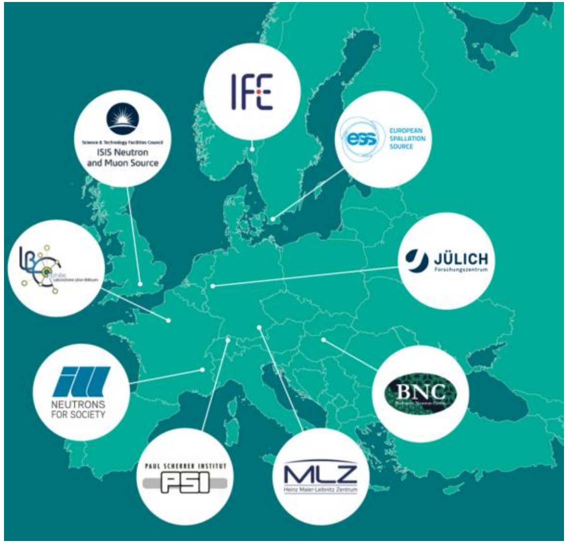
Science Project 8: PaNOSC – Dynamics of biological processes using Small Angle Scattering

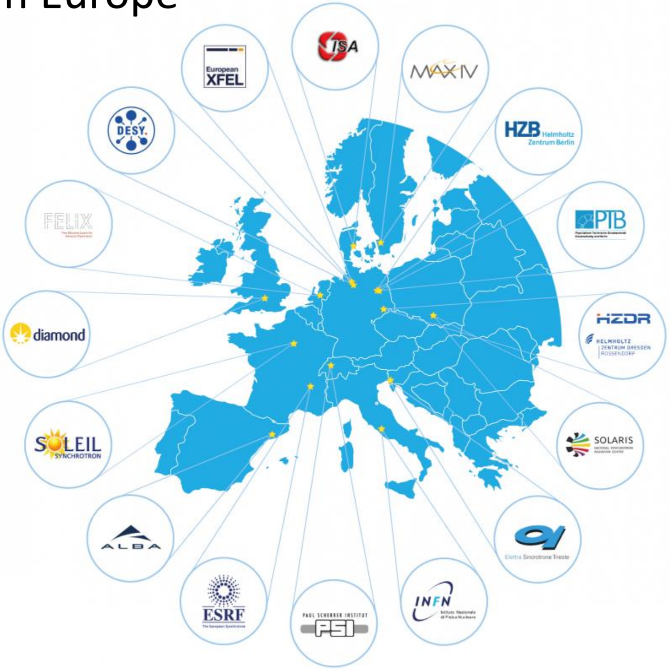
Miguel A. González, Irina Safiulina

Institut Laue-Langevin, Grenoble (France)

PaNOSC: What is it?

- **Photon and Neutron Open Science Cloud** (Grant no. 823852 in EU Horizon 2020 program)
- ExPaNDS (Grant no. 857641, PaNOSC complement for national facilities)
- Neutron (LENS) and synchrotron (LEAPS) facilities in Europe


IFE
Science & Technology Facilities Council
ISIS Neutron and Muon Source
ess EUROPEAN SPALLATION SOURCE
LLB
JÜLICH Forschungszentrum
NEUTRONS FOR SOCIETY
BNC
PAUL SCHERRER INSTITUT PSI
MLZ Heinz Maier-Leibnitz Zentrum


European XFEL
ISA
MAX IV
DESY.
HZB Helmholtz Zentrum Berlin
FELIX
PTB
diamond
HZDR HELMHOLTZ ZENTRUM DRESDEN ROSSENDORF
SOLEIL SYNCHROTRON
SOLARIS
ALBA
Elettra Sincrotrone Trieste
ESRF
PAUL SCHERRER INSTITUT PSI
INFN

EOSC Future

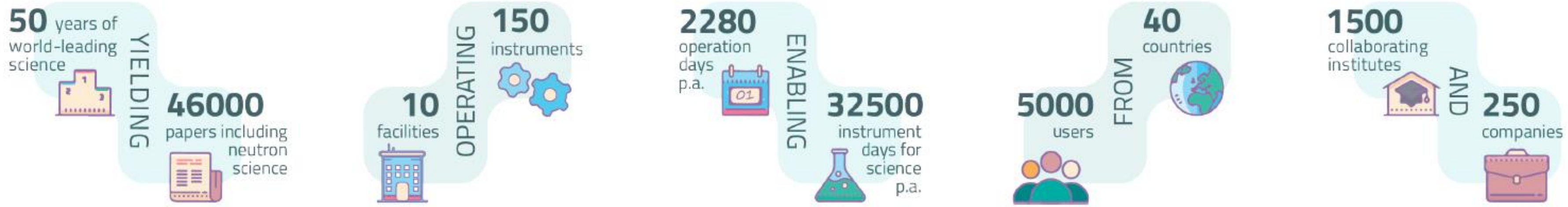
panosc photon and neutron open science cloud

ExPaNDS

egi

PaNOSC: What is it?

Neutron science in Europe:



50 years of world-leading science YIELDING 46000 papers including neutron science

10 facilities OPERATING 150 instruments

2280 operation days p.a. ENABLING 32500 instrument days for science p.a.

5000 users FROM 40 countries

1500 collaborating institutes AND 250 companies

Photon science in Europe:

Since 1960s

35000 researchers

5000+ publications/year

5 Nobel prizes

19 facilities
>300 end-stations
>1 million hours beam time/year

PaNOSC: The challenge of diversity

- Very diverse scientific communities
- 1000s of individual experiments
- Data and metadata?
  - FAIR data ✓
  - Instrument metadata ✓
  - Experiment metadata ✓ ?
  - Sample metadata ? ✗
  - Other characterization ✗

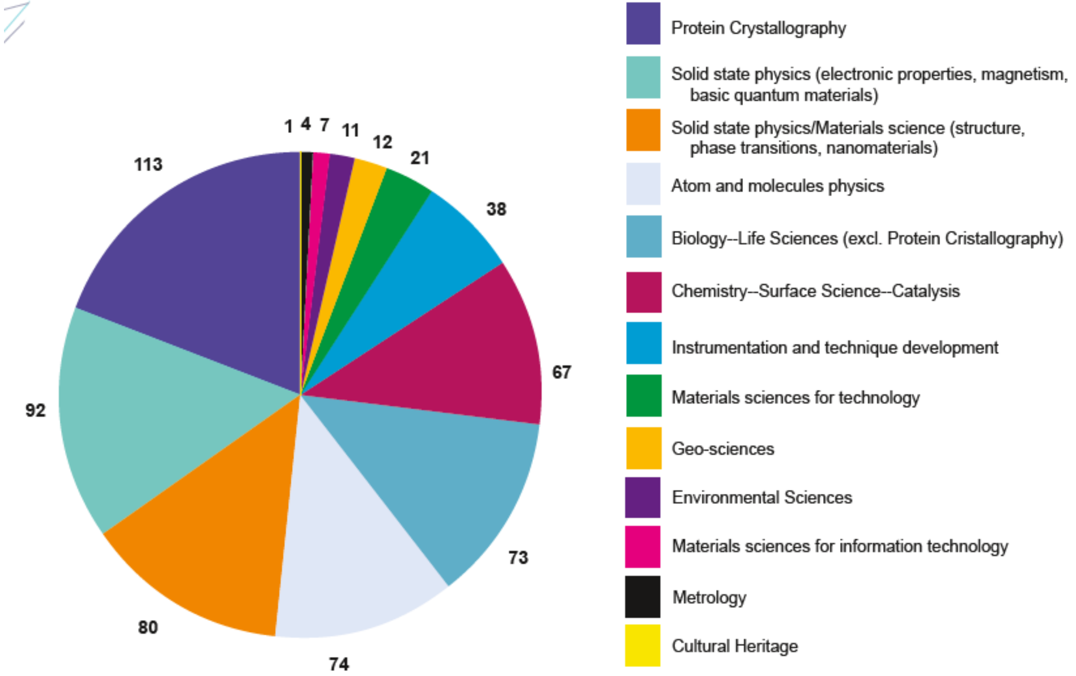
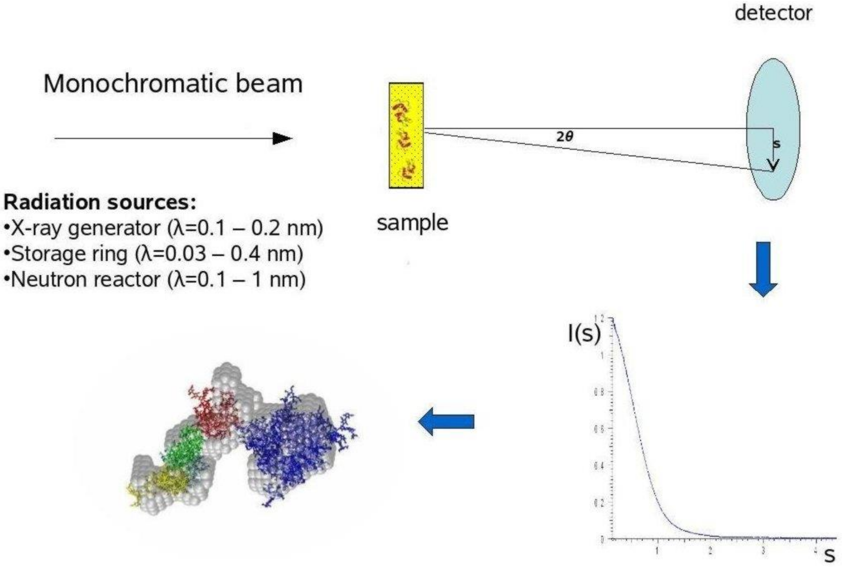


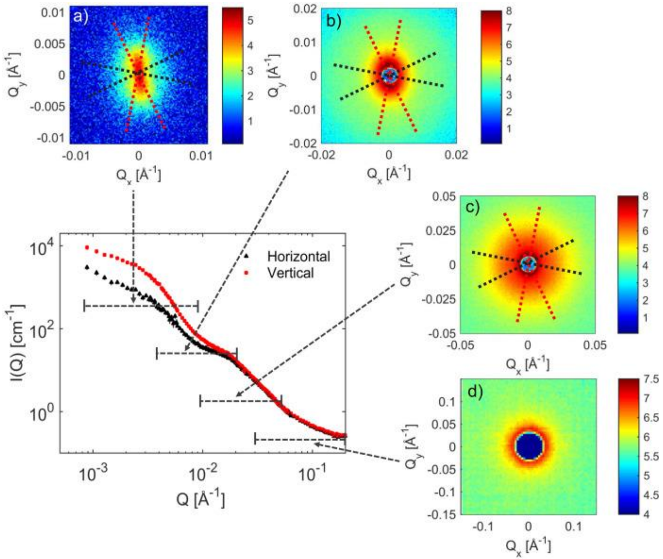
Fig. 1: Use of LEAPS labs for a broad science community. Example: number of publications at SOLEIL in 2018, by scientific field

EOSC Future: Scientific project

Focus in a single technique: Small Angle (Neutron/X-ray) Scattering (SANS/SAXS) and a single problem: Conformation of proteins and macromolecular complexes.



Radiation sources:
- X-ray generator (λ=0.1 – 0.2 nm)
- Storage ring (λ=0.03 – 0.4 nm)
- Neutron reactor (λ=0.1 – 1 nm)

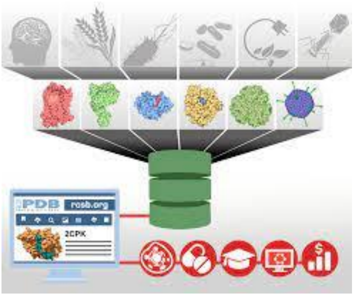


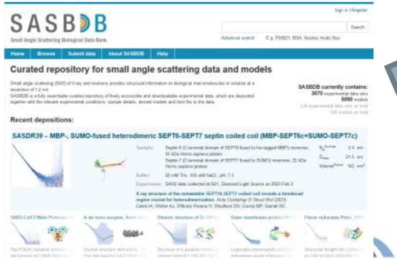
Tian et al, Food hydrocolloids (2020)

EOSC Future: Scientific project

Goal: Create an easy (semi-)automatic workflow allowing to download **reference structures** of proteins or biomacromolecular complexes and **experimental data** (from user facilities or curated databases) and combine them to obtain a set of plausible molecular conformations.

Uses: Teaching and facilitate use of SANS/SAXS techniques by non-experts.





Curated repository for small angle scattering data and models

Small angle scattering (SAS) of X-ray and neutrons provides structural information on biological macromolecules in solution at a resolution of 1-2 nm.
SASBDB is a fully searchable curated repository of freely accessible and downloadable experimental data, which are deposited together with the relevant experimental conditions, sample details, derived models and their fits to the data.

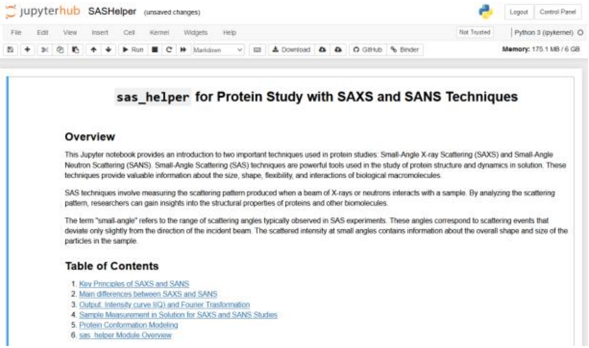
SASBDB currently contains:
3670 experimental data sets
5090 models

Recent depositions:

SASDR39 – MBP-, SUMO-fused heterodimeric SEPT6-SEPT7 septin coiled coil (MBP-SEPT6c+SUMO-SEPT7c)

Buffer: 50 mM Tris, 150 mM NaCl, pH 7.5

Experiment: SAXS data collected at B21, Diamond Light Source on 2023 Feb 3



`sas_helper` for Protein Study with SAXS and SANS Techniques

Overview

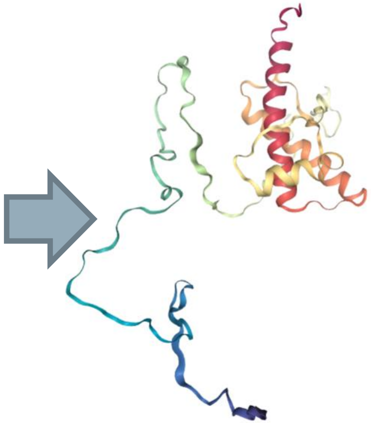
This Jupyter notebook provides an introduction to two important techniques used in protein studies: Small-Angle X-ray Scattering (SAXS) and Small-Angle Neutron Scattering (SANS). Small-Angle Scattering (SAS) techniques are powerful tools used in the study of protein structure and dynamics in solution. These techniques provide valuable information about the size, shape, flexibility, and interactions of biological macromolecules.

SAS techniques involve measuring the scattering pattern produced when a beam of X-rays or neutrons interacts with a sample. By analyzing the scattering pattern, researchers can gain insights into the structural properties of proteins and other biomolecules.

The term "small-angle" refers to the range of scattering angles typically observed in SAS experiments. These angles correspond to scattering events that deviate only slightly from the direction of the incident beam. The scattered intensity at small angles contains information about the overall shape and size of the particles in the sample.

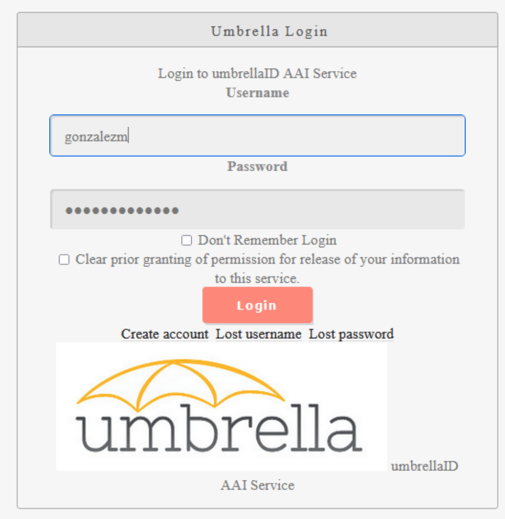
Table of Contents

1. Key Principles of SAXS and SANS
2. Main differences between SAXS and SANS
3. Output: Intensity curve I(Q) and Fourier Trasformation
4. Sample Measurement in Solution for SAXS and SANS Studies
5. Protein Conformation Modeling
6. sas_helper Module Overview



The demonstration in a nutshell

STEP 1:
http://replay.notebooks.egi.eu
login e.g. using UmbrellaID



Umbrella Login

Login to umbrellaID AAI Service

Username

gonzalezm

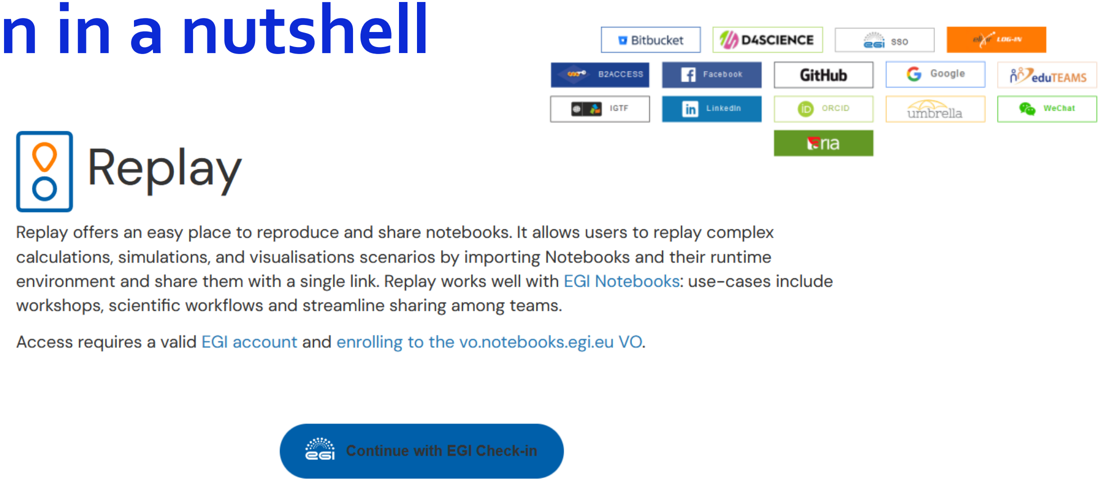
Password

••••••••••••••

☐ Don't Remember Login

☐ Clear prior granting of permission for release of your information to this service.

umbrellaID AAI Service



Replay

Replay offers an easy place to reproduce and share notebooks. It allows users to replay complex calculations, simulations, and visualisations scenarios by importing Notebooks and their runtime environment and share them with a single link. Replay works well with EGI Notebooks: use-cases include workshops, scientific workflows and streamline sharing among teams.

Access requires a valid EGI account and enrolling to the vo.notebooks.egi.eu VO.

UMBRELLAID: _THE_ DIGITAL IDENTITY FOR PHOTON AND NEUTRON USERS

umbrellaID is a digital identity, developed for use by **Photon and Neutron users**. Globally **recognised** and **secure**.

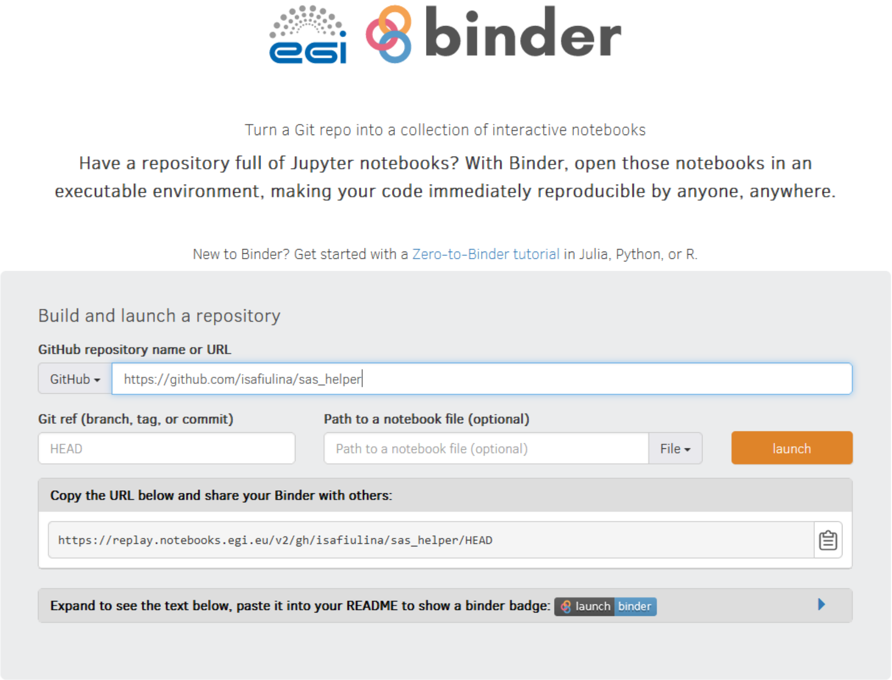
Nice! How can I use it?

umbrellaID is built on top of the existing local IT infrastructures and it **links its identity** in a once-only action directly to the local identity at the respective facility.

The demonstration in a nutshell

STEP 2:
https://github.com/isafiulina/sas_helper
binder

TO DO: Findable in EOSC Marketplace!



Turn a Git repo into a collection of interactive notebooks

Have a repository full of Jupyter notebooks? With Binder, open those notebooks in an executable environment, making your code immediately reproducible by anyone, anywhere.

New to Binder? Get started with a Zero-to-Binder tutorial in Julia, Python, or R.

Build and launch a repository

GitHub repository name or URL: GitHub ▾ https://github.com/isafiulina/sas_helper

Git ref (branch, tag, or commit): HEAD

Path to a notebook file (optional): Path to a notebook file (optional) File ▾

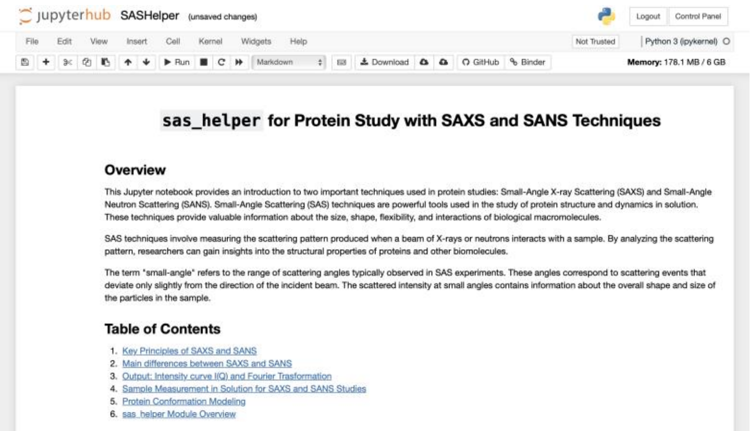
Copy the URL below and share your Binder with others:

```
https://replay.notebooks.egi.eu/v2/gh/isafiulina/sas_helper/HEAD
```

Expand to see the text below, paste it into your README to show a binder badge: launch binder

The demonstration in a nutshell

STEP 3:
Jupyter Notebook environment
(upload data or go to Notebook)



`sas_helper` for Protein Study with SAXS and SANS Techniques

Overview

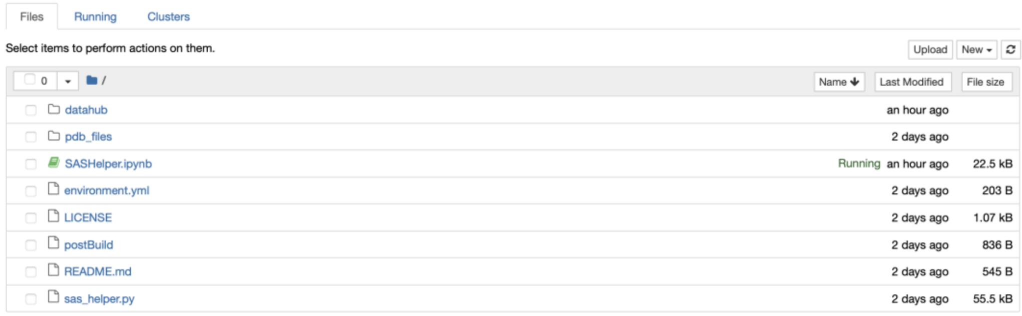
This Jupyter notebook provides an introduction to two important techniques used in protein studies: Small-Angle X-ray Scattering (SAXS) and Small-Angle Neutron Scattering (SANS). Small-Angle Scattering (SAS) techniques are powerful tools used in the study of protein structure and dynamics in solution. These techniques provide valuable information about the size, shape, flexibility, and interactions of biological macromolecules.

SAS techniques involve measuring the scattering pattern produced when a beam of X-rays or neutrons interacts with a sample. By analyzing the scattering pattern, researchers can gain insights into the structural properties of proteins and other biomolecules.

The term "small-angle" refers to the range of scattering angles typically observed in SAS experiments. These angles correspond to scattering events that deviate only slightly from the direction of the incident beam. The scattered intensity at small angles contains information about the overall shape and size of the particles in the sample.

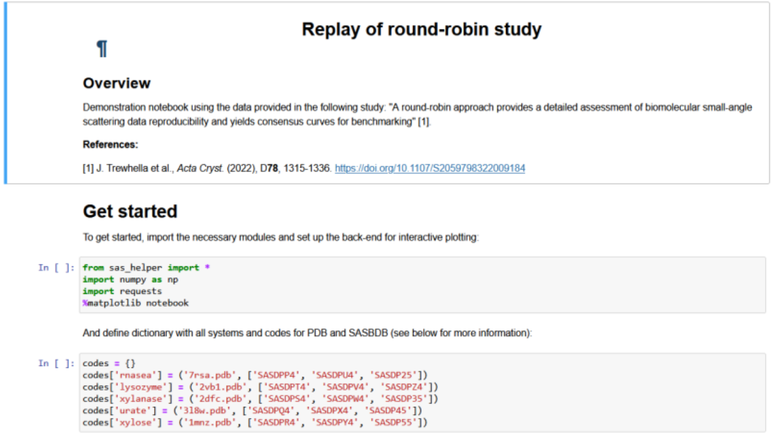
Table of Contents

1. Key Principles of SAXS and SANS
2. Main differences between SAXS and SANS
3. Output: Intensity curve I(Q) and Fourier Trasformation
4. Sample Measurement in Solution for SAXS and SANS Studies
5. Protein Conformation Modeling
6. sas_helper Module Overview



Select items to perform actions on them.

Name		Last Modified	File size
datahub		an hour ago	
pdb_files		2 days ago	
SASHelper.ipynb	Running	an hour ago	22.5 kB
environment.yml		2 days ago	203 B
LICENSE		2 days ago	1.07 kB
postBuild		2 days ago	836 B
README.md		2 days ago	545 B
sas_helper.py		2 days ago	55.5 kB



Replay of round-robin study

¶

Overview

Demonstration notebook using the data provided in the following study: "A round-robin approach provides a detailed assessment of biomolecular small-angle scattering data reproducibility and yields consensus curves for benchmarking" [1].

References:

[1] J. Trewhella et al., *Acta Cryst.* (2022), D**78**, 1315-1336. https://doi.org/10.1107/S2059798322009184

Get started

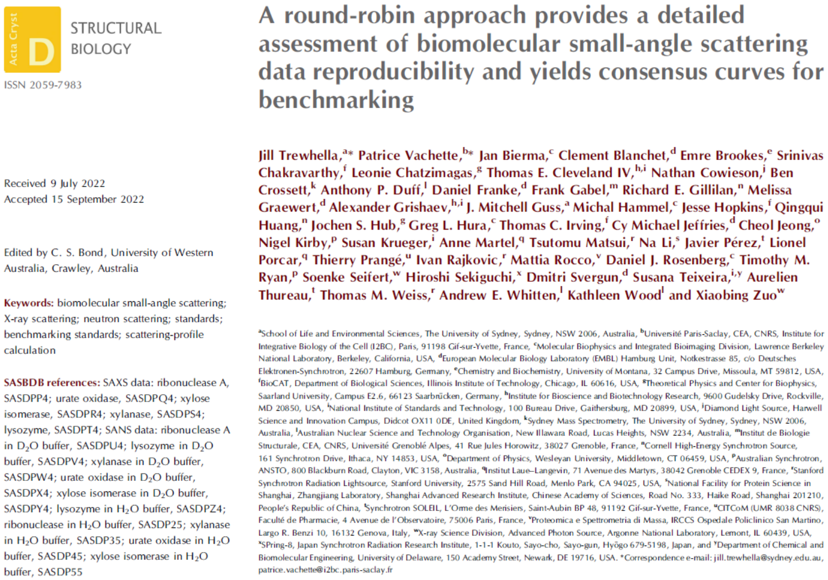
To get started, import the necessary modules and set up the back-end for interactive plotting:

```
In [ ]: from sas_helper import *
        import numpy as np
        import requests
        %matplotlib notebook
```

And define dictionary with all systems and codes for PDB and SASBDB (see below for more information):

```
In [ ]: codes = {}
        codes['rnasea'] = ('7rsa.pdb', ['SASDPP4', 'SASDPU4', 'SASDP25'])
        codes['lysozyme'] = ('2vb1.pdb', ['SASDPT4', 'SASDPV4', 'SASDPZ4'])
        codes['xylanase'] = ('2dfc.pdb', ['SASDPS4', 'SASDPW4', 'SASDP35'])
        codes['urate'] = ('3l8w.pdb', ['SASDPQ4', 'SASDPX4', 'SASDP45'])
        codes['xylose'] = ('1mnz.pdb', ['SASDPR4', 'SASDPY4', 'SASDP55'])
```

The demonstration in a nutshell



Acta Cryst D
STRUCTURAL
BIOLOGY

ISSN 2059-7983

Received 9 July 2022
Accepted 15 September 2022

Edited by C. S. Bond, University of Western Australia, Crawley, Australia

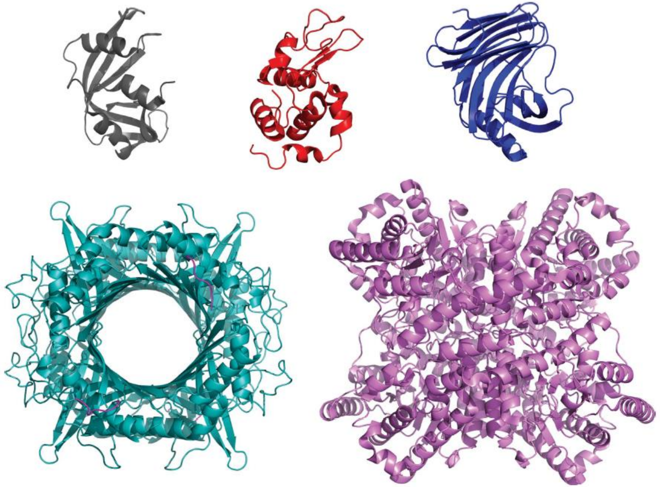
Keywords: biomolecular small-angle scattering; X-ray scattering; neutron scattering; standards; benchmarking standards; scattering-profile calculation

SASBDB references: SAXS data: ribonuclease A, SASDPP4; urate oxidase, SASDPQ4; xylose isomerase, SASDPR4; xylanase, SASDPS4; lysozyme, SASDPT4; SANS data: ribonuclease A in D_2O buffer, SASDPU4; lysozyme in D_2O buffer, SASDPV4; xylanase in D_2O buffer, SASDPW4; urate oxidase in D_2O buffer, SASDPX4; xylose isomerase in D_2O buffer, SASDPY4; lysozyme in H_2O buffer, SASDPZ4; ribonuclease in H_2O buffer, SASDP25; xylanase in H_2O buffer, SASDP35; urate oxidase in H_2O buffer, SASDP45; xylose isomerase in H_2O buffer, SASDP55

A round-robin approach provides a detailed assessment of biomolecular small-angle scattering data reproducibility and yields consensus curves for benchmarking

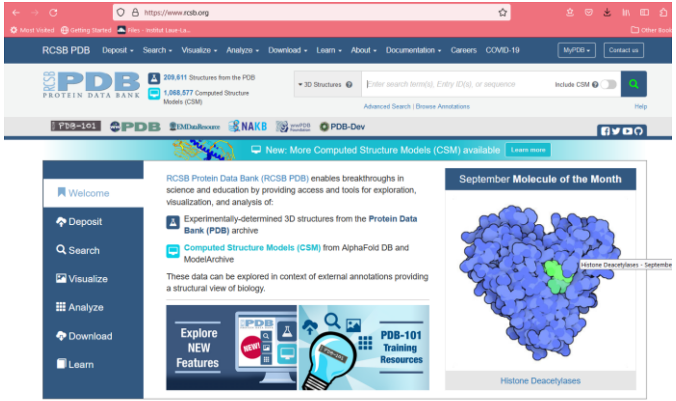
Jill Trewhella,[a*] Patrice Vachette,[b*] Jan Bierma,[c] Clement Blanchet,[d] Emre Brookes,[e] Srinivas Chakravarthy,[f] Leonie Chatzimagas,[g] Thomas E. Cleveland IV,[h,i] Nathan Cowieson,[j] Ben Crossett,[k] Anthony P. Duff,[l] Daniel Franke,[d] Frank Gabel,[m] Richard E. Gillilan,[n] Melissa Graewert,[d] Alexander Grishaev,[h,i] J. Mitchell Guss,[a] Michal Hammel,[c] Jesse Hopkins,[f] Qingqui Huang,[n] Jochen S. Hub,[g] Greg L. Hura,[c] Thomas C. Irving,[f] Cy Michael Jeffries,[d] Cheol Jeong,[o] Nigel Kirby,[p] Susan Krueger,[i] Anne Martel,[q] Tsutomu Matsui,[r] Na Li,[s] Javier Pérez,[t] Lionel Porcar,[q] Thierry Prangé,[u] Ivan Rajkovic,[r] Mattia Rocco,[v] Daniel J. Rosenberg,[c] Timothy M. Ryan,[p] Soenke Seifert,[w] Hiroshi Sekiguchi,[x] Dmitri Svergun,[d] Susana Teixeira,[i,y] Aurelien Thureau,[t] Thomas M. Weiss,[r] Andrew E. Whitten,[l] Kathleen Wood[l] and Xiaobing Zuo[w]

[a]School of Life and Environmental Sciences, The University of Sydney, Sydney, NSW 2006, Australia, [b]Université Paris-Saclay, CEA, CNRS, Institute for Integrative Biology of the Cell (I2BC), Paris, 91198 Gif-sur-Yvette, France, [c]Molecular Biophysics and Integrated Bioimaging Division, Lawrence Berkeley National Laboratory, Berkeley, California, USA, [d]European Molecular Biology Laboratory (EMBL) Hamburg Unit, Notkestrasse 85, c/o Deutsches Elektronen-Synchrotron, 22607 Hamburg, Germany, [e]Chemistry and Biochemistry, University of Montana, 32 Campus Drive, Missoula, MT 59812, USA, [f]BioCAT, Department of Biological Sciences, Illinois Institute of Technology, Chicago, IL 60616, USA, [g]Theoretical Physics and Center for Biophysics, Saarland University, Campus E2.6, 66123 Saarbrücken, Germany, [h]Institute for Bioscience and Biotechnology Research, 9600 Gudelsky Drive, Rockville, MD 20850, USA, [i]National Institute of Standards and Technology, 100 Bureau Drive, Gaithersburg, MD 20899, USA, [j]Diamond Light Source, Harwell Science and Innovation Campus, Didcot OX11 0DE, United Kingdom, [k]Sydney Mass Spectrometry, The University of Sydney, Sydney, NSW 2006, Australia, [l]Australian Nuclear Science and Technology Organisation, New Illawara Road, Lucas Heights, NSW 2234, Australia, [m]Institut de Biologie Structurale, CEA, CNRS, Université Grenoblé Alpes, 41 Rue Jules Horowitz, 38027 Grenoble, France, [n]Cornell High-Energy Synchrotron Source, 161 Synchrotron Drive, Ithaca, NY 14853, USA, [o]Department of Physics, Wesleyan University, Middletown, CT 06459, USA, [p]Australian Synchrotron, ANSTO, 800 Blackburn Road, Clayton, VIC 3158, Australia, [q]Institut Laue–Langevin, 71 Avenue des Martyrs, 38042 Grenoble CEDEX 9, France, [r]Stanford Synchrotron Radiation Lightsource, Stanford University, 2575 Sand Hill Road, Menlo Park, CA 94025, USA, [s]National Facility for Protein Science in Shanghai, Zhangjiang Laboratory, Shanghai Advanced Research Institute, Chinese Academy of Sciences, Road No. 333, Haike Road, Shanghai 201210, People's Republic of China, [t]Synchrotron SOLEIL, L'Orme des Merisiers, Saint-Aubin BP 48, 91192 Gif-sur-Yvette, France, [u]CITCoM (UMR 8038 CNRS), Faculté de Pharmacie, 4 Avenue de l'Observatoire, 75006 Paris, France, [v]Proteomica e Spettrometria di Massa, IRCCS Ospedale Policlinico San Martino, Largo R. Benzi 10, 16132 Genova, Italy, [w]X-ray Science Division, Advanced Photon Source, Argonne National Laboratory, Lemont, IL 60439, USA, [x]SPring-8, Japan Synchrotron Radiation Research Institute, 1-1-1 Kouto, Sayo-cho, Sayo-gun, Hyōgo 679-5198, Japan, and [y]Department of Chemical and Biomolecular Engineering, University of Delaware, 150 Academy Street, Newark, DE 19716, USA. *Correspondence e-mail: jill.trewhella@sydney.edu.au, patrice.vachette@i2bc.paris-saclay.fr



The demonstration in a nutshell

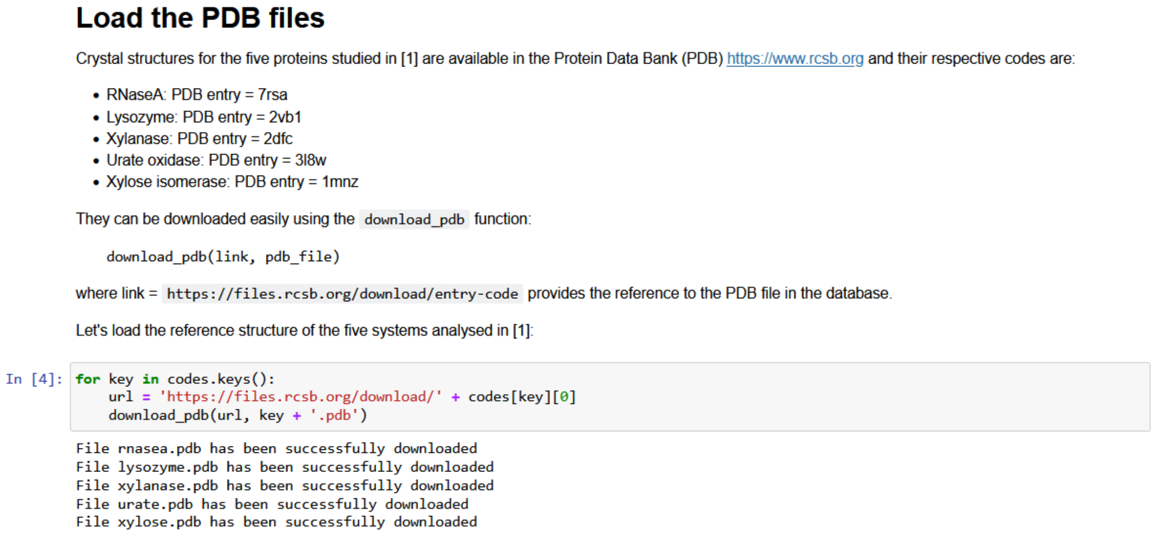
STEP 4:
Get protein reference structure
from Open Data Base


https://www.rcsb.org
RCSB PDB
Deposit
Search
Visualize
Analyze
Download
Learn
About
Documentation
Careers
COVID-19
PDB PROTEIN DATA BANK
209,611 Structures from the PDB
1,068,577 Computed Structure Models (CSM)
Welcome
September Molecule of the Month
Histone Deacetylases
Explore NEW Features
PDB-101 Training Resources
New: More Computed Structure Models (CSM) available

RCSB Protein Data Bank (RCSB PDB) enables breakthroughs in science and education by providing access and tools for exploration, visualization, and analysis of:

- Experimentally-determined 3D structures from the Protein Data Bank (PDB) archive
- Computed Structure Models (CSM) from AlphaFold DB and ModelArchive

These data can be explored in context of external annotations providing a structural view of biology.



Load the PDB files

Crystal structures for the five proteins studied in [1] are available in the Protein Data Bank (PDB) https://www.rcsb.org and their respective codes are:

- RNaseA: PDB entry = 7rsa
- Lysozyme: PDB entry = 2vb1
- Xylanase: PDB entry = 2dfc
- Urate oxidase: PDB entry = 3l8w
- Xylose isomerase: PDB entry = 1mnz

They can be downloaded easily using the `download_pdb` function:

```
download_pdb(link, pdb_file)
```

where link = `https://files.rcsb.org/download/entry-code` provides the reference to the PDB file in the database.

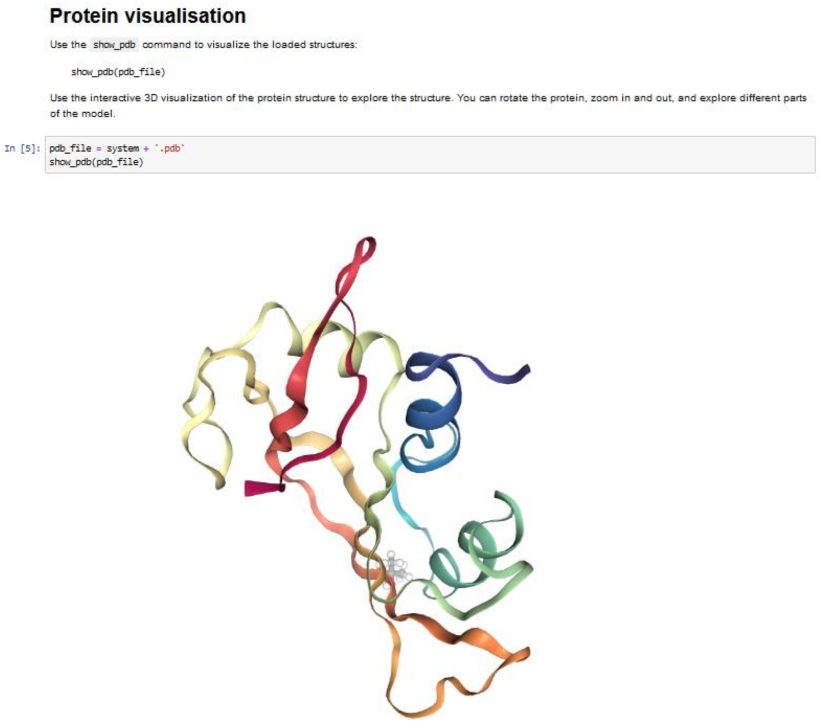
Let's load the reference structure of the five systems analysed in [1]:

```
In [4]: for key in codes.keys():
            url = 'https://files.rcsb.org/download/' + codes[key][0]
            download_pdb(url, key + '.pdb')
```

```
File rnasea.pdb has been successfully downloaded
File lysozyme.pdb has been successfully downloaded
File xylanase.pdb has been successfully downloaded
File urate.pdb has been successfully downloaded
File xylose.pdb has been successfully downloaded
```

The demonstration in a nutshell

STEP 5:
Visualize loaded structure



Protein visualisation

Use the `show_pdb` command to visualize the loaded structures:

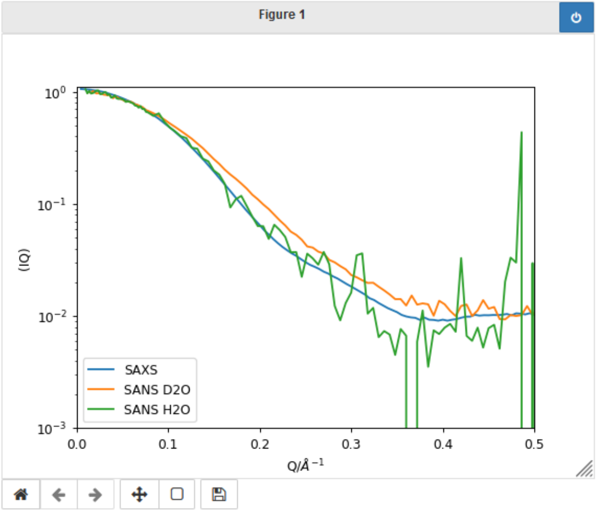
```
show_pdb(pdb_file)
```

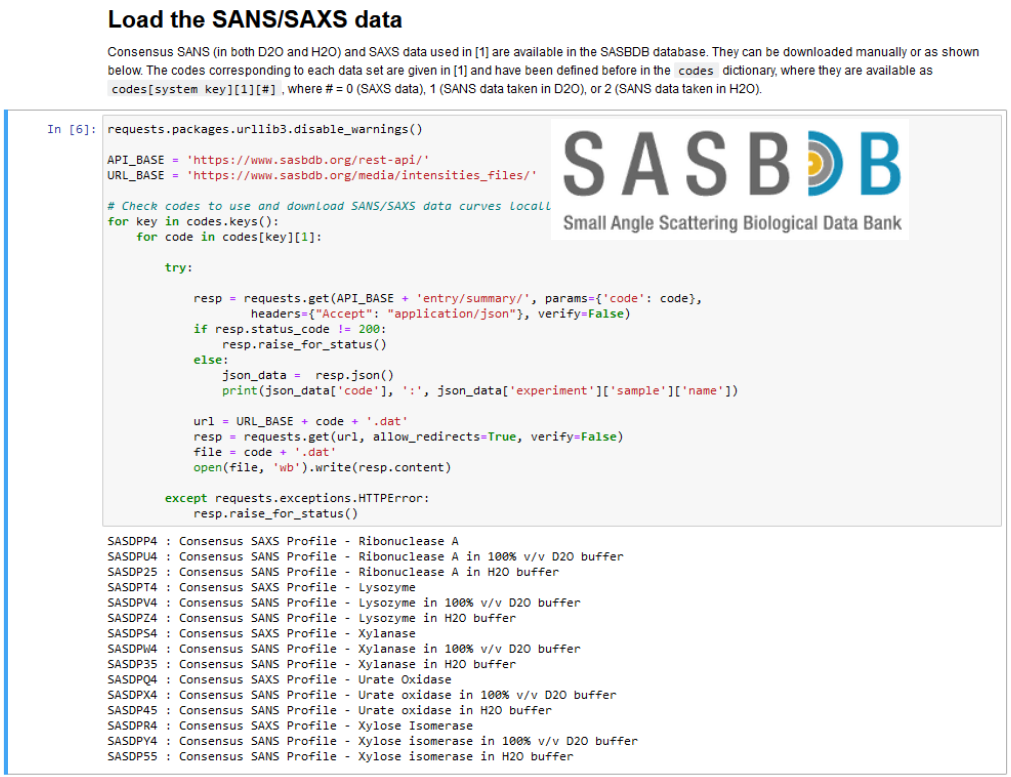
Use the interactive 3D visualization of the protein structure to explore the structure. You can rotate the protein, zoom in and out, and explore different parts of the model.

```
In [5]: pdb_file = system + '.pdb'
        show_pdb(pdb_file)
```

The demonstration in a nutshell

STEP 6:
Load experimental data
and plot them





Load the SANS/SAXS data

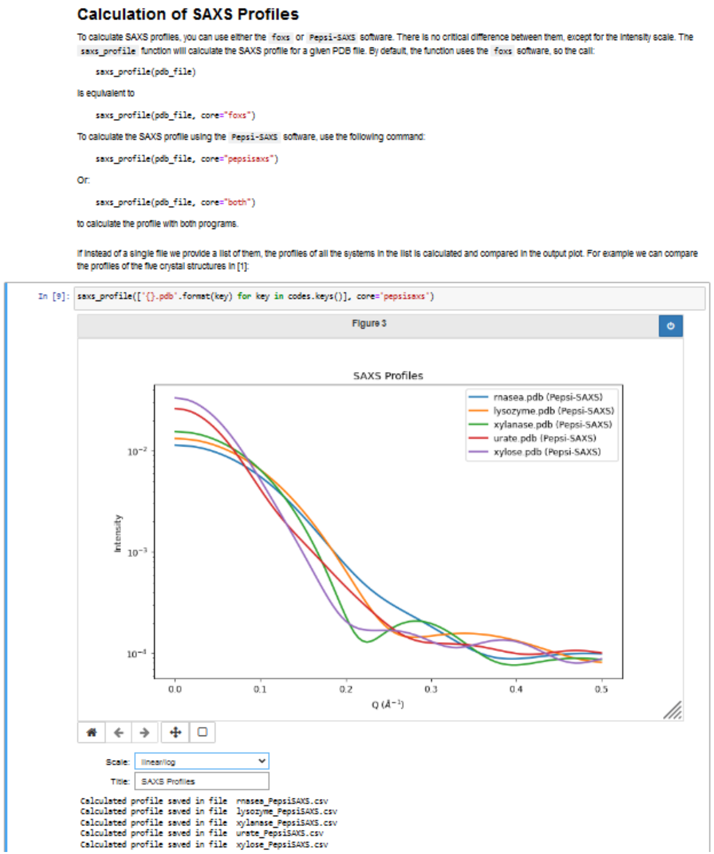
Consensus SANS (in both D2O and H2O) and SAXS data used in [1] are available in the SASBDB database. They can be downloaded manually or as shown below. The codes corresponding to each data set are given in [1] and have been defined before in the `codes` dictionary, where they are available as `codes[system key][1][#]`, where # = 0 (SAXS data), 1 (SANS data taken in D2O), or 2 (SANS data taken in H2O).

```
In [6]: requests.packages.urllib3.disable_warnings()

        API_BASE = 'https://www.sasbdb.org/rest-api/'
        URL_BASE = 'https://www.sasbdb.org/media/intensities_files/'

        # Check codes to use and download SANS/SAXS data curves locall
        for key in codes.keys():
            for code in codes[key][1]:

                try:

                    resp = requests.get(API_BASE + 'entry/summary/', params={'code': code},
                           headers={"Accept": "application/json"}, verify=False)
                    if resp.status_code != 200:
                        resp.raise_for_status()
                    else:
                        json_data =  resp.json()
                        print(json_data['code'], ':', json_data['experiment']['sample']['name'])

                    url = URL_BASE + code + '.dat'
                    resp = requests.get(url, allow_redirects=True, verify=False)
                    file = code + '.dat'
                    open(file, 'wb').write(resp.content)

                except requests.exceptions.HTTPError:
                    resp.raise_for_status()
```

```
SASDPP4 : Consensus SAXS Profile - Ribonuclease A
SASDPU4 : Consensus SANS Profile - Ribonuclease A in 100% v/v D2O buffer
SASDP25 : Consensus SANS Profile - Ribonuclease A in H2O buffer
SASDPT4 : Consensus SAXS Profile - Lysozyme
SASDPV4 : Consensus SANS Profile - Lysozyme in 100% v/v D2O buffer
SASDPZ4 : Consensus SANS Profile - Lysozyme in H2O buffer
SASDPS4 : Consensus SAXS Profile - Xylanase
SASDPW4 : Consensus SANS Profile - Xylanase in 100% v/v D2O buffer
SASDP35 : Consensus SANS Profile - Xylanase in H2O buffer
SASDPQ4 : Consensus SAXS Profile - Urate Oxidase
SASDPX4 : Consensus SANS Profile - Urate oxidase in 100% v/v D2O buffer
SASDP45 : Consensus SANS Profile - Urate oxidase in H2O buffer
SASDPR4 : Consensus SAXS Profile - Xylose Isomerase
SASDPY4 : Consensus SANS Profile - Xylose isomerase in 100% v/v D2O buffer
SASDP55 : Consensus SANS Profile - Xylose isomerase in H2O buffer
```

The demonstration in a nutshell

STEP 7:
Calculate SAXS and SANS profiles.
For SANS check differences between H/D.



Calculation of SAXS Profiles

To calculate SAXS profiles, you can use either the `foxs` or `Pepsi-SAXS` software. There is no critical difference between them, except for the intensity scale. The `saxs_profile` function will calculate the SAXS profile for a given PDB file. By default, the function uses the `foxs` software, so the call:

```
saxs_profile(pdb_file)
```

is equivalent to

```
saxs_profile(pdb_file, core="foxs")
```

To calculate the SAXS profile using the `Pepsi-SAXS` software, use the following command:

```
saxs_profile(pdb_file, core="pepsisaxs")
```

Or:

```
saxs_profile(pdb_file, core="both")
```

to calculate the profile with both programs.

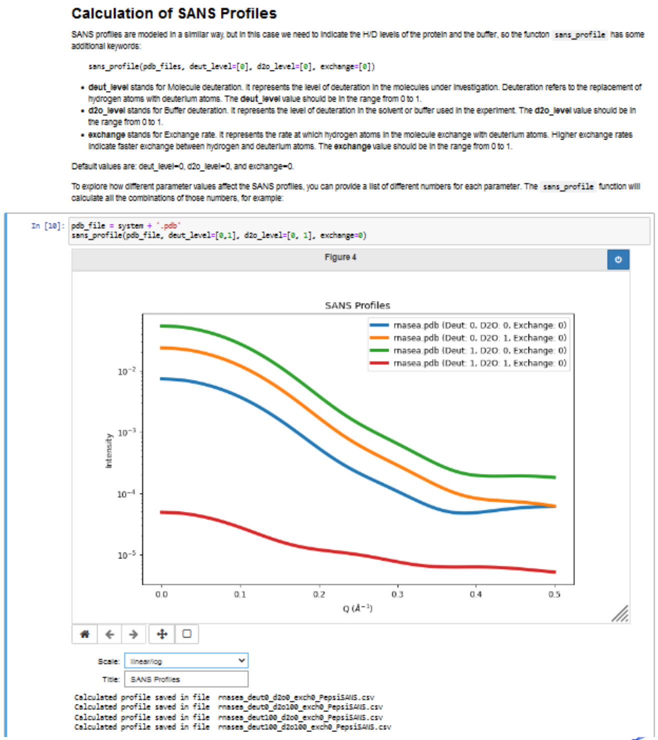
If instead of a single file we provide a list of them, the profiles of all the systems in the list is calculated and compared in the output plot. For example we can compare the profiles of the five crystal structures in [1]:

```
In [9]: saxs_profile(['{}.pdb'.format(key) for key in codes.keys()], core='pepsisaxs')
```

Scale: linear/log

Title: SAXS Profiles

```
Calculated profile saved in file  rnasea_PepsiSAXS.csv
Calculated profile saved in file  lysozyme_PepsiSAXS.csv
Calculated profile saved in file  xylanase_PepsiSAXS.csv
Calculated profile saved in file  urate_PepsiSAXS.csv
Calculated profile saved in file  xylose_PepsiSAXS.csv
```



Calculation of SANS Profiles

SANS profiles are modeled in a similar way, but in this case we need to indicate the H/D levels of the protein and the buffer, so the functon `sans_profile` has some additional keywords:

```
sans_profile(pdb_files, deut_level=[0], d2o_level=[0], exchange=[0])
```

- **deut_level** stands for Molecule deuteration. It represents the level of deuteration in the molecules under investigation. Deuteration refers to the replacement of hydrogen atoms with deuterium atoms. The **deut_level** value should be in the range from 0 to 1.
- **d2o_level** stands for Buffer deuteration. It represents the level of deuteration in the solvent or buffer used in the experiment. The **d2o_level** value should be in the range from 0 to 1.
- **exchange** stands for Exchange rate. It represents the rate at which hydrogen atoms in the molecule exchange with deuterium atoms. Higher exchange rates indicate faster exchange between hydrogen and deuterium atoms. The **exchange** value should be in the range from 0 to 1.

Default values are: deut_level=0, d2o_level=0, and exchange=0.

To explore how different parameter values affect the SANS profiles, you can provide a list of different numbers for each parameter. The `sans_profile` function will calculate all the combinations of those numbers, for example:

```
In [10]: pdb_file = system + '.pdb'
         sans_profile(pdb_file, deut_level=[0,1], d2o_level=[0, 1], exchange=0)
```

Figure 4
SANS Profiles
rnasea.pdb (Deut: 0, D2O: 0, Exchange: 0)
rnasea.pdb (Deut: 0, D2O: 1, Exchange: 0)
rnasea.pdb (Deut: 1, D2O: 0, Exchange: 0)
rnasea.pdb (Deut: 1, D2O: 1, Exchange: 0)
Intensity
10^{-2}
10^{-3}
10^{-4}
10^{-5}
0.0
0.1
0.2
0.3
0.4
0.5
Q ($Å^{-1}$)

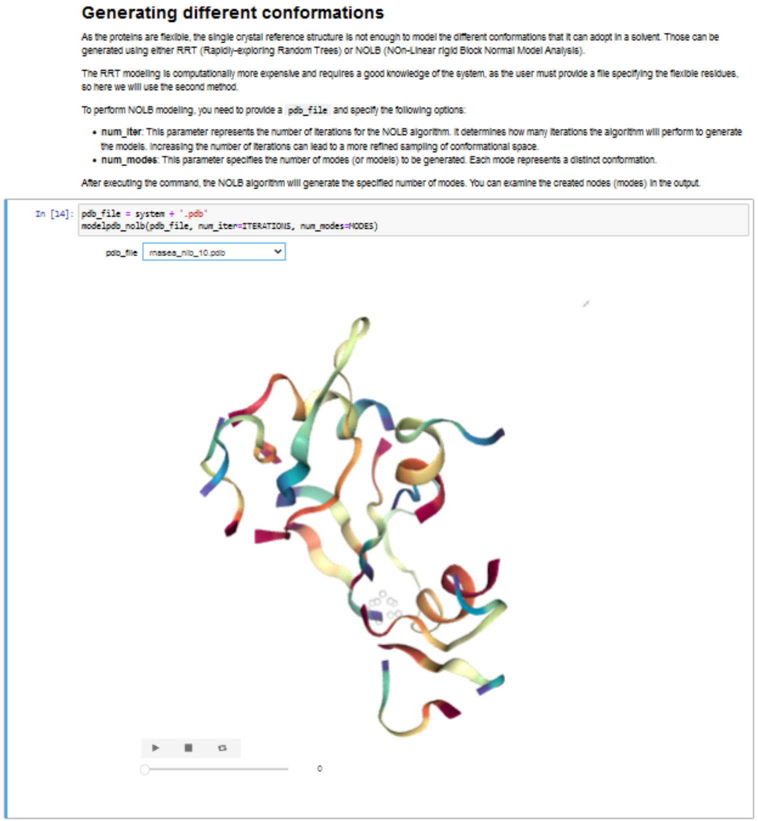
Scale: linear/log

Title: SANS Profiles

```
Calculated profile saved in file  rnasea_deut0_d2o0_exch0_PepsiSANS.csv
Calculated profile saved in file  rnasea_deut0_d2o100_exch0_PepsiSANS.csv
Calculated profile saved in file  rnasea_deut100_d2o0_exch0_PepsiSANS.csv
Calculated profile saved in file  rnasea_deut100_d2o100_exch0_PepsiSANS.csv
```

The demonstration in a nutshell

STEP 8:
Generate alternative conformations from the reference structure by moving smartly the flexible residues of the protein



Generating different conformations

As the proteins are flexible, the single crystal reference structure is not enough to model the different conformations that it can adopt in a solvent. Those can be generated using either RRT (Rapidly-exploring Random Trees) or NOLB (NOn-Linear rigid Block Normal Model Analysis).

The RRT modeling is computationally more expensive and requires a good knowledge of the system, as the user must provide a file specifying the flexible residues, so here we will use the second method.

To perform NOLB modeling, you need to provide a `pdb_file` and specify the following options:

- **num_iter**: This parameter represents the number of iterations for the NOLB algorithm. It determines how many iterations the algorithm will perform to generate the models. Increasing the number of iterations can lead to a more refined sampling of conformational space.
- **num_modes**: This parameter specifies the number of modes (or models) to be generated. Each mode represents a distinct conformation.

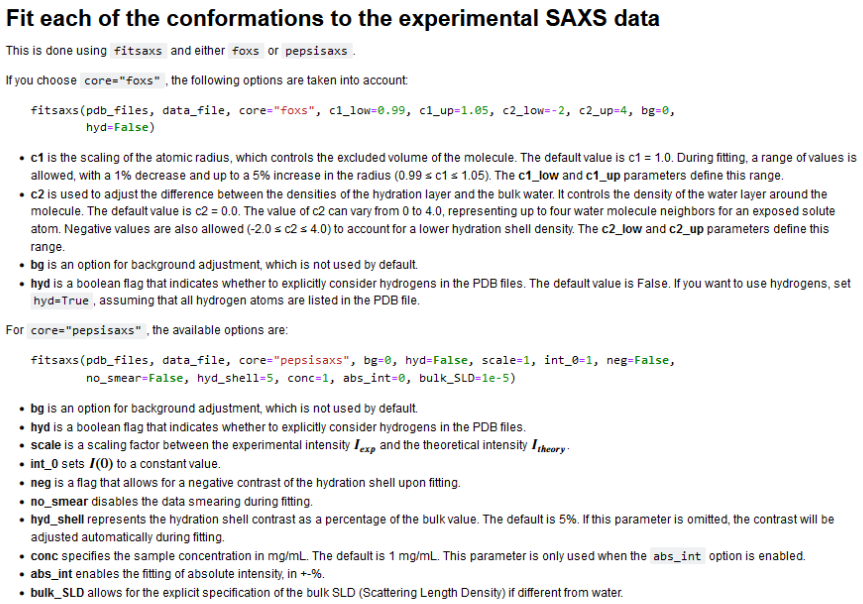
After executing the command, the NOLB algorithm will generate the specified number of modes. You can examine the created nodes (modes) in the output.

In [14]:
```
pdb_file = system + '.pdb'
modelpdb_nolb(pdb_file, num_iter=ITERATIONS, num_modes=MODES)
```

pdb_file: masea_nlb_10.pdb

The demonstration in a nutshell

STEP 9:
Fit calculated profiles to SAXS and SANS data, in order to select more likely/abundant conformations.



Fit each of the conformations to the experimental SAXS data

This is done using `fitsaxs` and either `foxs` or `pepsisaxs` .

If you choose `core="foxs"` , the following options are taken into account:

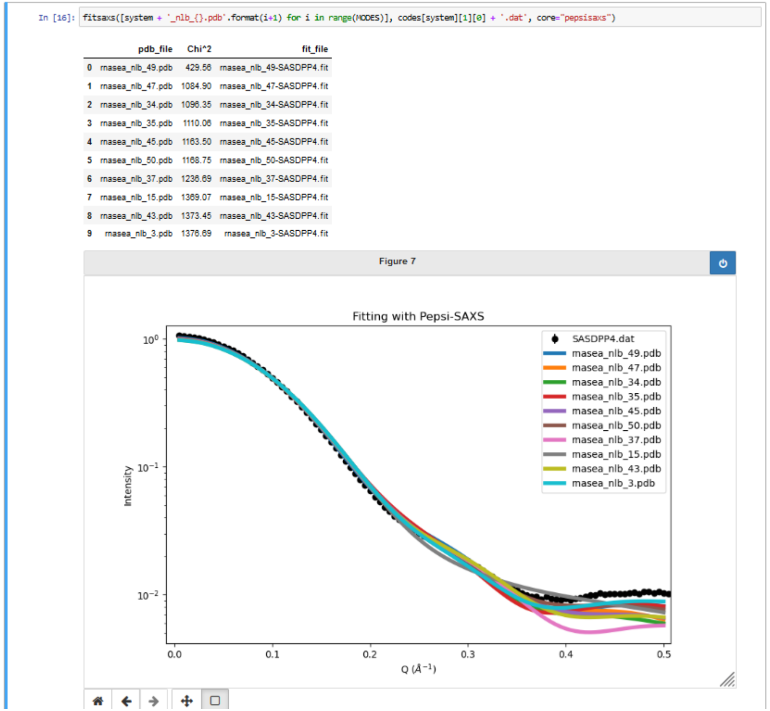
```
fitsaxs(pdb_files, data_file, core="foxs", c1_low=0.99, c1_up=1.05, c2_low=-2, c2_up=4, bg=0,
        hyd=False)
```

- **c1** is the scaling of the atomic radius, which controls the excluded volume of the molecule. The default value is c1 = 1.0. During fitting, a range of values is allowed, with a 1% decrease and up to a 5% increase in the radius (0.99 ≤ c1 ≤ 1.05). The **c1_low** and **c1_up** parameters define this range.
- **c2** is used to adjust the difference between the densities of the hydration layer and the bulk water. It controls the density of the water layer around the molecule. The default value is c2 = 0.0. The value of c2 can vary from 0 to 4.0, representing up to four water molecule neighbors for an exposed solute atom. Negative values are also allowed (-2.0 ≤ c2 ≤ 4.0) to account for a lower hydration shell density. The **c2_low** and **c2_up** parameters define this range.
- **bg** is an option for background adjustment, which is not used by default.
- **hyd** is a boolean flag that indicates whether to explicitly consider hydrogens in the PDB files. The default value is False. If you want to use hydrogens, set `hyd=True` , assuming that all hydrogen atoms are listed in the PDB file.

For `core="pepsisaxs"` , the available options are:

```
fitsaxs(pdb_files, data_file, core="pepsisaxs", bg=0, hyd=False, scale=1, int_0=1, neg=False,
        no_smear=False, hyd_shell=5, conc=1, abs_int=0, bulk_SLD=1e-5)
```

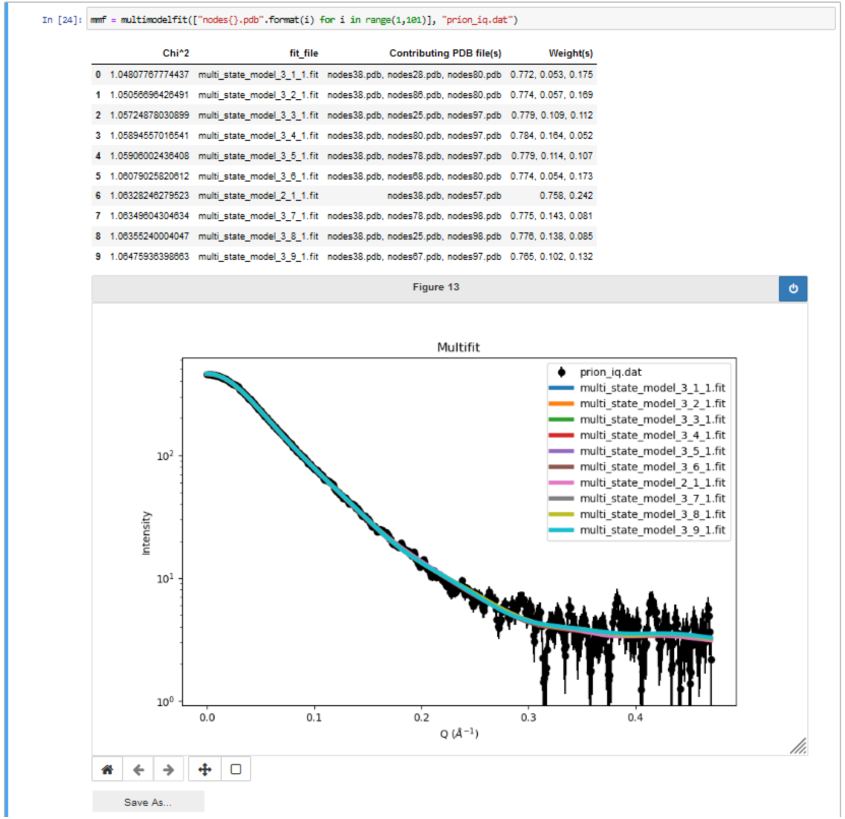
- **bg** is an option for background adjustment, which is not used by default.
- **hyd** is a boolean flag that indicates whether to explicitly consider hydrogens in the PDB files.
- **scale** is a scaling factor between the experimental intensity I_{exp} and the theoretical intensity I_{theory}.
- **int_0** sets $I(0)$ to a constant value.
- **neg** is a flag that allows for a negative contrast of the hydration shell upon fitting.
- **no_smear** disables the data smearing during fitting.
- **hyd_shell** represents the hydration shell contrast as a percentage of the bulk value. The default is 5%. If this parameter is omitted, the contrast will be adjusted automatically during fitting.
- **conc** specifies the sample concentration in mg/mL. The default is 1 mg/mL. This parameter is only used when the `abs_int` option is enabled.
- **abs_int** enables the fitting of absolute intensity, in +-%.
- **bulk_SLD** allows for the explicit specification of the bulk SLD (Scattering Length Density) if different from water.



```
In [16]: fitsaxs([system + '_nlb_{}.pdb'.format(i+1) for i in range(MODES)], codes[system][1][0] + '.dat', core="pepsisaxs")
```

	pdb_file	Chi^2	fit_file
0	rnasea_nlb_49.pdb	429.56	rnasea_nlb_49-SASDPP4.fit
1	rnasea_nlb_47.pdb	1084.90	rnasea_nlb_47-SASDPP4.fit
2	rnasea_nlb_34.pdb	1096.35	rnasea_nlb_34-SASDPP4.fit
3	rnasea_nlb_35.pdb	1110.06	rnasea_nlb_35-SASDPP4.fit
4	rnasea_nlb_45.pdb	1163.50	rnasea_nlb_45-SASDPP4.fit
5	rnasea_nlb_50.pdb	1168.75	rnasea_nlb_50-SASDPP4.fit
6	rnasea_nlb_37.pdb	1236.69	rnasea_nlb_37-SASDPP4.fit
7	rnasea_nlb_15.pdb	1369.07	rnasea_nlb_15-SASDPP4.fit
8	rnasea_nlb_43.pdb	1373.45	rnasea_nlb_43-SASDPP4.fit
9	rnasea_nlb_3.pdb	1376.69	rnasea_nlb_3-SASDPP4.fit

The demonstration in a nutshell



```
In [24]: mmf = multimodelfit(["nodes{}.pdb".format(i) for i in range(1,101)], "prion_iq.dat")
```

	Chi^2	fit_file	Contributing PDB file(s)	Weight(s)
0	1.04807767774437	multi_state_model_3_1_1.fit	nodes38.pdb, nodes28.pdb, nodes80.pdb	0.772, 0.053, 0.175
1	1.05056696426491	multi_state_model_3_2_1.fit	nodes38.pdb, nodes86.pdb, nodes80.pdb	0.774, 0.057, 0.169
2	1.05724878030899	multi_state_model_3_3_1.fit	nodes38.pdb, nodes25.pdb, nodes97.pdb	0.779, 0.109, 0.112
3	1.05894557016541	multi_state_model_3_4_1.fit	nodes38.pdb, nodes80.pdb, nodes97.pdb	0.784, 0.164, 0.052
4	1.05906002436408	multi_state_model_3_5_1.fit	nodes38.pdb, nodes78.pdb, nodes97.pdb	0.779, 0.114, 0.107
5	1.06079025820612	multi_state_model_3_6_1.fit	nodes38.pdb, nodes68.pdb, nodes80.pdb	0.774, 0.054, 0.173
6	1.06328246279523	multi_state_model_2_1_1.fit	nodes38.pdb, nodes57.pdb	0.758, 0.242
7	1.06349604304634	multi_state_model_3_7_1.fit	nodes38.pdb, nodes78.pdb, nodes98.pdb	0.775, 0.143, 0.081
8	1.06355240004047	multi_state_model_3_8_1.fit	nodes38.pdb, nodes25.pdb, nodes98.pdb	0.776, 0.138, 0.085
9	1.06475936398663	multi_state_model_3_9_1.fit	nodes38.pdb, nodes67.pdb, nodes97.pdb	0.765, 0.102, 0.132

Figure 13
Multifit
prion_iq.dat
multi_state_model_3_1_1.fit
multi_state_model_3_2_1.fit
multi_state_model_3_3_1.fit
multi_state_model_3_4_1.fit
multi_state_model_3_5_1.fit
multi_state_model_3_6_1.fit
multi_state_model_2_1_1.fit
multi_state_model_3_7_1.fit
multi_state_model_3_8_1.fit
multi_state_model_3_9_1.fit
Intensity
10^2
10^1
10^0
0.0
0.1
0.2
0.3
0.4
Q ($Å^{-1}$)
Save As...

STEP 10:

Multi model fit using combinations of most likely conformations.

Get comprehensive output

Interactive plotting

Save the result

EOSC Future and PaNOSC: Conclusions

- Photon and Neutron facilities are committed to FAIR principles and open data.
- Several facilities (but not all) have already on-boarded their data repositories and even provide infrastructures to work with the data. But at present they all require to be a registered user of the facility to have access.
- Big challenges due to the diversity of science done in PaN facilities and the “user facility” status (work mainly on samples provided by users, and whole knowledge of the sample is only available to the external group, not to the facility).
- Need to find ways to open data/resources to “non-users” and improve FAIRness of stored data (electronic logbooks, incentives to users to improve sample description, etc.)
- Proof-of-concept for a well defined case. Hopefully extensible to other communities: powder diffraction, spectroscopy, materials science, etc.

THANK YOU!