## 1 Name, Surname

Dominik Gront

## 2 Held diplomas, scientific / arts degrees - with the name, place and year of acquisition, and the title of doctoral dissertation

| | |
|---|---|
| June 23, 2001 | Master of Science diploma, Faculty of Chemistry, University of Warsaw<br>*„New and classical Monte Carlo methods for studying protein thermodynamics"* |
| April 16, 2006 | Doctoral degree in chemistry, specialization: theoretical chemistry, Faculty of Chemistry, University of Warsaw<br>*„Algorithm for protein structure modeling based on databases of known protein sequences and structures."* |

## 3 Information on current and previous employment in research /art institutions

| | |
|---|---|
| October 2006 - September 2007 | Faculty of Chemistry, University of Warsaw, Warsaw, Poland |
| October 2007 - September 2008 | Molecular Physiology and Biological Physics, University of Virginia, Charlottesville, VA, USA |
| since October 2008 | Faculty of Chemistry, University of Warsaw, Warsaw, Poland |

## 4 Indication of achievement under Art. 16 Paragraph 2 of the Act of 14 March 2003 Academic Degrees and Title, and Degrees and Title in Art(Dz. U. N o 65, item. 595 with amendments)

### 4.1 The title of the scientific achievement

*„Development of novel algorithms for protein modeling and their implementation in BioShell software package"*

## 4.2 Publications comprising the academic achievement

H1. **D. Gront***, S. Kmiecik, A. Kolinski, *„Backbone building from quadrilaterals: A fast and accurate algorithm for protein backbone reconstruction from alpha carbon coordinates."*, J. Comput. Chem. 2007; 28 1593-1597

H2. **D. Gront***, A. Kolinski, *„Efficient scheme for optimization of parallel tempering Monte Carlo method."*, J. Phys.: Condens. Matter, 2007; 19 036225

H3. A. Sikorski* and **D. Gront**, *„Thermodynamic properties of polypeptide chains. Parallel tempering Monte Carlo simulations"*, Acta Physica Polonica B, 2007 38 1899-1908

H4. **D. Gront***, A. Kolinski, *„T-Pile - a package for thermodynamic calculations of biomolecules."*, Bioinformatics; 2007; 23: 1840 - 1842.

H5. **D. Gront*** and A. Kolinski, *„Comparative modeling without implicit sequence alignments"*, Bioinformatics, 2007; 23 2522

H6. **D. Gront***, A. Kolinski, *„Utility library for structural bioinformatics"*, Bioinformatics, 2008; 24 584

H7. **D. Gront***, A. Kolinski, *„A fast and accurate methods for predicting short-range constraints in protein models"*, J. Comput. Aided Mol. Des, 2008 *DOI 10.1007/s10822-008-9213-8*

H8. P. Gniewek, A. Kolinski, **D. Gront**, *„Optimization of profile-to-profile alignment parameters for one-dimensional threading"*, J.Comput. Biol., 2012 19:879-886, doi: 10.1089/cmb.2011.0307

H9. **D. Gront***, P. Wojciechowski, M. Blaszczyk, A. Kolinski, *„Bioshell Threader: protein homology detection based on sequence profiles and secondary structure profiles"*, Nucl. Acid Res., 2012, doi: 10.1093/nar/gks555

H10. **D. Gront***, S. Kmiecik, M. Blaszczyk, D. Ekonomiuk, and A. Kolinski, *„Optimization of protein models"*, WIREs Comput Mol Sci, 2012; 2 479-493 doi: 10.1002/wcms.1090

H11. P. Gniewek, A. Kolinski, A. Kloczkowski, **D. Gront***, *„BioShell-Threading: versatile Monte Carlo package for protein threading"* BMC Bioinformatics 2014 15:22

H12. L. Wieteska*, M. Ionov, J. Szemraj, A. Kolinski, C. Feller, **D. Gront*** *„Improving thermal stability of thermophilic L-threonine aldolase from Thermatoga maritima"* J. of Biotechnology, 2015 199:69-76, doi: 10.1016/j.jbiotec.2015.02.013

## 4.3 Discussion of the scientific / artistic goals of the above publication / publications and the results achieved together with a discussion of their possible use

### Introduction

Progress in science and technological development are inextricably linked. Basic sciences create novel experimental tools, which in turn open new research areas. The first electronic computers, built in the fifties of the previous century, contributed to an intensive development of many scientific fields and gave rise to several new ones and, most of all, informatics. The foundations of informatics were laid both by theoretical works and practical solutions such as the Von Neumann architecture[1] and the first programming languages[2]. The two founding methods of molecular modeling, namely, molecular dynamics[3] (as a discrete-type modeling at first) and the Metropolis scheme[4] were published in the fifties of the XX century.

In the same years there was a constant progress in molecular biology. The genetic code has been deciphered[5], the structures of DNA[5] and myoglobin[6] have been revealed. The basic principles of protein structure have been described[7–9]. Protein sequencing became automated and the first protein sequence database was published, in a book form, by Margaret Dayhoff[10].

These studies were facilitated to some extent by the use of the first computers. They were employed, for example, to solve the X-ray protein structure and to assemble the sequencing results of DNA fragments[11]. As more and more protein sequences became available, scientific papers discussing gene and protein evolution started to appear. The first phylogenetic tree was published in 1967[12]. At last, in 1970, Needleman and Wunsch[13] published the global alignment algorithm. This gave birth to a new scientific discipline - bioinformatics, although the term itself came into being almost ten years later[14].

In the following decades, the directions of bioinformatics development were deter-

[1] von Neumann, J. Tech. rep. (Philadelphia, PA, USA, 1945), 1–43.
[2] Backus, J. W. *et al.* in *Papers Presented at the February 26-28, 1957, Western Joint Computer Conference: Techniques for Reliability* (Los Angeles, California, 1957), 188–198.
[3] Alder, B. J. & Wainwright, T. E. *The Journal of Chemical Physics* **27,** 1208–1209 (1957).
[4] Metropolis, N. *et al. The Journal of Chemical Physics* **21,** 1087–1092 (1953).
[5] Gamow, G. *et al. Advances in biological and medical physics* **4,** 23–68 (1956).
[6] Kendrew, J. C. *et al. Nature* **181,** 662–666 (1958).
[7] Pauling, L. & Corey, R. B. *Proceedings of the National Academy of Sciences* **37,** 251–256 (1951).
[8] Pauling, L. *et al. Proceedings of the National Academy of Sciences* **37,** 205–211 (1951).
[9] Ramachandran, G. N. *et al. Journal of molecular biology* **7,** 95–99 (1963).
[10] Dayhoff, M. O. (Silver Spring, Md., 1965).
[11] Dayhoff, M. O. & Ledley, R. S. in *Proceedings of the December 4-6, 1962, Fall Joint Computer Conference* (Philadelphia, Pennsylvania, 1962), 262–274.
[12] Fitch, W. M. & Margoliash, E. *Science (New York, N.Y.)* **155,** 279–284 (1967).
[13] Needleman, S. B. & Wunsch, C. D. *Journal of molecular biology* **48,** 443–453 (1970).
[14] Hogeweg, P. *PLoS Comput Biol* **7,** e1002021+ (2011).

mined by an increasing number of data in the form of sequences and biomolecular structures. Data collection and analysis became the major goal of bioinformatics. To achieve this, appropriate software had to be designed. At the same time the software for modeling the structure and dynamics of biomolecules started to be developed. At first, both areas were clearly separated. Bioinformatics relied on the acquired datasets while molecular modeling – on the principles of physics. Soon it turned out that conclusions inferred from gene evolution could be as valid as models acquired based on physical principles.

Currently there are a lot of initiatives devoted to creating bioinformatics and molecular modeling software. Among the modeling tools one should mention molecular modeling software packages, the Modeller[15], ICM[16], UNRES[17] and Rosetta[18] programs, and the family of lattice models developed by prof. Koliński's group (e.g., SICHO[19] and CABS[20]). On the other hand, BioPerl[21], BioPhyton[22], BioJava[23] and BioRuby[24] are typical bioinformatics packages. In 2004 these packages[1] provided only a modest set of functions operating on protein structures. Limited as they were, these packages proved indispensable for the Author to efficiently complete the doctoral thesis. The lack of more advanced tools was an impulse that initiated the development of the BioShell software package.

The scientific goal of the series of publications was to create a consistent and complete software, which would help to solve diverse problems in the field of structural bioinformatics and biomolecular structure modeling. The author himself implemented a nearly all of the source code and the resulting computational tools are now used by at least several research laboratories all over the world. The presented publications describe construction of the consecutive versions of the software and description of the most important algorithms implemented. Also included are publications presenting examples of the use of this software.

[1] BioPerl i BioPython only; BioJava i BioRuby have not yet existed at that time

[15] Šali, A. & Blundell, T. L. *Journal of Molecular Biology* **234,** 779–815 (1993).
[16] Abagyan, R. *et al. J. Comput. Chem.* **15,** 488–506 (1994).
[17] Liwo, A. *et al. The Journal of Chemical Physics* **115,** 2323–2347 (2001).
[18] Rohl, C. A. *et al.* in *Numerical Computer Methods, Part D* 66–93 (Department of Biochemistry and Howard Hughes Medical Institute, University of Washington, Seattle, Washington 98195, USA., 2004).
[19] Kolinski, A. & Skolnick, J. *Proteins* **32,** 475–494 (1998).
[20] Kolinski, A. *Acta biochimica Polonica* **51,** 349–371 (2004).
[21] Stajich, J. E. *et al. Genome research* **12,** 1611–1618 (2002).
[22] Chapman, B. & Chang, J. *SIGBIO Newsl.* **20,** 15–19 (2000).
[23] Holland, R. C. G. *et al. Bioinformatics* **24,** 2096–2097 (2008).
[24] Goto, N. *et al. Bioinformatics* **26,** 2617–2619 (2010).

## The BioShell Software

*Version 1.x –programs executed from a Unix command line.Unix*

Design of the first version of the software package, published in 2006[25], was entirely based on the Unix operating system. The software package consisted of several programs controlled by appropriate commands from the command line. In the initial version, `BioShell` facilitated protein dynamics simulations in the reduced `CABS`[20] model. It served to prepare the input files and to analyze the result trajectories. Other modules of the software package served to calculate the potential of mean force on the basis of statistics derived from the already known protein structures. In 2007 the software package comprised the following programs:

`strc` - (**str**ucture **c**onverter) performs conversion between various formats of biomolecular structure files.

`str_calc` - (**str**ucture **calc**ulator) performs calculations on protein structures: contact maps, $\Phi$, $\Psi$ dihedral angles of the backbone chain, $\omega$ and of the side chains $\chi$, etc.

`rms_calc` - (**rms calc**ulator) calculates the optimal superimposition of two protein structures

`clust` - (**clust**ering) used in cluster analysis

`alignc` - (**align**ment **c**onverter) performs the conversion between different sequence alignments.

`praline` - (**pr**ofile **ali**g**ne**r) serves to find the optimal alignment of two sequences or sequence profiles.

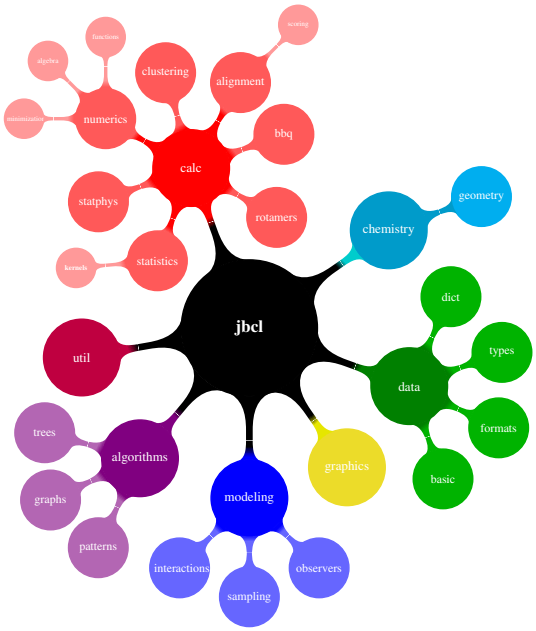
A great effort was made to integrate the software package with the standard commands of the Unix system, such as `grep`, `sed` or `awk`.

*Version 2.x - the library of modules for scripting languages*

The software described above fulfilled its task perfectly but its upgrading to perform novel tasks posed several problems. The most serious one was to unequivocally yet flexibly define the order of the performed operations. For that, after obtaining the doctoral degree, the Author started to work on the next version of the software package. The basic idea behind the software architecture has been completely changed. The new version of `BioShell` became mainly a library of functions called from the Python[H6] language scripts. This solved the problem of computation control and widened the scope of functions available to the users. The new version of the package retained the programs

[25] Gront, D. & Kolinski, A. *Bioinformatics* **22,** 621–622 (2006).

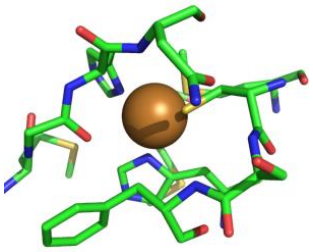
operating in the previous version with two new additions, `PsiBlastSearch` i `PsiBlastAnalyse`[2], that served to analyze the protein sequence space close to the target sequence.




**Ryc. 1: Hierarchical structure of the BioShell software library** (referred as `jbcl` - Java BioComputing Library). Individual modules are grouped into functional packages. For example, algorithms operating on graphs were assigned to `jbcl.algorithms.graphs`. The aligning procedures can be found in`jbcl.calc.alignment`,and the alignment score evaluation functions - in `jbcl.calc.alignment.scoring`.

The software is available for download from its website:`bioshell.pl`together with a comprehensive documentation, example scripts, test data sets, etc.

```
#!/usr/bin/env jython

import sys
# Here we import two BioShell modules: PDB (to read PDB files) ...
from jbcl.data.formats import PDB
# ... and Neighborhood to look for spatial neighbours.
from jbcl.calc.structural import Neighborhood

inputFile = sys.argv[1] # PDB file name is the parameter of this script
reader = PDB(inputFile)
allAtoms = reader.getStructure().getAtomsArray()

n = Neighborhood(allAtoms)
cuResidues = protein.findResidues("␣CU␣")

for cuResidue in cuResidues :
  cuAtom = cuResidue.getAtomsArray()[0]
  nn = n.findNeighborsArray(cuAtom,4.0)
  residueSet = set()
  for atom in nn : residueSet.add( atom.getOwner() )

  for residue in residueSet :
    for line in PDB.createPdbLines( residue ) : print line
```



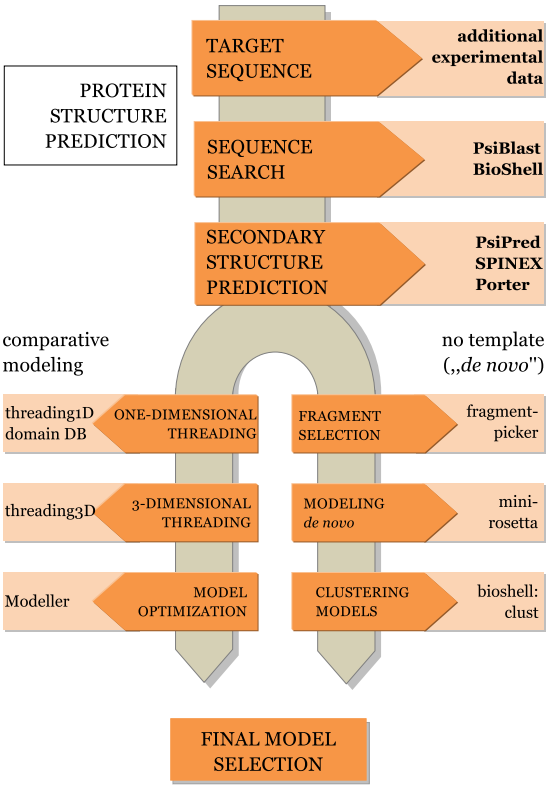
**Ryc. 2: Example of a script that uses the `BioShell` library** *(on the left)*. This script uploads PDB files and searches all cuprum atoms (recognized by the names of the atoms). It determines the spatial environment of every atom, i.e., the amino acid residues located no further than 4 Åaway - and the results are downloaded in the PDB format. *Shown above* is an example of a structure fragment cut out from the 2AZA deposit (azurin) using the script described above.

[2] The second version of the package was implemented in JAVA programming language. Names of all programs, classes and modules were capitalised in accordance to the standard naming convention in JAVA

## Applications

### *Modeling of protein structures*

Since many years Author's main scientific interest has remained protein structure modeling. Most package modules were created with this goal in mind. These procedures perform the required calculations either directly or automate the work of external programs such as `PsiBlast`, `Modeller`[15] or `Rosetta`[18]. The `BioShell` package has been used several times during the biannual CASP[3] experiments: in 2004 (CASP6), 2006 (CASP7), 2010 (CASP9) and 2014 (CASP11). This experiment is in fact a competition in which theory groups try to predict experimentally solved protein structure as accurately as possible. The actual protein structures are revealed after the end of the competition. The modeling protocol used by the Author has changed over those years along with the acquired experience and implementation of new algorithms. A schematic modeling protocol adopted during CASP11[26] is presented in Fig. 3. Some computational procedures employed in this protocol are described in the subsequent part of this Summary together with applications of the `BioShell` package. In the final ranking[4] of template-based modeling categories with 80 predicted domains, the `BioShell-server` group was classified in 41st position. The `BioShell` group, taking part in the FM (free modeling) category took the 40th position. Altogether, 123 expert groups and 84 servers took part in the CASP contest.




**Ryc. 3: Schematic representation of protein structure modeling**. Diagram (adopted from[26]) illustrates the scheme of protein structure modeling employed in the `CASP11` experiment. The algorithm starts from the target protein sequence. The first step consists of database search for proteins with similar amino acid sequences (most probably homological proteins). These sequences will constitute the template for modeling. Based on multiple sequence alignments certain structural features, such as the secondary structure or side-chain solvent exposure, can also be predicted. This information is used to compute the alignment between the templates and the target protein. The obtained alignments are then refined (3D threading) and serve as the basis for building structural models. The last step comprises analysis and selection of the generated models.

[3] Critical Assessment of Protein Structure Prediction Methods; `http://predictioncenter.org/`

[4] http://predictioncenter.org/casp11/zscores_final.cgi

[26] Strumillo, M. *et al.* in. **18** (2014), 379–384.

In the recent years the package has been used in tasks associated with rational protein engineering. These two applications are quite similar and use the same computational methods. In the first one the protein sequence is known and the goal is to model the protein structure. In the second one the sequence is searched for based on a given tertiary structure.

Molecular modeling is not the only application of this software package. Its substantial part is devoted to protein sequence and structure analysis. The package also provides many numerical and statistical procedures. Examples of its usage are summarized below.

### *Sequence search*

`PSIBLAST` is a standard program used to search protein sequence databases. It has been also used in publications summarized in this Summary. This program gives good results when the evolutionary distance between sequences is not very large. If it is, the best solution is to run the program multiple times, often in an iterative way[27]. The `PsiBlastSearch` and `PsiBlastAnalyse` programs were written within two weeks in the summer of 2010, during the CASPP9 experiment, specifically to overcome this problem. In this way, a procedure which constitutes the first stage of comparative modeling and which was until then performed manually in prof. Baker's lab, became fully automated. `PsiBlastAnalyse` automates the performance of the `PSIBLAST` tool, starting it with various initial parameters. The PsiBlastAnalyse processes and analyzes the results. The analysis includes filtering of the selected sequences according to multiple criteria as well as sequence clustering. The final result is a non-redundant set of sequences homological to the sequence of the modeled protein, which is then used to split the target sequence into (potential) domains and to construct a sequence profile. This procedure was intensively used during the CASP9 experiment and has been also employed to design mutants of Treonine Aldolase (the **H12.** publication)

### *Sequence-based prediction*

Sequence profile calculated as described above is used by multiple tools to predict certain structural features of an unknown protein e.g. its secondary structure or solvent exposure of particular amino acid residues. `BioShell` automates the work of the following programs: PsiPred, Porter, SAM, Jufo and SpineX. The obtained results serve as input data for sequence alignment and at the model building stage.

### *Sequence alignment building and optimization.*

A necessary step in comparative protein modeling is finding an appropriate template i.e., a protein (presumably a homolog or a structural analog) whose structure has already been determined experimentally. A proper alignment of these two proteins is of uppermost importance. In the literature one can find numerous methods to solve this

[27] Margelevicius, M. & Venclovas, Ä. *BMC Bioinformatics* **6,** 185+ (2005).

problem. In general, they can be divided into four groups: *(i)* sequence alignment of a pair of proteins, *(ii)* sequence alignment of the whole protein family, *(iii)* alignment of two sequence profiles, *(iv)* alignment of target sequence with the template structure.

The *(i)* method is not very precise and works well only when the proteins are closely related. In order to employ the *(ii)* approach one has to unequivocally identify amino acid sequences of proteins that belong to the investigated protein family. In expert hands, this method yields perfect results; however, more often than not, it requires manual intervention into the aligning process[28]. As one of the tasks of the `BioShell` software was to fully automate the process of biomolecular modeling, the methods implemented in the package were the easily automated methods *(iii)* and *(iv)* , also known as protein threading methods. The first method, 1- dimensional protein threading, was published in 1987[29] and the second one five years later[30]. The term "threading" came into use just then. For comparison, the `PsiBlast` program[31] was published in 1997. These two protein threading variants differ diametrically. While the 1-dimensional variant represents an "ordinary" alignment of two sequence profiles calculated using dynamic programming the 3-dimensional variant is an NP-complete problem. Several approximation methods to solve this problem have been proposed in the literature. All these methods, however, require huge computational resources. On the other hand, the `PsiBlast` program mentioned above, facilitated a quick search of databases for sequences similar to the target sequence and turned into a useful tool to create sequence profiles. In consequence, the use of PsiBlast largely simplified the 1-dimensional threading procedures. Due to all these factors the profile alignment algorithms almost entirely displaced the 3-dimensional threading methods. Until recently `Raptor`[32] was the only publicly available 3D program. This program, rated highly by the participants of the consecutive CASP experiments, makes use of a linear programming algorithm.

The `BioShell` Threading 3D$^{\mathbf{H11}}$ method is based on an entirely different principle. The problem of alignment optimization has been considered in categories of molecular modeling. The alignment score was substituted by energy that exploits both the sequence and the structural components. The alignment algorithm became sampling of the system state space. These states, or alignments, are defined as a list of continuous (gapless) blocks. The Monte Carlo moves consist of splitting, merging, shifting, shrinking or expanding the blocks. This algorithm is not rigorous; the input data might be any combination of sequences, sequence profiles or structures. For example, running the algorithm for a pair of profiles is equivalent to 1-D threading of a pair of structures and for a pair of structures - equivalent to structure alignment. The latter has been used to test

[28] Venclovas, C. & Margelevicius, M. *Proteins* **77 Suppl 9,** 81–88 (2009).
[29] Gribskov, M. *et al. Proceedings of the National Academy of Sciences of the United States of America* **84,** 4355–4358 (1987).
[30] Jones, D. T. *et al. Nature* **358,** 86–89 (1992).
[31] Altschul, S. F. *et al. Nucleic Acids Research* **25,** 3389–3402 (1997).
[32] Xu, J. *et al. J Bioinform Comput Biol* **1,** 95–117 (2003).

the method. In the **H11.** publication it was demonstrated that the `BioShell` Threading 3D program yields better structure alignments than `tm-align` – one of the best softwares written for this purpose.

The solutions implemented in `BioShell` Threading 3D obviously do not solve the NP problem and sampling of the alignment space requires substantial computational resources. Therefore the methodology used relies on two computer programs: one for 1-dimensional and the other for 3-dimensional threading. The first applies dynamic programming for sequence profile alignments of the template and target proteins. The profiles are supplemented with data about the secondary structure which greatly improves search sensitivity[H8.]. The method has been optimized to most reliably identify proteins (potential templates) that belong to the same SCOP family as the target protein. In order to produce the best possible alignments the 3-D threading calculations are performed only for the potential templates identified in the previous step. Also important is the fact that sampling of the state space can deliver the highly-ranked suboptimal alignments. On the basis of each of these alignments a structural model of the target protein is then built. This procedure was used by two research groups participating in the CASP11 experiment (`BioShell-server` and `BioShell-human` groups) and by Dr. Chen Keasar's[5] team (`keasar` group).

It is worth to mention that suboptimal alignments can be generated using other methods as well. This is usually done by modifying the algorithm of the alignment backtracking. The BioShell package has been supplemented with the implementation of the elegant Miyazawa algorithm[33], which preservers Boltzman distribution of the generated alignments. However, being a variant of dynamic programming, this algorithm does not allow to fully exploit data concerning the 3-dimensional template structure. For this reason it has been substituted by `BioShell` Threading 3D program mentioned above.

A totally different approach to comparative modeling has been proposed in publication **H5.**; in this approach there is no substantial need for an input alignment of the modeled protein and the template. The spatial structure of the template is projected onto a lattice in the CABS model. It is worth to mention here that all $\alpha$ carbon atoms lie on a cubic lattice with a constant of 0.61Å. An additional energy component, checking the fit between the modeled conformation and the lattice- projected template, has been introduced into the model. Energy prize (or sanction) is awarded if the template and target protein atoms lie in the same lattice nodes. This method works perfectly in modeling of small proteins but in the case of large molecules it requires substantial computational resources.

[5] Ben-Gurion University of the Negev, Be'er Sheva, Israel

[33] Miyazawa, S. *Protein Eng.* **8,** 999–1009 (1995).

### *Model construction*

Modeller is the program of choice for template-based model building and Rosetta – for *de novo* modeling. In addition, the CABS program is also used in both scenarios,. Implementation of the latter program, which uses reduced representation of the peptide chain, requires rebuilding of the atomic details of the model. The BBQ$^{\mathbf{H1}}$ program, which reconstructs the protein backbone, has been designed especially for this purpose. The side chains are reconstructed using the `scwrl`[34] program.

### *Cluster analysis*

Protein structure modeling usually yields multiple models. The next step is therefore cluster analysis, the goal of which is to select representative structures. For that purpose the `BioShell` package uses a hierarchical algorithm[35]. The `clust` program[6] routinely analyses datasets of tens of thousands structures. This tool has been created for the CASP6 experiment but has been also used during CASP7 and CASP11. In the publication[36] it was used for docking a short peptide derived from the C3D protein to the SH3-N domain. The hierarchical procedure is also employed in the `PsiBlastAnalyse` program to group similar protein sequences.

### *Computational procedures for biomolecular structures and numerical methods.*

The `BioShell` package offers a very wide scope of algorithms dedicated to protein structure analysis. It calculates and performs structure alignments. It also calculates various structural parameters: plane and torsion angles, distances, contact maps and hydrogen bond maps. The package also provides a wide set of numerical and statistical methods that serve for data processing, e.g., interpolation methods, bootstrap, histograms or kernel density estimators. Such functionality is not often found in this type of packages, and some functions are unique to the `BioShell` package. This is probably the reason why procedures operating on biomolecular structures are the preferred ones among the BioShell users[37,38].

### *Statistical potentials*

The functionality of BioShell has been tested several times to calculate statistical potentials. For example, in the **H7** publication statistical potentials describing local (i.e. involving several consecutive amino acid residues) geometry of the main chain of a given protein family. Statistical potentials are commonly used in biomolecular structure modeling. In their typical form they determine the probability of occurrence of

[6] originally published as HCPM - Hierarchical Clustering of Protein Models

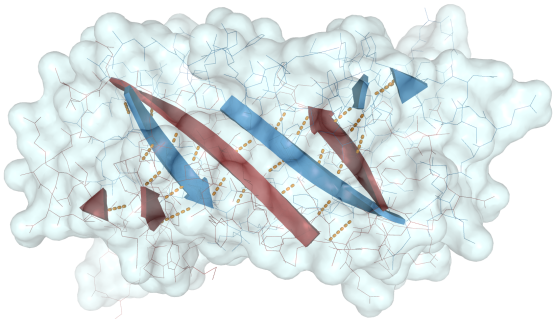
[34] Dunbrack, J. & Karplus, M. *Journal of Molecular Biology* **230,** 543–574 (1993).
[35] Gront, D. & Kolinski, A. *Bioinformatics* **21,** 3179–3180 (2005).
[36] Gront, D. *et al. Acta Pol Pharm* **63,** 436–438 (2006).
[37] Kim, H. & Kihara, D. *Proteins* **82,** 3255–3272 (2014).
[38] Chruszcz, M. *et al. Journal of Biological Chemistry* **287,** 7388–7398 (2012).

a certain structural property (e.g., atom-atom contact pair) depending on the type of amino acids. For example, the CABS program makes use of potentials that estimate the distance $R_{15}(A_2, A_3)$, that means between each $i$ and $i+4$ $\alpha$-carbon atom along the protein backbone. Apart from the distance, this function depends on the type of amino acids at positions $i+1$ and $i+3$. Once calculated, this function might be used to model any protein.

In the case of potentials defining a protein family, the energy function depends on the position in the protein sequence of the modeled protein. The sequence profiles[29] and sequence fragment libraries[39] employed in protein structure modeling are based on the same principles. As with the profile or sequence fragment, potential has to be calculated for each individual sequence of the modeled protein. In addition, databases should contain information about structures of proteins that belong to the same family as the query protein. Nonetheless, the potential provides much more details about the local conformation of the polypeptide chain.

*Topological analysis of interdigitated proteins*

The `BioShell` package has been also used for the AF2331[40] protein structure analysis (PDB deposit: `2FD0`). The protein is an atypical representative of the $\alpha+\beta$ class, because one of its $\beta$-sheets is formed alternately by fragments of the A and B chains. In that way, inter-chain interactions determine a substantial fraction of the secondary structure. In Fig. 4, which presents the selected $\beta$-strand of the AF2331 protein, one can notice that, when following the hydrogen bonds of the $\beta$-sheet, one encounters the `AABABABB` chains, changing the chain code as many as 5-times. The obvious question that arose during studies on `2FD0` was whether such topology has already been observed in other solved structures.



**Ryc. 4: Interdigitated $\beta$-sheet of `2FD0` protein**. The $\beta$-sheet is formed by 8 strands of `A` and `B` chains of the deposit, marked in blue and red, respectively. The orange dashed lines illustrate the hydrogen bond network.

The answer was given by a short script written in the BioShell environment. This script loaded a PDB file from which it read the information about $\beta$-strands and computed the hydrogen bond network. Then it built a graph in which the nodes were $\beta$-strands, colored according to the chain code, and the edges were hydrogen bonds. The final result represented the longest possible path leading through nodes of alternately changing colors. Analysis conducted on all protein structures solved at that time demonstrated that

[39] Gront, D. *et al. PloS one* **6,** e23294+ (2011).
[40] Wang, S. *et al. Protein Science* **18,** 2410–2419 (2009).

while the once-interdigitated $\beta$-sheets (the ABA type) are spotted quite frequently, the ABAB topology has been observed in about one hundred cases. A fourfold change of the protein chain was found only in three deposits. The ABABAB arrangement, present in AF2331, was found only in the 2HJ1 deposit and was the longest one observed.

*Canonical distribution of states -statistical description*

Biomolecular modeling is usually conducted in such a way so that the simulated system is described by a canonical ensemble. It is especially easy when the modeling procedure is based on the Monte Carlo model according to the Metropolis scheme mentioned in the Introduction. The result of modeling is then a set of conformations of the system the energy of which is defined by Boltzmann distribution. Thanks to the weighted histogram analysis method[41] implemented in the the `BioShell` package, it is possible to calculate the statistical sum $\mathcal{Z}(T)$ of the investigated system. The input data of the `MultiHist`$^{\mathbf{H4}}$ program of the `BioShell` package are energy values $\mathcal{E}$ observed during simulations carried in different temperatures and the result – the statistical sum $\mathcal{Z}(T)$ and the state density $\Omega(\mathcal{E})$. The `StatPhys` program, in turn, loads the $\Omega(\mathcal{E})$ values and the measured *observables* and calculates their canonical means in any given temperature. These programs have been used to study the lattice models of simple polymers$^{\mathbf{H3}}$ and also to analyze the results of protein simulations in the reduced CABS model$^{\mathbf{H2}}$ (the Monte Carlo method) or full-atom molecular dynamics.[42]

The simulations described above were performed using the Replica Exchange Monte carlo method[43]. In this method, due to simultaneous modeling of many replicas of the same system at different temperatures, exploration of the state space is more efficient. Replica exchange or, in other words, exchange of the copies of the system between different temperatures, makes it possible to cross the energy barriers fairly easily. The crucial thing, however, is the appropriate choice of the temperature set $\mathrm{T}_i$ in which the particular replicas are simulated. There are a lot of solutions described in the literature[44,45], none of them, however, guarantees the optimal flow of replicas through the temperature space. In the **H3** publication, a novel way of selecting the simulation temperatures has been presented. It is based on the observation that the probability of replica exchange $P(T_1 \rightarrow T_2)$ between temperatures $\mathrm{T}_1$ and $\mathrm{T}_2$ depends on the extent to which the state densities overlap at these temperatures. This probability can be calculated if the state density function $\Omega(\mathcal{E})$ is known. Determination of the $\mathrm{T}_i$ temperatures starts from REMC simulation in which the first approximation to the $\Omega(\mathcal{E})$ of the query system is built. Based on this state density, a set of temperatures in which the

[41] Ferrenberg, A. M. & Swendsen, R. H. *Physical Review Letters* **63,** 1195–1198 (1989).
[42] Wabik, J. *et al. International Journal of Molecular Sciences* **14,** 9893–9905 (2013).
[43] Geyer, C. J. in *Computing Science and Statistics: Proceedings of 23rd Symposium on the Interface Interface Foundation* (1991), 156–163.
[44] Rathore, N. *et al. The Journal of Chemical Physics* **122,** 024111+ (2005).
[45] Kofke, D. A. *The Journal of Chemical Physics* **117,** 6911–6914 (2002).

probability $P(T_i \rightarrow T_{i+1})$ is equal for each $i$ is calculated (via numerical integration). It can be demonstrated that this criterion ensures the fastest exchange of replicas between different temperatures.

### *In silico protein engineering*

The most recent application of the `BioShell` package is rational protein engineering. Publication **H12.** describes a successful modification of treonine aldolase from the bacterium *Thermotoga maritima* in order to increase stability of this enzyme. In living organisms this enzyme cleaves treonine into glycine and methanal and its active form is a homotetramer. For this reason the goal set in the **H12.** publication was to enforce the interactions between the amino acid chains. The first step of the theoretical part of the project consisted of assembling all sequences homological to the studied sequence. To achieve this, the `PsiBlastSearch` program, which starts the calculations performed by the `PsiBlast` program using different settings, was used. Results, analyzed with the `PsiBlastAnalyse` tool, gave 132 representative sequences. These sequences were used as queries in subsequent database searches. Finally, 100 000 sequences were found, 45% of which in the second search round. 52 sequences were found in thermophilic organisms. Multiple sequence alignment showed amino acid variability at every position of the polypeptide chain. The `StrCalc` program was used to analyze the crystallographic structure of the original enzyme (PDB: `1LW5` deposit). Based on the distance between atoms and spatial orientation of the amino acid side chains it was possible to select potential residues where new interactions, such as ionic bridges or disulfide bonds, could be introduced.

Treonine Aldolase has a rather peculiar quaternary structure i.e., two polypeptide chain pairs within a tetramer: pair `B` and `C`, and pair `A` and `D`, make contacts through amino acid residues located at the same position. For example, proline 56 in chain `A` is located only 4.5Åaway from proline 56 in chain `D`. Thanks to that it was possible to introduce as many as four cysteine residues and, in consequence, *two* disulfide bonds, into the tetramer by mutating only a single amino acid residue. Eighteen residues to be mutated were selected on the basis of a preliminary analysis. The structural models of mutant proteins were computed using `Modeller` and `Rosetta`. Finally, the best mutants were tested experimentally. Two of them (`P56C` and `A21C`) are significantly more stable than the native protein.

## 5 Description of other scientific achievements

Author also actively participates in development of the Rosetta modelling suite as one of 27 Principal Investigators affiliated in Rosetta Commons *Rosetta Commons*[7] This package consists of multiple programs with over 2.5 million lines of source code written in the C++language. The Rosetta software is used to model protein and RNA structures (both

[7] https://www.rosettacommons.org/

*de novo* and template-based modeling), to solve biomolecular structures based on fragmentary experimental data (mostly NMR and electron density maps EM) or to design new proteins. Author's most important contribution to this software is the design of an algorithm for building a fragment library[39]. Such library is indispensable for protein structure modeling using Rosetta. In addition, author incorporated into the Rosetta package procedures that allow to use SAXS data in modeling.