1 Imię i nazwisko

Dominik Gront

2 Posiadane dyplomy, stopnie naukowe/ artystyczne – z podaniem nazwy, miejsca i roku ich uzyskania oraz tytułu rozprawy doktorskiej

23 czerwca 2001	Magister chemii, Uniwersytet Warszawski, Wydział Chemii, Warszawa *„Klasyczne i nowe techniki Monte Carlo w termodynamice białek”*
16 kwietnia 2006	Doktor nauk chemicznych w zakresie chemii teoretycznej, Uniwersytet Warszawski, Wydział Chemii, Warszawa *„Opracowanie algorytmu do modelowania struktur białkowych na podstawie baz danych sekwencji i struktur”*

3 Informacje o dotychczasowym zatrudnieniu w jednostkach naukowych / artystycznych

październik 2006 - wrzesień 2007	Uniwersytet Warszawski, Wydział Chemii, Warszawa
październik 2007 - wrzesień 2008	Molecular Physiology and Biological Physics, University of Virginia, Charlottesville, VA, USA
od października 2008	Uniwersytet Warszawski, Wydział Chemii, Warszawa

4 Wskazanie osiągnięcia wynikającego z art. 16 ust. 2 ustawy z dnia 14 marca 2003 r. o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki (Dz. U. nr 65, poz. 595 ze zm.)

4.1 tytuł osiągnięcia naukowego

„Opracowanie nowych algorytmów modelowania białek i ich implementacja w pakiecie BioShell”

4.2 publikacje wchodzące w skład osiągnięcia naukowego

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 Thermotoga maritima”* J. of Biotechnology, 2015 199:69-76, doi: 10.1016/j.jbiotec.2015.02.013

4.3 omówienie celu naukowego/artystycznego ww. pracy/prac i osiągniętych wyników wraz z omówieniem ich ewentualnego wykorzystania.

Wprowadzenie

Postęp w nauce i rozwój technologii są nierozerwalnie ze sobą związane. Nauki podstawowe tworzą nowe narzędzia badawcze, które z kolei otwierają przed człowiekiem nieznane obszary badań. Pierwsze maszyny cyfrowe, powstałe w latach pięćdziesiątych minionego stulecia, przyczyniły się do intensywnego rozwoju istniejących dziedzin nauki. Zainicjowały też powstanie kilku nowych, przede wszystkim informatyki. U jej podwalin legły zarówno prace teoretyczne, jak i praktyczne rozwiązania: architektura von Neumanna[1] czy pierwsze języki programowania[2]. W latach pięćdziesiątych XX wieku opublikowane zostały dwie podstawowe metody modelowania molekularnego: dynamika molekularna[3] (na razie w ujęciu dyskretnym) oraz schemat Metropolisa[4].

W tych samych latach postępował rozwój biologii molekularnej. Rozszyfrowano kod genetyczny[5], poznano strukturę DNA[6] oraz mioglobiny[7]. Opisano także zasady rządzące strukturą białka[8–10]. Zautomatyzowano też metodę sekwencjonowania białek, co zaowocowało opublikowaniem - w formie książki - przez Margaret Dayhoff pierwszej sekwencyjnej bazy danych[11].

Badania te możliwe były również dzięki wykorzystaniu pierwszych komputerów. Zastosowano je między innymi do rozwiązywania rentgenograficznych struktur białek a także do składania fragmentarycznych wyników sekwencjonowania[12]. Sekwencji białek znano już coraz więcej; w literaturze zaczęto dyskutować więc ewolucję genów i białek. W roku 1967 opublikowano pierwsze drzewo filogenetyczne[13]. Wreszcie w 1970r. Needleman i Wunsch opublikowali algorytm globalnego uliniowienia sekwencji[14]. W ten sposób zrodziła się nowa nauka - *bioinformatyka* - choć na samą jej nazwę trzeba było jeszcze poczekać prawie dziesięć lat[15].

[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] Watson, J. D. & Crick, F. H. *Nature* **171,** 737–738 (1953).
[7] Kendrew, J. C. *et al. Nature* **181,** 662–666 (1958).
[8] Pauling, L. & Corey, R. B. *Proceedings of the National Academy of Sciences* **37,** 251–256 (1951).
[9] Pauling, L. *et al. Proceedings of the National Academy of Sciences* **37,** 205–211 (1951).
[10] Ramachandran, G. N. *et al. Journal of molecular biology* **7,** 95–99 (1963).
[11] Dayhoff, M. O. (Silver Spring, Md., 1965).
[12] Dayhoff, M. O. & Ledley, R. S. in *Proceedings of the December 4-6, 1962, Fall Joint Computer Conference* (Philadelphia, Pennsylvania, 1962), 262–274.
[13] Fitch, W. M. & Margoliash, E. *Science (New York, N.Y.)* **155,** 279–284 (1967).
[14] Needleman, S. B. & Wunsch, C. D. *Journal of molecular biology* **48,** 443–453 (1970).
[15] Hogeweg, P. *PLoS Comput Biol* **7,** e1002021+ (2011).

Kierunki rozwoju tej dziedziny w kolejnych dekadach związane były z gwałtownie powiększającymi się zasobami danych: sekwencjami i strukturami biomolekuł. Zadaniem bioinformatyki stało się te dane gromadzić i analizować. By temu sprostać, rozpoczęto tworzenie odpowiedniego oprogramowania. Równocześnie zaczęło powstawać oprogramowanie do modelowania struktury i dynamiki biomolekuł. Początkowo oba te obszary były wyraźnie rozgraniczone. Bioinformatyka bazowała na danych, modelowanie molekularne zaś - na zasadach fizyki. Okazało się jednak, że wnioskowanie oparte o ewolucję genów może być równie ważne co modele wyprowadzone z pierwszych zasad.

Aktualnie istnieje wiele inicjatyw poświęconych tworzeniu oprogramowania zarówno z dziedziny bioinformatyki jak i modelowania molekularnego. Spośród narzędzi do modelowania należy wspomnieć pakiety do dynamiki molekularej, programy Modeller[16], ICM[17], UNRES[18], Rosetta[19], czy rodzinę modeli siatkowych opracowanych w grupie prof. Kolińskiego (np. SICHO[20] oraz CABS[21]). Z kolei jako przykłady typowych pakietów bioinformatycznych wymienić należy BioPerl[22], BioPython[23], BioJava[24] i BioRuby[25]. W roku 2004 pakiety te[1], udostępniały jedynie bardzo skromny zestaw funkcji operujących na strukturach białek. Funkcjonalność taka była jednakże niezbędna piszącemu te słowa do sprawnego realizowania wykonywanej przezeń pracy doktorskiej. Było to impulsem, który zainicjował powstanie pakietu `BioShell`.

Celem naukowym przedstawionego cyklu prac było stworzenie spójnego i kompletnego pakietu oprogramowania, wspomagającego rozwiązywanie różnorodnych problemów z zakresu bioinformatyki strukturalnej oraz modelowania struktur biomolekuł. Praktycznie cały kod źródłowy napisany został przez habilitanta a powstałe narzędzia obliczeniowe wykorzystywane są w co najmniej kilkunastu laboratoriach na świecie. W pracach tych przedstawiono konstrukcję kolejnych wersji pakietu oraz opisano najważniejsze zaimplementowane w nim algorytmy. Do cyklu zostały włączone również prace prezentujące przykładowe zastosowania pakietu.

[1] jedynie BioPerl i BioPython; BioJava i BioRuby jeszcze wtedy nie istniały

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

Oprogramowanie BioShell

Wersja 1.x - programy uruchamiane z linii poleceń systemu Unix

Konstrukcja pierwszej wersji pakietu, opublikowanej w 2006 roku[26], w całości opierała się na idei funkcjonowania systemu Unix. Składał się on więc z kilku programów których działanie kontrolowane było odpowiednimi opcjami podawanymi z linii poleceń. W początkowej wersji `BioShell` ułatwiał prowadzenie symulacji dynamiki białek w modelu zredukowanym `CABS`[21]. Wykorzystywany był do przygotowywania plików wejściowych i analizowania wynikowych trajektorii. Inne moduły pakietu posłużyły do wyprowadzania potencjałów średniej siły na podstawie statystyk zebranych ze znanych struktur białkowych. W roku 2007 na pakiet składały się następujące programy:

`strc` - (**str**ucture **c**onverter) dokonujący konwersji pomiędzy formatami plików zapisujących struktury biomolekuł

`str_calc` - (**str**ucture **calc**ulator) wykonujący obliczenia na strukturach białek: mapy kontaktów, kąty łańcucha głównego Φ, Ψ, ω oraz łańcuchach bocznych χ, itp.

`rms_calc` - (**rms calc**ulator) obliczający optymalne nałożenie jednej struktury białka na drugą.

`clust` - (**clust**ering) do analizy skupień

`alignc` - (**align**ment **c**onverter) dokonujący konwersji pomiędzy formatami uliniowień sekwencyjnych

`praline` - (**pr**ofile **ali**gn**e**r) służący do znajdowania optymalnego uliniowienia dwóch sekwencji lub profili sekwencyjnych.

Duży nacisk położono na integrację pakietu ze standardowymi poleceniami systemu Unix, takimi jak `grep`, `sed` czy `awk`.

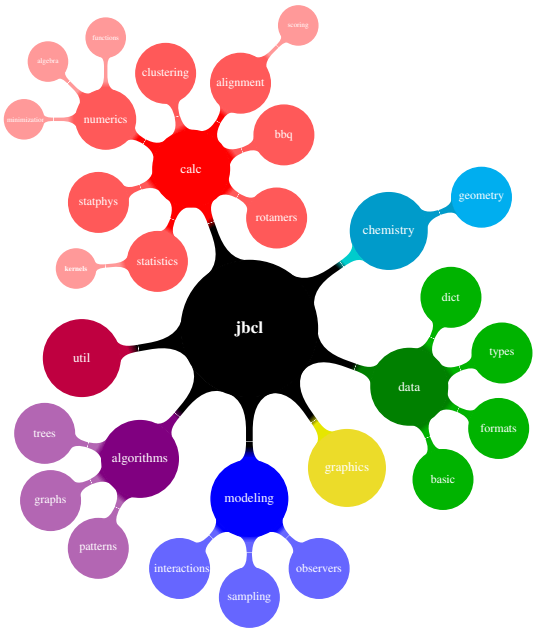
Wersja 2.x - biblioteka modułów dla języków skryptowych

Oprogramowanie opisane powyżej doskonale spełniało swoje zadanie, ale jego rozbudowa o nowe funkcje nastręczała kilku trudności. Najpoważniejszą z nich było jednoznaczne a zarazem elastyczne zdefiniowanie kolejności wykonywania zadanych operacji. Dlatego po uzyskaniu stopnia doktora, autor rozpoczął prace nad kolejną wersją pakietu. Diametralnej zmianie uległa filozofia, na której oparto konstrukcję oprogramowania. W nowej wersji `BioShell` stał się przede wszystkim biblioteką funkcji, wywoływanych ze skryptów napisanych w języku Python**H6**. Rozwiązało to całkowicie problem sterowania obliczeniami oraz znacznie poszerzyło wachlarz funkcji pakietu udostępnionych użytkownikowi. W nowej wersji pakietu odtworzono programy z wersji poprzedniej. Dodano też dwa nowe: `PsiBlastSearch` i `PsiBlastAnalyse`[2], służące do analizowania przestrzeni

[2] Wersja druga pakietu napisana została w języku Java. Nazwy wszystkich programów pakietu zmieniono, wprowadzając wielkie litery zgodnie z ogólnie przyjętymi w tym środowisku konwencją.

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

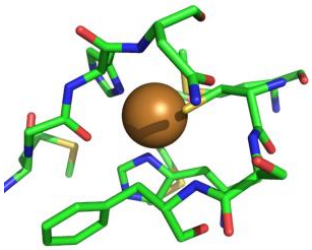
sekwencji białkowych w pobliżu zadanej sekwencji-celu.



Rysunek 1: **Hierarchiczna struktura biblioteki BioShell** (określonej tu `jbcl` - Java BioComputing Library). Poszczególne moduły pogrupowane zostały w pakiety odpowiadające ich funkcjonalności. Dla przykładu algorytmy operujące na grafach zawarto w `jbcl.algorithms.graphs`. Procedury do wyznaczania ulinowień - w `jbcl.calc.alignment`, a same funkcje oceniające uliniowienia - w `jbcl.calc.alignment.scoring`.

Oprogramowanie dostępne jest na stronie `bioshell.pl`, wraz z obszerną dokumentacją, przykładowymi skryptami, danymi testowymi itp.

```
#!/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
```



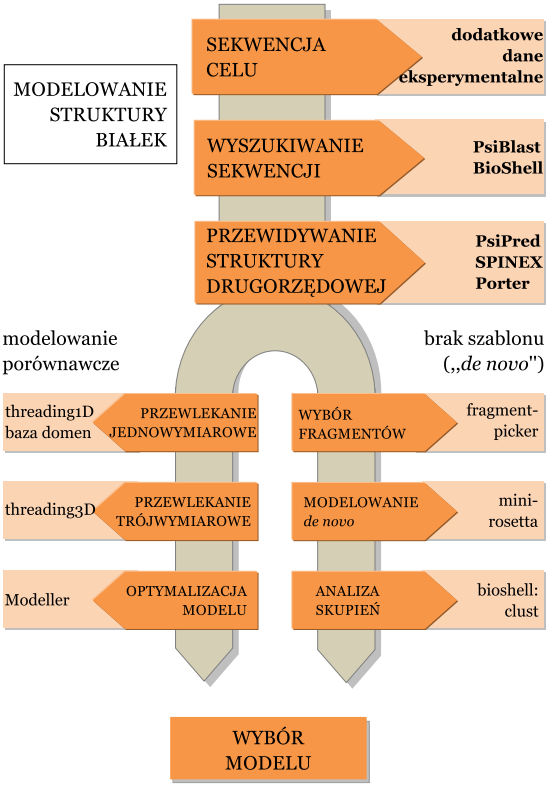
Rysunek 2: **Przykładowy skrypt** wykorzystujący bibliotekę `BioShell` *(po lewej)*. Skrypt ten ładuje plik PDB i wyszukuje wszystkie atomy miedzi (rozpoznawane po nazwie atomu). Dla każdego z tych atomów wyszukuje jego otoczenie w przestrzeni - reszty aminokwasowe odległe o nie więcej niż 4Å- a wyniki nagrywane są w formacie PDB. *Powyżej:* przykładowy fragment struktury wycięty z depozytu 2AZA (azuryna) za pomocą w/w skryptu.

Zastosowania

Modelowanie struktur białek

Od wielu lat głównym zainteresowaniem naukowym autora pozostaje modelowanie struktur białek. Większość modułów pakietu powstała w tym właśnie celu. Procedury te bezpośrednio wykonują potrzebne obliczenia, bądź automatyzują pracę zewnętrznych programów, takich jak `PsiBlast`, `Modeller`[16] czy `Rosetta`[19]. Pakiet `BioShell` wykorzysty-

wano kilkukrotnie podczas odbywających się co dwa lata eksperymentów CASP[3] w latach: 2004 (CASP6), 2006 (CASP7), 2010 (CASP9) i 2014 (CASP11). Eksperyment ten ma charakter konkursu, w którym uczestniczą grupy teoretyczne, starając się jak najdokładniej przewidzieć struktury białek wyznaczone eksperymentalnie. Faktyczne struktury są ujawnione dopiero po zakończeniu konkursu. Wykorzystywany przez autora protokół modelowania zmieniał się istotnie przez te lata wraz ze zdobywanym doświadczeniem i implementacją nowych algorytmów. Na Ryc. 3 przedstawiono schemat postępowania przyjęty podczas CASP11[27]. Niektóre z wykorzystanych w nim procedur obliczeniowych opisano szerzej w dalszej części autoreferatu, pokazując jednocześnie rolę oprogramowania `BioShell`. W końcowym rankingu[4] kategorii modelowania w oparciu o szablon[5], uwzględniającym 81 przewidzianych domen grupa `BioShell-server` została sklasyfikowana na 41 pozycji. Startująca w kategorii **FM** (*Free Modelling*) grupa `BioShell` zajęła 40 pozycję. Ogółem z konkursie CASP11 wzięło udział 123 grupy eksperckie i 84 serwery



Rysunek 3: **Schemat modelowania struktur białek** Diagram (zaadaptowany z pracy[27]) obrazuje schemat modelowania struktur białek, jaki wykorzystano w trakcie eksperymentu `CASP11`. Algorytm startuje z sekwencji białka-celu. Pierwszym etapem jest przeglądanie baz danych w poszukiwaniu białek o podobnych sekwencjach (najprawdopodobniej homologicznych). Będą one potencjalnymi szablonami do modelowania. Na podstawie stworzonego uliniowienia wielu sekwencji przewidywane są również pewne cechy strukturalne, np. struktura drugorzędowa, czy wyeksponowanie grup bocznych badanego białka do rozpuszczalnika. Informacja ta wykorzystywana jest do obliczenia uliniowienia pomiędzy sekwencją białka modelowanego a szablonami. Uliniowienia te są następnie udokładniane (przewlekanie 3D), a na ich podstawie budowane są modele strukturalne. Ostatni etap to analiza i selekcja uzyskanych modeli.

W ostatnich latach pakiet wykorzystywany jest również w zadaniach związanych z racjonalną inżynierią białek. Oba te cele są dość podobne i wykorzystują te same metody obliczeniowe. W pierwszym jednak znana jest sekwencja białka a celem modelowania jest jego struktura. W drugim zaś zakłada się konkretną strukturę trójwymiarową białka a poszukuje sekwencji.

[3] Critical Assessment of Protein Structure Prediction Methods; `http://predictioncenter.org/`
[4] http://predictioncenter.org/casp11/zscores_final.cgi
[5] **TBM**, *Template Based Modelling*

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

Modelowanie molekularne to nie jedyna rola pakietu. Spora jego część poświęcona jest bowiem analizie sekwencji i struktur białek. Pakiet udostępnia też wiele standardowych procedur numerycznych i statystycznych. Przykładowe zastosowania również podsumowano poniżej.

Wyszukiwanie sekwencji

Standardowo do przeszukiwania sekwencyjnych baz danych wykorzystywany jest program `PSIBLAST`. Stosowany był on także w pracach podsumowanych w niniejszym Autoreferacie. Program ten doskonale sprawdza się w przypadku, gdy odległość ewolucyjna pomiędzy sekwencjami nie jest duża. W innych przypadkach najlepiej sprawdza się wielokrotne, często iteracyjne wykorzystanie tego programu[28]. W tym właśnie celu powstały programy `PsiBlastSearch` i `PsiBlastAnalyse` napisane w niecałe dwa tygodnie latem 2010 r, w trakcie trwającego właśnie konkursu CASP9. W ten sposób autor niniejszego Autoreferatu w pełni zautomatyzował wykonywaną dotychczas manualnie w laboratorium prof. Bakera procedurę, stanowiącą pierwszy etap modelowania porównawczego. Program `PsiBlastSearch` automatyzuje pracę narzędzia `PSIBLAST`, uruchamiając go z różnymi wartościami parametrów startowych. Program `PsiBlastAnalyse` przetwarza i analizuje uzyskane wyniki. Analiza ta obejmuje filtrowanie znalezionych sekwencji wg wielu kryteriów oraz analizę skupień. Ostatecznym wynikiem jest różnorodny[6] zbiór sekwencji homologicznych do sekwencji modelowanego białka, wykorzystany następnie do podziału sekwencji celu na (potencjalne) domeny oraz przy konstrukcji profilu sekwencyjnego. Procedura ta była intensywnie wykorzystywana w trakcie eksperymentu CASP9, została też zastosowana przy projektowaniu mutantów białka aldolazy treoninowej (praca **H12.**).

Predykcje w oparciu o sekwencje

Profil sekwencyjny obliczony w opisanym powyżej etapie, wykorzystywany jest przez szereg narzędzi, których celem jest predykcja pewnych cech strukturalnych nieznanego białka, np jego struktury drugorzędowej oraz wyeksponowania poszczególnych reszt do rozpuszczalnika. `BioShell` automatyzuje pracę programów: PsiPred, Porter, SAM, Juffo i SpineX. Uzyskane wyniki służą następnie jako dane wejściowe do tworzenia uliniowienia oraz na etapie budowy modelu.

Tworzenie i optymalizacja uliniowień

Niezbędnym elementem modelowania porównawczego białka-celu jest znalezienie szablonu, czyli wystarczająco podobnego białka (w założeniu: homologa bądź analoga strukturalnego), którego struktura została już wyznaczona eksperymentalnie. Bardzo istotne

[6] ang. *non-redundant*

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

jest również zbudowanie poprawnego uliniowienia tych dwóch białek. W piśmiennictwie znaleźć można bardzo wiele metod rozwiązywania tego problemu. Ogólnie rzecz ujmując, można je podzielić na cztery grupy: *(i)* Uliniowienie sekwencyjne pary białek, *(ii)* ulinowienie sekwencyjne całej rodziny, *(iii)* uliniowienie dwóch profili sekwencyjnych oraz *(iv)* uliniowienie sekwencji celu ze strukturą szablonu.

Metoda *(i)* jest mało dokładna i sprawdza się tylko w przypadku, gdy białka są bardzo blisko spokrewnione. Aby zastosować podejście *(ii)*, trzeba jednoznacznie wskazać sekwencje białek, należące do badanej rodziny. W rękach eksperta metoda ta daje doskonałe wyniki; najczęściej wymaga jednak manualnej ingerencji w tworzone uliniowienie[29]. Ponieważ jednym z celów postawionych przed oprogramowaniem `BioShell` była jak najpełniejsza automatyzacja procesu modelowania biomolekuł, w pakiecie zaimplementowano łatwe do zautomatyzowania metody *(iii)* oraz *(iv)*, zwane przewlekaniem białek. Pierwszą z nich - przewlekanie jednowymiarowe - opublikowano w roku 1987[30], drugą pięć lat później[31]. Wtedy też pojawiła się nazwa "przewlekanie"[7]. Dla porównania, program PsiBlast[32] opublikowano w roku 1997. Wspomniane dwa warianty przewlekania różnią się diametralnie. O ile przypadek jednowymiarowy to „zwykłe" uliniowienie dwóch profili sekwencyjnych obliczane metodą dynamicznego programowania, to wariant trójwymiarowy jest problemem NP-zupełnym. W literaturze zaproponowano kilka przybliżonych metod jego rozwiązywania. Wszystkie one wymagają jednak ogromnych zasobów obliczeniowych. Z drugiej strony, wspomniany już program `PsiBlast` umożliwił szybkie wybieranie z baz danych sekwencji podobnych do białka-celu i stał się wygodnym narzędziem do tworzenia profili sekwencyjnych. W konsekwencji istotnie uprościł procedury przewlekania jednowymiarowego. Czynniki te spowodowały, że algorytmy uliniowienia profili wyparły prawie zupełnie metody przewlekania trójwymiarowego. Jeszcze do niedawna jedynym publicznie dostępnym programem 3D był `Raptor`[33]. Metoda ta, wysoko oceniana w kolejnych konkursach przewidywania struktury białek CASP, wykorzystuje algorytm programowania liniowego.

Metoda `BioShell` Threading 3D$^{\mathbf{H11}}$ oparta jest na zupełnie innej zasadzie. Problem optymalizacji uliniowienia potraktowany został jak zagadnienie z dziedziny modelowania molekularnego. Ocenę uliniowienia (*ang. score*) zastąpiła energia, wykorzystująca zarówno człony sekwencyjne jak i strukturalne. Algorytmem uliniowienia stało się probkówanie przestrzeni stanów układu. Stany te - czyli uliniowienia - zapisane są jako lista ciągłych (niezawierających przerw) bloków. Ruchy Monte Carlo polegają na dzieleniu, łączeniu, przesuwaniu, skracaniu oraz wydłużaniu bloków. Algorytm ten jest bardzo

[7] ang. *threading*

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

ogólny. Jako dane wejściowe można użyć dowolne połączenie sekwencji, profilu sekwencyjnego albo struktury. Dla przykładu, uruchomienie algorytmu dla pary profili jest równoważne przewlekaniu 1D a dla pary struktur - zagadnieniu uliniowienia struktur. To ostatnie zagadnienie posłużyło do przetestowania opisywanej metody. W pracy **H11.** pokazano, że `BioShell` Threading 3D znajduje lepsze uliniowienia struktur niż `tm-align` - jedno z najlepszych programów stworzonych do tego celu.

Rozwiązania zaimplementowane w programie `BioShell` Threading 3D oczywiście nie rozwiązują ściśle problemu NP, a próbkowanie przestrzeni uliniowień wymaga znacznych zasobów. Dlatego stosowana metodyka zakłada wykorzystanie dwóch programów: przewlekania jednowymiarowego oraz przewlekania trójwymiarowego. Pierwszy z nich wykorzystuje programowanie dynamiczne do uliniowienia profili sekwencyjnych obliczonych dla białek: szablonu i celu. Profile są uzupełnione informacją o strukturze drugorzędowej, co istotnie poprawia czułość wyszukiwania**H8**. Metodę tą zoptymalizowano tak, aby z jak największą wiarygodnością wskazywała białka (potencjalne szablony) należące do tej samej rodziny SCOP, co białko-cel. Obliczenia przewlekania trójwymiarowego prowadzone są tylko dla potencjalnych szablonów wyłonionych w poprzednim etapie i mają na celu uzyskanie jak najlepszego uliniowienia. Bardzo istotny jest również fakt, że próbkowanie przestrzeni stanów dostarcza także wysoko ocenionych uliniowień sub-optymalnych. Na podstawie każdego z nich budowany jest strukturalny model białka-celu. Procedura ta wykorzystana została przez dwa zespoły badawcze, biorące udział w konkursie CASP11: (grupy `BioShell-server` i `BioShell-human`) oraz zespół dra Chena Keasara[8] (grupa `keasar`).

Warto zaznaczyć, że uliniowienia suboptymalne mogą być generowane także innymi metodami. Najczęściej uzyskuje się je odpowiednio modyfikując algorytm odtwarzania ścieżki uliniowienia[9]. Do pakietu `BioShell` dołączono implementację eleganckiego algorytmu Miyazawy[34], który zachowuje rozkład Boltzmanna generowanych uliniwień. Algorytm ten jednakże, jako wariant programowania dynamicznego, nie umożliwia wykorzystania pełnej informacji o trójwymiarowe strukturze szablonu. Dlatego też został zastąpiony przez opisany powyżej program `BioShell` Threading 3D.

Zupełnie inne podejście do modelowania porównawczego zaproponowano w pracy **H5.**; w podejściu tym wejściowe uliniowienie pomiędzy modelowanym białkiem a jego szablonem w zasadzie nie jest potrzebne. Przestrzenna struktura szablonu zrzutowana jest na siatkę w modelu CABS. Warto w tym miejscu zaznaczyć, że w tym modelu wszystkie atomy węgli α leżą na sieci kubicznej o stałej 0.61Å. Do modelu wprowadzono dodatkowy człon energii, oceniający dopasowanie pomiędzy aktualną modelowaną konformacją a siatkowym rzutem szablonu. Nagroda (bądź kara) energetyczna przyznawana jest,

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

[9] ang. *backtracking*

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

gdy atomy szablonu i celu znajdują się w tych samych węzłach sieci. Metoda ta doskonale sprawdza się w przypadku modelowania małych białek, jednakże dla dużych molekuł wymaga znacznych zasobów obliczeniowych.

Budowa modelu

Do budowy modelu w oparciu o szablon wykorzystywany jest program Modeller, przypadkach modelowania *de novo* zaś - Rosetta. Dodatkowo w obu scenariuszach wykorzystywany jest także program CABS. Zastosowanie tego ostatniego, w którym łańcuch polipeptydowy zapisany jest w reprezentacji zredukowanej, pociąga za sobą konieczność odbudowy detali atomowych modelu. Specjalnie w tym celu powstał program BBQ$^{\mathbf{H}\alpha}$, który rekonstruuje szkielet główny białka. Grupy boczne odtwarzane są programem `scwrl`[35].

Analiza skupień

Wynikiem modelowania struktur białek jest zazwyczaj bardzo wiele modeli. Kolejnym etapem jest zatem analiza skupień, której celem jest wybór reprezentatywnych struktur. W pakiecie `BioShell` zaimplementowano algorytm hierarchiczny[36]. Program `clust`[10] rutynowo analizuje zbiory po kilkanaście a nawet kilkadziesiąt tysięcy struktur. Narzędzie to zostało napisane na potrzeby CASP6, a wykorzystano je także w konkursach CASP7 oraz CASP11. W pracy[37] posłużył do analizowania wyników dokowania krótkiego peptydu pochodzącego z białka C3D do domeny SH3-N. Procedura hierarchiczna jest również wykorzystana w programie `PsiBlastAnalyse` do grupowania podobnych sekwencji białek.

Procedury obliczeniowe dla struktur biomolekuł oraz metody numeryczne

Pakiet `BioShell` oferuje bardzo szeroki wachlarz algorytmów dedykowanych do analizy struktur białek. Oblicza ich nałożenia i uliniowienia. Wyznacza też różne parametry strukturalne: kąty płaskie i torsyjne, odległości, mapy kontaktów i mapy wiązań wodorowych. Pakiet `BioShell` udostępnia również szeroki zasób metod numerycznych i statystycznych, służących do przetwarzania danych, np. metody interpolacji, *bootstrap*, histogramy czy też estymatory jądrowe. Funkcjonalność ta jest nieczęsto spotykana w takich pakietach, a niektóre funkcje są unikalne dla pakietu BioShell. Być może dlatego właśnie inne laboratoria wykorzystujące ten pakiet w swoich badaniach, przede wszystkim sięgają po procedury operujące na strukturach biomolekuł[38,39].

Potencjały statystyczne

[10] w oryginalne pracy opublikowano go pod nazwą HCPM - Hierarchical Clustering of Protein Models

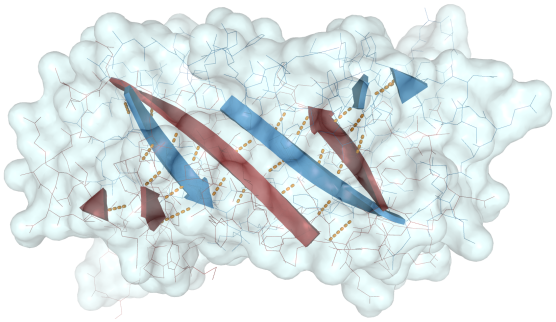
[35] Dunbrack, J. & Karplus, M. *Journal of Molecular Biology* **230,** 543–574 (1993).
[36] Gront, D. & Kolinski, A. *Bioinformatics* **21,** 3179–3180 (2005).
[37] Gront, D. *et al. Acta Pol Pharm* **63,** 436–438 (2006).
[38] Kim, H. & Kihara, D. *Proteins* **82,** 3255–3272 (2014).
[39] Chruszcz,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

Funkcjonalność ta została wykorzystana kilkukrotnie do wyprowadzania potencjałów statystycznych. Dla przykładu, w pracy **H7** zaproponowano potencjały statystyczne opisujące lokalną (tj. obejmującą kilka następujących po sobie reszt aminokwasowych) geometrię łańcucha głównego w konkretnej rodzinie białek. Potencjały statystyczne są powszechnie wykorzystywane w modelowaniu struktur biomolekuł. W typowym ujęciu opisują one prawdopodobieństwo pojawienia się pewnej własności strukturalnej (np. kontaktu między atomami), uzależnione od typów aminokwasów. Dla przykładu, w programie CABS wykorzystywane są potencjały oceniające odległość $R_{15}(A_2, A_3)$, czyli między każdym i-tym a $i+4$-tym węglem α wzdłuż łańcucha głównego białka. Oprócz wspomnianej odległości, funkcja ta zależy od typu aminokwasów na pozycjach $i+1$ oraz $i+3$. Raz wyprowadzona, może być wykorzystywana do modelowania białka o dowolnej sekwencji.

W przypadku takich potencjałów opisujących rodzinę białek, funkcja energii zależy od pozycji w sekwencji modelowanego białka. Na tej samej zasadzie oparte są m.in. profile sekwencyjne[30] oraz biblioteki fragmentów, wykorzystywanych w modelowaniu struktur białek[40]. Podobnie jak profil czy fragmenty, potencjał taki musi być wyprowadzony oddzielnie dla każdej sekwencji modelowanego białka. Dodatkowo, bazy danych muszą zawierać informację o strukturach białek należących do tej samej rodziny, co białko badane. Uzyskany potencjał dostarcza jednak znacznie dokładniejszego opisu lokalnej konformacji łańcucha polipeptydowego.

Analiza topologii białek splecionych[11]

Pakiet `BioShell` został także wykorzystany do analizy struktury białka AF2331[41] (depozyt w PDB: `2FDO`). Białko to jest nietypowym reprezentantem klasy $\alpha+\beta$, bowiem jedną z dwóch jego β-kartek tworzą naprzemiennie fragmenty łańcuchów A oraz B. W ten sposób oddziaływania pomiędzy łańcuchami odpowiadają za jego znaczną część struktury drugorzędowej. Z ryc. 4, przedstawiającej wybrane β-wstęgi tego białka można odczytać, że podążając za wiązaniami wodorowymi β-kartki, odwiedzamy łańcuchy `AABABABB`, zmieniając kod łańcucha aż pięciokrotnie. Oczywistym pytaniem, które nasunęło się w trakcie badań nad `2FDO` było, czy topologia taka pojawiła się już w znanych strukturach.



Rysunek 4: **Spleciona β-kartka białka** `2FDO`. Tworzy je osiem wstęg z łańcuchów `A` oraz `B` depozytu, oznaczonych odpowiednio kolorami niebieskim i czerwonym. Pomarańczowe przerywane linie ilustrują sieć wiązań wodorowych.

[11] ang. *interdigitated*

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

Odpowiedź na nie przyniósł krótki skrypt napisany w środowisku BioShell. Skrypt ten wczytywał plik PDB, z którego odczytywał informację o β wstęgach, i obliczał sieć wiązań wodorowych. Następnie budował graf, którego wierzchołkami były β wstęgi pokolorowane wg kodu łańcucha a krawędziami - wiązania wodorowe. Ostatecznym wynikiem była najdłuższa możliwa w tym grafie ścieżka wiodąca przez węzły o naprzemiennie różnych kolorach. Analiza przeprowadzona na wszystkich znanych ówcześnie strukturach białek pokazała, że o ile pojedynczo-splecione β wstęgi (układ typu `ABA`) są często spotykane, to już topologie `ABAB` (trzykrotna zmiana kodu łańcucha) obserwowano około stukrotnie. Czterokrotna zmiana łańcucha (`ABABA`) pojawiła się tylko w 3 depozytach. Układ `ABABAB`, wykryty w AF2331, pojawił się jeszcze tylko w depozycie `2HJ1` i był najdłuższym zaobserwowanym.

Statystyczny opis stanów w rozkładzie kanonicznym

Modelowanie biomolekuł często prowadzone jest tak, aby symulowany układ opisywany był zespołem kanonicznym. Jest to szczególnie łatwe, gdy procedura modelowania oparta jest o metodę Monte Carlo wg wspomnianego już we wstępie schematu Metropolisa. Wynikiem modelowania jest wtedy zbiór konformacji układu o energiach opisanych rozkładem Boltzmanna. W pakiecie `BioShell` zaimplementowano *metodę ważonych histogramów*[42], dzięki której możliwe jest wyznaczenie sumy statystycznej $\mathcal{Z}(T)$ badanego układu. Danymi wejściowymi do programu `MultiHist`**H4** pakietu `BioShell` są wartości energii $\mathcal{E}$ zaobserwowane podczas symulacji w różnych temperaturach a wynikiem - suma statystyczna $\mathcal{Z}(T)$ oraz gęstości stanów $\Omega(\mathcal{E})$. Program `StatPhys` z kolei wczytuje $\Omega(\mathcal{E})$ oraz zmierzone *observable*, dla których wyznacza średnie kanoniczne w dowolnej temperaturze. Programy te wykorzystano do badania sieciowych modeli prostych polimerów**H3** a także do analizowania wyników symulacji białek w zredukowanym modelu CABS**H2** (metodą Monte Carlo) a także pełnoatomową dynamiką molekularną[43].

Symulacje te prowadzone były metodą wymiany replik Monte Carlo[44] (REMC[12]). W metodzie tej, dzięki jednoczesnemu modelowaniu replik tego samego układu w wielu temperaturach, eksploracja przestrzeni stanów jest znacznie wydajniejsza. Wymiana replik, czyli przenoszenie kopii układu pomiędzy temperaturami, pozwala na względnie łatwe przekraczanie barier energetycznych. Kluczowy jest tu jednak odpowiedni wybór zbioru temperatur $\{T_i\}$, w których symulowane są poszczególne repliki. W literaturze opisano wiele rozwiązań[45,46], żadne z nich jednak nie gwarantuje optymalnego przepływu replik pomiędzy temperaturami. W pracy **H3** zaproponowano nowatorski sposób wyboru temperatur do prowadzenia symulacji. Opiera się on na spostrzeżeniu,

[12] Replica Exchange Monte Carlo

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

że prawdopodobieństwo zamiany replik $P(T_1 \rightarrow T_2)$ pomiędzy temperaturami T_1 i T_2 zależy od nakrywania się gęstości stanów w tych temperaturach. Prawdopodobieństwo to można obliczyć, o ile znana jest funkcja gęstości stanów $\Omega(\mathcal{E})$. Ustalanie temperatur T_i rozpoczyna się więc od symulacji REMC w której budowane jest pierwsze przybliżenie do $\Omega(\mathcal{E})$ badanego układu. Na podstawie tej gęstości stanów oblicza się (poprzez numeryczne całkowanie) taki zestaw temperatur, w którym prawdopodobieństwa $P(T_i \rightarrow T_{i+1})$ były równe dla każdego i. Można pokazać, że takie kryterium gwarantuje najszybszą podróż replik pomiędzy temperaturami.

Inżynieria białek in silico

Najnowszym zastosowaniem pakietu `BioShell` jest racjonalna inżynieria białek. W pracy **H12.** opisano udaną modyfikację aldolazy treoninowej z bakterii *Thermotoga maritima* mającą na celu zwiększenie stabilności tego enzymu. Białko to w żywych organizmach rozkłada treoninę do glicyny i metanalu i w formie aktywnej jest homotetramerem. Dlatego w pracy **H12.** za cel obrano wzmocnienie oddziaływań pomiędzy łańcuchami. Pierwszym etapem części teoretycznej projektu było zgromadzenie wszystkich znanych sekwencji homologicznych. Wykorzystano do tego program `PsiBlastSearch`, który uruchamia obliczenia programem `PsiBlast` z różnymi ustawieniami. Wyniki przeanalizowano narzędziem `PsiBlastAnalyse` za pomocą którego wybrano 132 reprezentatywne sekwencje. Sekwencje te wykorzystano jako zapytania do kolejnej serii poszukiwań w bazie danych. Ostatecznie znaleziono ponad 100 000 podobnych sekwencji w tym około 45% w drugiej rundzie. 52 z tych sekwencji pochodziło z organizmów termofilnych. Na ich podstawie stworzono uliniowienie wielu sekwencji, pokazujące zmienność aminokwasową na każdej pozycji w białku. Program `StrCalc` wykorzystano do przeanalizowania struktury krystalograficznej wyjściowego enzymu (depozyt PDB: `1LW5`). Na podstawie odległości między atomami i wzajemnej orientacji przestrzennej łańcuchów bocznych reszt aminokwasowych wytypowano potencjalne miejsca do wprowadzenie nowych oddziaływań: mostków jonowych i wiązań disulfidowych.

Struktura czwartorzędowa aldolazy treoninowej jest dość szczególna: pary łańcuchów `B` i `C` oraz `A` i `D` tetrameru kontaktują się resztami o tych samych numerach. Dla przykładu, prolina 56 z łańcucha `A` jest oddalona tylko o 4.5Å od P56 w łańcuchu `D`. Dzięki temu możliwe było wprowadzenie do tetrameru aż czterech reszt cysteiny i jednocześnie *dwóch* wiązań disulfidowych za pomocą jednej tylko mutacji. Po wstępnej analizie do wprowadzania mutacji wybrano 18 pozycji. Modele struktur mutantów obliczono programami `Modeller` i `Rosetta`. Ostatecznie 10 najlepszych mutantów przetestowano eksperymentalnie. Dwa z nich (`P56C` oraz `A21C`) wykazują znacząco większą stabilność niż białko dzikie.

5 Omówienie pozostałych osiągnięć naukowo-badawczych

Habilitant jest również jednym z dwudziestu siedmiu *Principal Investigators* zrzeszonych w *Rosetta Commons*[13] i bierze czynny udział w rozwoju pakietu oprogramowania naukowego Rosetta. Na pakiet ten składa się wiele programów, liczących obecnie w sumie ponad 2,5 miliona linii kodu źródłowego napisanego w języku C++. Oprogramowanie to służy do modelowania struktur białek i RNA (zarówno *de novo* jak i w oparciu o szablony), rozwiązywania struktur biomolekuł z fragmentarycznych danych eksperymentalnych (głównie NMR oraz map gęstości elektronowej EM) oraz do projektowania nowych białek. Główny wkład habilitanta to stworzenie algorytmu do budowania biblioteki fragmentów[40], niezbędnej w przypadku modelowania struktur białek tą metodą. Dodatkowo, zaimplementował on umożliwiające wykorzystanie danych SAXS w modelowaniu.

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