Dear GROMACS,

My name is Anna Sinelnikova and I am a 3rd year PhD student in the department of Physics and Astronomy at Uppsala University. I am reaching out to you because I am interested in establishing a collaboration with GROMACS since my topic is not mainstream in my own department. Below I give an overview of my work so far and I would be very thankful if you would be interested in discussing possible joint projects. I will tell you about one idea I have.

Physics

As you know the simulation of proteins is a very complicated task because the huge amount of particles involved, and the fact that all particles interact with each other makes the problem scale badly. This means that even on supercomputers you soon run into a limit.

A first step to improve the situation is to optimize the algorithms, and to then optimize the algorithms for a given hardware. Here GROMACS is definitely doing great. Modern programming practice together with large efforts devoted to increasing of performance makes you a leader in this field.

However, as well as computational power, the optimization has its own limits. Another way to improve the situation is to review the physical models you use.

A model where all atoms interact with all other atoms is a very simple, but inefficient physical model. It is not optimal on two sides: many parameters to fit and a lot of forces to simulate. To avoid these problems one can consider *coarse graining* methods, where blocks of particles are taken into account, instead of working with every single particle.

In my PhD I am working on a big polymer project. The crucial part of it is a coarse graining approach for modelling and simulating polypeptide chains. We simulate only C_α-backbones – the skeleton of the protein. The remaining atoms in a polypeptide which are not constructing the backbone appear in our model as just a few parameters of an effective Hamiltonian, but do not participate directly in our simulations.

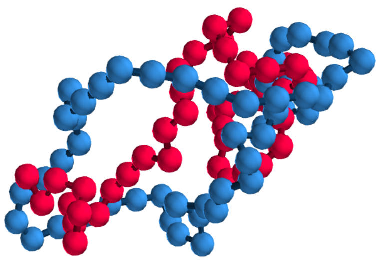
Figure 1: Two chains with interaction simulate and plotted by PCMC project.

Another idea of our method is to separate secondary and tertiary structures of proteins. We have built an effective Hamiltonian, inspired by the Abelian Higgs model with spontaneous symmetry breaking. It provides solitonic solutions which correspond to units of secondary structures, such as α-helices or β-strands. In addition we have long range interaction which allows us to simulate, for example, van der Waals forces. This part is responsible for the tertiary structure of polypeptides. Then we use Markov chain Monte Carlo to obtain the thermodynamical equilibrium states of the described model. By tuning the parameters of both parts we can model different phases of a given chain [1].

As a result we have a big advantage – much less atoms to simulate as well as much fewer parameters in the system compared to a model where all atoms interact with each other [2][3]. Instead of working with every atom we can work with subchains and blocks, where the interaction with the solvent also can be taken into account in the parameters of the model, without creating extra terms in the Hamiltonian.

Programming

In my work I am focused not only on the result of the calculations, but also on the development of tools which can be used and easily updated or extended in the future.

At the moment I have 4 git project (3 of them are in open access on GitHub). All of them share data through my own file format (.pca) for polypeptide backbone chains. The pca-file contains xyz-coordinates of C_α-atoms of one or more chains.

The four projects are:

1. PDB parser in C++ [https://github.com/Anny-Moon/pdb2xyz][1], can create a pca-file of any protein from the rcsb.org database or any pdb-file. This parser is a command line program and further functions such as treatment of repeating atoms and saving in other file formats such as the conventional .xyz-format is described in the man page.

2. Plotter for pca-files in Python [https://github.com/Anny-Moon/PlotterPyPCA]. This program is still under development. It uses MatplotLib or Mayavi (Figure 1) for making interactive plots or videos of protein backbones.

3. Program for Scaling Polymers in C++ [https://github.com/Anny-Moon/PCA]. In the paper [4] we introduced a novel idea for investigating polymer chain phases by using renormalization group transformation. This program is a working code which was used to obtaining all the data for the original paper. It can perform the scaling transformation and plot all supplementary figures using data from any pca-file.

4. PCMC (Polymer Chain Monte Carlo) in C++ and OpenCL (not in open access). This is the main ongoing project which was described in the previous section.

Possible collaboration

By looking in your code and in the paper [5] I see the same philosophy as I have. The idea to write a convenient API which can be simply used both by ordinary users and advanced developers seems very attractive to me. In my opinion the future of developing scientific software should go exactly in this direction. Instead of producing programs for some very specific problem, one should build a convenient tool which can provide solutions for a broad variety of problems or that can become a part of more complex tools.

A potential collaboration would be interesting for me from both bio-physical and programming point of views. I would like to become part of a team and see how a lot of skilled people can join together for producing something really great.

At the same time I hope that I can also bring something useful to the project. Beside physics, I could make input in code. For example my knowledge of OpenCL goes inline with your politics of "resources are scarce". Just imagine how much more efficient GROMACS can be if it can use any graphic cards, not only NVIDIA ones. For example, Intel coprocessors become more and more popular, and one can expect them to become a real competitor to NVIDIA in the nearest future.

Even if CUDA C is enough for supercomputers nowadays, the target should be personal computers and laptops as well. If you can use Intel graphic cards, for example, then you can reach higher perfomance on Mac laptops and desktops. It can make GROMACS the most desirable tool for projects like "Foldit@home".

26 March 2018 Anna Sinelnikova

[1]The prototype of this parser was a pure C program [https://github.com/Anny-Moon/Simple_PDB_parser].

References

[1] A. Sinelnikova, A. J. Niemi, M. Ulybyshev, Phys. Rev. **E92** 032602 (2015)

[2] Xubiao Peng, Adam K. Sieradzan, and Antti J. Niemi Phys. Rev. **E94** 062405 (2016)

[3] Jiaojiao Liu, Jin Dai, Jianfeng He, Antti J. Niemi, and Nevena Ilieva Phys. Rev. **E95** 032406 (2017)

[4] A. Sinelnikova, A. J. Niemi, M. Ulybyshev, Phys. Rev. **E97** 052107 (2018)

[5] M. J. Abraham *et al.*, SoftwareX 1–2 (2015) 19-25

[6] https://github.com/Anny-Moon