SasView : A Small Angle Scattering Analysis Software Package

Andrew Jackson, European Spallation Source
on behalf of the SasView Collaboration

SINE 2020 WP 10 Meeting, Grenoble, 24th – 25th April 2017

Analysis Software - Who's Job is it Anyway?

Analysis is where the science is → the USER'S JOB

Scattering is an analysis tool and part of providing the tool should be the necessary software → the FACILITY'S JOB

Data sat on disk is useless to EVERYBODY

We need to work together!

Analysis Software - Who's Job is it Anyway?

But … where are the resources?

More pressing tasks for all of us ...

Facility scientists & staff : maintaining & improving instrumentation, bringing in and supporting users, dealing with instrument control and data reduction software …

Users : data analysis should be high priority ... But in fact the priority is publication. Users want to get rapidly from data to answer – if it is going to need major investment of time then low priority. Especially for "one-time" and industrial users.

Writing shareable data analysis software is bottom of the heap ...

… need to pool resources.

A little history …

Where did SasView come from?

DANSE project output
~ 8.5% of funds were for SANS
Kickoff meeting August 2006

Heritage: NIST IGOR macros

Continuity …

NIST Supported initial transition from NSF funding

Expansion ...

NIST supported initial transition from NSF funding

Now 8 active facilities
ORNL, ISIS, NIST, ESS, ILL, TUD/RID, ANSTO and Diamond

SINE2020 funding two staff at ESS
First major investment since DANSE
http://sine2020.eu

Development Model

Open, Collaborative, Community Development

Code is open source and publicly hosted at Github
Released under BSD 3-clause license

Bug and Enhancement Ticket System

Bi-weekly developer calls

Code Camps

1st at NIST April 2013
2nd at ISIS April 2014
3rd at ESS Feb 2015
4th at TU Delft/RID March 2016
5th at ORNL October 2016
6th at ILL/ESRF April 2017

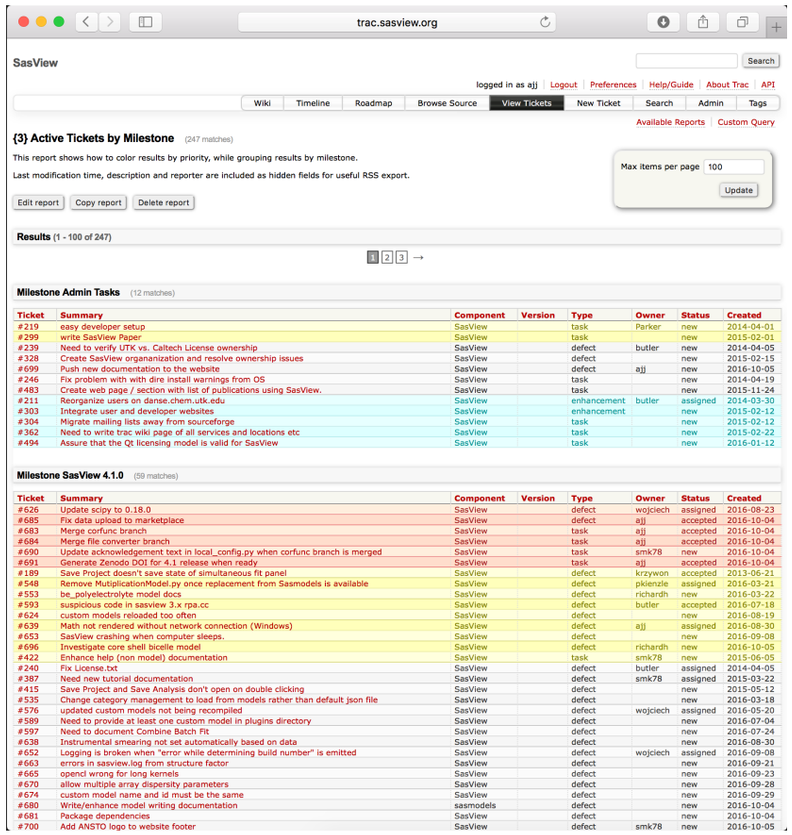
5 Year Roadmap

http://www.sasview.org http://github.com/SasView

Development Model

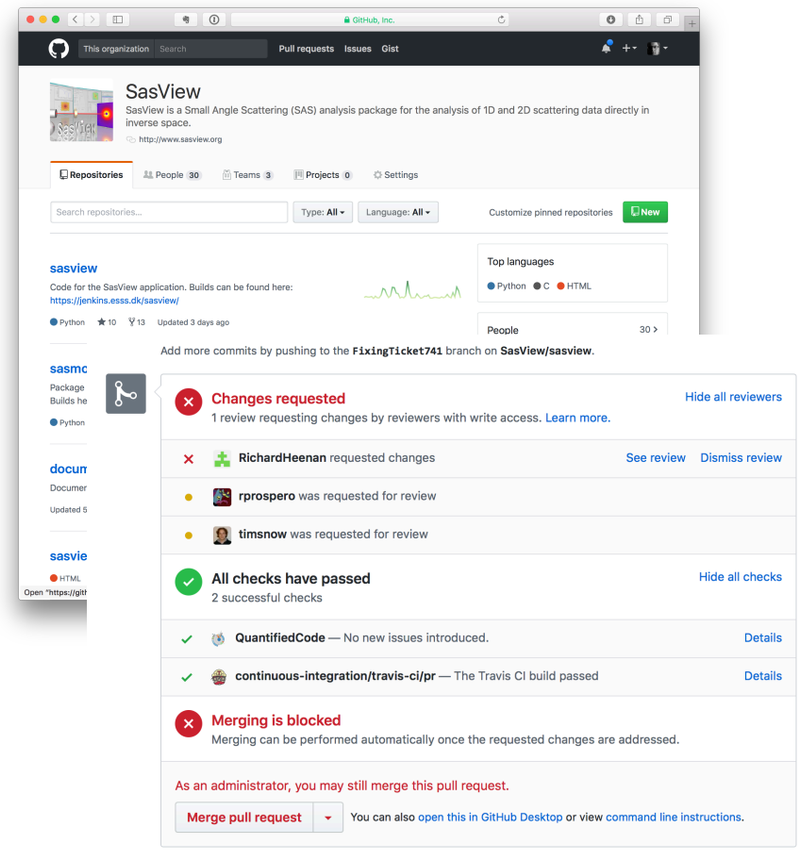
www.sasview.org

Task and bug tracking. Developer wiki trac.sasview.org



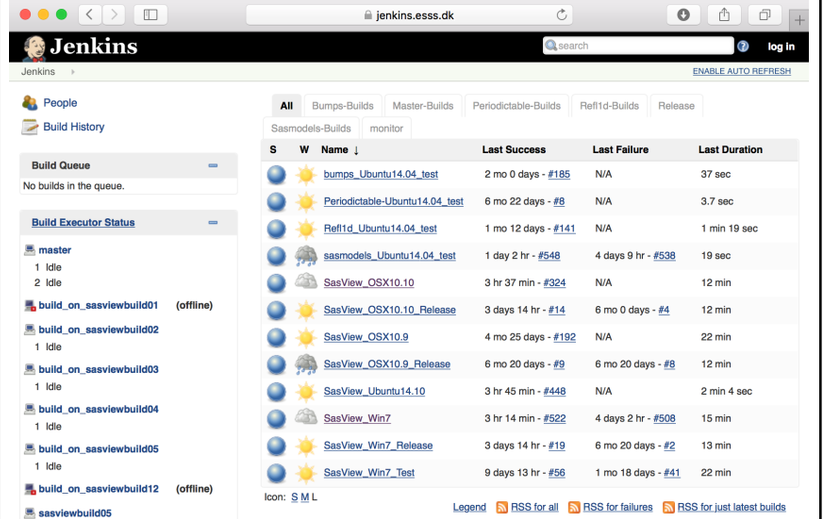
Hosted by UT

Code Hosting and some CI/testing of PRs



Hosted by GitHub etc ...

Automated Builds build.sasview.org



Hosted by ESS

Development Model

Communications

Mailing lists

`management@sasview.org`

`help@sasview.org`

`developers@sasview.org`

`users@sasview.org`

Current Development Team

"Management" Team

- Paul Butler (NIST)
- Mathieu Doucet (ORNL)
- Andrew Jackson (ESS)
- Steve King (STFC)

Code Camp VI hosted by ESRF & ILL – April 4th – 11th 2017

- Jurrian Bakker (TUD)
- Wim Bouwman (TUD)
- Miguel Gonzales (ILL)
- Richard Heenan (STFC)
- Dirk Honecker (ILL)
- Paul Kienzle (NIST)
- Jeff Kryzwon (NIST)
- Ricardo Leal (ORNL)
- David Mannicke (ANSTO)
- Andrew Nelson (ANSTO)
- Torben Nielsen (ESS)
- Lewis O'Driscoll (STFC)
- Steve Parnell (TUD)
- Wojciech Potrzebowski (ESS)
- Piotr Rozyczko (ESS)
- Tim Snow (Diamond)
- Adam Washington (STFC)

- and thanks to the many previous contributors, particularly Jae Hie Cho and Alina Gervaise

What Can SasView Do?

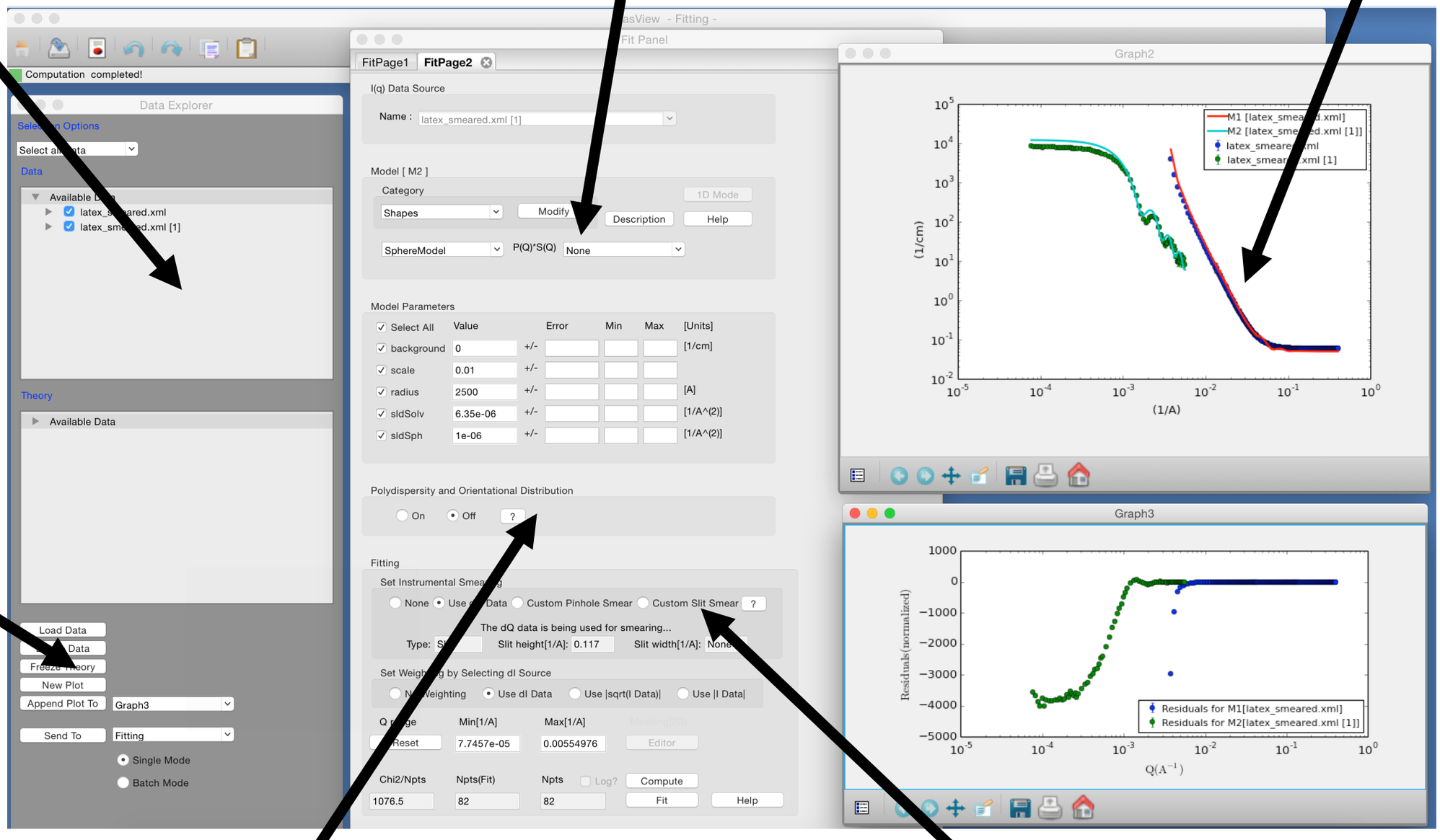
1D Analysis

Data management

Common data formats supported, including NXCansas & *cansas1D*

Wide choice of built-in models (> 70)
P(Q), S(Q) & P(Q)*S(Q)

Simultaneous fitting



Analysis Tool Choice & Plotting

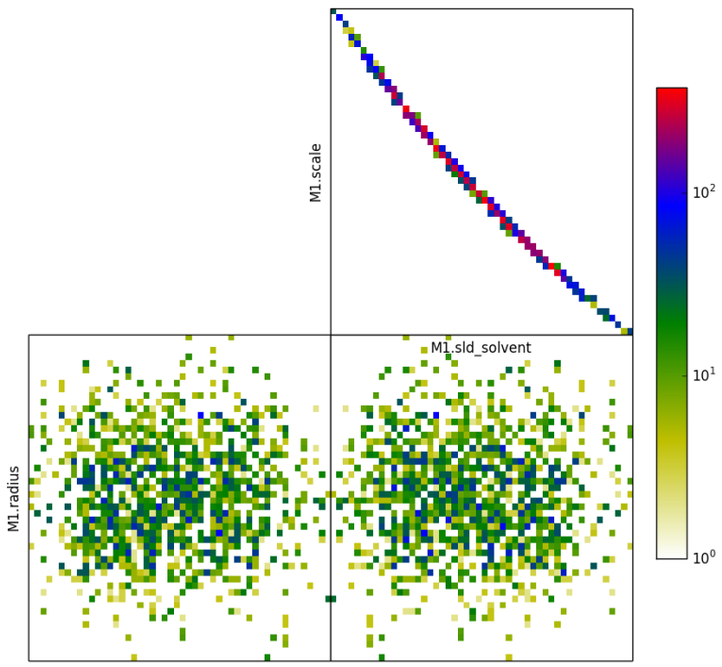
Generic parameter polydispersity

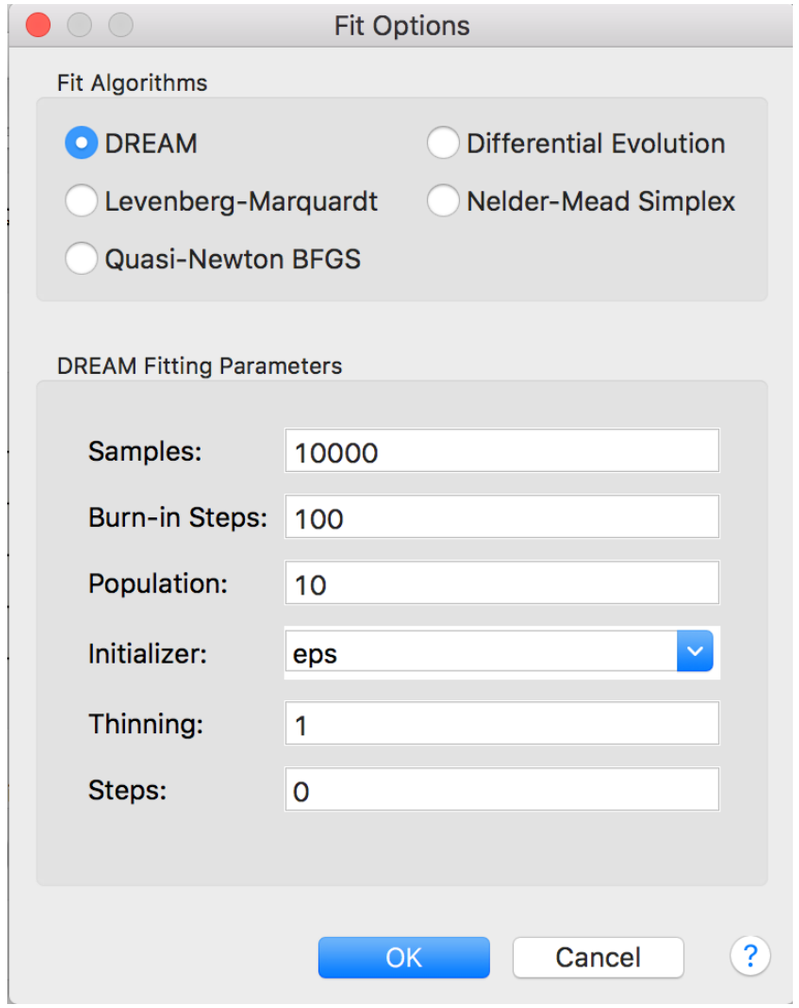
Choice of distribution and distribution parameters

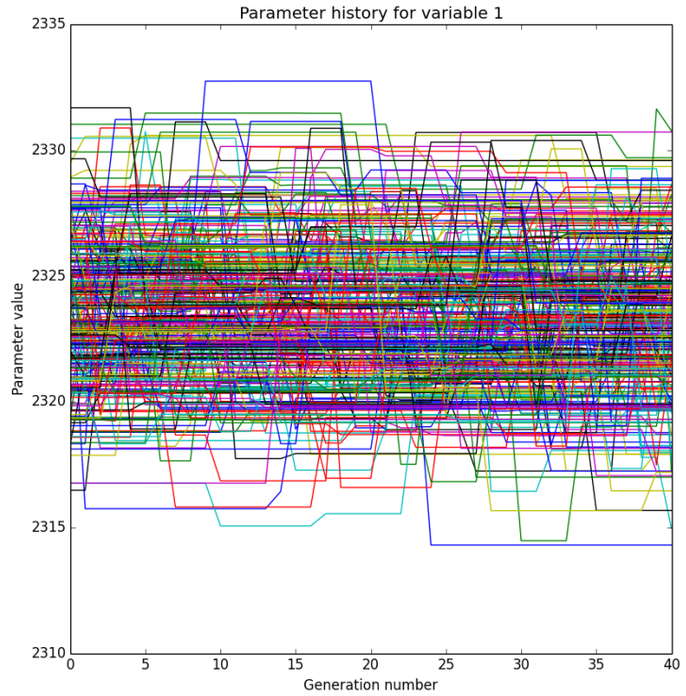
Resolution smearing (pinhole and slit)

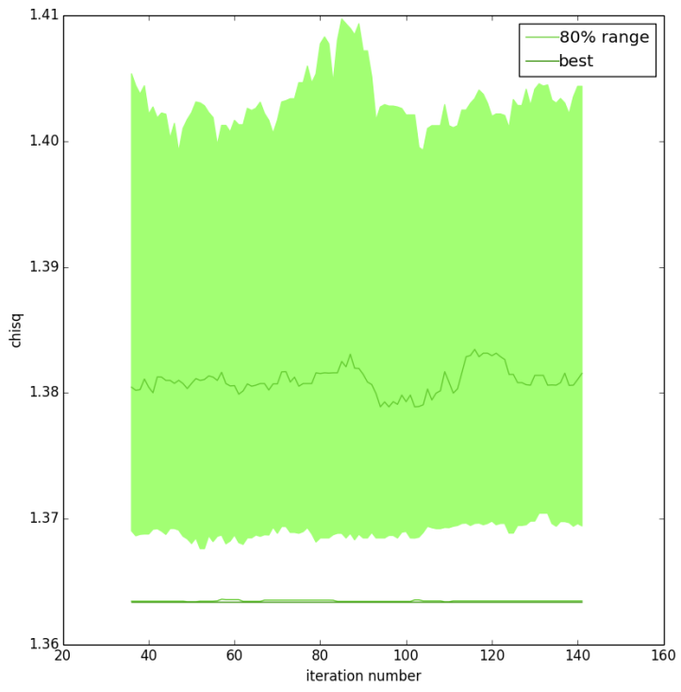
Automatically from data or provide parameters

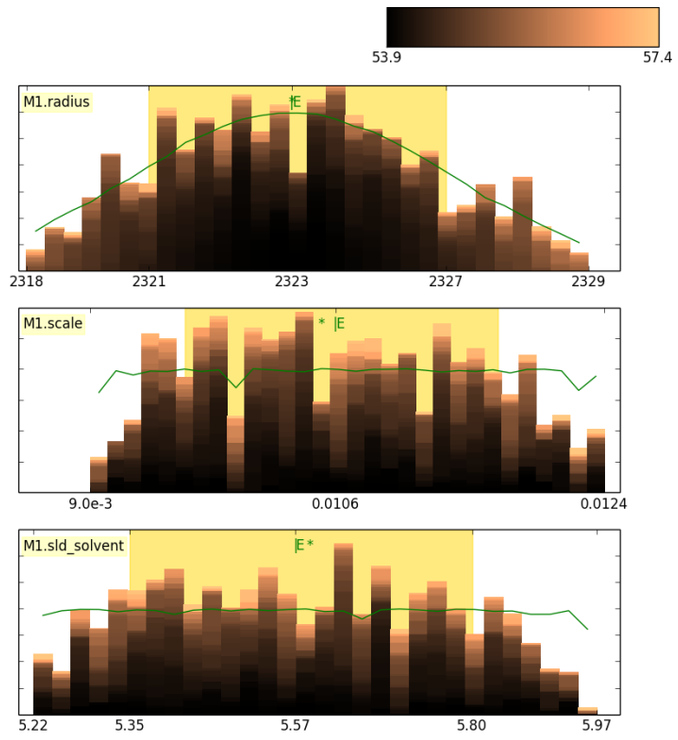
Fitting Algorithms





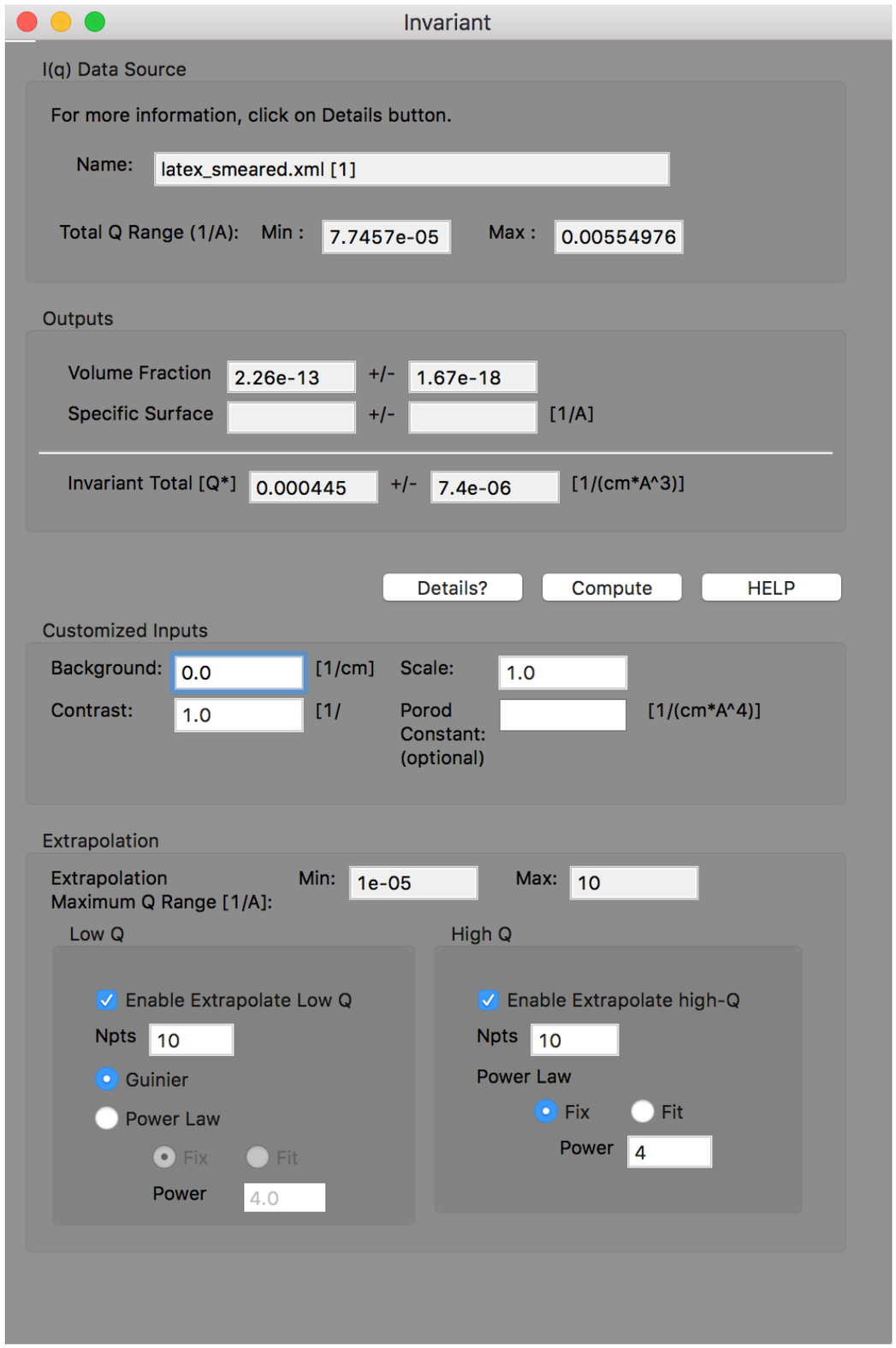


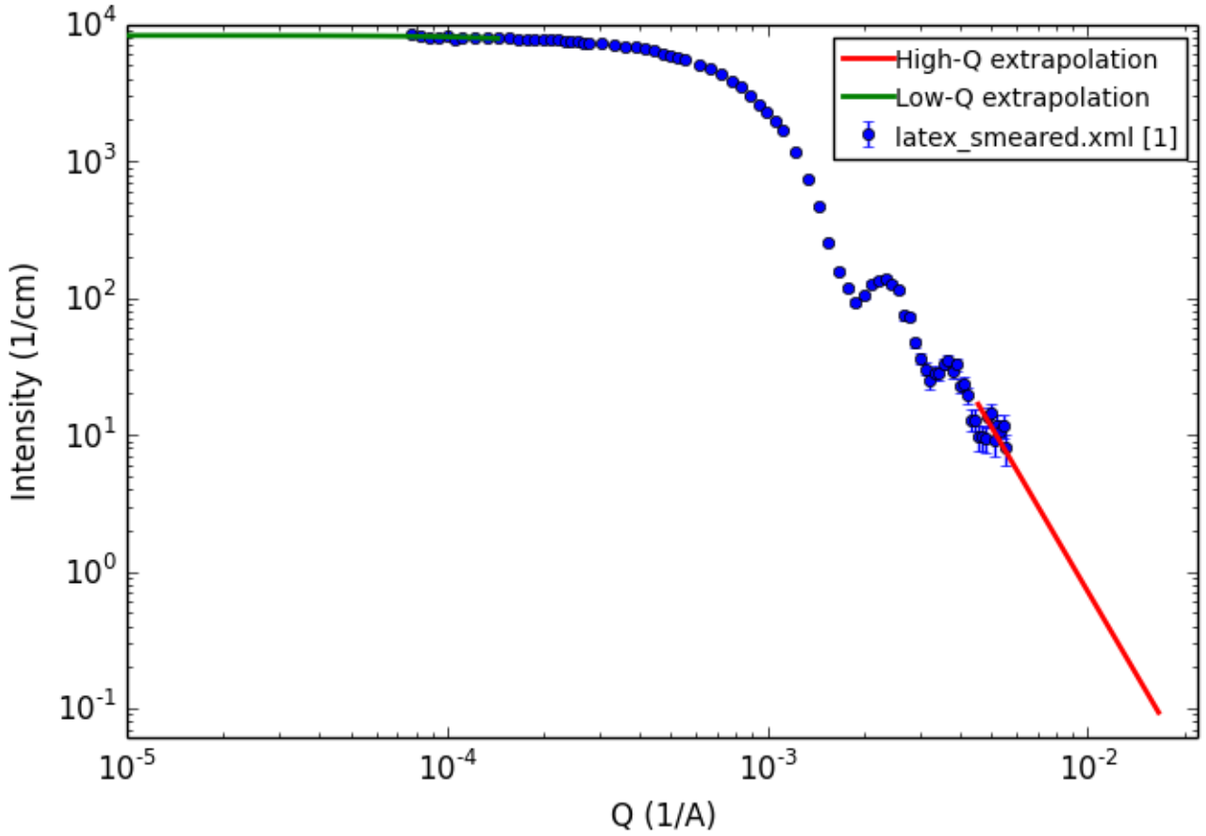


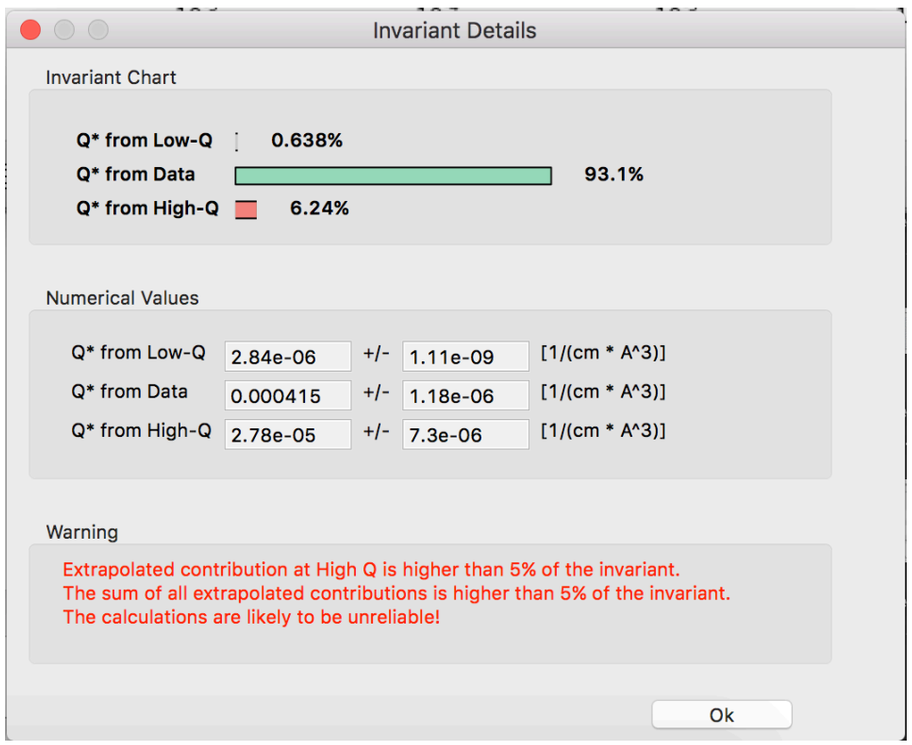


Uses bumps package from P. Kienzle

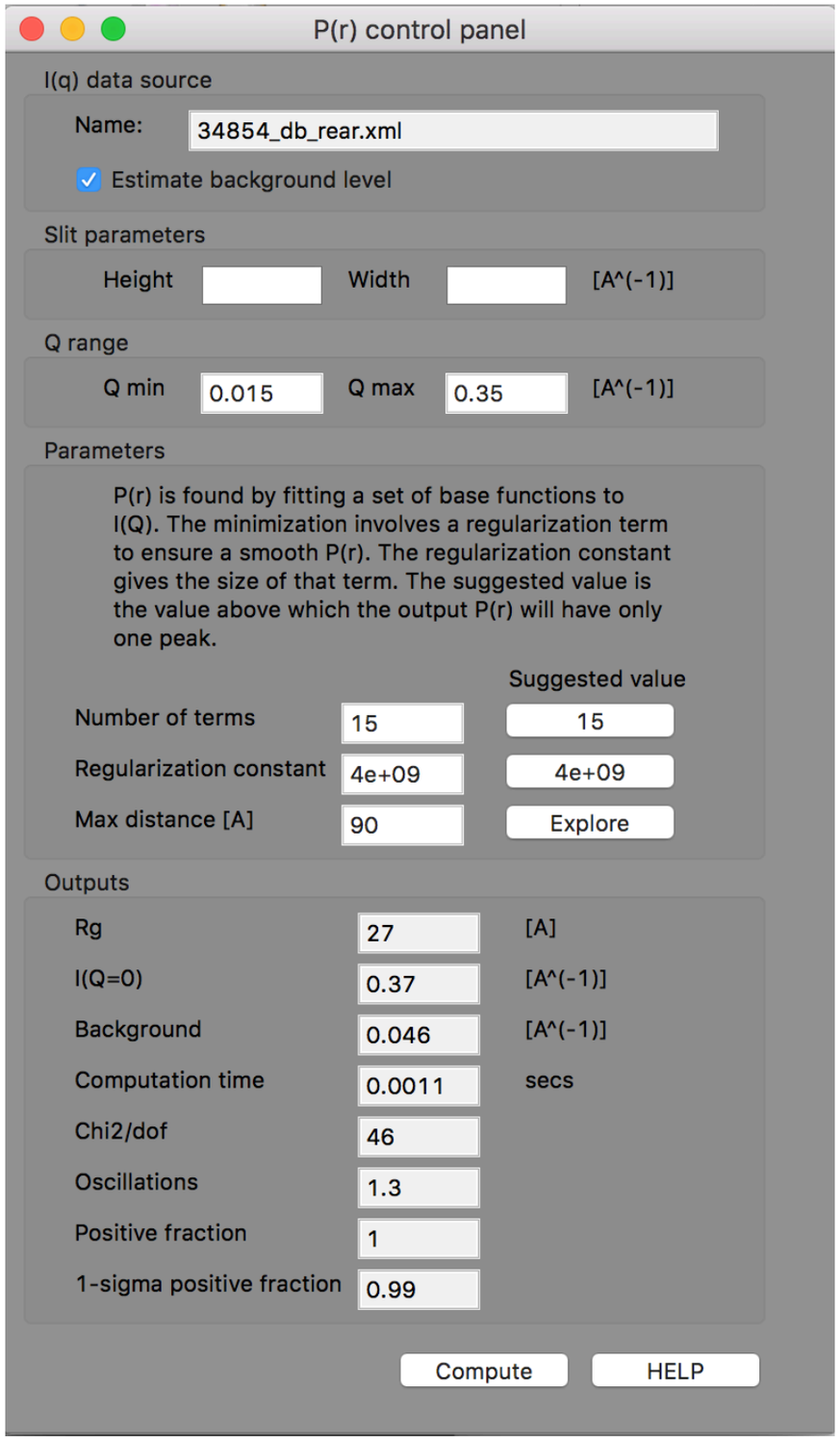
Invariant Calculation

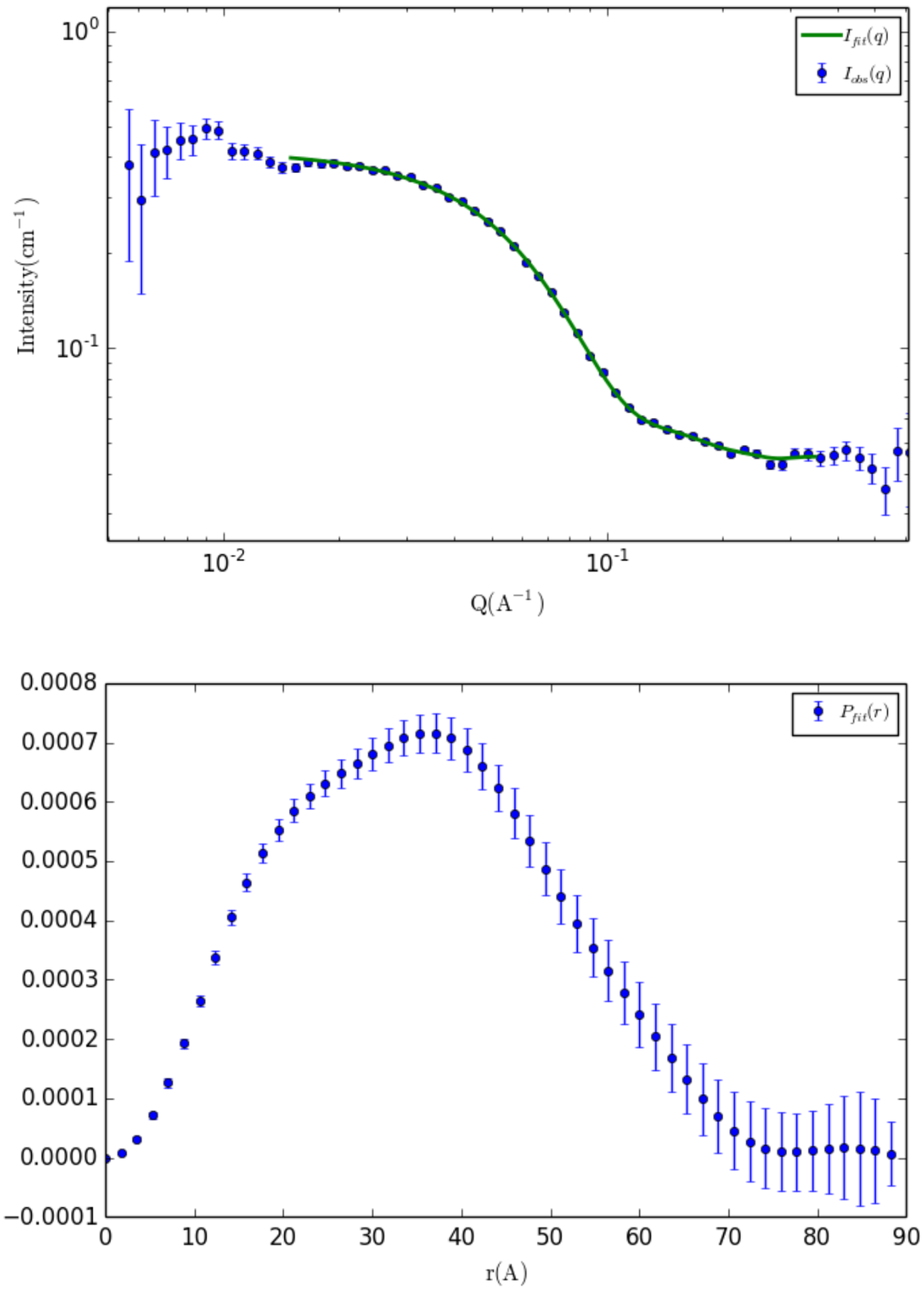




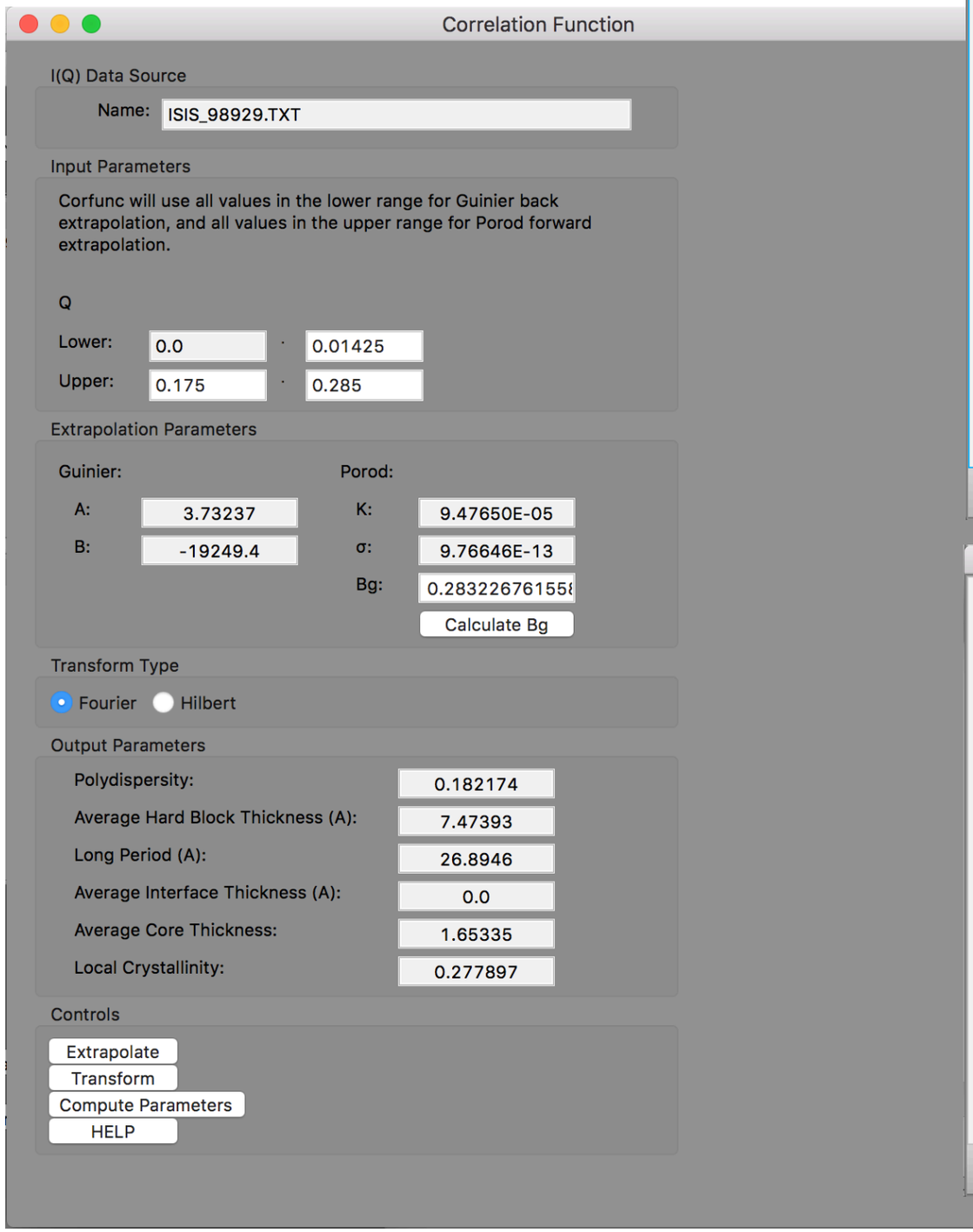


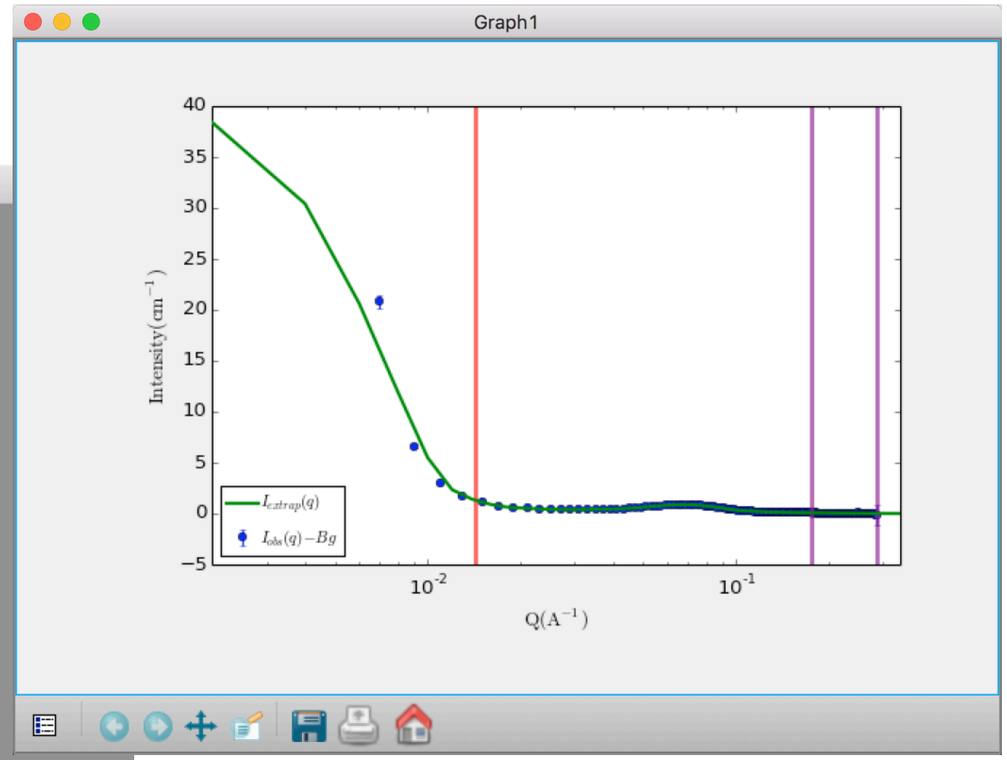
P(r) Inversion

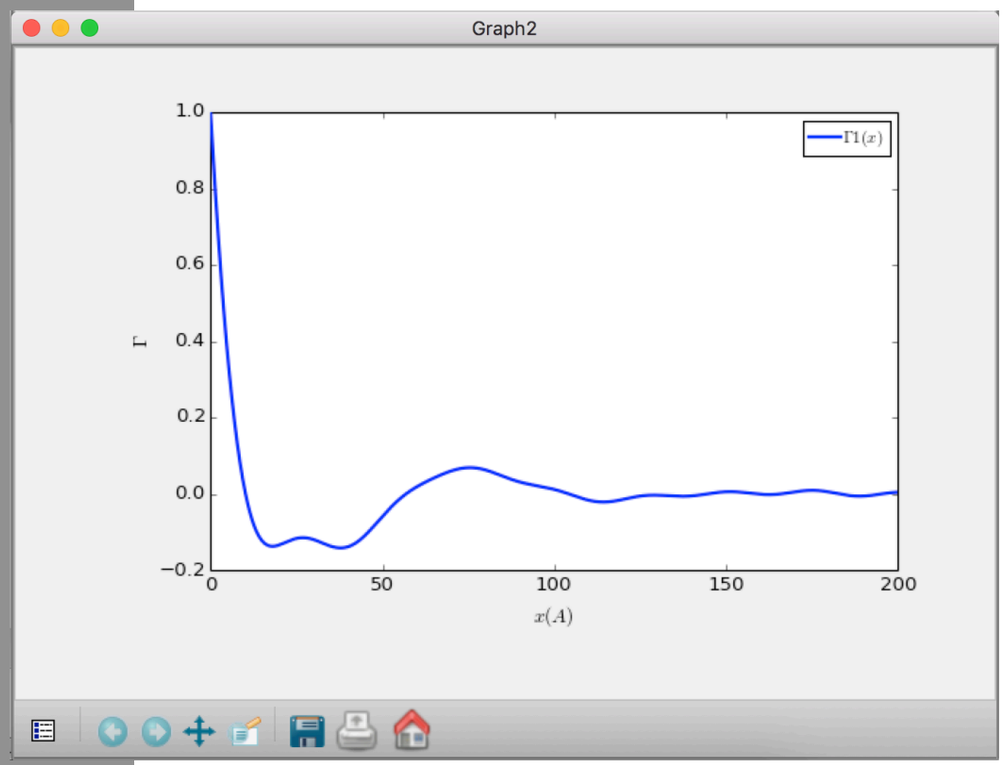




Correlation Function Analysis (new!)







SESANS Analysis

Fit panel - Active Fitting Optimizer: Levenberg-Marquardt

FitPage1

I(q) Data Source

Name : sphere2micron.ses

Model [M1]

Category

Sphere | Modify | Description | 1D Mode | Help

sphere | P(Q)*S(Q) | None

Model Parameters

Parameter	Value		Error	Min	Max	[Units]
☑ scale	0.005323	+/-	5.2808e-0	0	inf	
☐ background	0	+/-		-inf	inf	1/cm
☐ sld	1	+/-		-inf	inf	1e-6/Ang^2
☐ sld_solvent	6	+/-		-inf	inf	1e-6/Ang^2
☑ radius	10383	+/-	114.45	0	inf	Ang

Polydispersity and Orientational Distribution

On | Off | ?

Fitting

Set Instrumental Smearing

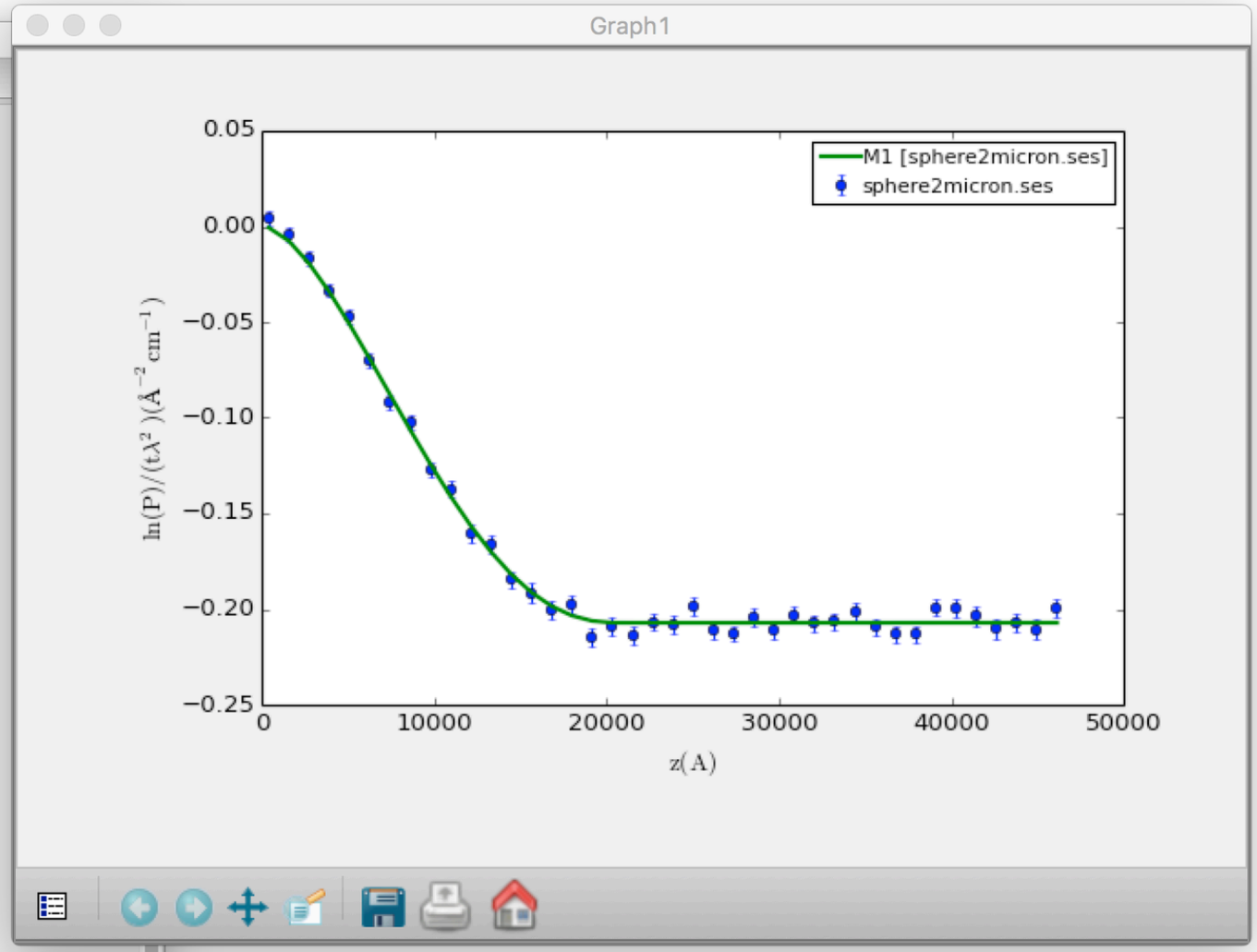
None | Use dQ Data | Custom Pinhole Smear | Custom Slit Smear | ?

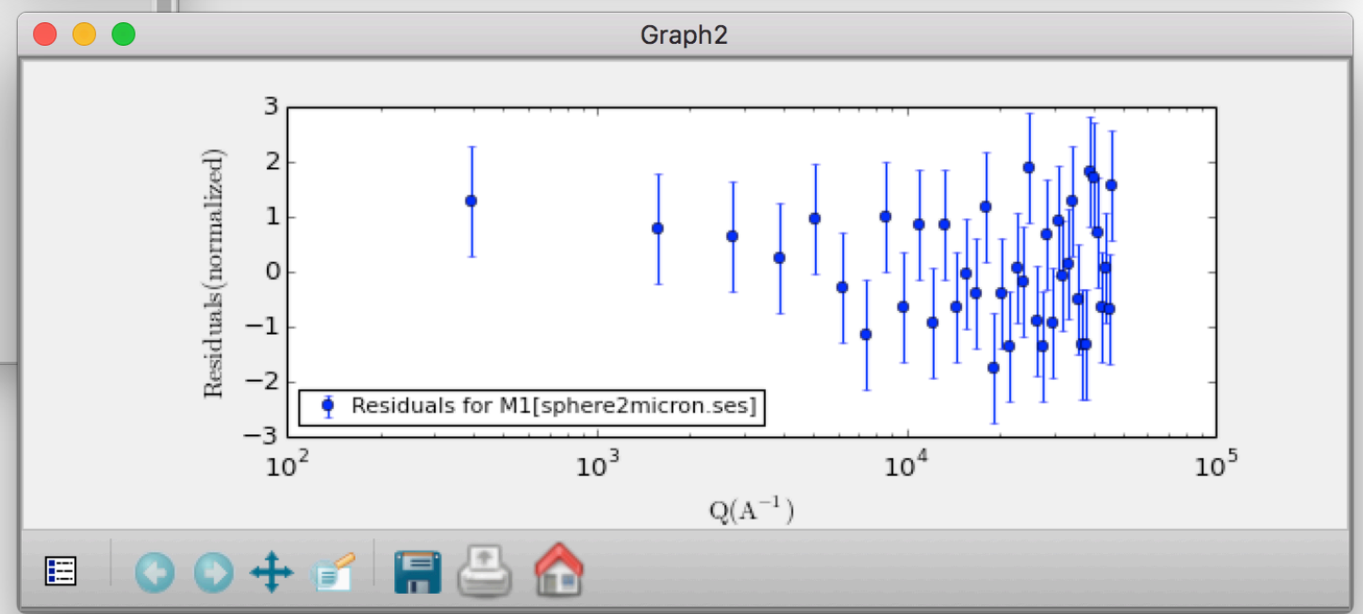
The dQ data is being used for smearing...

Type: Pinhole | 19.578 | dQ[%]: 2304.9

Set Weighting by Selecting dI Source

No Weighting | Use dI Data | Use |sqrt(I Data)| | Use |I Data|

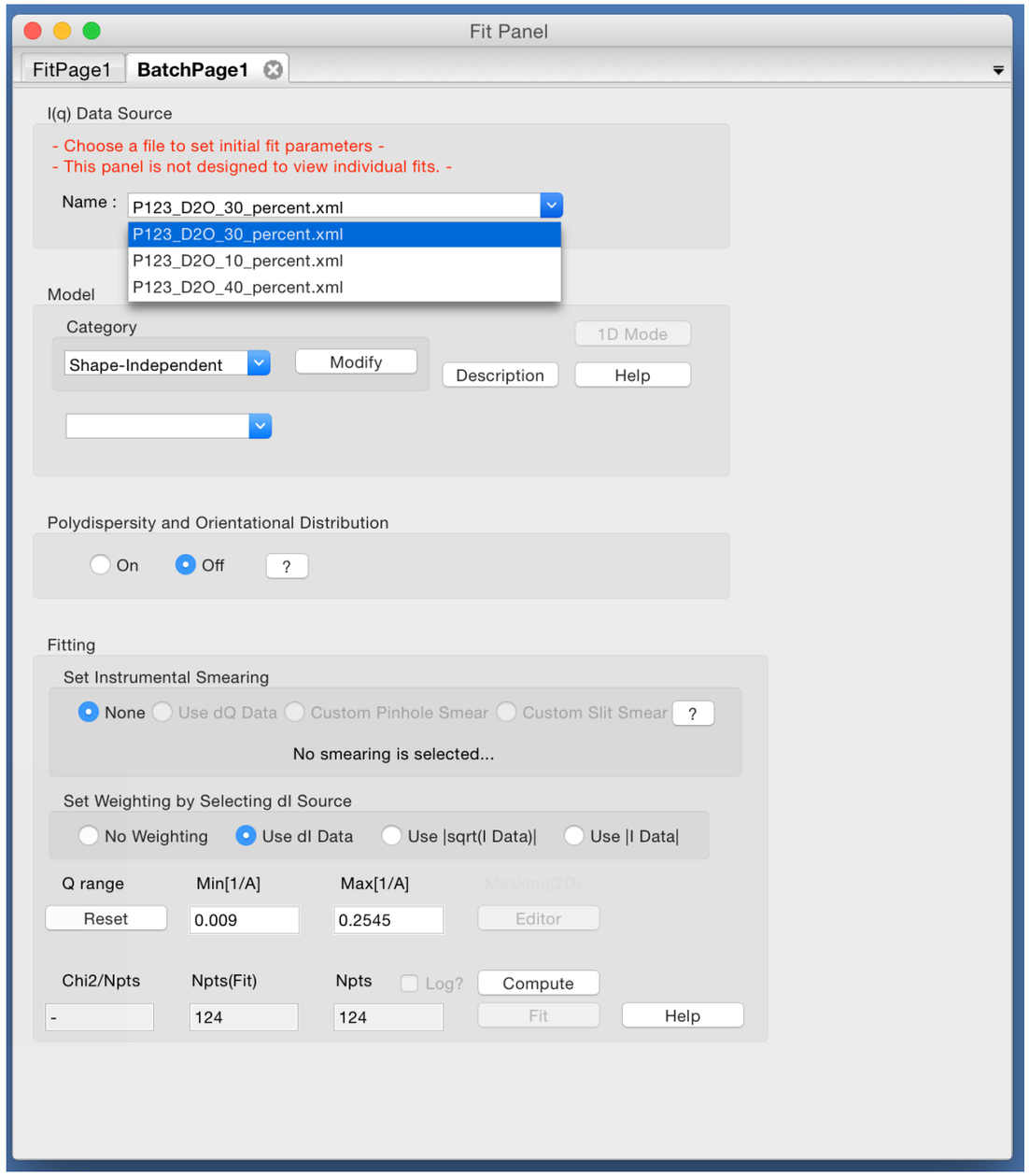




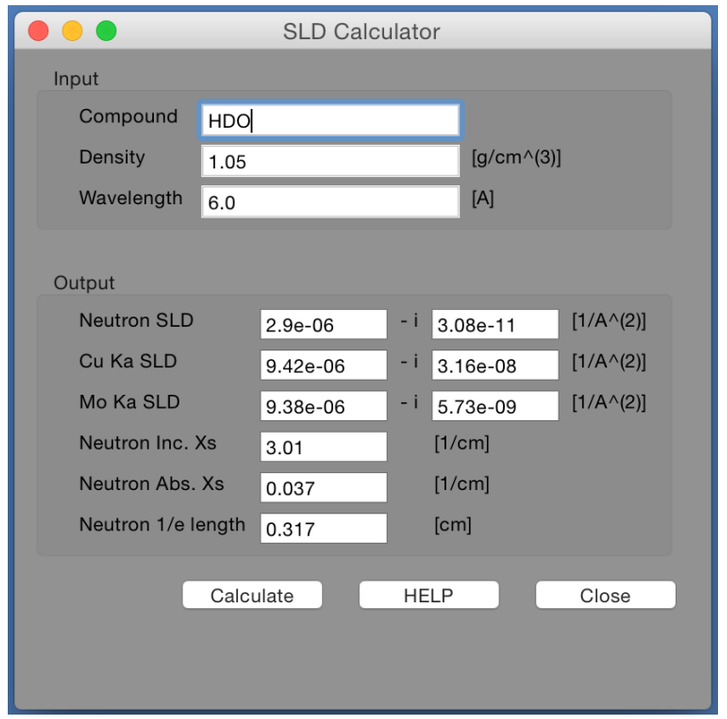
Automatic Hankel Transform of SANS models

Other Features

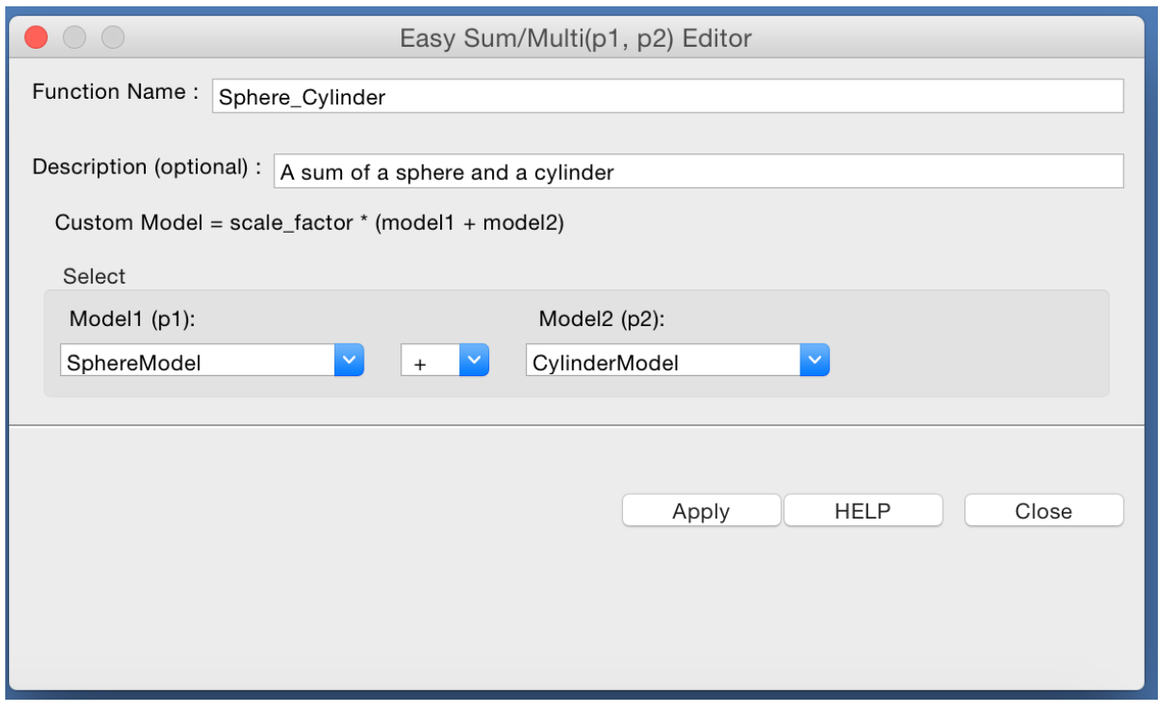
Batch fitting …



SLD Calculator …



Sum & Multiplication of Models …



2D Modelling

Oriented systems & Magnetic Scattering

Cite this: *Soft Matter*, 2011, **7**, 9992

www.rsc.org/softmatter

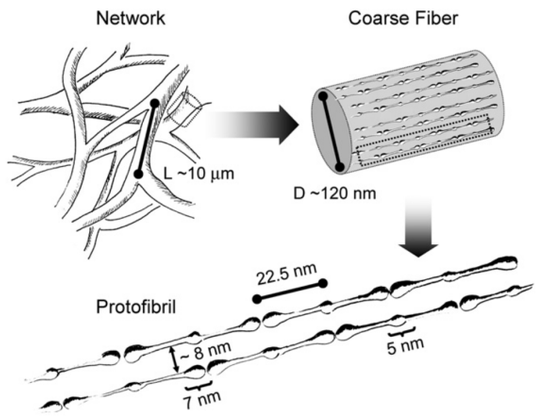
PAPER

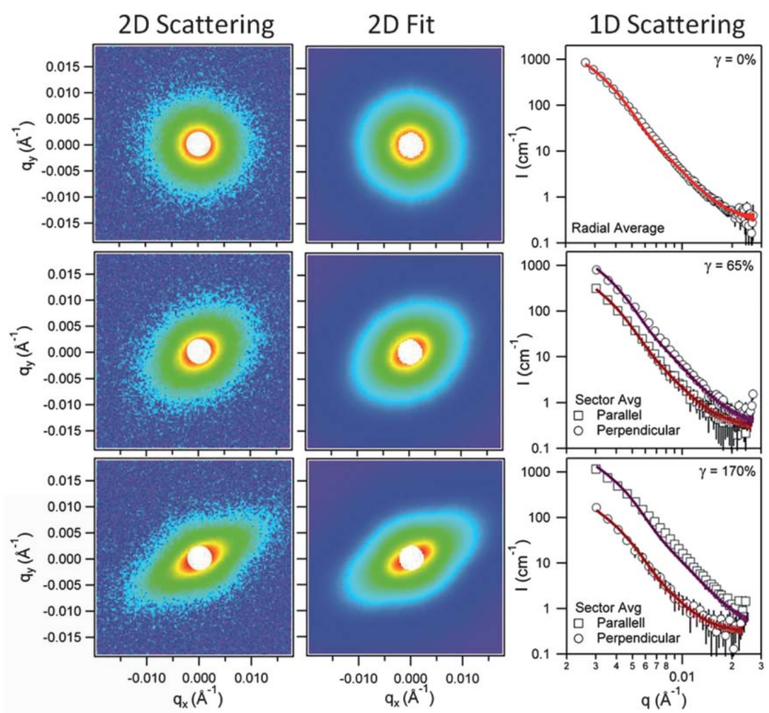
In situ neutron scattering study of structural transitions in fibrin networks under shear deformation

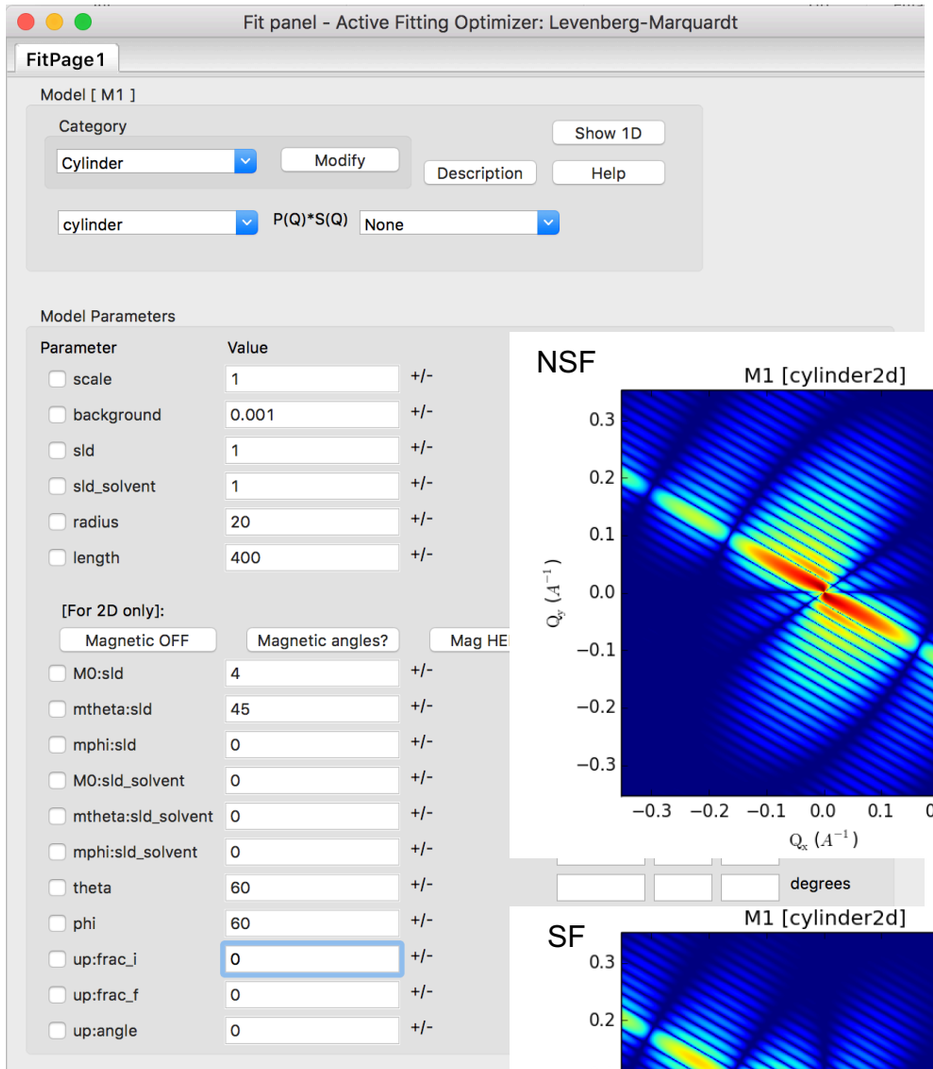
Katie M. Weigandt,[a] Lionel Porcar[bc] and Danilo C. Pozzo*[a]

Received 23rd June 2011, Accepted 5th August 2011
DOI: 10.1039/c1sm06176c

Small angle neutron scattering (SANS) is used to decipher th
biopolymer networks by directly measuring the structural res
deformation. A special Couette shear cell is used to systemati
a fibrin clot over strain values in the range of $\gamma = 1$–170%. T
hardening response of coarse fibrin gels occurs in two distinc
mechanical signatures that are separated by an intermediate s
($\gamma < 10\%$) there is a measurable increase in the shear modulus
there are no significant changes to the clot structure. At higher
hardening regime is directly correlated to significant fiber alig
determined directly from two-dimensional fits to the anisotro
monotonically in the high-strain regime. The results suggest tl
of fibrin clots are the result of a reduction of lateral entropic flu
between bending and stretching at higher strain values.







Creating Models

Old Way

- Write a model in python and drop it in plugin folder
  - Easy but no polydispersity available
- Write a model in C and incorporate into SasView
  - Difficult and need to recompile whole programme

New Way

- Models distributed in separate package (sasmodels)
  - separation of models from GUI
  - simpler addition of models by users
  - speed! GPU and parallel processing when using C
- All models work the same
- Write in python and/or C and drop in plugin folder

Creating Models

Pure Python

```
r"""
This model calculates a simple power law with a flat background.

Definition
----------

.. math::

    I(q) = \text{scale} \cdot q^{-\text{power}} + \text{background}

Note the minus sign in front of the exponent. The exponent *power*
should therefore be entered as a **positive** number for fitting.

Also note that unlike many other models, *scale* in this model
is NOT explicitly related to a volume fraction. Be careful if
combining this model with other models.


References
----------

None.
"""

from numpy import inf, errstate

name = "power_law"
title = "Simple power law with a flat background"

description = """
    Evaluates the function
    I(q) = scale * q^(-power) + background
    NB: enter power as a positive number!
    """
category = "shape-independent"
```

- Documentation
- Description
- Parameters
- Calculation
- Tests

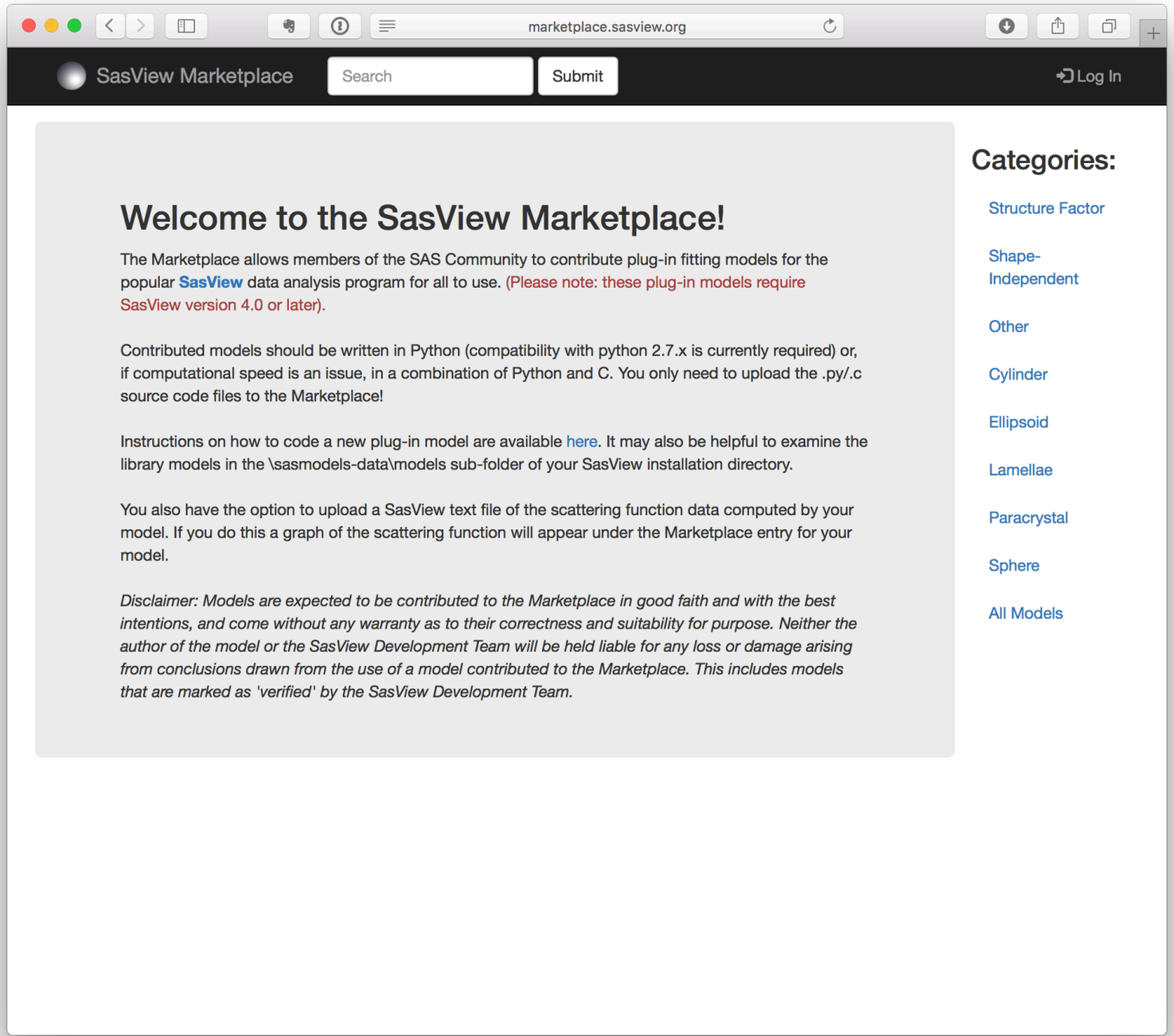
```
#             ["name", "units", default, [lower, upper], "type", "description"]
parameters = [["power", "", 4.0, [-inf, inf], "", "Power law exponent"]]

# NB: Scale and Background are implicit parameters on every model
def Iq(q, power):
    # pylint: disable=missing-docstring
    with errstate(divide='ignore'):
        result = q**-power
    return result
Iq.vectorized = True  # Iq accepts an array of q values

demo = dict(scale=1.0, power=4.0, background=0.0)

tests = [
    [{'scale': 1.0, 'power': 4.0, 'background': 0.0},
     [0.0106939, 0.469418], [7.64644e+07, 20.5949]],
    ]
```

SasView Marketplace

Users can share their models with the community

Documentation & example output

Models can be tested and included in future releases of sasmodels & SasView



http://marketplace.sasview.org

SasView 4.1 is Out!

www.sasview.org

Official Releases available for Windows and Mac

Models

New models

New model package (sasmodels)

Separation of models from GUI

Simpler addition of models by users

Speed! GPU and parallel processing

Correlation Function Analysis

CCP13 corfunc algorithm

Documentation

Enhanced, updated documentation for models

SESANS

Automatic transform of SANS model to P(z)

Plotting and fitting of SESANS data from GUI

Example scripts for fitting SESANS data

Simultaneous fitting of SANS & SESANS

Future ...

UI Refactoring ("SasView 5.0")

Move to QT - current and well supported toolkit

Complete separation of GUI and calculation code

Provision of CLI & updated Python API

Sasmodels Enhancements

Return F(q) from models

Beta approximation

Coherent sums

Constraints refactor

Multi-GPU support

Interface with SasFit models

Documentation

Tutorials

Manual

And much more!

See Roadmap and Tickets

Questions?