pyPr 1.0 Documentation 27-July-2024

Revised 9-Nov-2024

Overview

The pyPr program calculates a pair distance distribution, P(r), for protein samples from x-ray solution scattering data, I(q) using a direct transform. Various techniques are used to alleviate artifacts that might occur due to the limited range of intensity measurements.

Tests performed with synthetic data obtained from known P(r) functions, and with resolution limits typical of SAXS data, showed that the methodology could accurately reproduce the correct P(r). Tests performed with experimental data showed that the method typically results in a P(r) very similar to that produced by calculations with the indirect transform program, GNOM.

The pyPr distribution includes the pyPr program (*pyPr_v#.py*), a GNU GPLv3 license (*COPYING.txt*), and an example input file (*example_pyPr_ip.txt*). The most current version of this documentation and a summary of the terms and conditions for use of the program are available from internet site **http://saxs2shapes.com**.

How to run pyPr

The pyPr program is currently distributed as single file, *pyPr_v#.py*, as Python script. The program was developed and tested on the Windows 11 operating system using python 3.12 but should execute on any computer system on which a python interpreter is installed. The pyPr program may be launched via a double-click on the program icon (from the Windows operating system) or from the command line. For example,

'your-path-to-python-interpreter' ‘your path to the *pyPr_v#.py* program’

The operation of pyPr is controlled by a keyword file, for example, *example_pyPr_ip.txt.* The user is prompted for the name of this file after executing pyPr. A full path may be given to this file if it is not located in the same folder as the pyPr program. The pyPr input parser reads lines from this file that contain a keyword as the first item and an associated parameter as the second item.

In addition to the terminal display, statistical information on the run is captured in an output file, ‘*id’_results.txt*, where *id* is an identification parameter provided in the keyworded input file as the **id** parameter. Run times for pyPr operation are typically fractions of a second and the terminal display then lingers for a few seconds before closing.

Keyword Input

Only three parameters, specified by the **id, data_file** and **rg** keywords are required to run pyPr. All other parameters will be set to default values if they are not provided.

All keywords are followed by a single parameter.

Possible keywords are:

id A prefix added to all output file names. This option may be useful for distinguishing sets of output files resulting from different runs carried out within the same folder.

data_file The name of the file containing intensity data. This file should be in the same location as the parameter file so no path should be given. The data file parser assumes three columns of numbers corresponding to q, I and sd(I). The units of q may either be Å^{-1} or nm^{-1} and the program will automatically convert to Å^{-1} if necessary. Files generated by GNOM and CRYSOL will also be recognized, and intensities extracted from them. The CRYSOL option can be useful for generating a model P(r) to compare with an experimentally determined P(r).

rg An estimate for the radius of gyration, Rg, in Å units. The values of Rg and I0 are recalculated within the program and this estimate is used to set the resolution limit of the Guinier region for this recalculation.

dmax Option to set the value of dmax in Å units. If not provided, the program will make and apply an estimate of the minimum likely value of dmax. The program will ensure that the P(r) tapers to zero at dmax.

qmin Option to set the minimum resolution of acceptable data. Units are the same as for the input data. This option may be useful if a Guinier analysis shows irregularities at very low resolution. By default, all low q data are included.

qmax Option to set the maximum resolution of acceptable data. Units are the same as for the input data. This option may be useful if data extends beyond the limit of acceptable random noise. By default, all high q data are included.

pr_step Option to set the interval with which P(r) will be calculated and output. Units are Å and the default value is 1Å.

ref_pr_file Option to specify the name of a previously calculated P(r) file to compare with the result of the current calculation. This file may be obtained from a previous pyPr run or GNOM run. This file is expected to be found in the same location as the parameter file so no path should be given. The file parser assumes three columns of numbers corresponding to r, P(r) and sd(P(r)). The units of r may either be Å or nm and the program will automatically convert them to Å.

qmax_guinier Option to set the maximum q value for data in the Guinier zone. By default (if thus keyword is not given), a value of 1.3/Rg is used based on the input value for Rg. This option might be occasionally useful for highly elongated proteins where a value closer to 1.0/Rg might be more appropriate and linearity of the Guinier zone breaks down. The units of this parameter are Å^{-1}.

Outputs and recommended use

pyPr is intended to provide a reliable calculation of P(r) by the direct transform method with results of similar quality to the most widely used indirect transform programs. The program should simplify the comparison of P(r) functions obtained from different data sets, perhaps from proteins in the presence of different ligands. Another application is to compare the experimental P(r) with the calculation of P(r) from an atomic model. In this case the calculation would typically use synthetic data calculated from atomic coordinates with CRYSOL with data limited to the experimental limit.

The following output files are placed in the same location as the input parameter file and may be identified via the **id** parameter. Files containing I or P(r) information may be viewed using PRIMUS or any graph plotting software.

‘id’_start_pr.dat The ‘raw’ P(r) function plotted to 4.5xRg without any cutoff or adjustment. This file contains r, P(r) and sd(P(r)) and is useful for visually assessing the quality of the result and possible value of d_{max}.

‘id '_final_pr.dat The P(r) function smoothly truncated to zero at d_{max}. This file contains r, P(r) and sd (P(r)).

‘id '_filter_pr.dat A P(r) function that was filtered to remove noise and structural detail. This file contains r, P(r) and sd(P(r)). It is used by the pyPr program to infer a value for d_{max}.

‘id '_i.int A compilation of the input data and scaled back-transform of the P(r) after truncation at d_{max}. This file contains q, I and sd(I).

‘id '_i_and_pr.out A file containing data fields for both the input intensities and final P(r) that mimics some aspects of the output file from the GNOM program. This file is read as if it is a GNOM output file by the plotting program PRIMUS. It may also be used as an input to the 3D reconstruction program SHAPES. It is not read by the reconstruction programs DAMMIN and GASBOR.

‘id'_with_ref_pr.int A file containing P(r) calculated in the pyPr run and a reference P(r), if given. This file contains columns for r, experimental P(r), and reference P(r). (The extension *.int* is used so as to facilitate convenient viewing in the PRIMUS program by setting the plot option to absolute scale.)

‘id '_results.txt A text file capturing statistical information from the pyPr run.