DESY summerstudent programme 2008

Machine learning in determination of the radius of gyration

Tim Bommersbach*

Supervisor: Daniel Franke

23/7-16/9/08

*tim.bommersbach@gmx.de

Contents

1 Introduction **3**

2 Theoretical background - Small-angle X-ray scattering **3**
2.1 Definition[1] 3
2.2 Radius of gyration[2] 3

3 My project **5**
3.1 Sample creation 5
3.2 The algorithm 6
3.3 Test runs and results 6

4 Literature **9**

1 Introduction

I worked in the SAXS group of EMBL lead by Dmitri Svergun. My task was to develop a self learning algorithm which calculates the radius of gyration from a scattering curve. Therefor I used the protein databank which includes the perfect simulated scattering data of about 20000 different proteins.

2 Theoretical background - Small-angle X-ray scattering

2.1 Definition[1]

Small-angle X-ray scattering (SAXS) is a small-angle scattering (SAS) technique to study samples with elastic scattering of X-rays (wavelength 0,1 to 0,2 nm). The angular range of the scattered radiation contains information about the size, the shape and characteristic parameters of macromolecules. The structural resolution of SAXS is between 5 and 25 nm, so that the atomic structure of a macromolecule or a protein cannot be studied . A great advantage of SAXS over crystallography for the study of biological material is that the sample does not need to have a crystalline structure.

2.2 Radius of gyration[2]

An example for a real scattering curve is illustrated in figure 1 where I is the scattered intensity and $s = \frac{4\pi \sin(\theta)}{\lambda}$ is a function from the scattering angle 2θ.

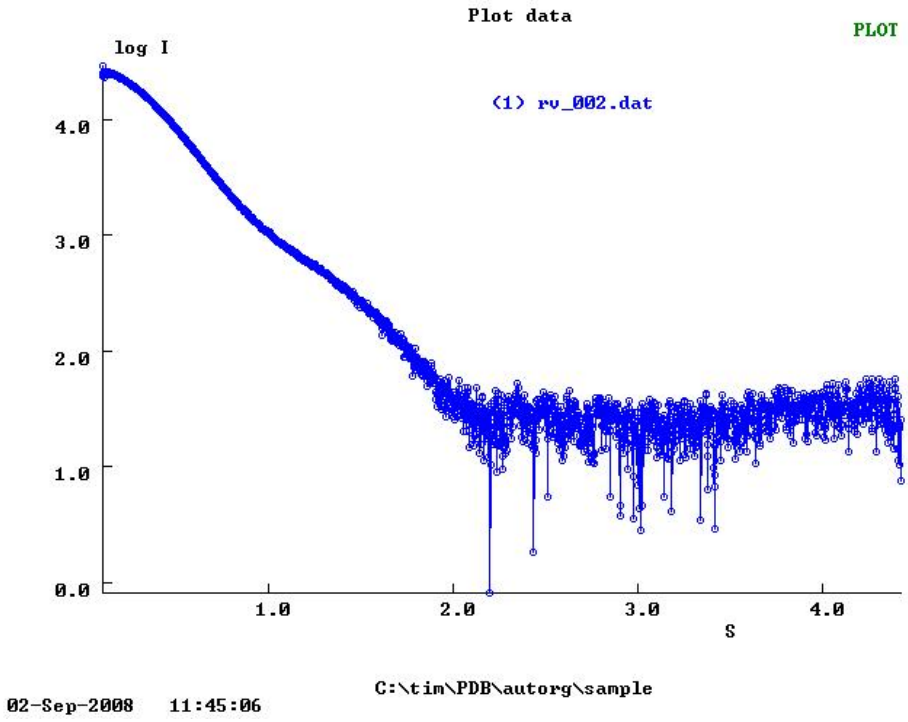

Figure 1: Example for a real scattering curve

In Guinier Approximation the scattered intensity I can be calculated by the expression

$$I(s) = I(0)\exp(-\frac{1}{3}R_g^2 s^2). \tag{2.1}$$

If you use the natural logarithm, you will get the expression

$$\ln(I(s)) = \ln(I(0)) - \frac{1}{3}R_g^2 s^2. \tag{2.2}$$

That means that for small angles the intesity I(s) as a function of s^2 is a straight line $y = mx + n$ with the slope

$$m = -\frac{1}{3}R_g^2. \tag{2.3}$$

You can transpose this expression which leads to the radius of gyration

$$R_g = \sqrt{-3m}. \tag{2.4}$$

3 My project

3.1 Sample creation

First I downloaded all data from the protein databank and ran a program named crysol. This program delivers readable datatables from the non-readable files of the databank, so that I could work with these tables. I ran crysol in a s range upto $s_max = 0,1\frac{1}{nm}$. An example for the produced files is illustrated in figure 2, where you can see a simulated curve in a s range between 0 and $0,1\frac{1}{nm}$.

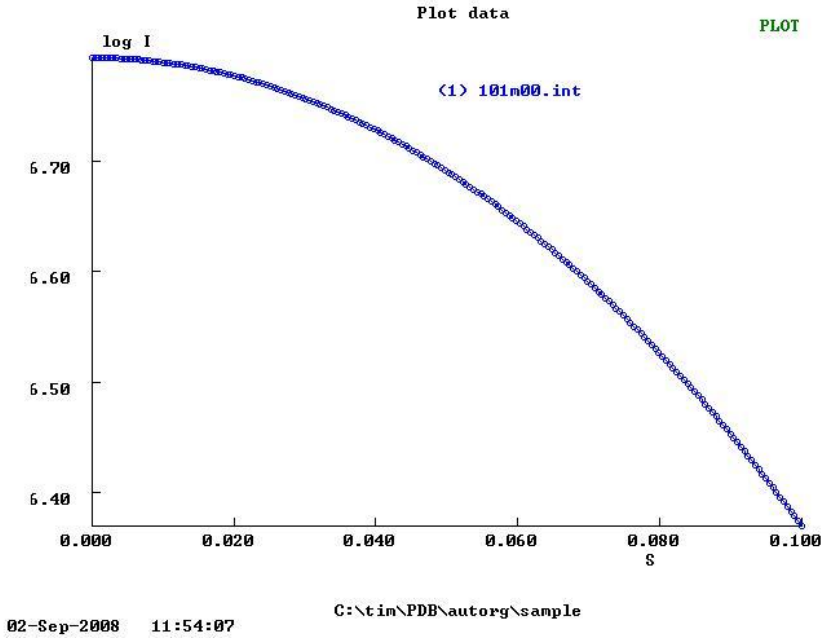

Figure 2: Example for a simulated curve of the protein databank

On the other hand crysol provides the theoretical radius of gyration of the different proteins, therefor you can use the program RGui to build a histogram for the R_g's. The resultant histogram is illustrated in figure 3 where the R_g's are plotted on the x-axis scaled in Angstrom. For my further work i have chosen a sample of hundred proteins in an R_g range from 15 to 25 Angstrom based on the results of the histogram.

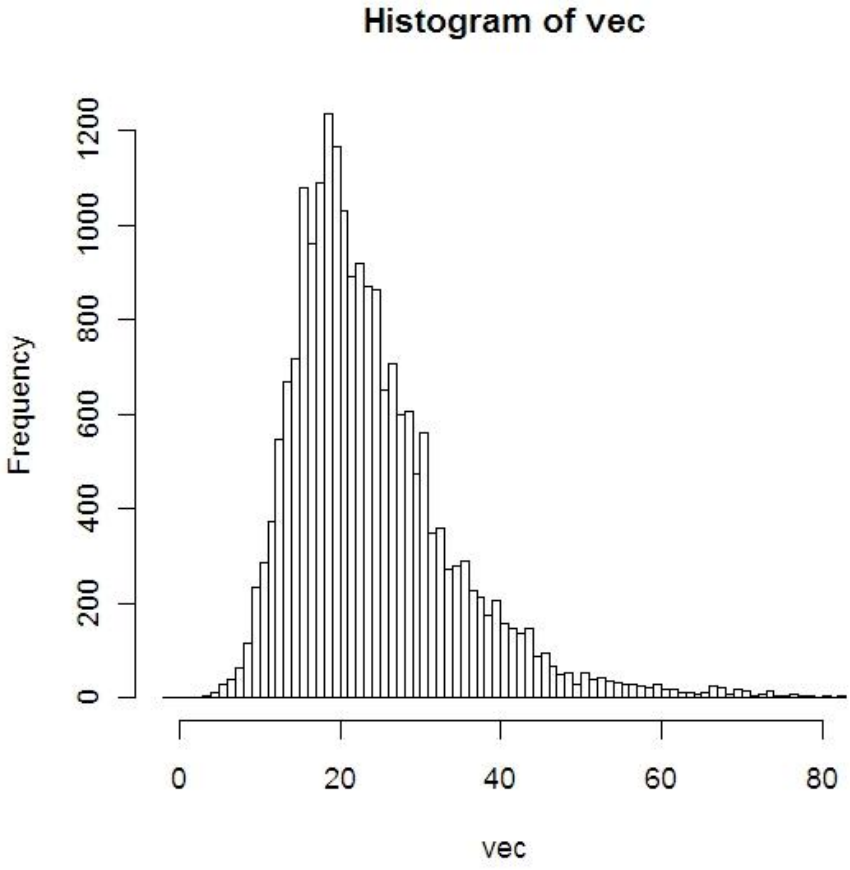

Figure 3: Histogram of the R_g distribution

3.2 The algorithm

In my further work I used the programs Primus and RGui. With Primus I could identify the optimal s range of each data which provides the best linear fit for the R_g calculation.
With RGui I wrote a source code for the self learning algorithm. In the first part it uses a set of training files($\approx$10) with the optimal s range to fit a logistic regression model, in this way it learns from the sample.
In the second part the algorithm requires some test files, I worked with three files, for determining their s ranges with the logistic regression model of the first set of files. After the s ranges have been determined it was possible to calculate the R_g's of the files respectively the proteins.

3.3 Test runs and results

An example for a result is illustrated in the following figure. The line plotted in black stands for the expected s ranges of the three test files identified with Primus, the green line represents

the s ranges calculated by the self learning algorithm.

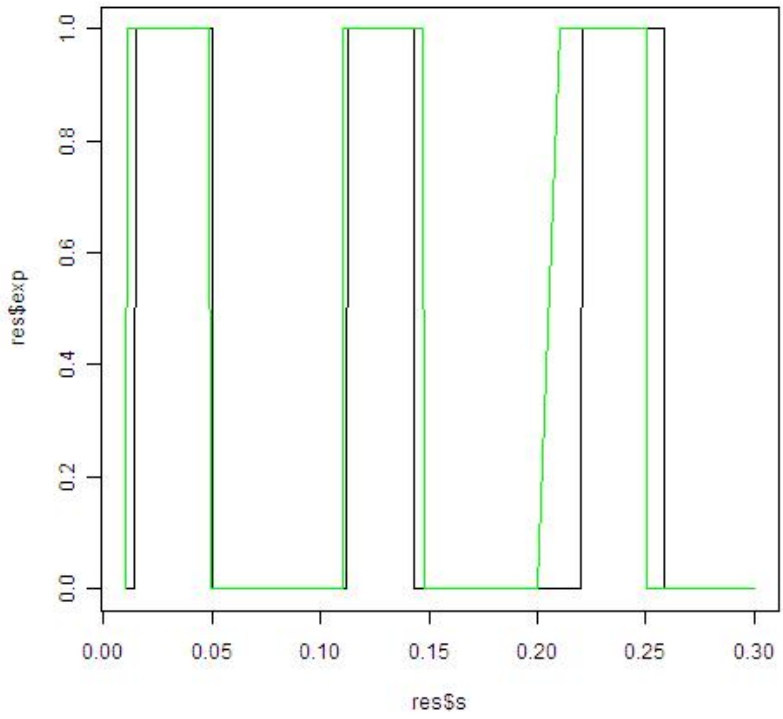

Figure 4: Example for the expected (black) and the calculated (green) s ranges of the three test files (both set of files have perfect intensity

This algorithm should not only work for perfect files of the protein databank, but also for real scattering curves. So I ran it for the perfect files with some artificially induced noise to the intensity of both files. The runs proceeded in many combinations of noise because I used different magnitudes of noise (perfect intensity, low noise, middle noise, high noise).
In the next step I used real data for training (5 files) and test (2 files). Two results for two different sets of test files are presented in the following figures, where you see the predicted probabilty, that this s value belongs to the optimal range for the determination of R_g, on the left. On the right you see the conclusion for the optimal range of the logistic regression model. In the second example I could not identifiy R_g region with Primus for this data. The algorithm approved this suspicion.
After some more test runs I concluded, that the algorithm works well, because it provides good results, but it works slowly in RGui.
The next step would be an implementation of the algorithm in a faster programming language like FORTRAN, but I had not enough time in summer student programme.

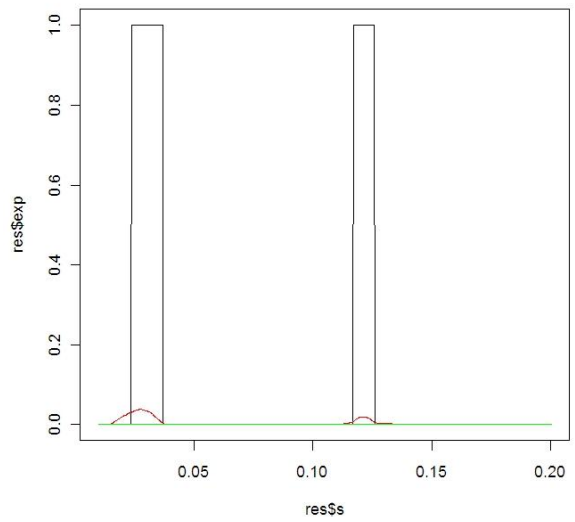

Figure 5: First example: Predicted probabilities (red) by the logistic regression model

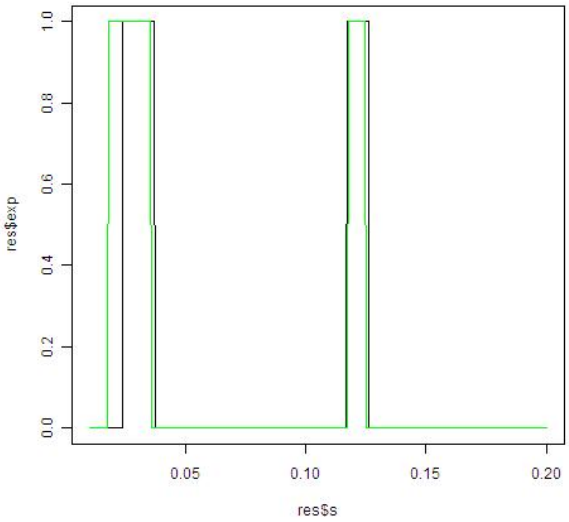

Figure 6: Comparison between the R_g ranges calculated by Primus and the self learning algorithm

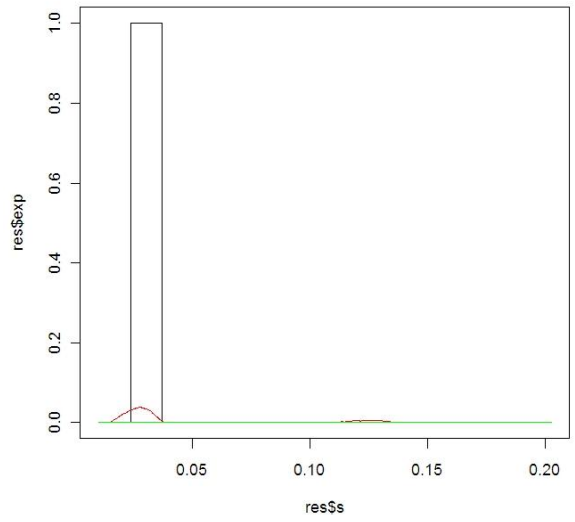

Figure 7: Predicted probabilities (red) by the logistic regression model

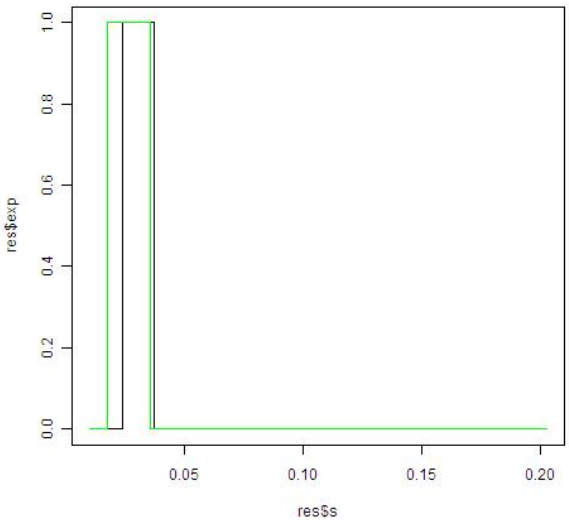

Figure 8: Comparison between the R_g ranges calculated by Primus and the self learning algorithm

4 Literature

References

[1] http://de.wikipedia.org/wiki/Saxs

[2] http://hasylab.desy.de/e77/e18402/e28817/e18403/e35070/Gehrke-SAXS.pdf

September 11, 2008