0;0;0;0;0;0;0;-;
6  C_p_mol;poly2;22.5566;0.0013;0;0;0;-1.8195e+06;0;0;0;no
   source;
7  Em_func;optics_lambda;1;0;0;0;0;0;0;0;0;no source;
8  H_v;const;790776.6;0;0;0;0;0;0;0;0;no source;
9  molar_mass;const;0.012011;0;0;0;0;0;0;0;0;no source;
10 molar_mass_v;const;0.036033;0;0;0;0;0;0;0;0;no source;
11 p_v;notSet;0;0;0;0;0;0;0;0;0;notSet;
12 p_v_ref;const;61.5;0;0;0;0;0;0;0;0;no source;
13 T_v_ref;const;3000;0;0;0;0;0;0;0;0;no source;
```

Property	Variable name	Unit	Description
$\boldsymbol{\alpha_T}$	alpha_T_eff	-	Effective thermal accommodation coefficient (conduction)
$\boldsymbol{\varepsilon}$	eps	-	Total emissivity for radiation heat transfer
$\boldsymbol{\varrho_p}$	rho_p	kg/m³	Particle density
$\boldsymbol{\theta_e}$	theta_e	-	Thermal accommodation coefficient (evaporation)
$\boldsymbol{C_{p,mol}}$	C_p_mol	J/mol/K	Molar heat capacity
$\boldsymbol{E(m)}$	Em	-	E(m) absorption value at wavelength λ
$\boldsymbol{E(m)}$	Em_func	-	E(m) absorption function at wavelength λ
$\boldsymbol{\Delta H_v}$	H_v	J/mol	Enthalpy of evaporation
$\boldsymbol{M}$	molar_mass	kg/mol	Molar mass
$\boldsymbol{M_v}$	molar_mass_v	kg/mol	Molar mass of vapor species
$\boldsymbol{p_v}$	p_v	Pa	Vapor pressure function
$\boldsymbol{p_v^*}$	p_v_ref	Pa	Reference pressure for Clausius-Clapeyron
$\boldsymbol{T_v^*}$	T_v_ref	K	Reference temperature for Clausius-Clapeyron

5.2 Gases

Example:

```
1  Nitrogen;Gas;1.0;Standard properties;
2  C_p_mol;const;29.124;0;0;0;0;0;0;0;0;no source;
3  molar_mass;const;0.0280134;0;0;0;0;0;0;0;0;NIST Chemistry
   WebBook, SRD 69;
```

Property	Variable name	Unit	Description
$C_{p,\mathrm{mol}}$	C_p_mol	J/mol/K	Specific heat capacity (isobaric)
M	molar_mass	kg/mol	Molar mass

5.3 GasMixtures

Example:

```
1  Ar/N2-50/50%;GasMixture;1.0;Standard properties;
2  gas_component;Argon;gases/argon.txt;0.5;
3  gas_component;Nitrogen;gases/nitrogen.txt;0.5;
4  L;notSet;0;0;0;0;0;0;0;0;0;notSet;
5  gamma_eqn;notSet;0;0;0;0;0;0;0;0;0;notSet;
6  therm_cond;notSet;0;0;0;0;0;0;0;0;notSet;
```

Property	Variable name	Unit	Description
κ_a	therm_cond	W/m/K	Thermal conductivity
L	L	m	Mean free path
γ	gamma_eqn	-	Custom heat capacity ratio

Example:

```
gas_component;Argon;gases/argon.txt;0.5;
gas_component;Nitrogen;gases/nitrogen.txt;0.5;
```

Element	Example	Description
<identifier>	gas_component	Identifier, indicating that the lines describes a component of the mixture
<database_name>	Argon	Gas name; should be the same as in the specific gas database file.
<database_location>	gases/argon.txt	Path to the gas database file.
<composition>	0.5	Percentage in the mixture, in this case 50 %. All components should add up to 100%

5.4 LIISettings

The LIISettings file structure is explained earlier in section 4.1 . For each LII detector setup (different bandpass filters, different detectors) a new LIISettings file can be created. This allows switching between different setups for the analysis.

6 Signal Processing

Processing steps can be individually ordered and parameters can be set individually for every measurement run, for a group or globally. This allows the systematic processing of all loaded measurement runs. The following processing plugins are available and can be applied multiple times in a signal processing chain.

6.1 Processing Plugins

The following processing plugins are available within LIISim:

- Arithmetic Operations
- Baseline Correction
- Calibration (gain, sensitivity)
- ND-Filter Correction
- Get Section
- Moving Average
- Multi Signal Average
- Normalize
- Overwrite
- Resolution Reducer
- Signal Arithmetic
- Simple Data Reducer
- Swap Channels
- Transfer
- Temperature Calculator

6.1.1 Arithmetic Operations

Perform arithmetic operations on signal trace. The operation will be applied to every data point in a signal.

Parameters:	
Operation	The following operations are available: - Multiplication - Division - Addition - Subtraction
Channel	Selects the channel used for the operation
Value	The value for the operation (e.g. multiply/divide by this value)

6.1.2 Baseline Correction

“Baseline” provides two options for signal offset correction:
i) **For gated PMTs**: the signal before the gate opening can be used to calculate the offset. This is done by averaging a user-defined signal range and subtract this from the signal.

ii) **For ungated PMTs**, the baseline offset for each channel can be defined in the LIISettings file. During processing these values are automatically retrieved and subtracted from the signal trace.

Parameters	
Calculate offset from	Selects if the baseline correction should be calculated from all signals or from each signal separately, LIISettings uses the "offset" values for each channel from the LIISettings-file
Start [ns]	Start time in nanoseconds
End [ns]	End time in nanoseconds

6.1.3 Calibration (gain, channel sensitivity)

Calibration values are stored in the LIISettings file for each channel i.
The gain correction factor y_i and channel-sensitivity calibration values are multiplied with the signal traces giving calibrated signal traces.

x_i is the channel gain voltage, $x_{i,\mathrm{ref}}$ the gain reference voltage and A_i the gain calibration value for the respective channel (coefficient B_i is currently not used).

Parameters	
Operation	Following operations are available: - *Gain (exp)*: Performs gain correction using the function: $y_i = \exp\left(-A_i \ \ln\left(x_i / x_{i,\mathrm{ref}}\right)\right)$ - *Gain (log(10)):* Performs gain correction using the function: $y_i = 10^{\left(-A_i \ \log\left(\frac{x_i}{x_{i,\mathrm{ref}}}\right)\right)}$ - *Channel Sensitivity*: Performs sensitivity correction using the PMT sensitivity values saved in the LIISettings
Source of gain values	If Gain (exp) or Gain (log(10) is selected: Uses either the values stored in the run's settings file or manually entered values
Channel 1 Channel 2 Channel 3 Channel 4	If manual is selected as source of gain values, these values are entered here.

6.1.4 ND-Filter Correction

This plugin uses the transmission values from the current LIISettings according to the selected filter identifier. The identifier can be automatically selected from the MRun settings file or manually defined. Each signal trace is divided by the respective filter transmission values.

Parameters	
Filter Value Source	*Copy from run settings*: uses the filter identifier embedded in the run settings *Manual*: allows a manual entered filter transmission
Filter transmission (%)	If manual is selected as source, this filter transmission value is used

6.1.5 Get Section

Select a specified section from a signal. The following processing steps are applied only on this section.

Parameters	
Start from peak	Select this option to let LIISim auto detect a signal peak and use this as start time (experimental)
Reset time offset	Sets the start time of the selected section to zero.
Start [ns]	Section start time in nanoseconds
End [ns]	Section end time in nanoseconds

6.1.6 Moving Average

Performs a moving average over a signal.

Parameters	
Channel	The channel to perform the moving average on (or 'all' for all channels)
Window size	Window size used for averaging. ('5' means 2 datapoints before and 2 datapoints after each datapoints is used for the averaging.

6.1.7 Multi Signal Average

Performs an average over multiple signals. This processing step calculates an average signal from all signals. For all following processing steps this average signal is used. This is similar as using an average function on the oscilloscope).

Parameters	
Start signal	First signal number to include in the average
End signal	Last signal number to include in the average

6.1.8 Normalize

Normalizes a signal trace to the peak or a specific value.

Parameters	
Operation	Select if you want to normalize the signal by peak or by value
Channel	The channel to perform the normalization on (or 'all' for all channels)
Value	If "by value" is chosen, this value is used for normalization

6.1.9 Overwrite

Overwrite a channel by another channel. This plugin can be used to pass signals from Raw processing chain to absolute processing chain.

Parameters	
Source	The source signal type for the signal
Channel A	The destination channel (will be overwritten)
Channel B	The source signal

6.1.10 Resolution Reducer

Artificially reduce the vertical resolution of a signal trace. This plugin can be used to simulate the influence of lower detection resolution (e.g. Resolution of data recorded by a 12bit oscilloscope can be reduced to 8bit to see if the quality is still sufficient).

Parameters	
Resolution	Target bit resolution
Range	The vertical resolution range, mapped to the selected resolution

6.1.11 Signal Arithmetic

Used to perform arithmetic operations across multiple channels.

Parameters	
Operation	The following operations are available: - Multiplication (A*B) - Division (A/B) - Division (B/A) - Addition (A+B) - Subtraction (A-B) - Subtraction (B-A)
Channel A	The destination channel (will be overwritten)

Channel B	The source signal

6.1.12 Simple Data Reducer

This plugin reduces the data size by removing n data points consecutively. This plugin can be used to decrease computing time for data evaluation.

Parameters	
Skip data points	The number of data points which will be removed between two remaining data points

6.1.13 Swap Channels

Used to swap two channels.

Parameters	
Channel A Channel B	The channels which should be swapped

6.1.14 Transfer

Copies the final calculation data from the raw processing chain to the absolute processing chain.

6.1.15 Temperature calculation (Two-Color / Spectrum)

For the calculation of temperature traces, two methods are implemented in LIISim: Two-color pyrometry and spectral fitting of Planck's law.

Two-color pyrometry requires calibrated LII signals at two independent wavelengths:

$$T_\mathrm{p} = \frac{h\,c_0}{k_\mathrm{B}}\left(\frac{1}{\lambda_2}-\frac{1}{\lambda_1}\right)\left[\ln\left(\frac{I_\lambda(\lambda_1,T_\mathrm{p})}{I_\lambda(\lambda_2,T_\mathrm{p})}\frac{E(m_{\lambda_2})}{E(m_{\lambda_1})}\left(\frac{\lambda_1}{\lambda_2}\right)^6\right)\right]^{-1} \quad (1)$$

where T_p is the particle temperature, h the Planck constant, c_0 the speed of light in vacuum, k_B the Boltzmann constant, λ_i the bandpass center-wavelength of the detector, I_λ the calibrated intensity, and $E(m_\lambda)$ the absorption function. It needs to be noted that this formula is derived using the Wien approximation and differentiates in this aspect from a spectral fitting.

Spectrum: When two or more spectral detection channels are available, a spectral fit including all channels can be performed. For this analysis, Planck's law corrected for deviation from black-body behavior is fitted to the data points of all selected channels:

$$I(\lambda,T_\mathrm{p}) = f_V G_\lambda C_{\lambda,\mathrm{abs}}\frac{2hc_0^2}{\lambda^5}\left[\exp\left(\frac{h\,c_0}{k_\mathrm{B}\lambda\,T_\mathrm{p}}\right)-1\right]^{-1} \quad (2)$$

Where f_V is the volume fraction during the given laser shot, G_λ scales the emitted intensity based on the geometry and optical efficiencies of the detector, and $C_{\lambda,\mathrm{abs}}$ is the wavelength-dependent absorption cross-section given by the Rayleigh limit of Mie theory:

$$C_{\lambda,\mathrm{abs}}(d_\mathrm{p}) = \frac{\pi^2\, d_\mathrm{p}^3}{4}\frac{E(m_\lambda)}{\lambda} \tag{3}$$

In LIISim, wavelength-independent properties of equation (2) and (3) are summarized in a general scaling factor C, which simplifies equation (2) to:

$$I(\lambda, T_\mathrm{p}) = C\,\frac{E(m_\lambda)}{\lambda}\frac{2hc_0^2}{\lambda^5}\left[\exp\left(\frac{h\,c_0}{k_\mathrm{B}\lambda\,T_\mathrm{p}}\right) - 1\right]^{-1} \tag{4}$$

The **TemperatureCalculator**-Plugin uses the result from the absolute/raw processing chain as input. This can be specified within the plugin's settings. Further, it can be selected which channels should be used for analysis and which optical properties should be used for the calculation/fitting.

6.2 Linking of processing steps

You can link the parameters of plugins over multiple runs – so if you change one parameter from one run, it also changes the parameter of another run. Before adding any plugin, you can choose between:

- 'all' (A)
- 'group' (G)
- 'single' (S)

link mode. 'All' will link the parameters between all loaded runs. 'Group' will link them between runs inside the same group. 'Single' will not link them. For a GUI example see section 7.5.3.

7 Graphical User Interface (GUI)

7.1 Home screen

The home screen (Figure 7-1) shows a log window and gives access to load/save functions for session, program settings and signal data.

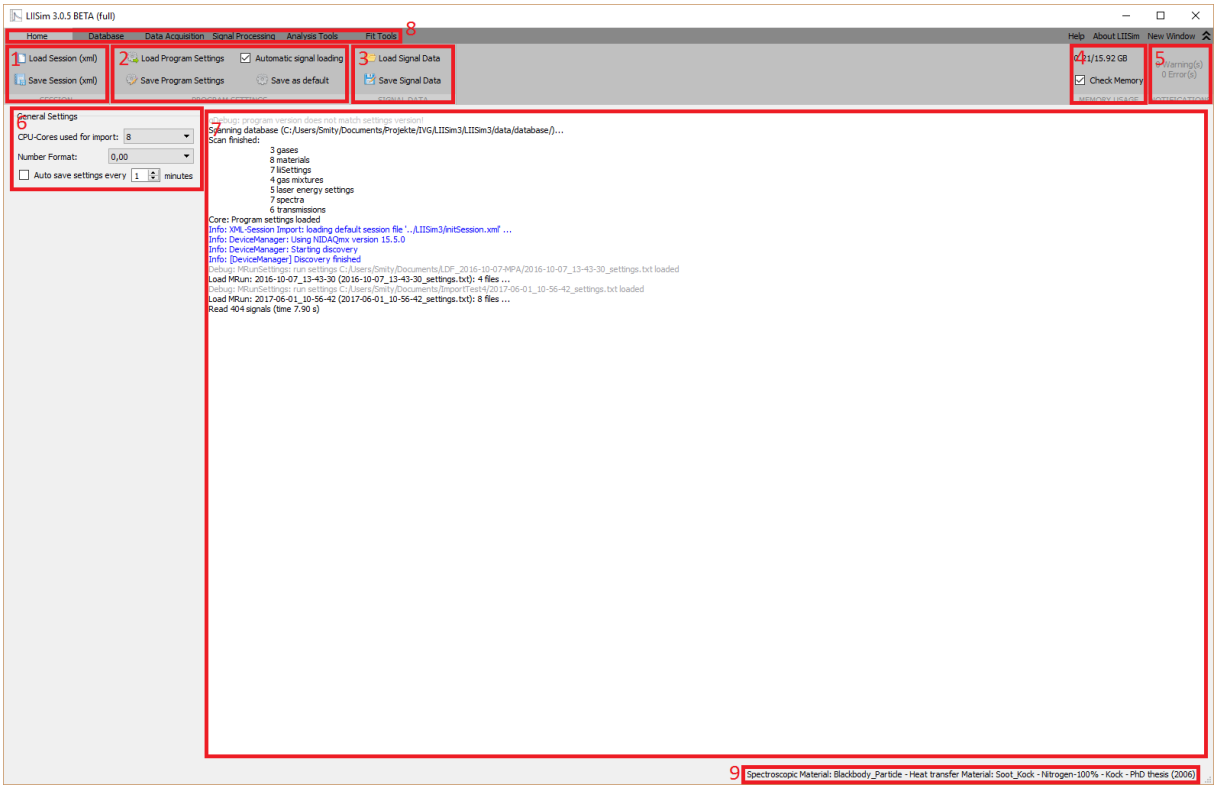


Figure 7-1 Home screen

(1) Session

Through these buttons, a session file (.xml) can be saved or restored. The program state (called a session) contains information about loaded signal data (file names and folders) and signal processing steps (type of processing steps, parameters, linked steps, …).

Saving/loading a session can be helpful if different data sets need to be analyzed, but require different signal processing steps.

(2) Program settings

Program settings are saved in .ini-files. The default settings are automatically loaded on program start and saved in the "defaultSettings.ini". Individual program settings can be saved and loaded.

"Save as default" can be used to save the current program settings in the defaultSettings.ini file.

"Automatic signal loading" determines if signals from the previous session are automatically loaded on program start.

(3) Signal data

Opens the load or save dialog for signals. See section 7.2 or 7.3 for further information.

(4) Memory widget (visible in all tabs)

Shows the currently used and available RAM. If the “Check Memory“-check-box is checked, LIISim tries to calculate the RAM usage before calculation to prevent program crashes.

(5) Notification widget (visible in all tabs)

Displays the current warning/error count. If clicked, opens a window which shows all warnings/errors since start and resets the count.

(6) General settings

“CPU-Cores for import”: Sets the number of CPU-cores which are used during import.

“Auto save settings every X minutes”: If checked, saves the session and program settings every X minutes, where X is the number selected in the spin box.

(7) Log

Shows all log messages since start.

(8) Navigation

(9) Status bar

7.2 Signal Import

Signal import/export dialog can be accessed through the home screen (see section 7.1).

7.2.1 CSV formatted signals

To import signals in CSV format, you should choose the 'CSV AUTO' tab in the Signal Import dialog **(1)**. The dialog will look similar to the one below.

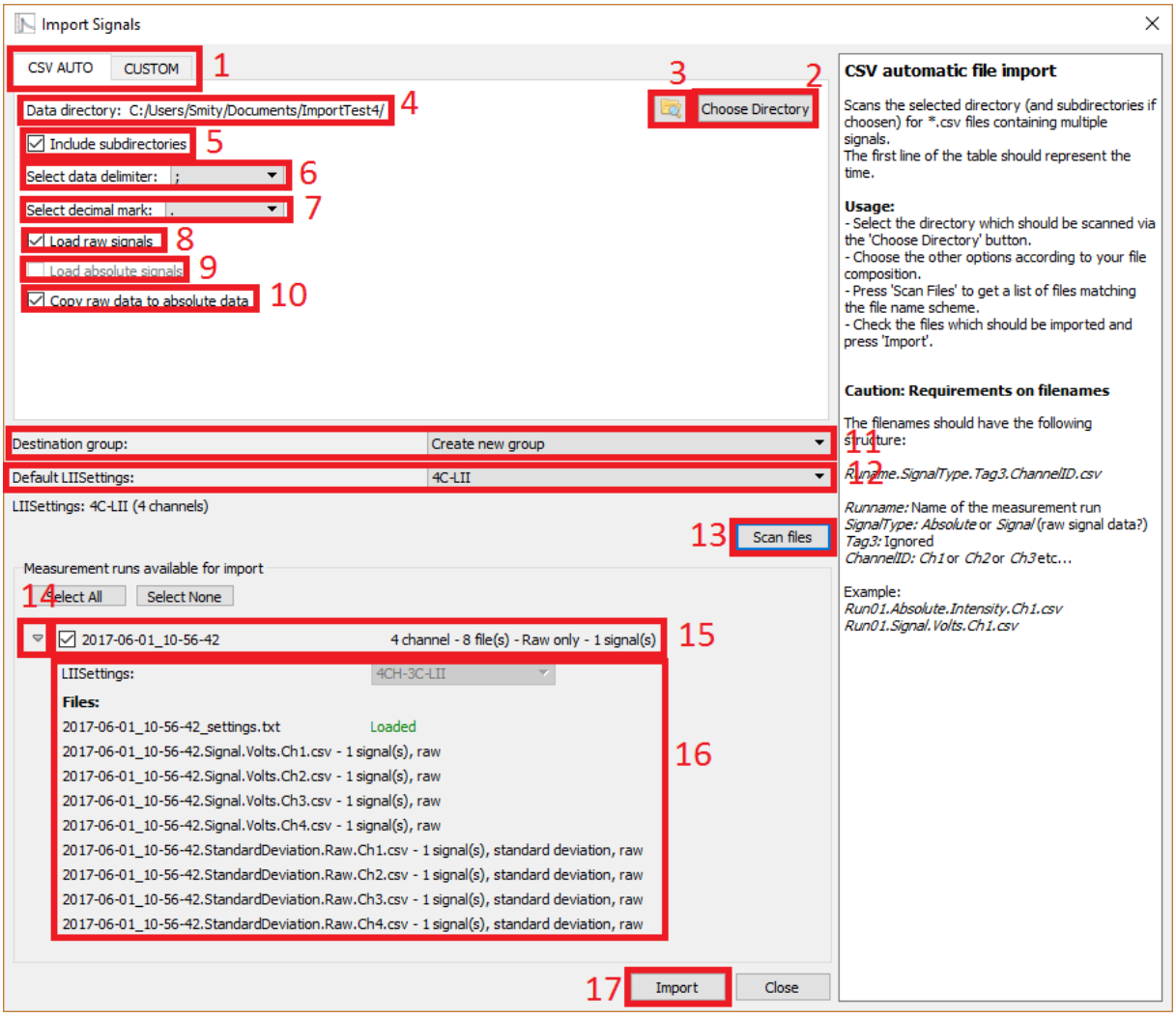


Figure 7-2 CSV import window

The following steps are necessary to import signal data:

- First, you need to specify the directory where the files are located. This is done using the "Choose Directory" button **(2)**. **(4)** shows you the currently selected directory and with **(3)** you can open the selected directory in your file explorer.
- If you want to include files in subdirectories in your import, enable the checkbox at **(5)**.
- Select your data delimiter **(6)** and decimal mark **(7)** according to your data.
- If you want to import raw signal data, enable the checkbox at **(8)**. In general, you should leave this checkbox enabled.
- If you want to load absolute signal data, enable the checkbox at **(9)**. If you have no absolute signal data, leave this checkbox disabled, and enable the **(10)** checkbox, so the raw signal data is copied over as absolute signal.
- Choose the destination group where the measurement run should be added **(11)**.

- Choose the default LIISettings to use, if not defined elsewhere **(12).**
- Click “Scan Files” **(13)**. LIISim will now search the selected directory for files matching the naming scheme.
- If files are found, they are grouped according to their run name and shown with some additional information **(15)**. The checkbox in front of the name defines if the run should be loaded or not.
- Via a click on **(14)** you can toggle the details of a run **(16)**. Here you can set the LIISettings at a per run level, if there are no run settings available for the particular run.
- If you have set up everything you want, click “Import” **(17)** to let LIISim import the selected runs.

Important: To be recognized by the import routine, the CSV files should have the following name scheme with the tags below separated by dots:

<Runname>.<SignalType>.<Tag>.<ChannelID>.csv

Example: Run3.Signal.Anything.Ch2.csv

<Runname>	Can be any text, but must not contain a full stop/period. Will be used as run name and should be the same across one measurement run.
<SignalType>	Can be either ***Signal*** for raw signal data or ***Absolute*** for absolute signal data.
<Tag>	Can be any text, but must not contain a full stop/period. Will be ignored.
<ChannelID>	*Ch#*, with the # replaced by the channel number, ranging from 1 to 4.

7.2.2 Custom formatted signals

To import existing *.txt or *.dat files, you should use the custom import mode. Open the Signal Import dialog and select the “CUSTOM” tab **(1)**. The dialog will look similar like the one below.

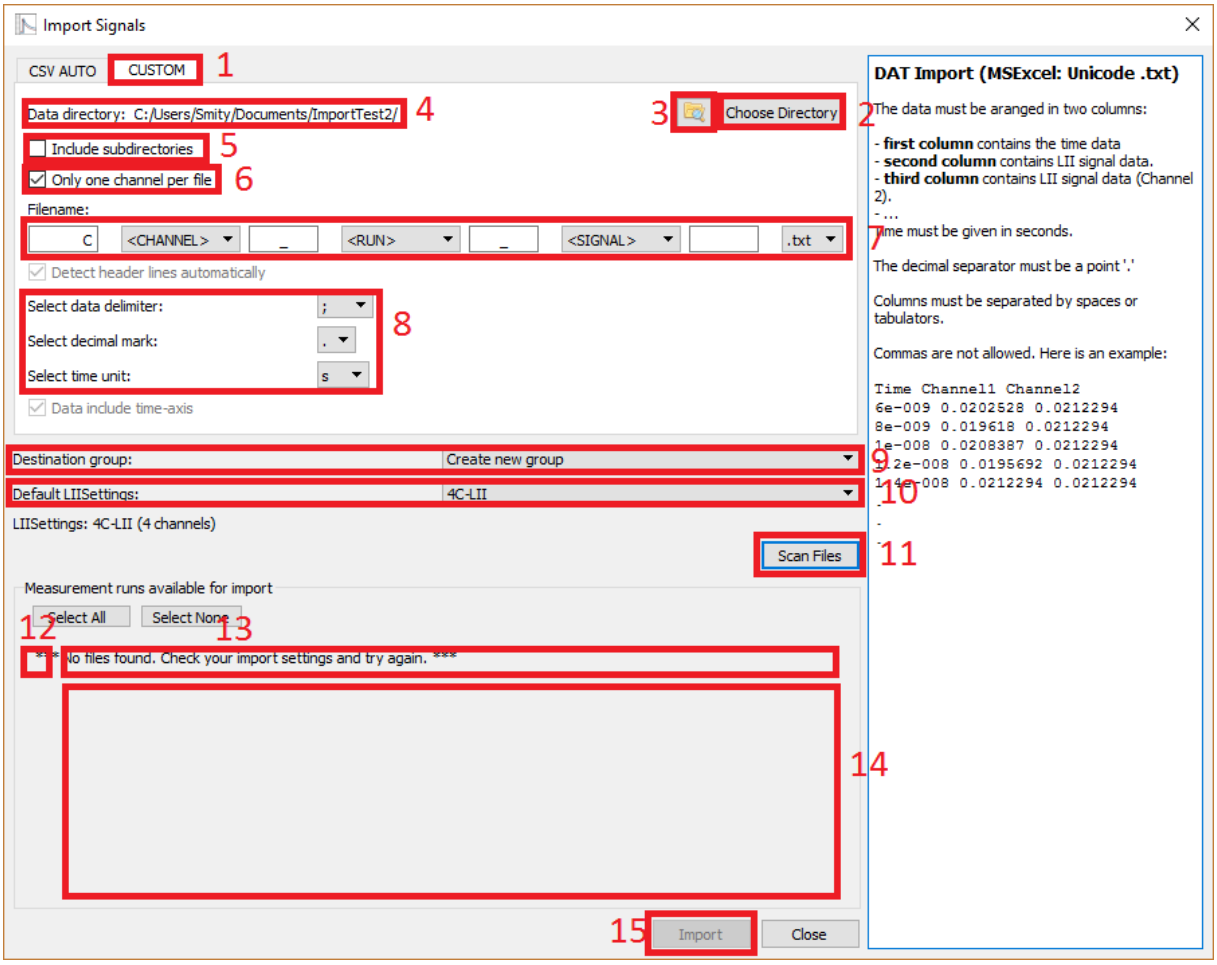


Figure 7-3 Custom import window

- First, you need to select the directory where the files are located with the “Choose Directory”-button **(2)**. You will see your currently selected directory at **(4).** With **(3)** you can open the currently selected directory in your operating system file browser.
- If you want to load files from multiple directories, you can enable this by selecting the “Include subdirectories” checkbox **(5)**. Note that these files all need to have the same naming scheme, which will be defined later.
- When your data is separated into multiple files, with only one channel per file, you should select the checkbox at **(6)**. Note that you need to select <CHANNEL> next step, otherwise the file scan will return wrong results and a signal import will fail.
- Now you can select the naming scheme of your files **(7).**

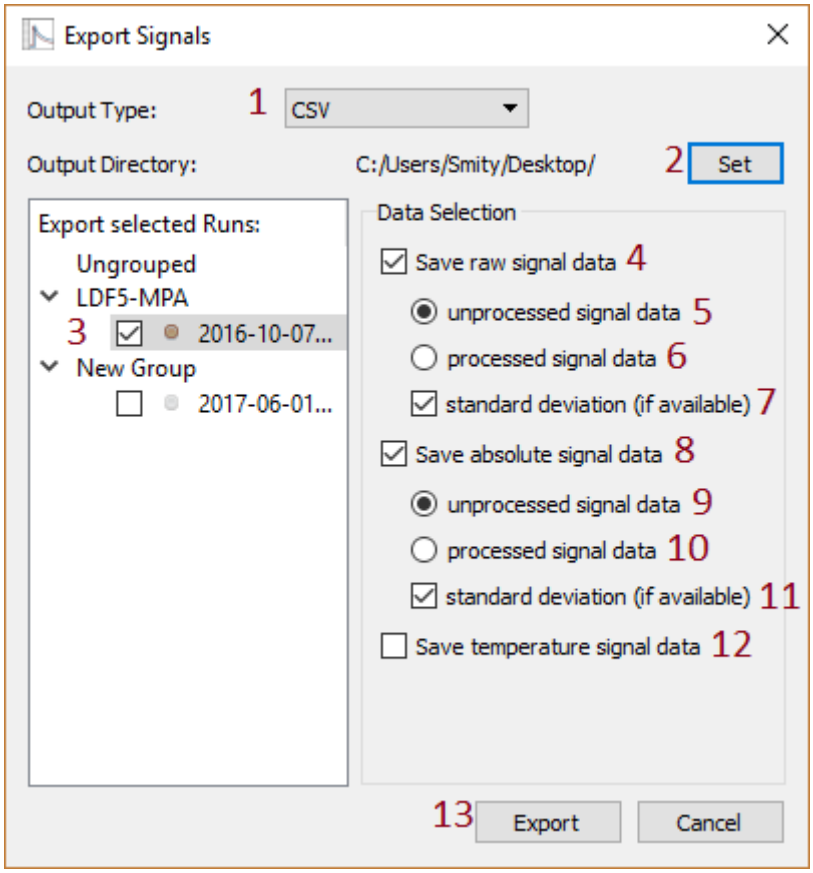
- The following placeholder are available and help with the automatic detection of custom file names:

Tag	Description	Example
<channel>	Channel identifier (successive numbers, ranging from 1 to x depending on LIISettings)	Run3_CH1_0001.txt
<signal>	Signal identifier (successive numbers)	signal_0001.txt signal_1.txt …
<run>	This string will be used as name inside LIISim	Run3_002.txt CH1_Run20.txt
<integer>	Any number, not used in a particular way	Run3_002_1000mbar.txt CH1_Run20_10HAB.txt
<…>		

- Select the format in which your data is available here **(8)**. Wrong selection might lead to empty signals after import.
- Select the group to which the run should be added after import **(9)**.
- Select your default LIISettings **(10)**. You can change these on a per-run basis after the file scan in the run details.
- Click on “Scan Files” **(11)** to let LIISim scan for importable files.
- Select the runs you want to import. **(13)** Shows the name of the run and some additional information. If you click on **(12)**, you can open the details of that particular run **(14)**.
- If you have selected all runs you like to import, press “Import” **(15)**.

7.3 Signal Export

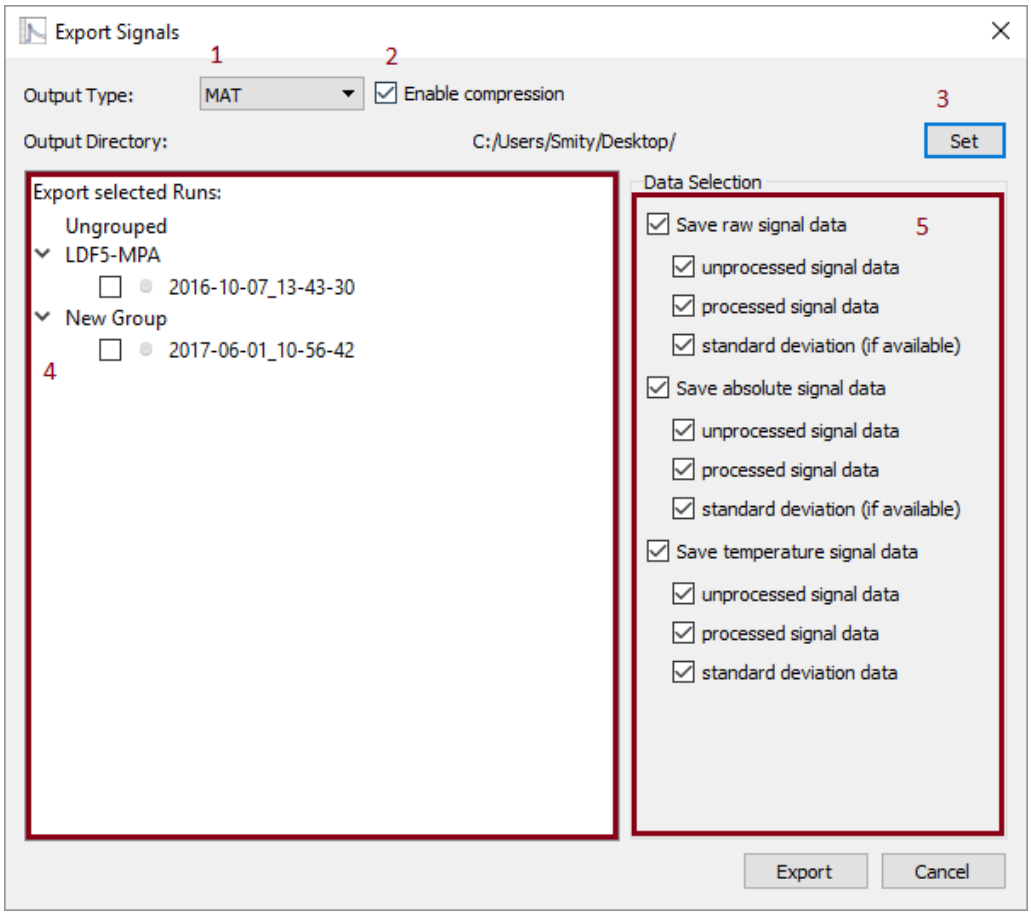
7.3.1 CSV Export



Steps to export a signal as a CSV file:

- Choose the right export type **(1)**, in this case 'CSV'.
- Select your target directory with **(2)**.
- Select the runs you would like to export in **(3)**
- Activate the checkbox at **(4)** to export raw signal data and choose either the unprocessed **(5)** or processed **(6)** signal. (Additionally, standard deviation data can be exported by activating the checkbox at (7).)
- For the absolute signal, this is done analog to the raw signal, by activating **(8)** and choosing unprocessed **(9)** or processed **(10)** signal data.
- To export temperature signal data, activate the checkbox at **(12)**.
- Finally, press 'Export' **(13)** to export the selected signal data.

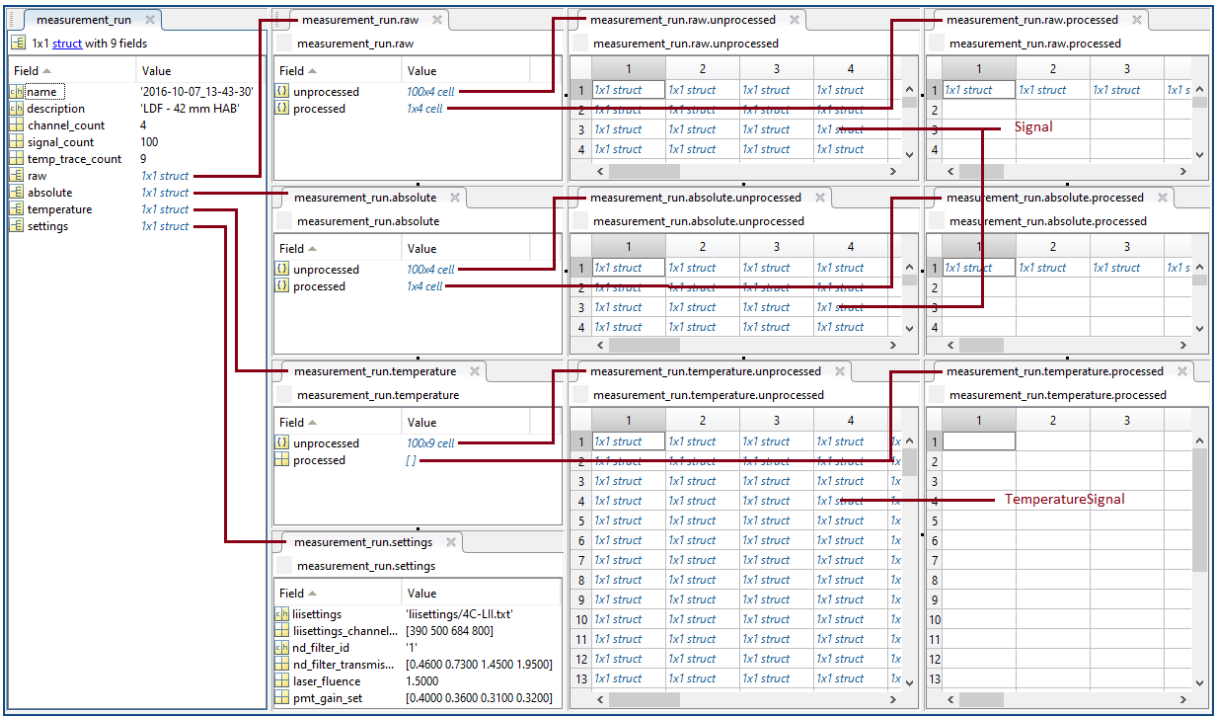
7.3.2 MATLAB Export

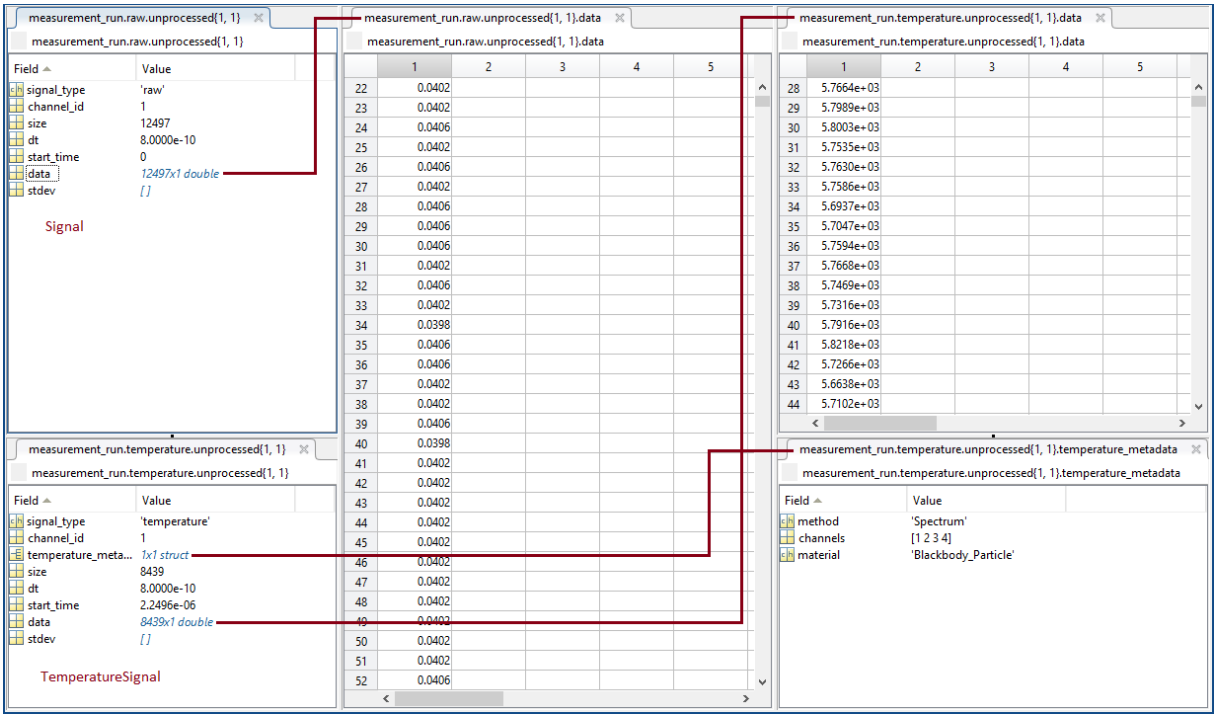


Steps to export a signal as a Matlab file:

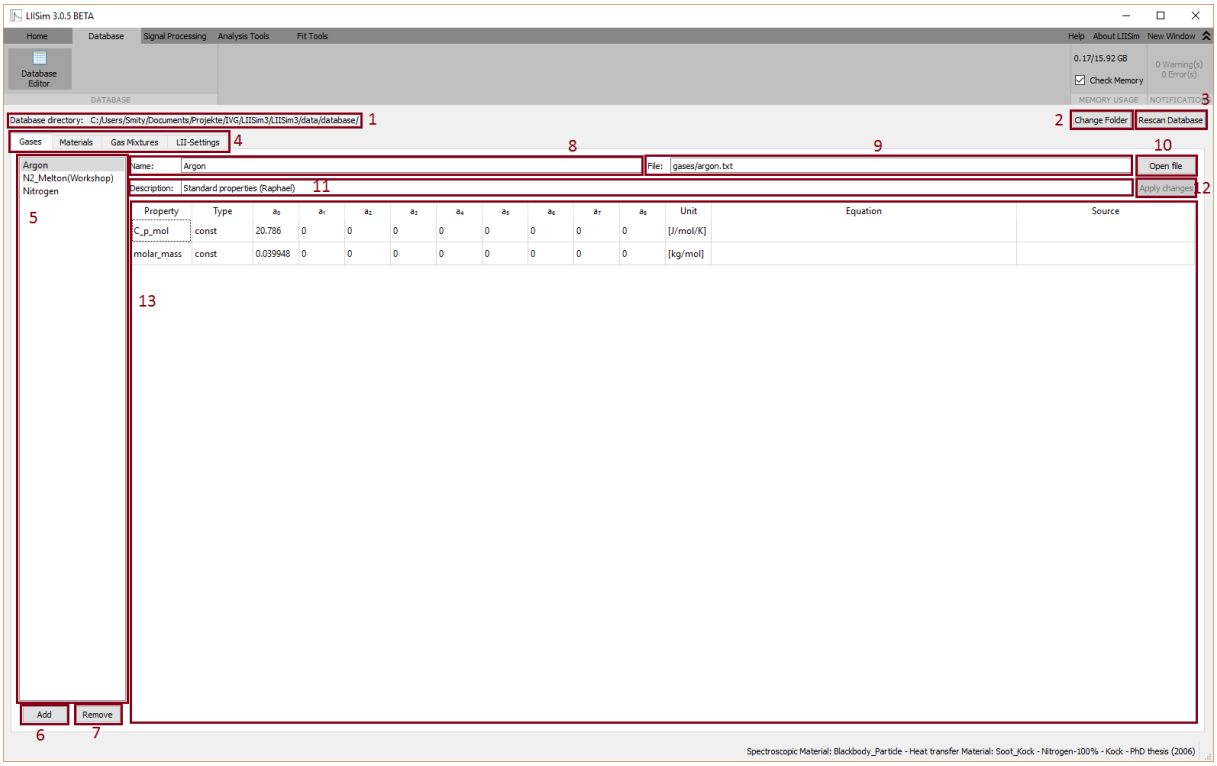
- Choose ‘MAT’ as export type **(1)**.
- Select if you like the file to be compressed **(2)**.
- Select your target directory **(3)**.
- Choose the runs you would like to export **(4)**.
- Select the data types you like to export **(5)**. See the CSV export for a more detailed description of the available data types. In the contrary to the CSV export, all data will be exported into one file.
- Press ‘Export’.

7.3.3 MATLAB File Structure





7.4 Database Editor



(1) **Database path**

Shows the currently selected database path.

(2) **Change folder**

Changes the current database path.

(3) **Rescan database**

Rescans the database folder for new / changed files. Will reload the complete database. If any manual changes to the text files are made, this command should executed.

(4) **Databases**

Switch between the different database categories.

(5) **List of database elements**

Lists the available database files in the current database category.

(6) **Add database element**

Creates a new database file from the currently selected category.

(7) **Remove database element**

Removes the database element and deletes the associated file. ***Note:*** *there is no warning before deletion.*

(8) **Element name**

Shows the element name. Can be edited.

(9) **Database element path / filename**

Shows the element path (from the database path) and filename. Can be edited.

(10) **Open file**

Opens the currently selected element with the default text editor.

(11) **Element description**

Shows the element description. Can be edited.

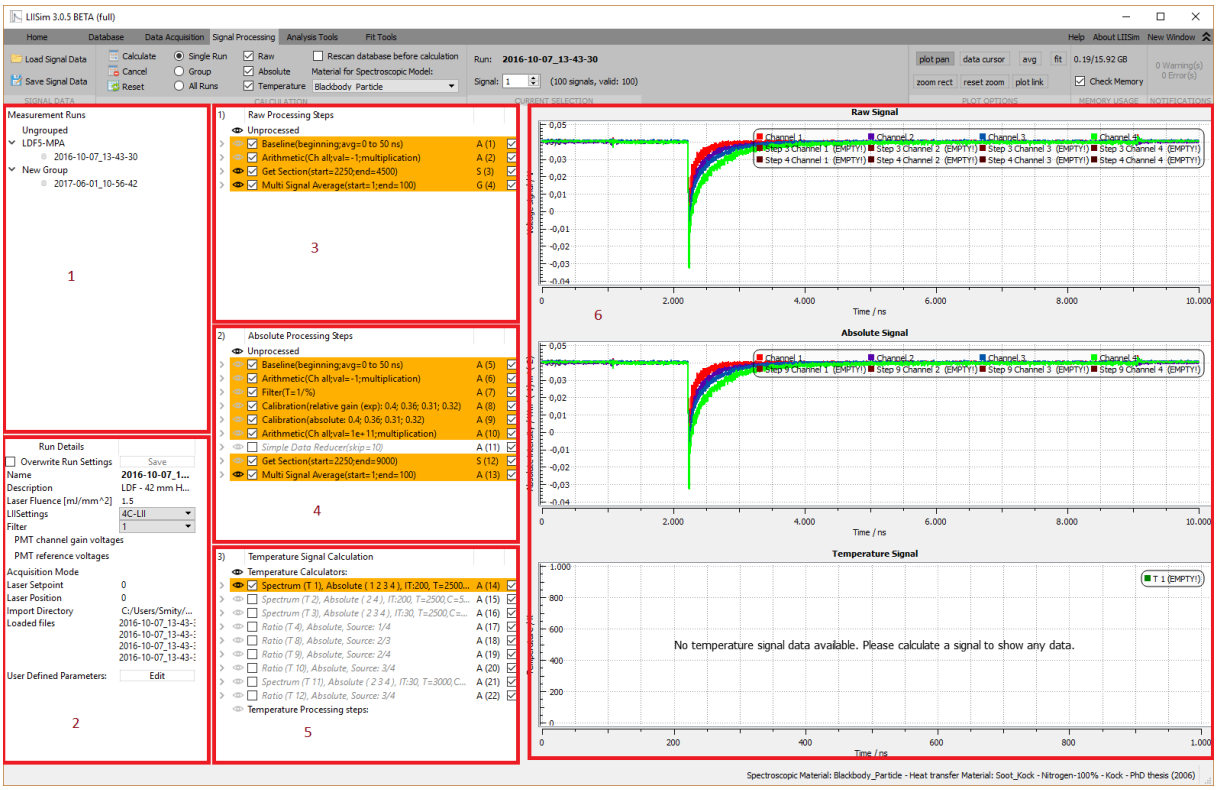
(12) **Apply changes**

Applies the changes made to the file.

(13) **Database element data**

The data defined in the element file.

7.5 Signal Processing



(1) **Measurement run list**

Shows all currently loaded runs. Groups can be used to order runs from different experiments. Parameters of processing chain can be set individually for every measurement run, by group or global.

(2) **Measurement run details**

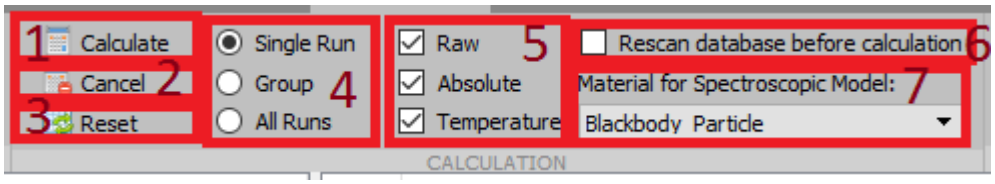
If a run is selected, this shows details and parameters saved in the run settings. See section 7.5.3.

(3) **Processing chain for raw signals**

(4) **Processing chain for absolute signals**

(5) **Processing chain for temperature calculations**

7.5.1 Calculation widget / current selection



(1) **Calculate**

Starts the calculation with the currently selected measurement runs.

(2) **Cancel**

Cancels any running calculation.

(3) **Reset**

Resets the data buffers and calculation stats (indicated by background color)

(4) **Calculation mode**

Select measurement runs to be calculated:

Single Run: Only the currently selected run will be calculated.

Group: All runs in the current group are processed, active group is determined by the currently selected run.

All Runs: Calculate all runs.

(5) **Calculation of signal types**

Select which signal type you would like to calculate. Processing chains can be disabled to increase computing speed.

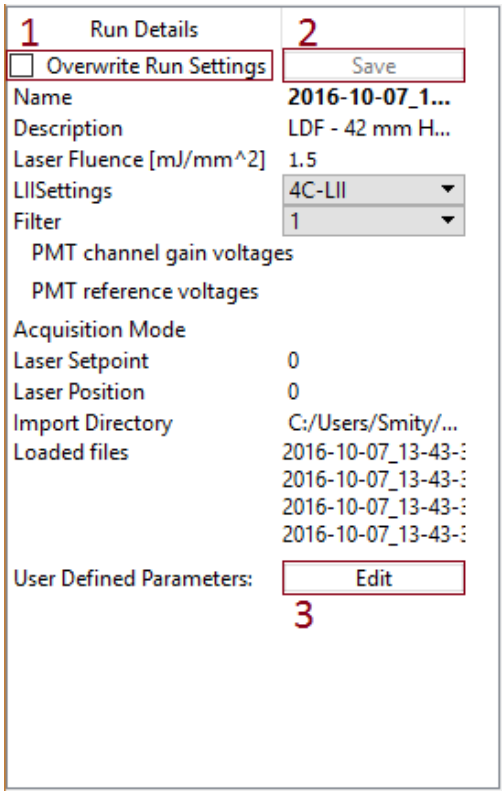
(6) **Rescan database before calculations**

If this is checked, the database is rescanned before each calculation cycle. Useful if you are experimenting with different material settings and editing database files manually.

(7) **Material for spectroscopic model**

Select the material from database, which will be used in temperature calculations in the “global” setting. Material contains information of optical properties, i.e. E(m).

7.5.2 Run details



This window is displayed everywhere in LIISim in the lower left corner and shows the meta data for the currently selected run. For details regarding the parameters, see section 4.2.1 (MRun settings).

(1) **Overwrite Run Settings**

If enabled, any changes in the run settings are saved at program exit. ***Caution:*** *This will overwrite the settings files for ALL loaded runs if enabled.*

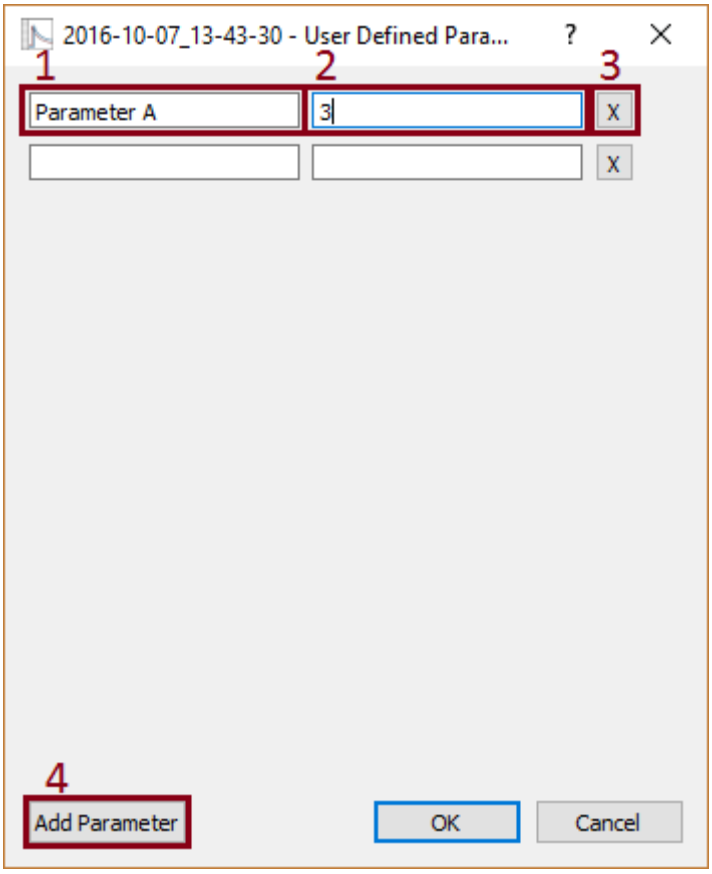
(2) **Save**

Saves the currently made changes **permanently**. The checkbox **(5)** must be enabled before saving and can be disabled afterwards.

(3) **Edit**

Opens the user-defined parameters editor, described in section 7.5.3.

7.5.3 User-defined parameters

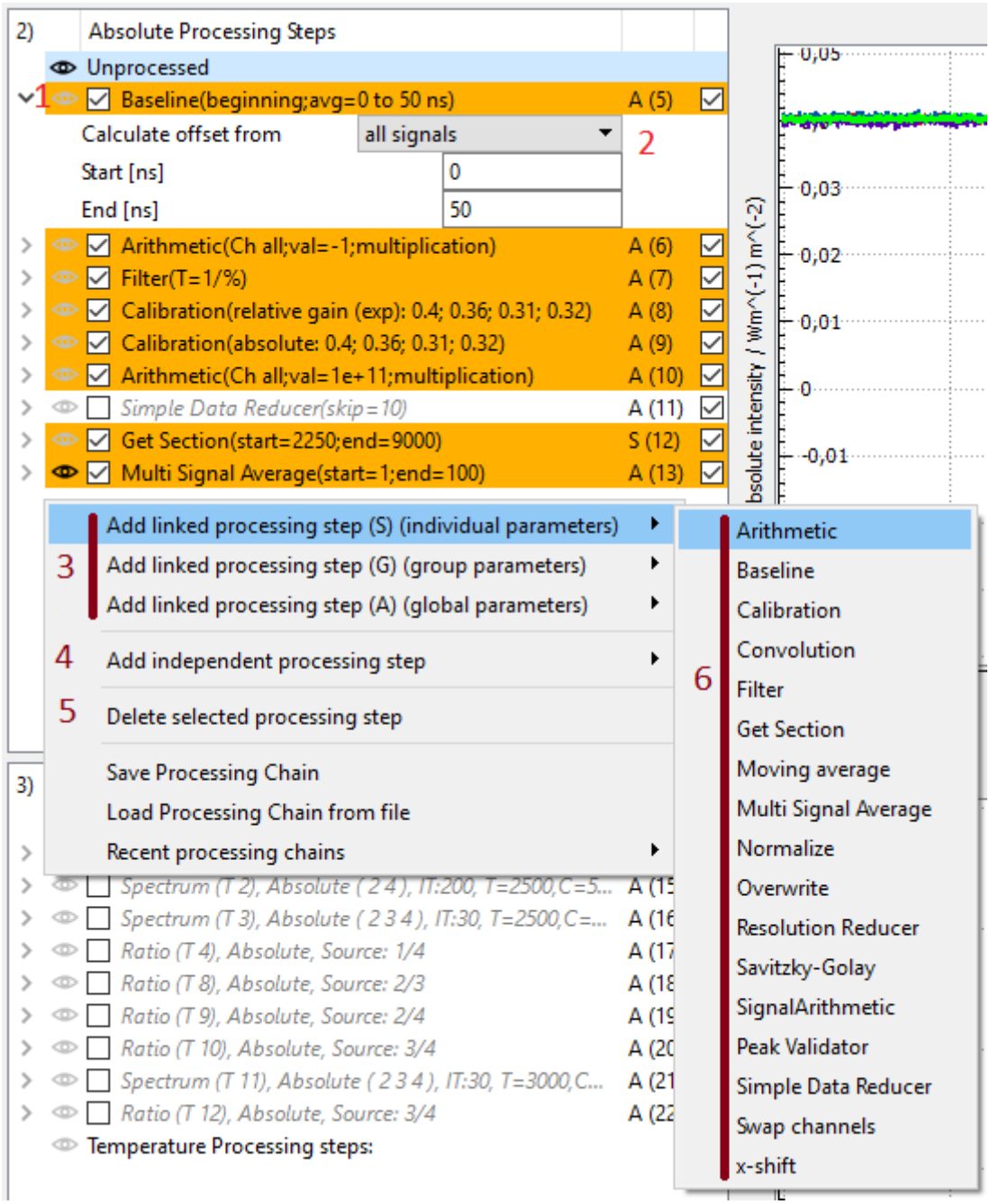


User-defined parameters (UDPs) can be used to add parameters which are not pre-defined (i.e. process pressure, HAB, …).

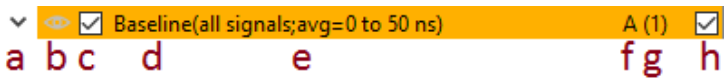
These UDPs will be displayed in the run details and can be used in the parameter analysis tool.

In the UDP window, **(1)** will be the parameter name, **(2)** respectively the parameter value. **(3)** deletes a parameter and **(4)** adds a new name-value field to the list.

7.5.4 Processing steps



(1) Plugin



(a) Toggles visibility of plugin settings

(b) Toggles the visibility of the processing step inside the plot

(c) Enables the plugin. If unchecked, the plugin will be skipped during calculation

(d) Plugin name

(e) Short parameter summary

(f) Link type (A, G, S) (see Linking of processing steps)

(g) Link identifier. (Indicates which Plugins are connected)

(h) Activate/Deactivate intermediate result buffer. If this is activated the result of this processing step is internally saved and can be visualized. If large data sets are processed, memory consumption can increase exponentially. In this case deactivation helps to reduce the memory used by LIISim and speeds up processing time.

(2) Plugin settings

Shows the current plugin settings.

(3) Add linked processing step (see Linking of processing 6.2)

Add a plugin to the processing chain. The plugin will be added to the processing chain of all runs. You can choose between different parameter link states (A, G and S).

(4) Add independent processing step

Adds a plugin only to the currently selected run. This processing step is not added to any other processing chain.

(5) Delete selected processing step

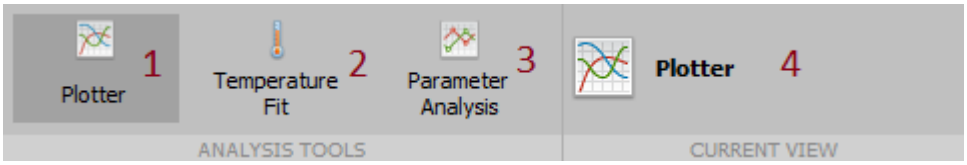
Removes the selected plugin from the processing chain.

(6) Plugin list

Shows all available plugins.

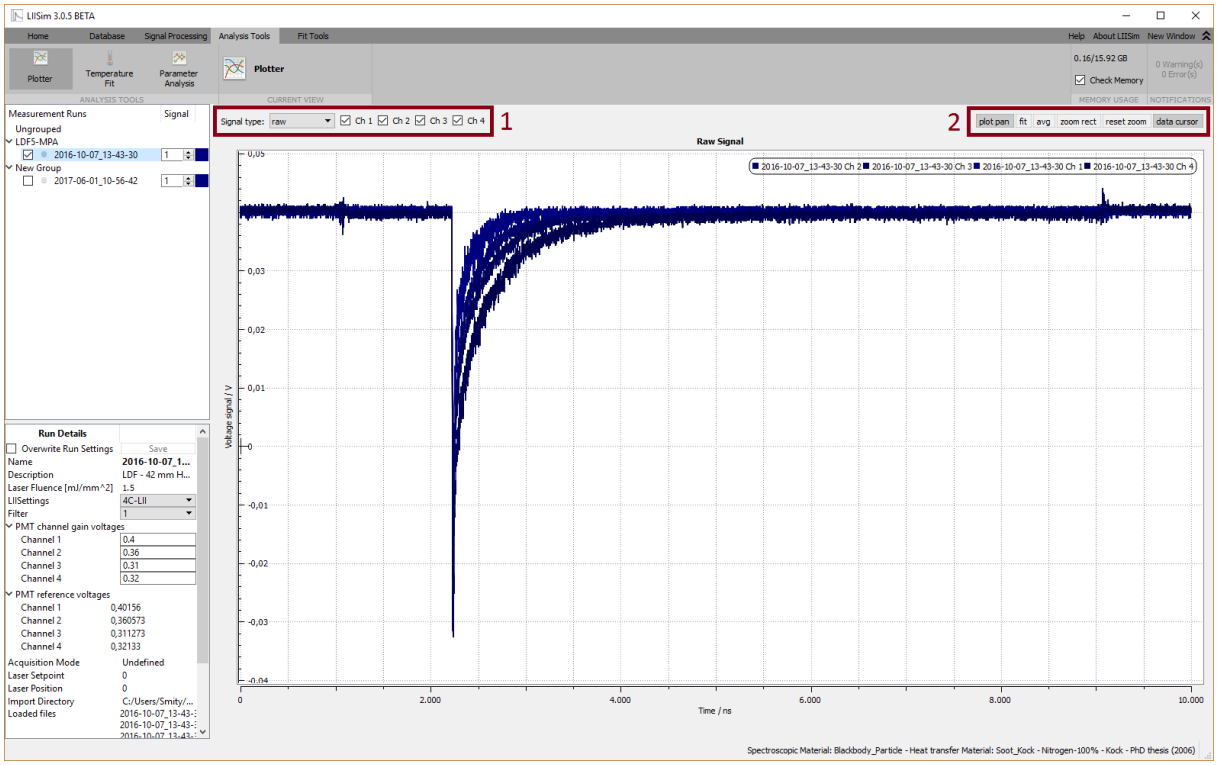
7.6 Analysis Tools

Analysis tools can be used to visualize previously processed data and extract key information for further analysis.



The different tools can be switched using **(1) (2)** and **(3)**. The currently selected tool is shown in **(4)**.

7.6.1 Plotter (Plot Tools)



The plot tools are used for comparison of different measurement runs (selectable in the left top list) for an in-depth analysis. The signal type and channel can be changed through **(1)**. The plot control **(2)** is described in section 7.8.1 (Plot controls).

7.6.2 Temperature Fit

This tool only works with temperature traces calculated with the “Spectrum”-method. “Two-Color” traces cannot be visualized.

First, a measurement run needs to be selected.

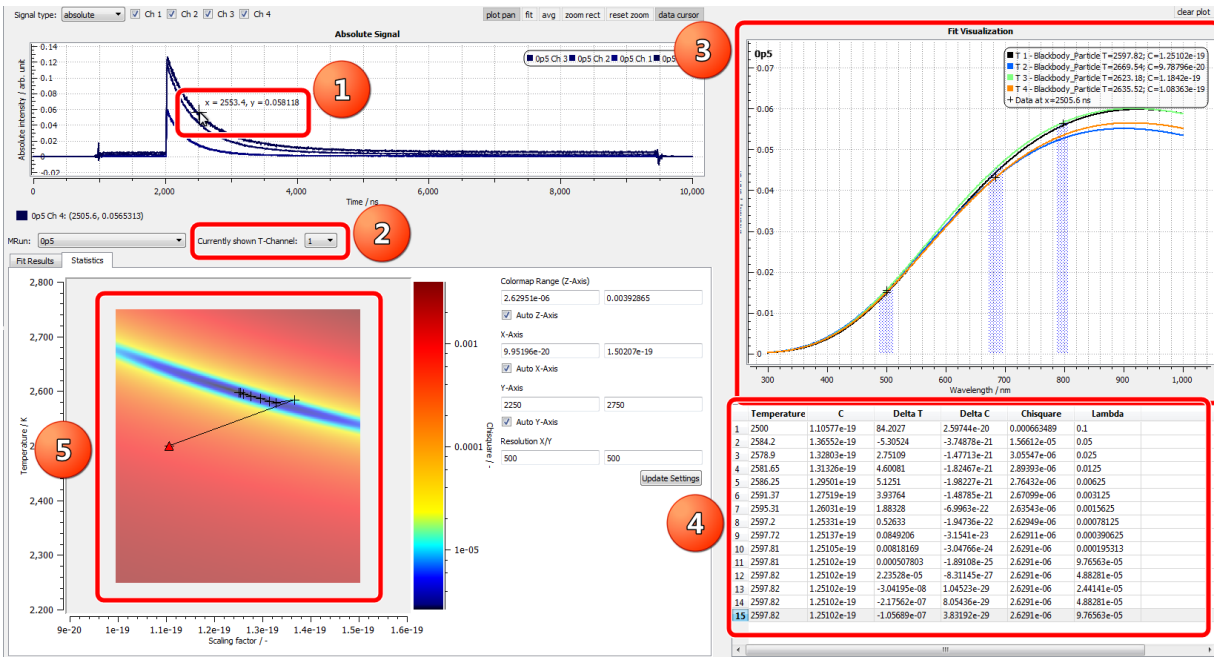
(1) Select now in the top left plot a data point you want to analyze. The temperature trace selected in **(2)** is then visualized.

(3) Compares the selected spectral fit to other available “Spectrum” temperature fits.

(4) Shows the spectral fitting iterations for the selected data point and **(4)** the chi-square map of the two parameters used during fitting (Temperature and Scaling factor).

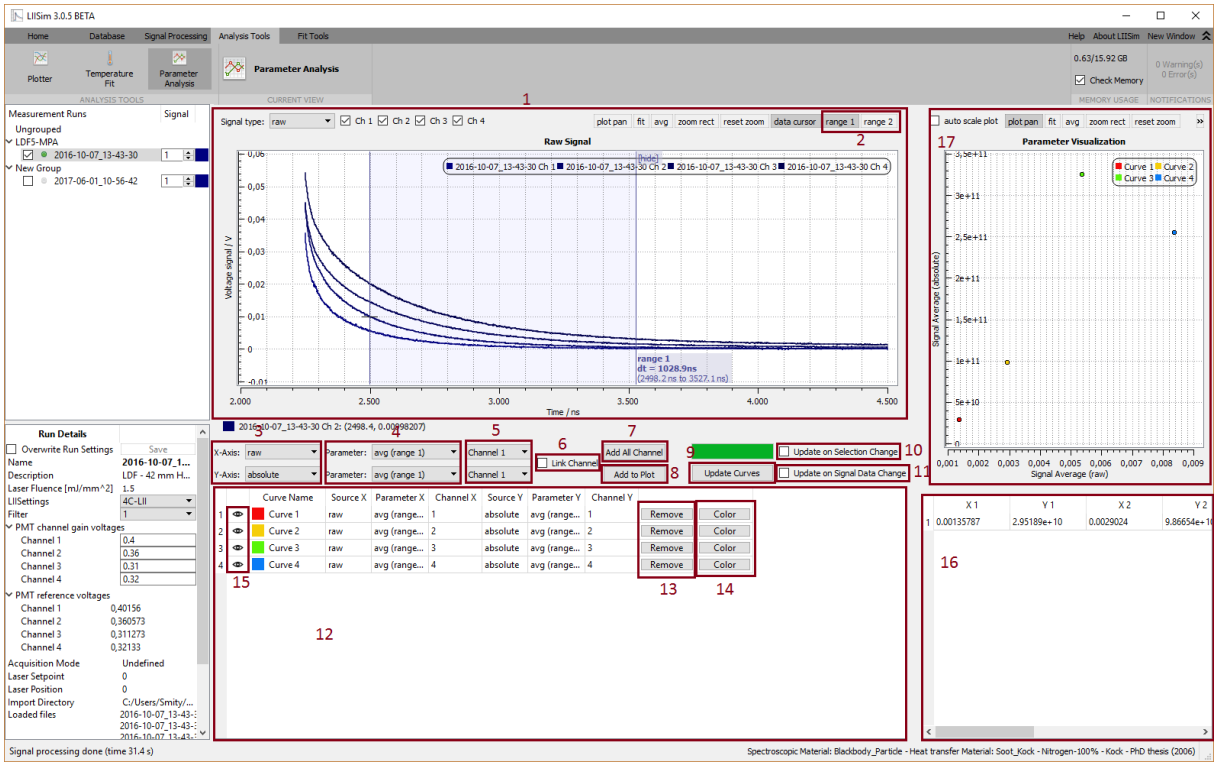
The crosses and lines in **(5)** indicate the iterations of the fitting procedure and the red triangle the initial conditions.

- Click on the map **(5)** to show the spectral fit in **(3)**.
- Click on any iteration in **(4)** to show the corresponding spectral fit in **(3)**



7.6.3 Parameter Analysis

The parameter analysis tool let you oppose different parameters (laser fluence, user-defined parameters, averaged signal range, …) against each other to analyze any correlations between measurement runs.



General:

(1) **Plotter**

See section 7.6.1 for the plotter window.

(16) **Result data table**

Displays the currently shown data point/curves as values. Values can be copied with CTRL-C or right-click and ‘copy’.

(17) **Result graph**

Shows the data points/curves as graphs.

How to use this tool:

- Select, which type of signal you would like to oppose with the dropdown menu at **(3)**.
- Choose the parameter **(4)** you like to compare.
- If needed for the chosen parameter, select a channel **(5)**. If X and Y-axis of the same channel should be compared, channels can be linked **(6).**
- For parameter “avg” or “integral” a range for calculation can be selected **(2).**
- Press **(8)** to add a data point/curve with the selected parameters or press **(7)** to add all possible one-to-one channel combinations.
- **(9)** lets you manually update the curves, useful to prevent unnecessary recalculation.

- **(10)** and **(11)** enable automatic updates when the run selection or the data itself is changed.
- The table at **(12)** shows all added curves. There you can:
    - Toggle the visibility of a curve with **(15)**
    - Change the color of a curve with **(13)**
    - Remove a curve with **(14)**

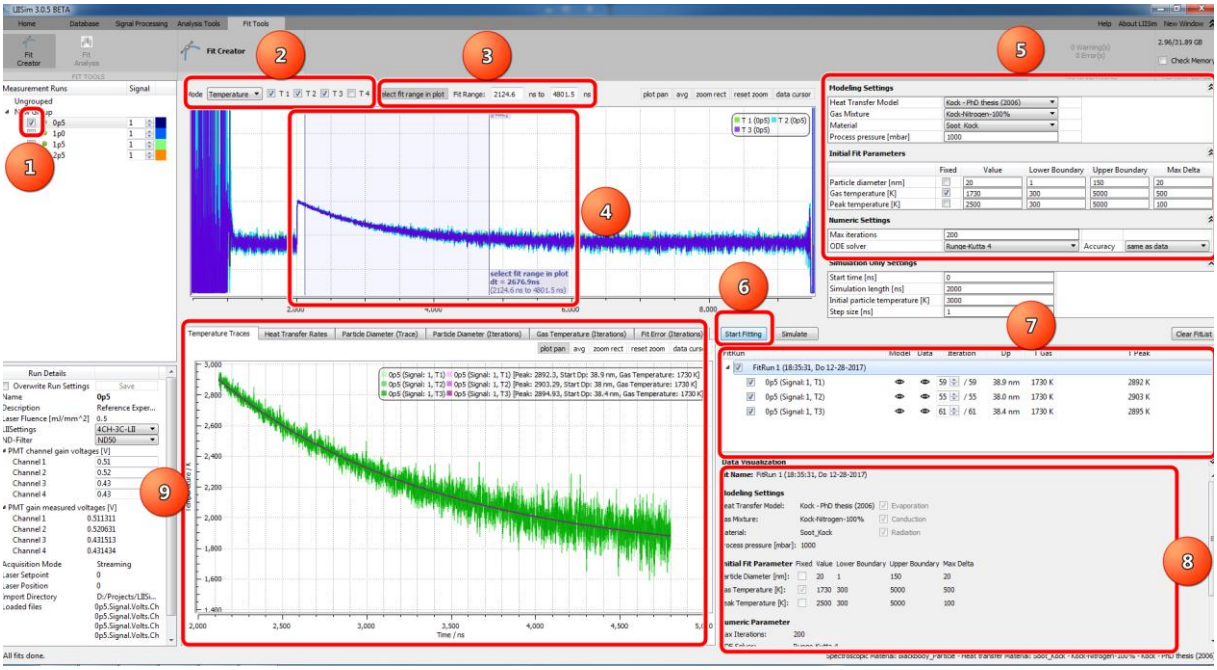
7.7 Fit Tools

The previously calculated temperature traces can be used for comparison with modeled temperature traces using various heat-transfer models and databases.

First, you need to select at least one measurement run **(1)** and the temperature traces to be used for the fitting **(2).**

The temperature traces should now appear in the plot and you can select a range using the button **(3)** and then select the range in the plot **(4)** or you can enter the range manually.

After adjusting the settings **(5)** for the fitting you can press “Start Fitting” **(6)** to perform a fit on all selected measurement runs and temperature traces.

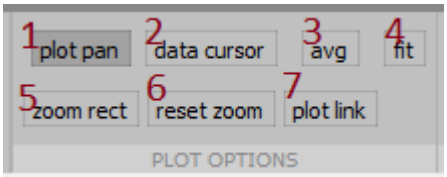


The result from this fitting is called a “FitRun” which is listed in the FitRun-List **(7).** You can select the temperature traces and look at the detailed information in **(8)** or at various plot tools in **(9).**

7.8 General tools

The following tools are available in multiple modules of LIISim.

7.8.1 Plot controls



(1) Plot pan

Select to pan the plot with the left mouse button pressed. The mouse wheel can be used to zoom in and out. If you press CTRL while using the mouse

wheel, you can zoom the x-axis only. If you press SHIFT, you zoom the y-axis only.

(2) **Data cursor**

Enables the selection of a single data point in a displayed curve. The cursor can be move with the left/right button.

(3) **Average Plot Tool**

Enables a range selection and calculates the average over the selected signal range.

(4) **Fit Plot Tool**

Fits a simple exponential function to the curve (experimental!)

(5) **Zoom**

Enables a rectangular zoom.

(6) **Reset zoom**

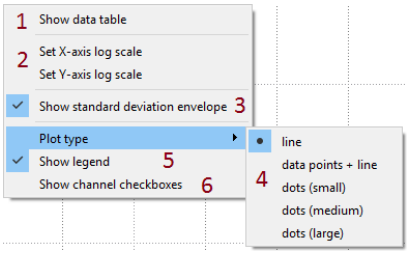
Resets the zoom so all curves are completely visible.

(7) **Plot link**

Links the plots, so any zoom or horizontal pan is applied to all plots.

7.8.2 Plot context menu

Right click on plot opens context menu. The following options are available from there:



(1) **Show data table**

Opens a data table which contains the currently displayed curve data.

(2) **Set X-axis log scale / Set Y-axis log scale**

Enables a logarithmic scaling of the x- and y-axis.

(3) **Show standard deviation envelope**

If standard deviation is available (calculated by 6.1.7 Multi Signal Average) envelope function from the standard deviation data can be toggled.

(4) **Plot type**

Shows different curve representation types and let them be changed.

(5) **Show legend**

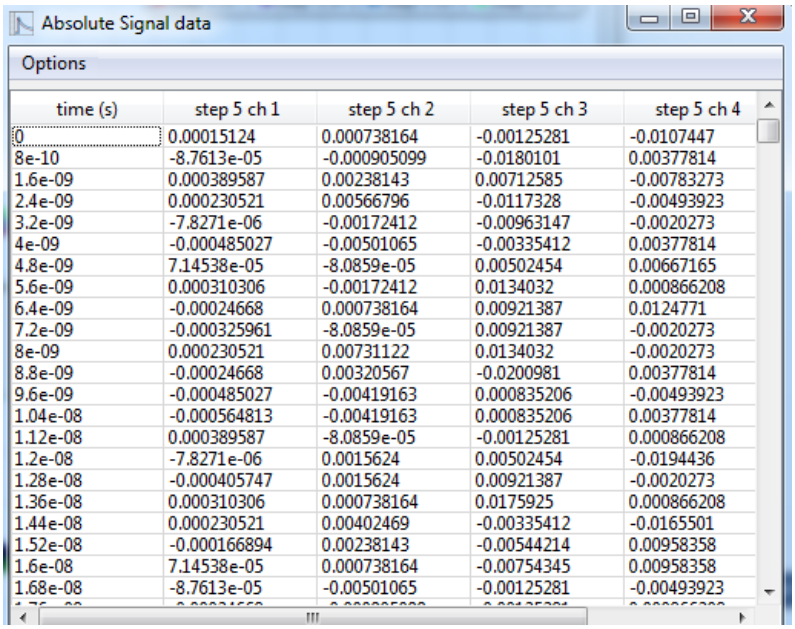
Toggles the visibility of the plot legend.

(6) **Show channel checkboxes**

Displays channel checkboxes below the plot for toggling visibility.

7.8.3 Plot data table

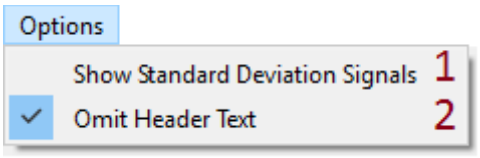
Opens via the plot context menu. Multiple values in the table can be selected with the mouse and copied either with CTRL-C or right-click and 'copy' and pasted into another spreadsheet software.



time (s)	step 5 ch 1	step 5 ch 2	step 5 ch 3	step 5 ch 4
0	0.00015124	0.000738164	-0.00125281	-0.0107447
8e-10	-8.7613e-05	-0.000905099	-0.0180101	0.00377814
1.6e-09	0.000389587	0.00238143	0.00712585	-0.00783273
2.4e-09	0.000230521	0.00566796	-0.0117328	-0.00493923
3.2e-09	-7.8271e-06	-0.00172412	-0.00963147	-0.0020273
4e-09	-0.000485027	-0.00501065	-0.00335412	0.00377814
4.8e-09	7.14538e-05	-8.0859e-05	0.00502454	0.00667165
5.6e-09	0.000310306	-0.00172412	0.0134032	0.000866208
6.4e-09	-0.00024668	0.000738164	0.00921387	0.0124771
7.2e-09	-0.000325961	-8.0859e-05	0.00921387	-0.0020273
8e-09	0.000230521	0.00731122	0.0134032	-0.0020273
8.8e-09	-0.00024668	0.00320567	-0.0200981	0.00377814
9.6e-09	-0.000485027	-0.00419163	0.000835206	-0.00493923
1.04e-08	-0.000564813	-0.00419163	0.000835206	0.00377814
1.12e-08	0.000389587	-8.0859e-05	-0.00125281	0.000866208
1.2e-08	-7.8271e-06	0.0015624	0.00502454	-0.0194436
1.28e-08	-0.000405747	0.0015624	0.00921387	-0.0020273
1.36e-08	0.000310306	0.000738164	0.0175925	0.000866208
1.44e-08	0.000230521	0.00402469	-0.00335412	-0.0165501
1.52e-08	-0.000166894	0.00238143	-0.00544214	0.00958358
1.6e-08	7.14538e-05	0.000738164	-0.00754345	0.00958358
1.68e-08	-8.7613e-05	-0.00501065	-0.00125281	-0.00493923

Figure 7-4 Plot data table

Plot data table options



(1) **Show Standard Deviation Signals**

Toggles the standard deviation signal data in the table

(2) **Omit Header Text**

If enabled, omits the header label text from the copied values

8 Troubleshooting

8.1 MSVCR120.dll was not found

If you see an error like “MSVCR120.dll was not found” please try to download and install the Visual C++ Redistributable Packages from the Microsoft website. It should fix the issue: https://www.microsoft.com/en-us/download/details.aspx?id=40784

9 Frequently Asked Questions

[empty]