

SasView

SINE2020

PSI meeting April 4th and 5th 2016

Wojtek Potrzebowski

Piotr Rozyczko

Torben Nielsen

Data Analysis and Modelling group

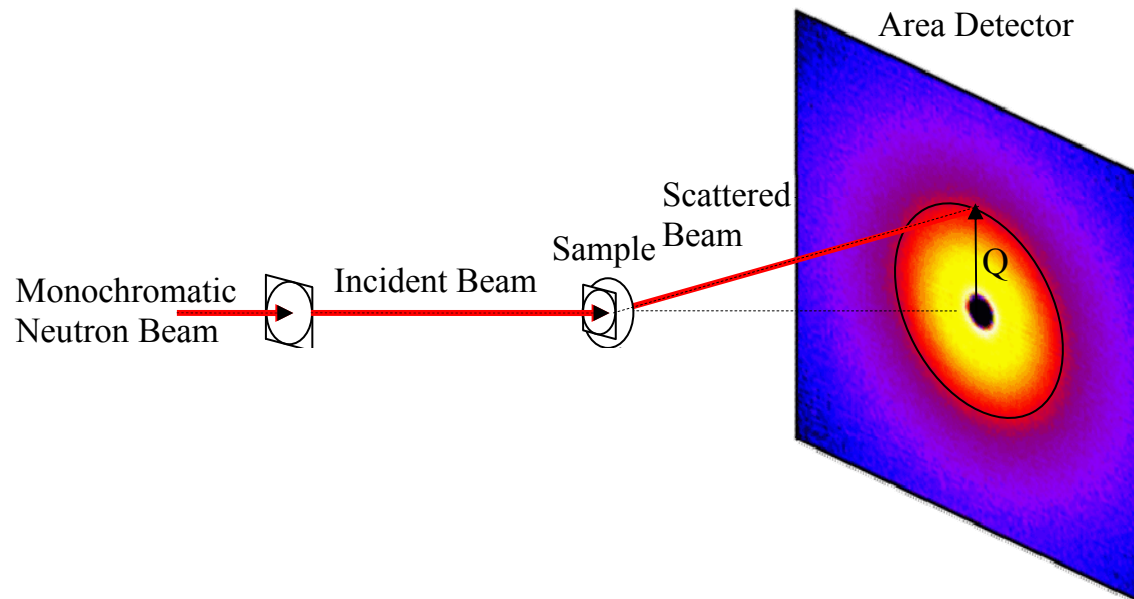
www.europeanspallationsource.se

22 June 2017

- What is Small Angle Neutron Scattering (SANS)?
- What is SasView and why do we use it?
- SasView workflows and organization
- SINE2020 tasks and SasView roadmap
- Live Neutron Data Analysis (LINDA)

Small Angle Neutron Scattering

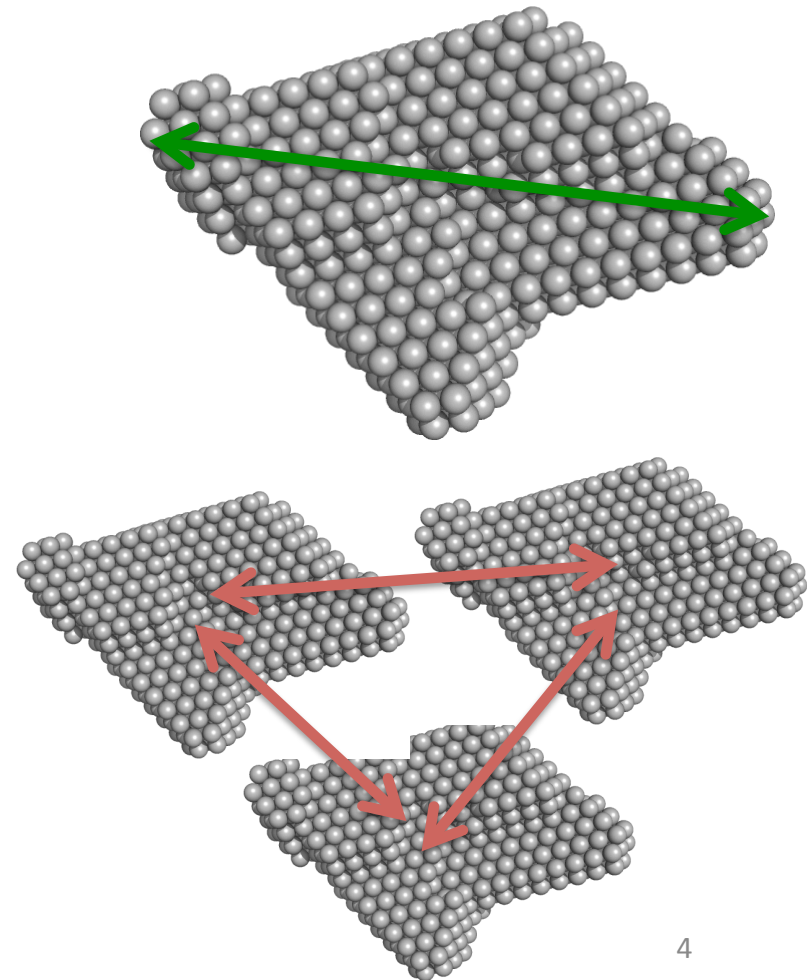
- Small-angle neutron scattering (SANS) uses **elastic neutron scattering** at small scattering angles to investigate the structure at a mesoscopic scale of about **1 - 100 nm**.



What can SANS tell me?

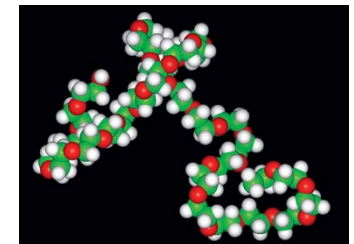
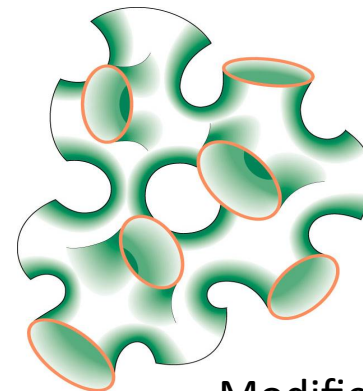
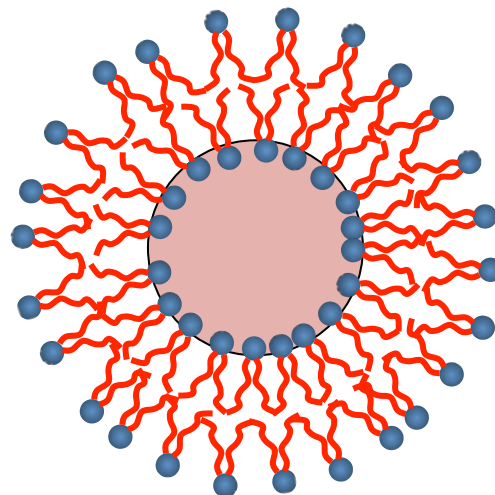
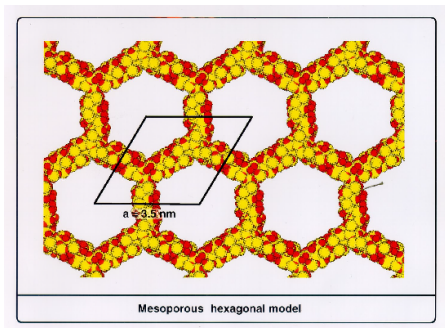
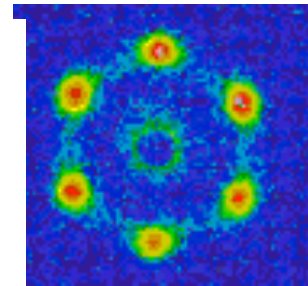
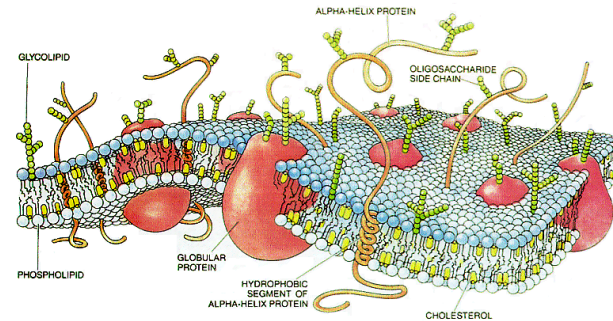
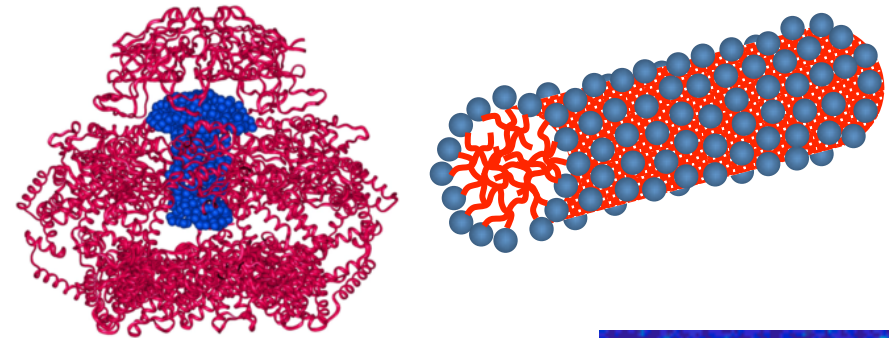
SANS provides information about:

- Size of particle
- Shape of particle
- Inter-particle interactions
- Morphological characteristics
- Surface area-to-volume ratios
- Orientation



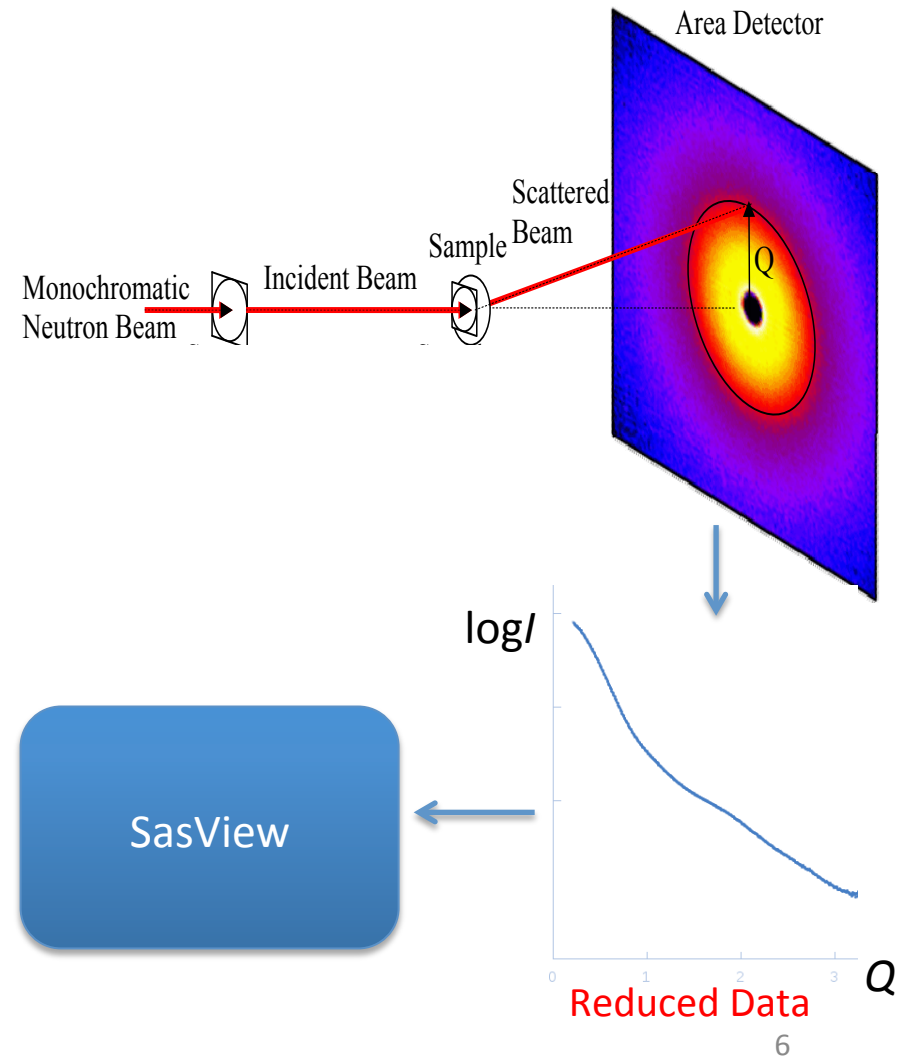
What can we study with SANS?

- Mesoporous structures
- Biological structures
- Polymers
- Colloids and surfactants
- Magnetic films and nanoparticles



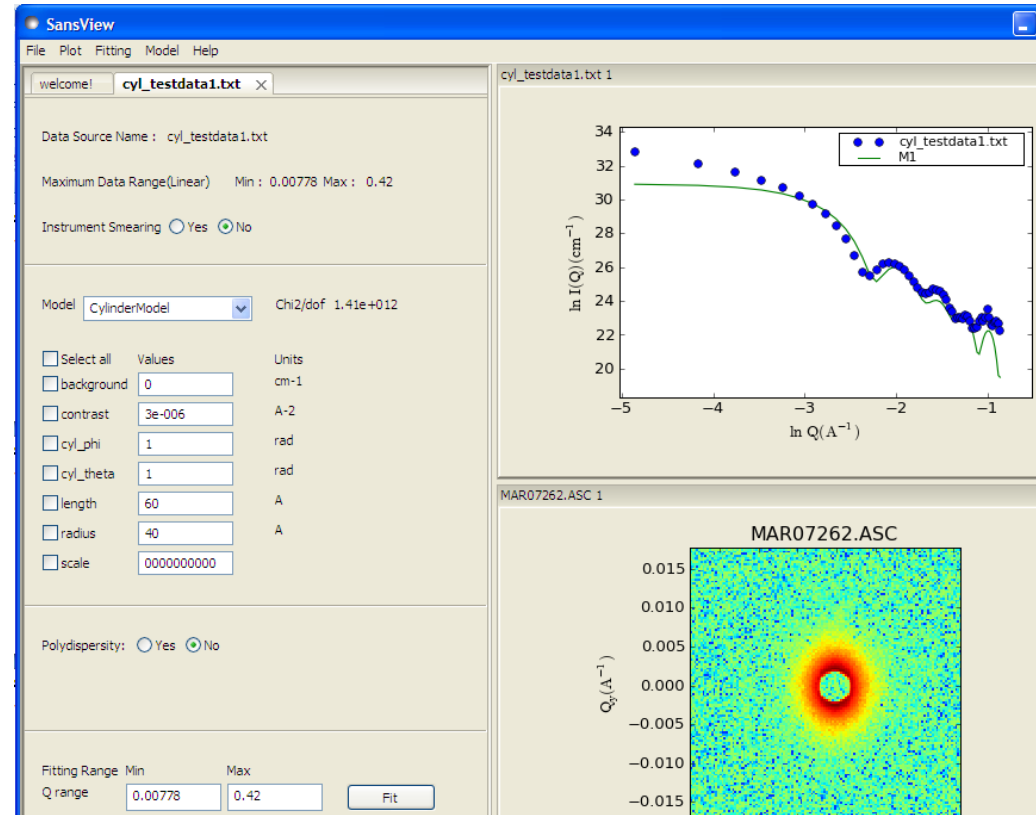
SasView Design Goals

- Operates on **reduced data** and thus should be instrument agnostic.
- Performs modeling in inverse space
- Data analysis toolbox:
 - Fitting models to data
 - $P(r)$ inversion
 - Model-independent analysis
- Plus some other useful tools



SasView - Fitting

- Handles 1D and 2D data
- Form and structure factors for various particle shapes
- Different optimizers (Bayesian Statistics)
- Allows polydispersity
- Simultaneous and batch fitting
- Addition of custom models

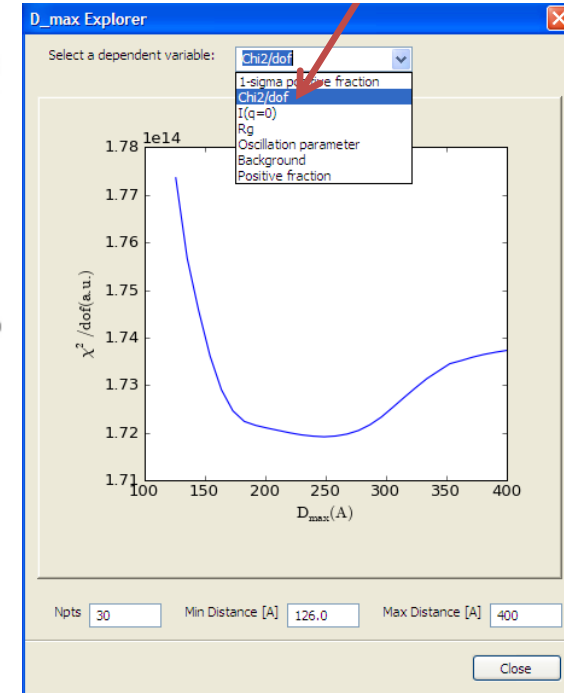
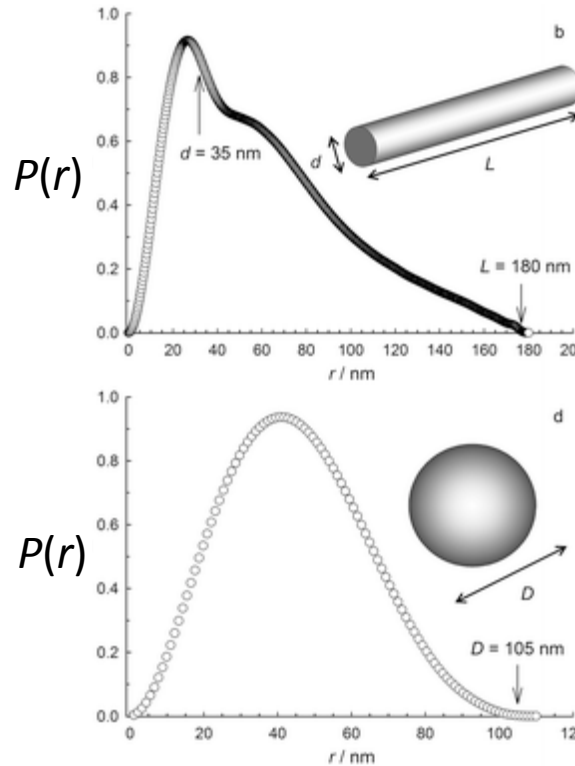


$$P(q, \alpha) = \frac{scale}{V} f^2(q) + bkg$$

$$f(q) = 2(\Delta\rho)V \sin(qL \cos \alpha / 2) / (qL \cos \alpha / 2) \frac{J_1(qr \sin \alpha)}{(qr \sin \alpha)}$$

SasView – P(r) inversion

- Calculates a real-space pair-distance distribution function
- Allows for estimation of background and regularization constant
- Dmax can be explored with respect to χ^2



F. Grrohn, Soft Matter, 2010,6, 4296-4302

<https://github.com/SasView/documents/Tutorial.pdf>

SasView – Model Independent Analysis

Model independent quantities

- The scattering invariant

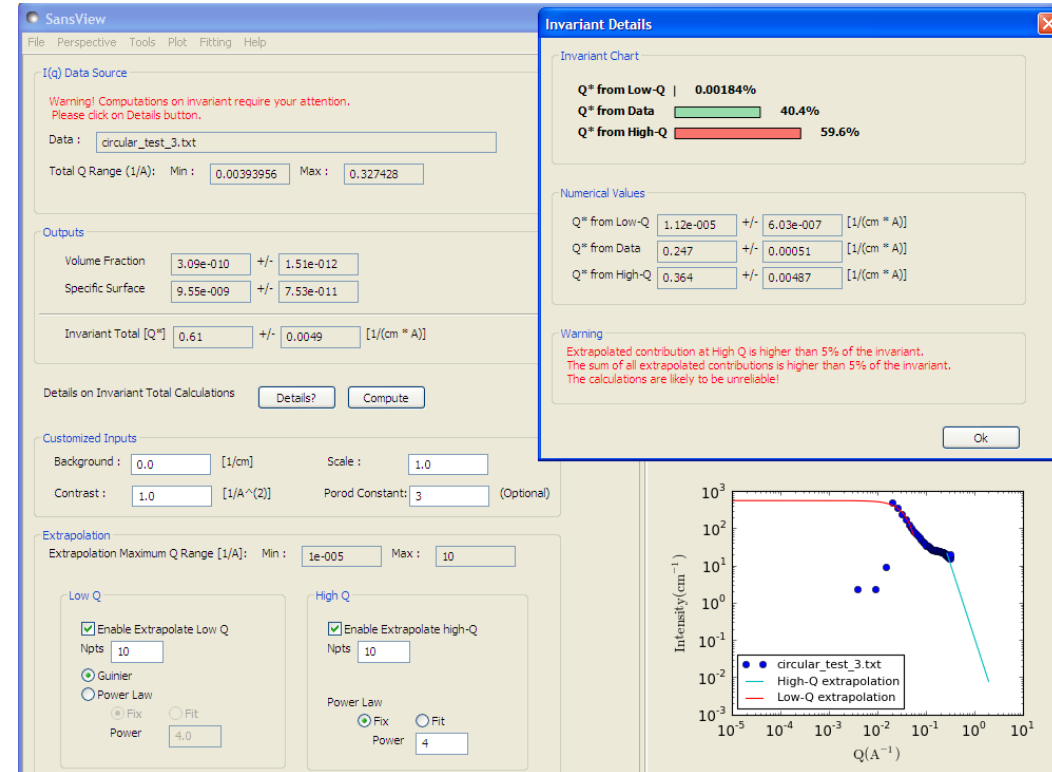
$$Q^* = \int_0^{\infty} (qg)I(q)dq$$

- The volume fraction

$$\phi(1 - \phi) = \frac{Q^*}{2\pi^2(\Delta\rho)^2} \equiv A$$

- The specific surface area

$$S_v = \frac{2\pi\phi(1 - \phi)C_p}{Q^*} = \frac{2\pi AC_p}{Q^*}$$



<https://github.com/SasView/documents/Tutorial.pdf>

SasView History

2006

DANSE (Distributed Analysis for Neutron Scattering Experiments) NSF funded

2011

Community driven project
Releases after code camps

2012

NIST and UMD funded one full time employee

2013

2014

2015

2016

SINE2020 two employees at ESS

2006

2011

2012

2013

2014

2015

2016

SasView 1.0 released
SasView 2.0 released
SasView 3.0 released

SasView 3.1 released

SasView 4.0 coming soon

SasView manpower 2016

Contributors:

- Around 25 part-time
- Main contribution at code camps
- Mostly instrument scientists
- 2 developers working full time SINE2020



Management TEAM:

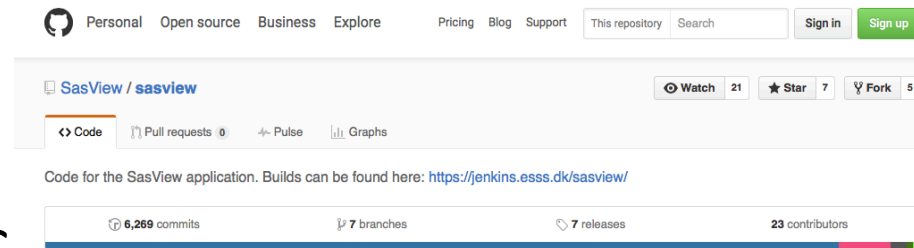
- Paul Butler, NIST
- Mathieu Doucet, SNS
- Andrew Jackson, ESS
- Steve King, ISIS

Facilities involved:

NIST
SNS
ISIS (joined 2012)
ILL (joined 2012)
Ansto
ESS (joined 2014)

SasView Development Workflow

- Code hosted at github
- Trac issue tracking system
- Build system hosted at ESS -DMSC
- Biweekly video conference
- Code camp once or twice per year
- Web-based and built-in documentation
- Tutorial
- Mailing lists



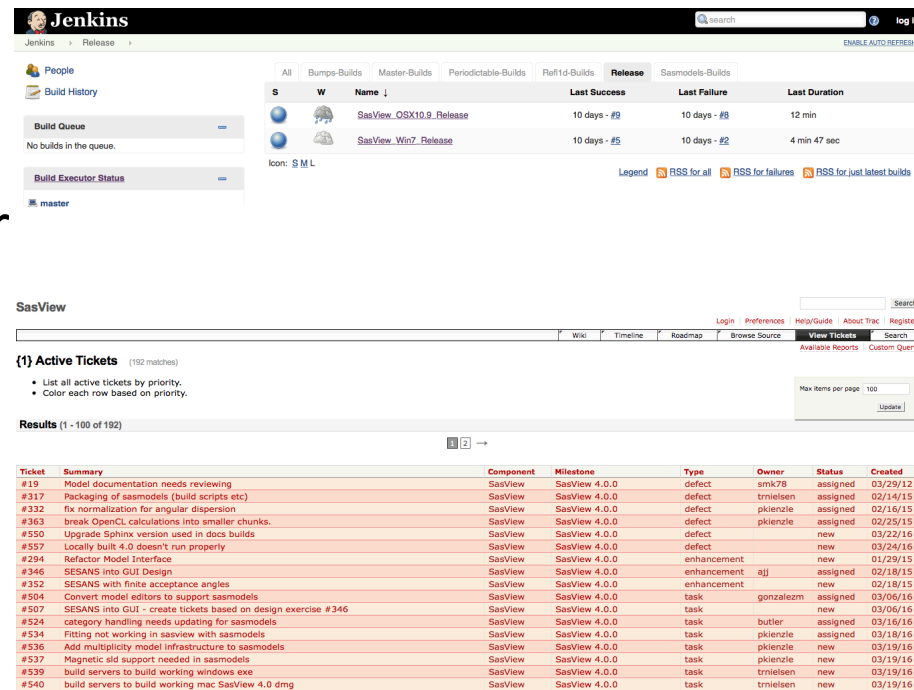
Personal Open source Business Explore Pricing Blog Support This repository Search Sign in Sign up

SasView / sasview Watch 21 Star 7 Fork 5

Code Pull requests Pulse Graphs

Code for the SasView application. Builds can be found here: <https://jenkins.esss.dk/sasview/>

6,269 commits 7 branches 7 releases 23 contributors



Jenkins Release

People Build History

Build Queue: No builds in the queue.

Build Executor Status: master

All	Bumps-Builds	Master-Builds	Periodic-Builds	Ref'd-Builds	Release	Sasmodels-Builds
S	W	Name ↓	Last Success	Last Failure	Last Duration	
		SasView_OSX10.9_Release	10 days - #9	10 days - #8	12 min	
		SasView_Win7_Release	10 days - #5	10 days - #2	4 min 47 sec	

Legend: RSS for all RSS for failures RSS for just latest builds

SasView

Search

Wiki Timeline Roadmap Browse Source View Tickets Search

Log in Preferences Help/Guide About Trac Register

(1) Active Tickets (192 matches)

- List all active tickets by priority.
- Color each row based on priority.

Results (1 - 100 of 192)

Ticket	Summary	Component	Milestone	Type	Owner	Status	Created
#19	Model documentation needs reviewing	SasView	SasView 4.0.0	defect	smk78	assigned	03/29/12
#317	Packaging of sasmodels (build scripts etc)	SasView	SasView 4.0.0	defect	trnielsen	assigned	02/14/15
#332	fix normalization for angular dispersion	SasView	SasView 4.0.0	defect	pkienzle	assigned	02/16/15
#363	break OpenCL calculations into smaller chunks.	SasView	SasView 4.0.0	defect	pkienzle	assigned	02/25/15
#550	Upgrade Sphinx version used in docs builds	SasView	SasView 4.0.0	defect	new	new	03/22/16
#557	Locally built 4.0 doesn't run properly	SasView	SasView 4.0.0	defect	new	new	03/24/16
#294	Refactor Model Interface	SasView	SasView 4.0.0	enhancement	new	new	01/29/15
#346	SESANS into GUI Design	SasView	SasView 4.0.0	enhancement	ajj	assigned	02/18/15
#352	SESANS with finite acceptance angles	SasView	SasView 4.0.0	enhancement	new	new	02/19/15
#504	Convert model editors to support sasmodels	SasView	SasView 4.0.0	task	gonzalez	assigned	03/04/16
#507	SESANS into GUI - create tickets based on design exercise #346	SasView	SasView 4.0.0	task	new	new	03/06/16
#524	category handling needs updating for sasmodels	SasView	SasView 4.0.0	task	butler	assigned	03/16/16
#534	Fitting not working in sasview with sasmodels	SasView	SasView 4.0.0	task	pkienzle	assigned	03/18/16
#536	Add multiplicity model infrastructure to sasmodels	SasView	SasView 4.0.0	task	pkienzle	new	03/19/16
#537	Magnetic aid support needed in sasmodels	SasView	SasView 4.0.0	task	pkienzle	new	03/19/16
#539	build servers to build working windows exe	SasView	SasView 4.0.0	task	trnielsen	new	03/19/16
#540	build servers to build working mac SasView 4.0 dmg	SasView	SasView 4.0.0	task	trnielsen	new	03/19/16

Can we do SasView better?

SasView version 4.0:

- GUI/backend interdependence -> difficult to modularize
- GUI dependent on wx-python -> difficult to maintain
- Limited number of unit tests -> needs to be extended
- Initial GPU implementation -> needs to be optimized
- Quite big collections of model functions -> can be extended

SasView Roadmap vs. SINE2020 goals

- SasView Roadmap

- **Post-CCIV, 4.0**

- Move models to new independent Sasmodels package
- Enable OpenCL GPU utilization for models
- Separation of the model calculation code from the GUI
- Begin work on integrating SESANS into the SasView GUI

- **Post-CCV, 4.w**

- Results reporting refactoring
- Project save refactoring

- **Post-CCVI, 4.x**

- Finish UI and code separation + testing
- Preferences/startup config refactoring

- **Post-CCX, 5.x**

- UI refactor work

- SINE2020 (2016-2017)

- Code modularization
- New API
- New GUI
- Optimization of algorithms for real time analysis
- Extension with SASFit models

SasView Roadmap vs. SINE2020 goals

- SasView Roadmap

- Post-CCIV, 4.0**

- Move models to new independent Sasmodels package
- Enable OpenCL GPU utilization for models
- Separation of the model calculation code from the GUI
- Begin work on integrating SESANS into the SasView GUI

- Post-CCV, 4.w**

- Results reporting refactoring
- Project save refactoring

- Post-CCVI, 4.x**

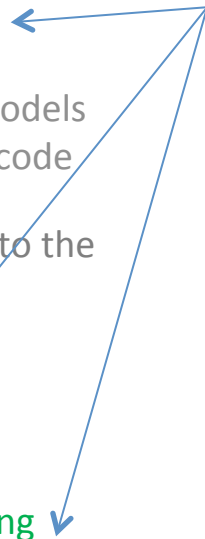
- Finish UI and code separation + testing
- Preferences/startup config refactoring

- Post-CCX, 5.x**

- UI refactor work

- SINE2020 (2016-2017)

- Code modularization
- New API
- New GUI
- Optimization of algorithms for real time analysis
- Extension with SASFit models



SasView Roadmap vs. SINE2020 goals

- SasView Roadmap

- **Post-CCIV, 4.0**

- Move models to new independent Sasmodels package
- Enable OpenCL GPU utilization for models
- Separation of the model calculation code from the GUI
- Begin work on integrating SESANS into the SasView GUI

- **Post-CCV, 4.w**

- Results reporting refactoring
- Project save refactoring

- **Post-CCVI, 4.x**

- Finish UI and code separation + testing
- Preferences/startup config refactoring

- **Post-CCX, 5.x**

- UI remake work

- SINE2020 (2016-2017)

- Code modularization

- **New API**

- New GUI

- Optimization of algorithms for real time analysis

- Extension with SASFit models

SasView Roadmap vs. SINE2020 goals

- SasView Roadmap

- **Post-CCIV, 4.0**

- Move models to new independent Sasmodels package
 - Enable OpenCL GPU utilization for models
 - Separation of the model calculation code from the GUI
 - Begin work on integrating SESANS into the SasView GUI

- **Post-CCV, 4.w**

- Results reporting refactoring
 - **Project save refactoring**

- **Post-CCVI, 4.x**

- Finish UI and code separation + testing
 - **Preferences/startup config refactoring**

- **Post-CCX, 5.x**

- **UI refactor work**

- SINE2020 (2016-2017)

- Code modularization

- New API

- **New GUI**

- Optimization of algorithms for real time analysis

- Extension with SASFit models

SasView Roadmap vs. SINE2020 goals

- SasView Roadmap

- **Post-CCIV, 4.0**

- Move models to new independent Sasmodels package
- **Enable OpenCL GPU utilization for models**
- Separation of the model calculation code from the GUI
- Begin work on integrating SESANS into the SasView GUI

- **Post-CCV, 4.w**

- Results reporting refactoring
- Project save refactoring

- **Post-CCVI, 4.x**

- Finish UI and code separation + testing
- Preferences/startup config refactoring

- **Post-CCX, 5.x**

- UI refactor work

- SINE2020 (2016-2017)

- Code modularization

- New API

- New GUI

- **Optimization of algorithms for real time analysis**

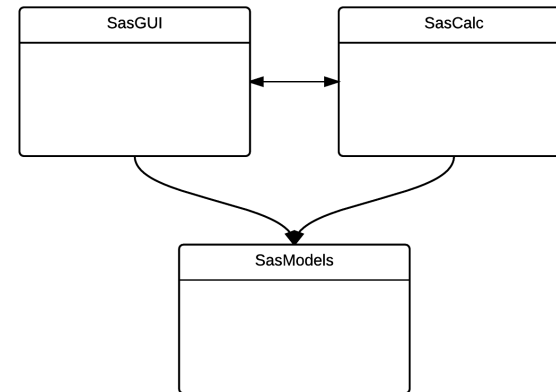
- Extension with SASFit models

Code refactoring

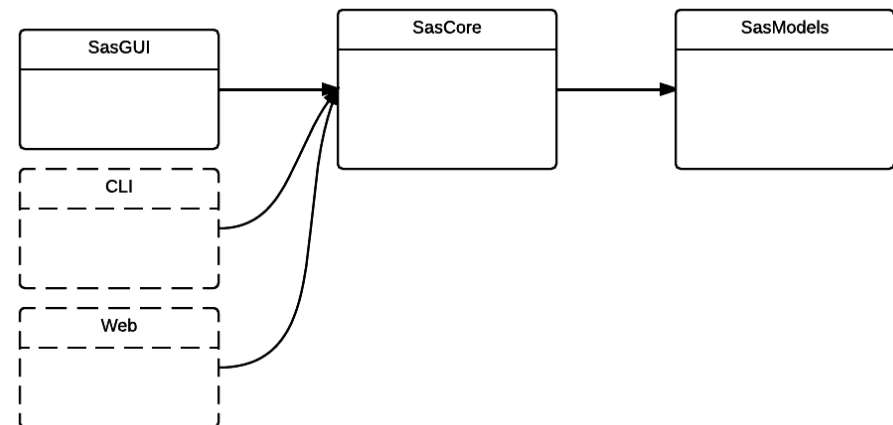
- Separate core functionality: calculations, data management (SasCore) from GUI (SasGUI)
 - SasCore can be used as a standalone module
 - No GUI dependencies in SasCore
 - No SasModels dependencies in SasGUI
- Need to define how to use SasCore modules (agree on API for internal perspectives and calculators)
- Write example scripts and API documentation

Module dependencies

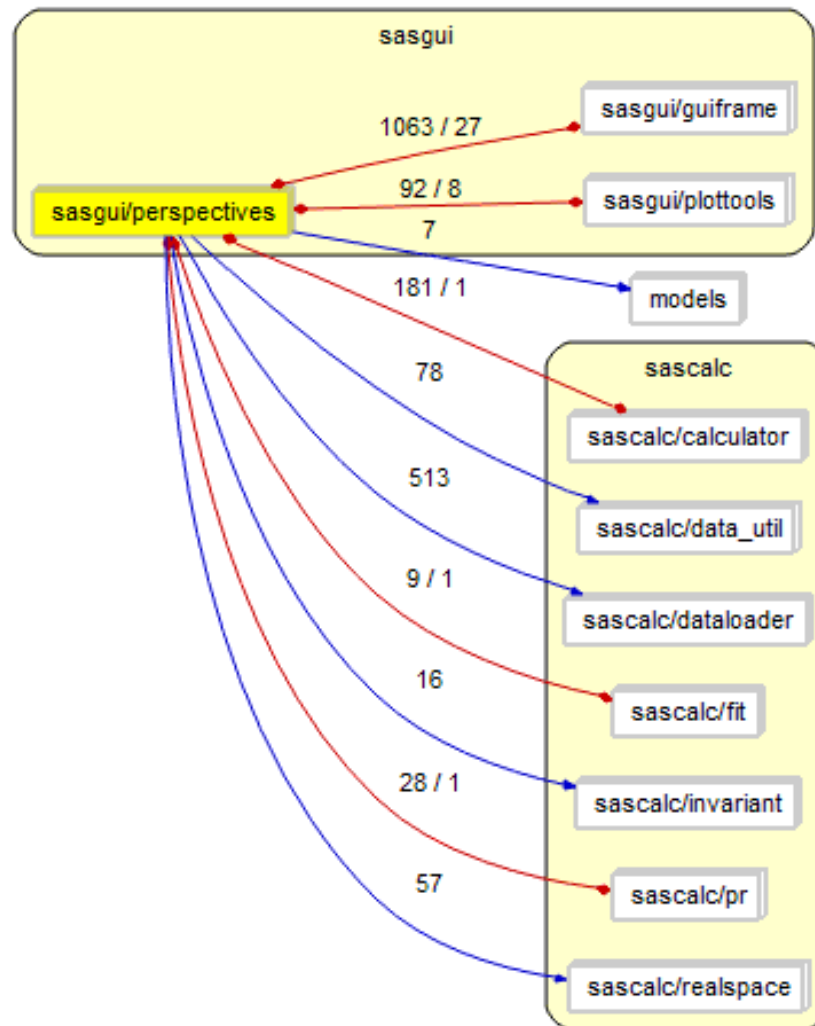
- **Currently :**
 - SasCore depends on some methods in SasGUI
 - SasModels is used by SasGUI



- **Proposed:**
 - Access to SasModels, either directly or through SasCore
 - SasView using SasCore methods exclusively
 - CLI access to all the SasCore functionality and models



Module dependencies



Using SasCalc as a module

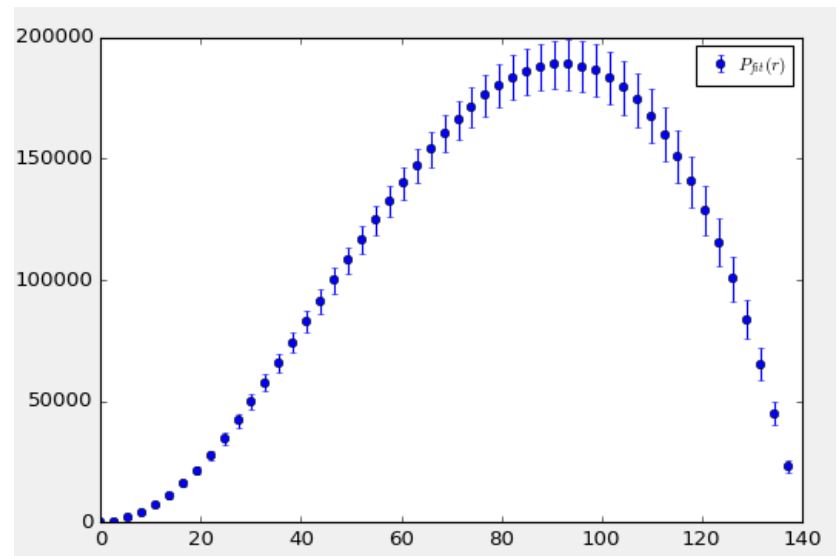
```
from sas.sascore.dataloader.loader import Loader
from sas.sascore.pr.invertor import Invertor
```

```
# Instantiate the file loader
loader = Loader()
test_data = loader.load("sphere_80.txt")
```

```
# Instantiate the invertor
pr = Invertor()
pr.x = test_data.x
pr.y = test_data.y
```

```
# Perform inversion and show graph
x, y = pr.invert()
```

```
import matplotlib.pyplot as plt
plt.plot(x, y)
plt.show()
```



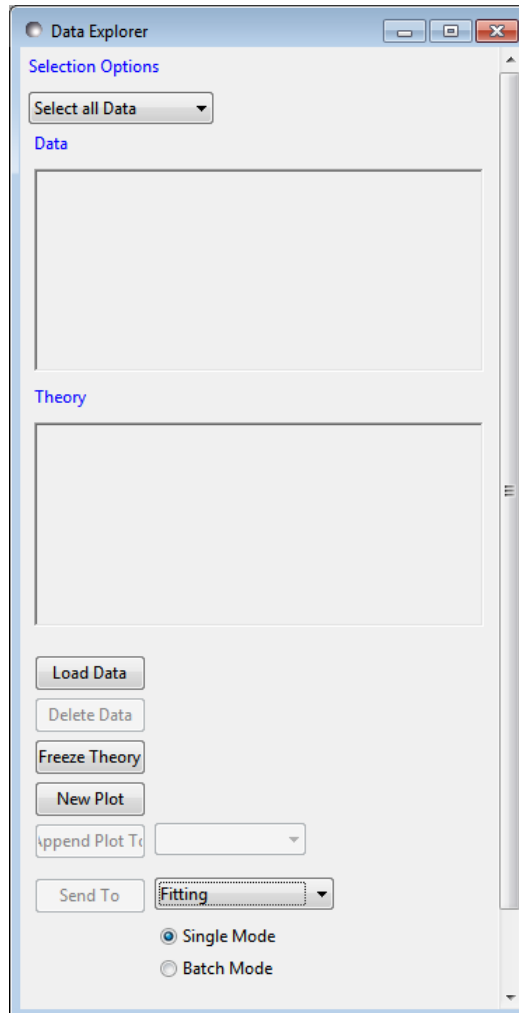
Need to (re)define how the calculators are to be used – agree on API for them and possibly standardize with existing codes (BornAgain/Mantid)

GUI modernization

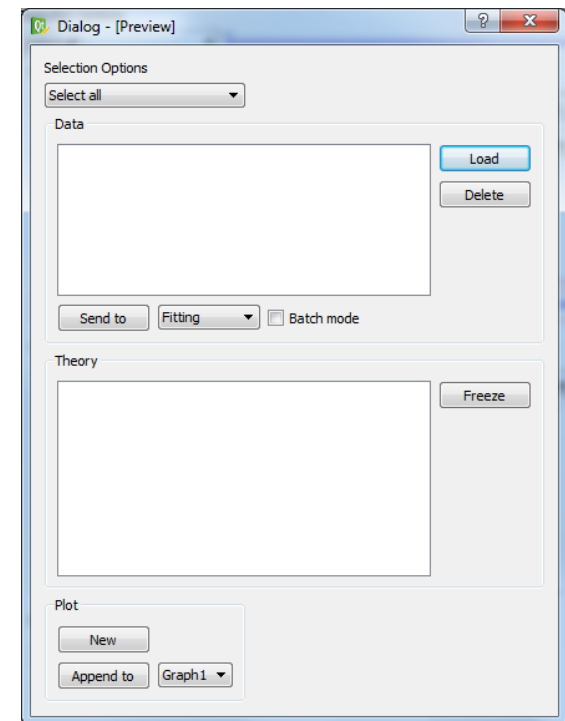
- With SasGUI refactored it is possible to work exclusively on the GUI part of SasView
- Proposed rewrite using PyQt
 - Platform consistency - dialogs look and behave the same across all platforms
 - Professional, more mature technology
 - Long term maintainability
 - Ease of development (Qt designer)
 - Clean separation of UI and code
 - Signals connected to slots automatically
 - Inherent MVC pattern in Qt simplifying data management
 - Native thread support
 - SINE2020 requirement for interoperability with other Qt based codes (Mantid, BornAgain)
 - Test driven development

Quick dialog prototyping

Current view

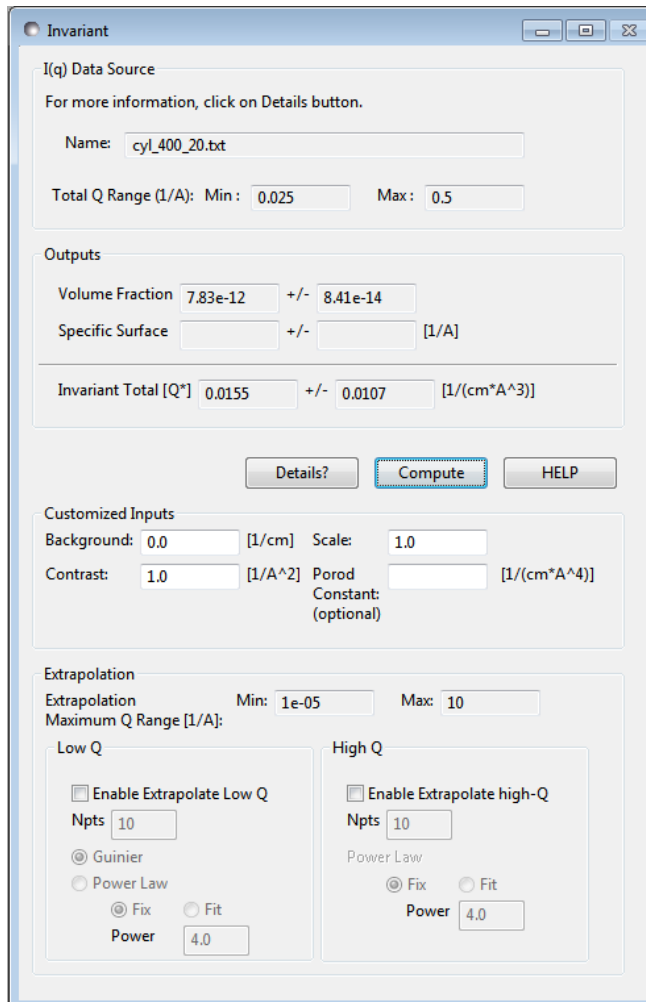


New look



Quick dialog prototyping

Current view



Invariant

I(q) Data Source
For more information, click on Details button.

Name:

Total Q Range (1/A): Min: Max:

Outputs

Volume Fraction +/-

Specific Surface +/- [1/A]

Invariant Total [Q*] +/- [1/(cm*A^3)]

Customized Inputs

Background: [1/cm] Scale:

Contrast: [1/A^2] Porod Constant: [1/(cm*A^4)] (optional)

Extrapolation

Extrapolation Min: Max:

Maximum Q Range [1/A]:

Low Q

Enable Extrapolate Low Q

Npts:

Guinier Power Law

Fix Fit

Power

High Q

Enable Extrapolate high-Q

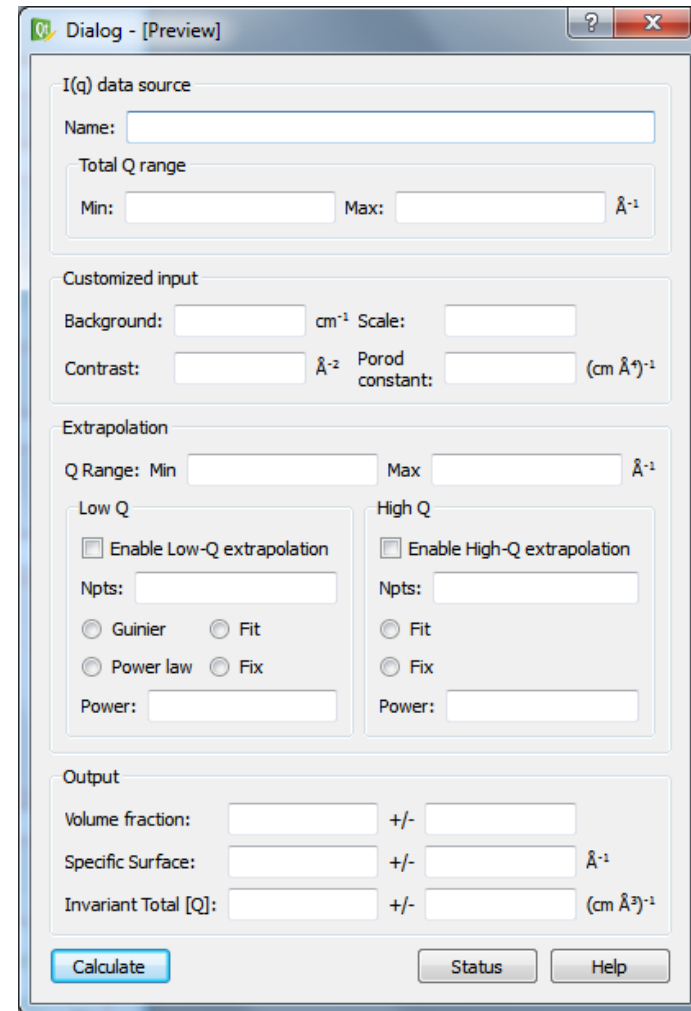
Npts:

Power Law

Fix Fit

Power

New look



Dialog - [Preview]

I(q) data source

Name:

Total Q range

Min: Max: Å⁻¹

Customized input

Background: cm⁻¹ Scale:

Contrast: Å⁻² Porod constant: (cm Å⁴)⁻¹

Extrapolation

Q Range: Min Max Å⁻¹

Low Q

Enable Low-Q extrapolation

Npts:

Guinier Fit

Power law Fix

Power:

High Q

Enable High-Q extrapolation

Npts:

Fit

Fix

Power:

Output

Volume fraction: +/-

Specific Surface: +/- Å⁻¹

Invariant Total [Q]: +/- (cm Å³)⁻¹

Release schedule

- Initial version for internal release: October 2016
- Available in the current release: 2nd half of 2017



SasFit integration

- Idea – reuse a large set of SasFit fitting functions in SasModels
- Create setup similar for SasModel – conversion Wiki, compare.sh script, etc.
- Investigate possible automation of the conversion or its parts
- Examine SasFit structure factor methods with the OZ equation
- In collaboration with PSI

SasFit integration

