Uniwersytet Warszawski
Wydział Matematyki, Informatyki i Mechaniki

Paweł Daniluk

Analiza podobieństwa struktur przestrzennych białek przy użyciu deskryptorów lokalnej struktury

rozprawa doktorska

Promotor rozprawy
prof. dr hab. Bogdan Lesyng
Wydział Fizyki
Uniwersytet Warszawski

Sierpień 2011

Oświadczenie autora rozprawy

Oświadczam, że niniejsza rozprawa została napisana przeze mnie samodzielnie.

data podpis autora rozprawy

Oświadczenie promotora rozprawy

Niniejsza rozprawa jest gotowa do oceny przez recenzentów.

data podpis promotora rozprawy

Streszczenie

W niniejszej rozprawie rozważamy zastosowanie deskryptorów lokalnej struktury do identyfikowania podobieństw, uliniawiania i klasyfikacji struktury białek. Uliniowienie struktur rozumiane jest jako dowolne odwzorowanie pomiędzy aminokwasami badanych struktur. Dopuszczalne są przestawienia sekwencyjne oraz odkształcenia strukturalne.

Przedstawiamy formalną definicję deskryptorów, ich podobieństwa, uliniowień i multi-uliniowień białek złożonych z uliniowień mniejszych wycinków struktury. Rozważamy problem znajdowania optymalnych uliniowień i dowodzimy NP-zupełności jego wariantów. Mimo teoretycznej wykładniczej złożoności obliczeniowej przedstawionych problemów podajemy wydajne algorytmy heurystyczne. Problem obliczania optymalnych uliniowień sprowadzamy do wyszukiwania maksymalnej kliki, rozwiązując go przy pomocy metody podziałów i ograniczeń, Monte-Carlo z wymianą replik oraz heurystyki opartej na twierdzeniu Motzkina-Strausa. Do znajdowania multi-uliniowień stosujemy algorytm ewolucyjny.

Zaimplementowane algorytmy zostały przetestowane na popularnych zbiorach uliniowień referencyjnych: SISY, RIPC oraz SISY-multiple i porównane z innymi publicznie dostępnymi wiodącymi metodami. Algorytm uliniawiający pary struktur na najtrudniejszym zbiorze testowym RIPC uzyskał średnią dokładność 77% (druga pod względem jakości metoda osiąga dokładność 60%). Algorytm znajdowania multi-uliniowień uzyskuje wyniki na poziomie najlepszych konkurencyjnych metod. Stworzony został również serwis internetowy (http://bioexploratorium.pl/EP) udostępniający opisane metody.

Słowa kluczowe

lokalne deskryptory struktury, uliniowienia strukturalne, porównywanie struktur, NP-zupełność, fragmenty strukturalne, przestawienia sekwencyjne, permutacje cyrkularne, multi-uliniowienia, wyszukiwanie klik, twierdzenie Motzkina-Strausa, algorytmy ewolucyjne

Klasyfikacja ACM

F.1.3 Complexity Measures and Classes, F.2.2 Nonnumerical Algorithms and Problems, J.3 Life and Medical Sciences

Abstract

In this thesis we discuss application of local descriptors of protein structure to identifying similarities, computing alignments, and carrying out classification of protein structures. Definition of structural alignment is expanded to cover any arbitrary mapping of residues. Segment swaps and structural flexibility are allowed.

We present formal definitions of descriptors, their similarity, structural alignments and multi-alignments assembled with alignments of small structural fragments. We consider a problem of finding optimal alignments and prove NP-completeness of its several variants. Despite their intractability we give efficient heuristic algorithms, which can be used to solve the aforementioned problems. We apply clique searching techniques to compute optimal alignments such as branch-and-bound algorithm, replica exchange Monte-Carlo, and a heuristic based on the Motzkin-Straus theorem. We use an evolutionary algorithm to compute multi-alignments.

Implementations have been tested on popular reference alignment sets: SISY, RIPC and SISY-multiple and compared with other leading, publicly available methods. The pairwise alignment algorithm achieves an accuracy of 77% on the most difficult RIPC set (the second best method tested achieves 60%). The multiple alignment algorithm achieves results on par with other leading methods. Presented methods are available online (http://bioexploratorium.pl/EP).

Keywords

local descriptors of protein structure, structural alignment, structure comparison, NP-completeness, structural fragments, segment swaps, circular permutations, multiple alignments, clique searching, Motzkin-Straus theorem, evolutionary algorithms

ACM Computing Classification

F.1.3 Complexity Measures and Classes, F.2.2 Nonnumerical Algorithms and Problems, J.3 Life and Medical Sciences

Pragnę podziękować:

promotorowi tej rozprawy prof. Bogdanowi Lesyngowi za cenne rady, stworzenie doskonałych warunków do pracy i nieskończone pokłady optymizmu,

Rodzicom za niezachwianą wiarę we mnie,

oraz:

mojej Żonie za miłość i niezmierzoną cierpliwość
i Łucji za to, że jest.

Wam tę pracę poświęcam.

Badania realizowane były przy wsparciu finansowym grantu MNiSW (N N301 243736) oraz projektu Biocentrum Ochota (POIG.02.03.00-00-003/09).

Spis treści

Rozdział 1

Wstęp

Białka są niezwykle zróżnicowaną klasą biopolimerów. Pełnią one fundamentalną rolę we wszystkich znanych organizmach żywych, biorąc udział w praktycznie każdym procesie życiowym komórki. Szczególnie istotną gałęzią badań szeroko rozumianej biochemii i biofizyki jest poznanie funkcji pełnionych przez konkretne białka i mechanizmów ich realizacji. To z kolei ma praktyczne znaczenie dla rozumienia między innymi procesów chorobotwórczych i przeciwdziałania im.

Powszechnie wiadomo, że biologiczna funkcja danego białka zależy w dużej mierze od jego struktury przestrzennej. Co więcej, można spodziewać się, że białka o podobnej strukturze mogą być spokrewnione ewolucyjnie i pełnić zbliżone funkcje. Ponieważ eksperymentalne określenie struktury białka jest niejednokrotnie łatwiejsze od rozpoznania funkcji, jaką pełni ono w żywym organizmie, odnajdowanie podobnej struktury o znanej funkcji może być użyteczną metodą przewidywania funkcji nowoodkrytego białka. Ponadto szczegółowe określenie podobieństwa struktur ze wskazaniem odpowiadających sobie regionów może pomóc w zidentyfikowaniu tych miejsc, które odpowiadają za realizowanie rozważanej funkcji.

Struktura białka zależy natomiast od jego sekwencji (w ustalonym środowisku fizyko-chemicznym). Jednakże białka o znacząco różnych sekwencjach mogą posiadać bardzo zbliżone struktury przestrzenne. Zjawisko to jest przejawem procesów molekularnej ewolucji, które w procesie selekcji prowadzą do struktur o określonych funkcjonalnych właściwościach, które muszą być dostatecznie stabilne z punktu widzenia fizyki. W szczególności możliwe jest, że wskutek ewolucji konwergentnej dwa białka niezależnie osiągną tę samą strukturę[13]. Możliwy jest również przypadek bardziej złożony, gdy struktury różnią się kolejnością występowania elementów w odpowiadających im sekwencjach i jednocześnie zachowują tę samą "architekturę" i funkcję[68].

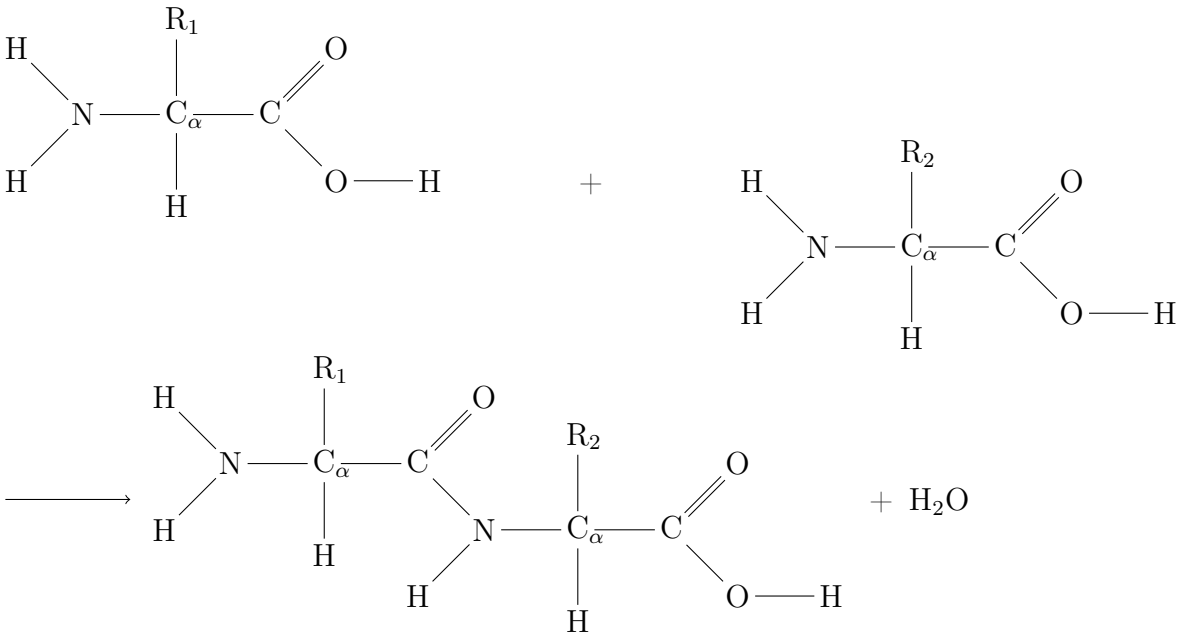
Porównywanie oraz możliwość klasyfikacji struktur przestrzennych białek z wykorzystaniem dobrze zdefiniowanych, zalgorytmizowanych procedur, ma zatem istotne znaczenie z punku widzenia pełniejszego rozumienia mechanizmów funkcjonowania białek, mechanizmów ewolucji molekularnej oraz związków między procesami ewolucji molekularnej a prawami fizyki, które wyznaczają warunki konieczne dla istnienia obserwowanych w przyrodzie struktur. Z tego punktu widzenia analiza podobieństwa struktur wydaje się być nawet istotniejsza od porównywania sekwencji.

W niniejszej rozprawie prezentujemy klasę metod służących do identyfikowania podobieństw pomiędzy parami struktur białek, jak również w większych zbiorach struktur. Do ich opracowania zastosowaliśmy formalizm lokalnych deskryptorów struktury białek[34]. Lokalne deskryptory są pewnymi wycinkami struktury, obejmującymi molekularne otoczenie wskazanego aminokwasu. Posłużą one do identyfikowania lokalnych podobieństw pomiędzy strukturami, które następnie posłużą do budowania globalnych odpowiedniości pomiędzy nimi.

W drugim rozdziale rozprawy podajemy formalną definicję deskryptorów i ich podobieństwa oraz dowód NP-zupełności problemu porównywania dwóch deskryptorów. Następnie w trzecim rozdziale definiujemy problem znajdowania optymalnego uliniowienia pary struktur i dowodzimy jego NP-zupełności. Przedstawiamy również wydajne algorytmy heurystyczne uliniawiające pary struktur i wyniki ich testów. Wreszcie czwarty rozdział zawiera opis problemu znajdowania optymalnych multi-uliniowień, analizę jego złożoności, opis zaprojektowanej procedury oraz analizę jej skuteczności. Rozdział drugi bazuje na metodologii opisanej w pracach[34, 33]. Podany w nim formalizm matematyczny oraz dowody twierdzeń zostały opracowane przez autora rozprawy. Algorytm porównywania deskryptorów jest rozszerzeniem metody zaprezentowanej w pracy[34], zaś jego implementacja została od podstaw wykonana przez autora rozprawy i była wykorzystana w pracach[41, 42, 66, 12]. Rozdziały trzeci i czwarty stanowią oryginalny wkład autora rozprawy.

1.1. Sekwencja i struktura białek

Niezwykle bogaty i zróżnicowany świat białek znajduje swój początek w sekwencjach RNA i DNA odpowiadających im genów. Z pewnym uproszczeniem można przyjąć, że każdy gen w żywym organizmie koduje pewne białko. Na podstawie sekwencji nukleotydów w procesie transkrypcji i translacji następuje w rybosomie synteza łańcucha reszt aminokwasowych tworzących białko, który następnie przyjmuje charakterystyczną dla



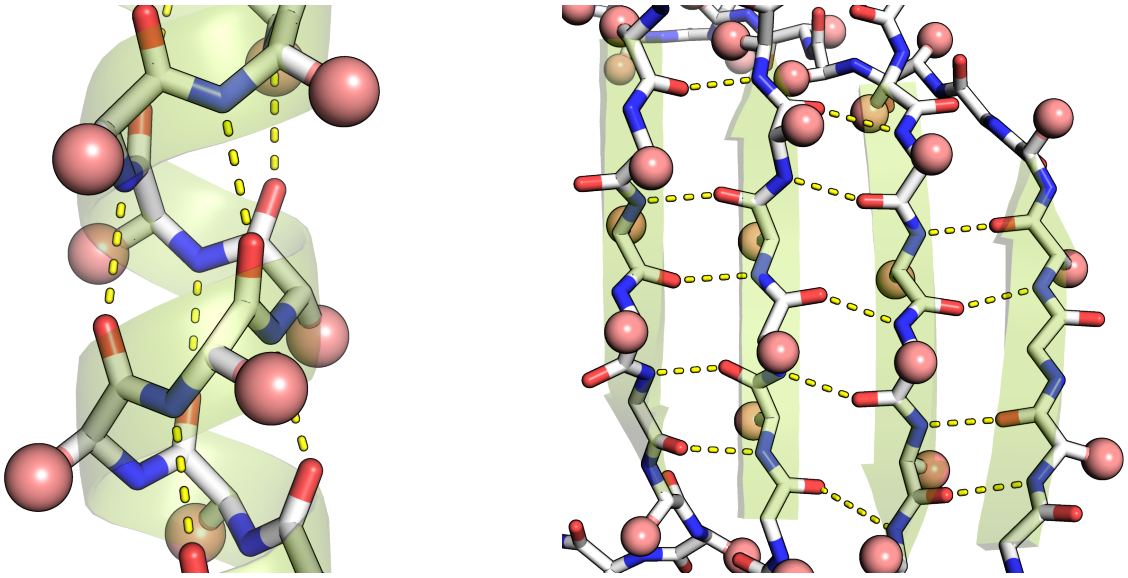
Rysunek 1.1: Aminokwasy mogą łączyć się ze sobą za pośrednictwem wiązań peptydowych tworząc biopolimery o dowolnej długości (R_1 i R_2 oznaczają łańcuchy boczne aminokwasów)

danego białka strukturę przestrzenną.

Białka są łańcuchami aminokwasów połączonych ze sobą wiązaniami peptydowymi. Aminokwas jest organicznym związkiem chemicznym posiadającym grupę karboksylową ($-COOH$) oraz grupę aminową ($-NH_2$), które w przypadku aminokwasów biogennych przyłączone są do atomu węgla C^α. Ponadto do węgla C^α może być przyłączony łańcuch boczny (R) zwany również resztą aminokwasową. Jest 20 biogennych aminokwasów, które różnią się między sobą właściwościami łańcuchów bocznych. Aminokwasy mogą tworzyć wiązania peptydowe i w ten sposób łączyć się w łańcuch. Wiązanie powstaje wtedy pomiędzy grupą karboksylową poprzednika a grupą aminową następnika (rys. 1.1). Białka są zróżnicowane pod względem wielkości. Pojedynczy łańcuch może liczyć od kilkudziesięciu do kilkudziesięciu tysięcy aminokwasów[1].

Cząsteczka białka po zsyntetyzowaniu ulega procesowi zwijania (ang. *folding*), który w ustalonym środowisku prowadzi do osiągnięcia konformacji natywnej. Proces ten jest powtarzalny, zaś struktura natywna zależy od sekwencji aminokwasów. Reszty aminokwasowe różnią się między sobą szeregiem cech fizyko-chemicznych takich jak wielkość, ładunek elektrostatyczny, kwasowość lub zasadowość, czy występowanie pier-

[1]Największe znane białko – tytyna liczy w wariancie występującym u człowieka 34350 aminokwasów (ponad pół miliona atomów).



(a) (b)

Rysunek 1.2: Elementy struktury drugorzędowej stabilizowane są przez wiązania wodorowe pomiędzy atomami tlenu(kolor czerwony) i azotu(niebieski). W helisach α (a) wiązania występują pomiędzy co trzecimi aminokwasami. W arkuszach β (b) występują pomiędzy aminokwasami leżącymi na sąsiadujących ze sobą wstęgach. Łańcuchy boczne są dla uproszczenia reprezentowane przez różowe kule.

ścienia aromatycznego. Jednym z najistotniejszych kryteriów podziału jest hydrofobowość. Niektóre reszty aminokwasowe charakteryzuje "awersja" do cząsteczek wody. Przeciwieństwem hydrofobowości jest polarność. Reszty polarne mają nierównomiernie rozmieszczone ładunki elektrostatyczne, co powoduje powstanie elektrycznego momentu dipolowego. Takie aminokwasy cechuje "powinowactwo" do wody. Oddziaływania aminokwasów ze środowiskiem wodnym oraz między sobą są kluczowe w zwijaniu się białka. Reszty hydrofobowe są zazwyczaj skierowane do środka cząsteczki, a polarne na zewnątrz.

Struktura białka jest stabilizowana przez różnego rodzaju wiązania pomiędzy łańcuchami bocznymi aminokwasów. W szczególności znaczącą rolę odgrywają wiązania wodorowe występujące pomiędzy atomami tlenu w grupach karbonylowych (>C=O) a atomami azotu. Wiązania te stabilizują helisy α, arkusze β (rys. 1.2) i inne podobne motywy strukturalne, które występują w niemal wszystkich strukturach białkowych. Często rozważa się hierarchiczny opis struktury białka, w którym najniższy poziom stanowi sekwencja aminokwasów (struktura pierwszorzędowa). Poziom organizacji obejmujący wspomniane wyżej elementy(helisy α, arkusze β) nazywa się strukturą drugorzędową.

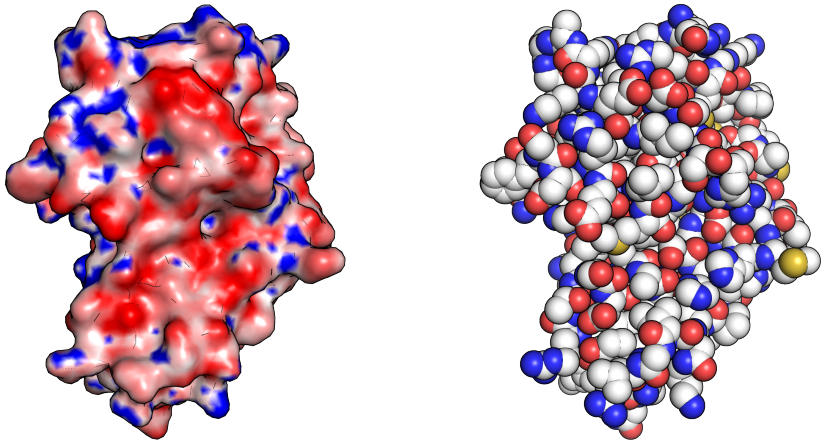
Interakcje pomiędzy aminokwasami bardziej odległymi w sekwencji odpowiadają za wzajemne ułożenie elementów struktury drugorzędowej i konformację całego łańcucha polipeptydowego tworząc strukturę trzeciorzędową. Zazwyczaj funkcjonalna cząsteczka białka składa się z kilku łańcuchów, a ich wzajemne położenie nazywa się strukturą czwartorzędową.

Niezależnie od liczby łańcuchów w strukturze czwartorzędowej białka, zazwyczaj dokonuje się podziału struktury na domeny białkowe. Domena jest częścią struktury, która wyodrębniona z białka posiada samoistną zdolność zachowania swojej konformacji. Zazwyczaj domena wycięta z łańcucha białka może również samoistnie zwinąć się do konformacji natywnej. Domeny można traktować jako cegiełki będące budulcem białek. Często się zdarza, że takie same domeny obserwowane są w różnych białkach. Ponieważ niejednokrotnie domeny samodzielnie mogą realizować pewne funkcje biologiczne, jest to mechanizm pozwalający na powstawanie białek o nowych funkcjonalnościach, będących kombinacją funkcji domen, z których się składają.

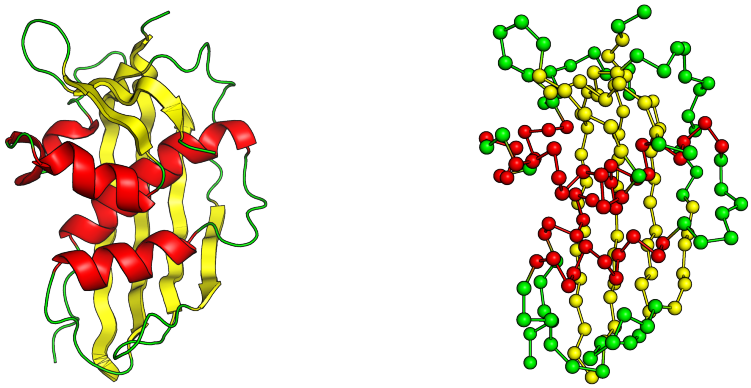
Hierarchizacja cech struktury białkowej znacząco ułatwia jej wizualizację. Człowiek nie jest zdolny obserwować ani analizować układu złożonego z kilkudziesięciu tysięcy atomów. Konieczne było zatem stworzenie uproszczonych sposobów wizualizacji struktur. Dosyć typowe jest posługiwanie się tzw. obrazem wstążkowym, który pokazuje kształt i położenie elementów struktury drugorzędowej. Model pełnoatomowy stosuje się zazwyczaj w sytuacji, gdy konieczne jest obserwowanie wycinków struktury odpowiedzialnych za interakcje z innymi białkami lub mniejszymi cząsteczkami (rys. 1.3).

Eksperymentalne określenie struktury białka jest dość złożonym eksperymentem. Najczęściej stosowaną metodą jest krystalografia rentgenowska, która polega na badaniu obrazu dyfrakcyjnego powstającego wskutek "prześwietlania" kryształu białka przy pomocy promieni rentgenowskich. Niestety nie zawsze jest możliwe uzyskanie kryształu badanego białka o zadowalającej jakości. Ponadto niejednokrotnie konformacja białka w krysztale różni się od występującej w formie rozpuszczonej. Alternatywną metodą jest spektroskopia magnetycznego rezonansu jądrowego oparta na wzbudzaniu spinów jąder atomów wodoru, węgla C^{13} lub azotu N^{15} w silnych polach magnetycznych, przy pomocy szybkich zmian drugiego pola magnetycznego, a następnie np. obserwowaniu powrotu do stanu równowagi termodynamicznej. Wynikiem eksperymentu jest zbiór odległości pomiędzy "obserwowanymi" atomami, na podstawie których można odtworzyć strukturę trzeciorzędową. Zazwyczaj w rezultacie uzyskuje się zespół możliwych konformacji. Zaletą tej metody jest możliwość badania białek rozpuszczonych w wodzie.

Obecnie w największej publicznie dostępnej bazie struktur białek Protein Data



(a) (b)



(c) (d)

Rysunek 1.3: Różne sposoby wizualizacji struktury przestrzennej białka: powierzchnia dostępna dla środowiska wodnego (a), model pełnoatomowy (b), schematyczny rysunek elementów struktury drugorzędowej (c), reprezentacja zdegenerowana (same atomy C_α) (d).

Bank[10] zdeponowane są ponad 73 tysiące eksperymentalnie pozyskanych struktur białek. Ponadto istnieją bazy danych, których podstawowym przeznaczeniem jest zdefiniowanie taksonomii struktur na podobieństwo systematyki gatunków w biologii. Szczególnie istotnym serwisem tego typu jest SCOP (*Structural Classification of Proteins*)[52]. W bazie SCOP domeny białkowe dzielone są na klasy na podstawie zawartości elementów struktury drugorzędowej, a następnie na foldy, które grupują struktury o zbliżonym układzie przestrzennym elementów struktury drugorzędowej. Fold może z kolei obejmować jedną lub więcej super-rodzin zawierających struktury spokrewnione ewolucyjnie i rodzin struktur o homologicznych sekwencjach. SCOP jest tworzony przy istotnym wkładzie wiedzy eksperckiej. Po ostatniej aktualizacji zawiera 3902 rodziny struktur pogrupowanych w 1195 foldów. Serwisem konkurencyjnym wobec SCOP jest tworzony półautomatycznie serwis CATH[55]. Nazwa CATH pochodzi od zastosowanych poziomów klasyfikacji:

- Klasa (ang. *Class*) – białka o zbliżonej zawartości elementów struktury drugorzędowej
- Architektura (ang. *Architecture*) – białka o podobnym układzie elementów struktury drugorzędowej, ale niekoniecznie ułożonych w tej samej kolejności w sekwencji
- Topologia (ang. *Topology*) – białka o znaczącym podobieństwie strukturalnym, lecz niekoniecznie powiązane ewolucyjnie – odpowiednik foldu w hierarchii SCOP
- Rodzina homologiczna (ang. *Homologous superfamily*) – białka o sekwencjach spokrewnionych ewolucyjnie (odpowiednik superrodziny w bazie SCOP).

Na szczególną uwagę zasługuje nieobecne w hierarchii SCOP pojęcie architektury. Istnieją bowiem białka o bardzo podobnej strukturze przestrzennej, które różnią się topologią[44].

1.2. Podobieństwo białek

Podstawową metodą określania stopnia podobieństwa sekwencji białek jest algorytm Needlemana-Wunscha[53]. Jest on oparty na programowaniu dynamicznym i ma złożoność obliczeniową $O(mn)$, gdzie m i n są długościami porównywanych sekwencji. Aby go zastosować konieczne jest określenie miary podobieństwa pomiędzy aminokwasami (np. w postaci macierzy substytucji[18, 30]) oraz kary za pominięcie aminokwasu

w dopasowaniu. Wynikiem obliczenia jest odwzorowanie pomiędzy aminokwasami o maksymalnej mierze zwane uliniowieniem (ang. *alignment*). Jeżeli s_1 i s_2 są porównywanymi sekwencjami, to ich uliniowienie ma postać pary sekwencji $\left\langle s_1^{\#}, s_2^{\#} \right\rangle$, która spełnia następujące warunki:

- $s_1^{\#}, s_2^{\#} \in (\Sigma \cup \{-\})^*$ [2], gdzie Σ jest zbiorem aminokwasów, a – znakiem specjalnym, który nazywać będziemy spacją,
- $|s_1^{\#}| = |s_2^{\#}|$,
- Sekwencje s_1 i s_2 można otrzymać z $s_1^{\#}$ i $s_2^{\#}$ przez pominięcie spacji,
- $\nexists_i s_1^{\#}(i) = s_2^{\#}(i) = -$, gdzie $s(i)$ oznacza symbol występujący na i-tym miejscu w sekwencji s.

Pewną wadą opisanego podejścia jest brak gwarancji, że tak zdefiniowana miara podobieństwa sekwencji będzie proporcjonalna do prawdopodobieństwa otrzymania jednej sekwencji z drugiej w drodze procesów ewolucji. Na przykład znacznie bardziej prawdopodobne jest pominięcie w kopiowaniu ciągłego fragmentu liczącego kilkanaście aminokwasów niż kilkunastu pojedynczych aminokwasów rozrzuconych w sekwencji białka. Opracowano wiele udoskonaleń pozwalających otrzymywać uliniowienia posiadające sens biologiczny, m.in. poprzez zapewnienie minimalnej długości fragmentów uliniowionych bez wstawiania spacji[45, 1], czy też rozważanie kontekstu sekwencyjnego aminokwasów[23].

Jak wspomnieliśmy powyżej, białka o podobnych (homomologicznych) sekwencjach mają zazwyczaj podobne struktury przestrzenne. Podobieństwo sekwencji wskazuje na ewolucyjne pokrewieństwo rozważanych białek. Zakłada się bowiem, że procesy ewolucji występujące na poziomie molekularnym polegają na stopniowym pojawianiu się zaburzeń w sekwencji genu kodującego białko, co z kolei prowadzi do zmian w jego strukturze i pojawianiu się zdolności do pełnienia nowych funkcji. Mutacje w sekwencji genu są niemożliwe do uniknięcia. Szczęśliwie większość z nich nie ma istotnego wpływu na naturę białka. W przeciwnym wypadku w przytłaczającej większości przypadków okazywałyby się one śmiertelne, a proces rozmnażania się organizmów byłby w zasadzie niemożliwy. Porównywanie struktur białek ma zatem największe znaczenie w sytuacji, gdy ich sekwencje nie są podobne.

Nie istnieje uniwersalna miara podobieństwa dwóch struktur. Jednak nader często używa się w tym kontekście odległości średniokwadratowej (ang. *Root Mean Squre*

[2]Symbol Σ^* oznacza zbiór wszystkich słów nad alfabetem Σ

Deviation – RMSD). Dla ustalonych, równolicznych zbiorów punktów odległość średniokwadratową definiuje się w następujący sposób:

$$RMSD(X,Y) = \min_{\substack{R \text{ - obrót w } \mathbb{R}^3 \\ T \in \mathbb{R}^3}} \sqrt{\frac{\sum_{i=1}^{n} |x_i - (Ry_i + T)|^2}{n}}$$

gdzie $X = (x_i)_{i=1}^n$, $Y = (y_i)_{i=1}^n$ są ciągami punktów z $\mathbb{R}^3$. Obliczenie RMSD wiąże się z koniecznością rozwiązania pozornie skomplikowanego problemu optymalizacyjnego. Istnieje jednak wydajny algorytm o złożoności kwadratowej ze względu na liczbę punktów[36, 37]. Podstawową wadą odległości średniokwadratowej w zastosowaniu do oceniania podobieństwa struktur jest niemożność zastosowania jej bez znajomości odwzorowania pomiędzy aminokwasami. W rzeczywistości sprowadza się to do równoczesnej optymalizacji dwóch zmiennych – liczba przyporządkowanych aminokwasów n jest maksymalizowana, przy równoczesnej minimalizacji odległości pomiędzy nimi. Przytoczyliśmy powyższy przykład, aby zademonstrować, że ocena podobieństwa strukturalnego powinna odbywać się w kontekście pewnego odwzorowania pomiędzy aminokwasami. Takie podejście jest również zgodne z założeniem, że poza ilościowym określeniem stopnia podobieństwa białek istotne jest jakościowe opisanie tego podobieństwa.

Wprawdzie struktura przestrzenna białka jest zdeterminowana przez jego sekwencję, jednakże nie znamy natury tej zależności. Dlatego przy obecnym stanie wiedzy struktura niesie ze sobą więcej informacji niż sama sekwencja. Niejednokrotnie zdarza się, że dwa białka mają podobny kształt, mimo że różnią się kolejnością występowania podobnych regionów w sekwencji[44]. Klasyczne rozumienie uliniowienia nie obejmuje takiego przypadku, a standardowe metody porównywania sekwencji nie pozwalają na wykrycie podobieństwa. Nierzadko jest ono jednak widoczne już przy wizualnej inspekcji rozważanych struktur (rys. 3.8). Najczęściej występują tzw. permutacje cyrkularne, które mogą powstawać m. in. wskutek duplikacji genów lub zmian zachodzących w łańcuchu białka podczas zwijania[68]. Powiemy, że jedno białko jest cyrkularną permutacją drugiego, jeżeli istnieje podział obydwu struktur na dwie podjednostki (odpowiednio $A_1{-}B_1$ i $A_2{-}B_2$) takie, że struktury $A_1{-}B_1$ i $B_2{-}A_2$ są podobne w sensie klasycznego uliniowienia (bez przestawień). Występują również bardziej skomplikowane przestawienia fragmentów w sekwencji, które powodowane są między innymi przez zmianę długości pętli łączących elementy struktury drugorzędowej, co z kolei wymusza ich przestawienia na skutek ograniczeń stereochemicznych[26].

Struktury białek nie powinny być traktowane jak obiekty sztywne. Wiele funkcji, które pełnią, realizowanych jest poprzez celowe zmiany konformacji przestrzennej[25,

19]. Również eksperymentalne procedury pomiaru struktury mogą dawać rozbieżne wyniki spowodowane naturą eksperymentu (por. rozdział 3.7.5). Dość istotne jest zatem uwzględnienie potencjalnych odkształceń podczas oceniania podobieństwa.

1.3. Metody porównywania struktur

Można wyróżnić dwie podstawowe metodologie porównywania struktur białek – globalną i lokalną. Pierwsza polega na iteracyjnym ulepszaniu uliniowienia i superpozycji struktur. Wychodząc od pewnego uliniowienia, oblicza się optymalne nałożenie odpowiadających sobie aminokwasów, a następnie w tak uzyskanej superpozycji identyfikuje się pary bliskich przestrzennie aminokwasów, traktując je jako uliniowienie w następnym kroku iteracji. Metody tego typu sprawdzają się, jeżeli w strukturach porównywanych białek nie występują odkształcenia i podobieństwo jest wystarczająco duże, aby proces był zbieżny do globalnego optimum.

Alternatywą dla podejścia globalnego są metody oparte na identyfikowaniu podobieństw lokalnych, z których w kolejnych fazach obliczenia budowane jest globalne uliniowienie. Jest wiele możliwości dekompozycji struktury i co za tym idzie, sposobów obliczania lokalnych podobieństw. Do najważniejszych należy badanie odległości pomiędzy aminokwasami (SSAP[56], DALI[32]), podobieństwa pojedynczych wycinków łańcucha głównego (CE[65]) lub elementów struktury drugorzędowej (VAST[46], MATRAS[38], GANGSTA[27]). Stosuje się również między innymi triangulację Delone (TOPOFIT[35]), sferyczne Fourierowskie rozwinięcia funkcji gęstości (3D-BLAST[48]), czy też pochodzące z badań nad widzeniem komputerowym metody haszowania obrazów (C_α-match[5]). Globalne uliniowienie jest następnie obliczane poprzez wybór możliwie licznego podzbioru lokalnych podobieństw, które są ze sobą zgodne. Definicja pojęcia zgodności oraz sposób przeszukiwania przestrzeni rozwiązań zależą od metody. Zazwyczaj konieczne jest, aby odpowiedniości pomiędzy aminokwasami wynikające z lokalnych podobieństw były tożsame na części wspólnej. Stosuje się również dodatkowe ograniczenia takie jak podobieństwo przekształceń koniecznych do nałożenia przestrzennego fragmentów[5] albo kolejność występowania w sekwencji białka. Do poszukiwania rozwiązania stosuje się algorytmy znajdowania izomorficznych podgrafów lub klik, klasteryzacji, programowanie dynamiczne i inne. Ze względu na złożoność obliczeniową związaną z kombinatorycznym rozmiarem przeszukiwanej przestrzeni zazwyczaj rezygnuje się z rozważania permutacji cyrkularnych i przestawień segmentów, nawet jeżeli metodologia pozwalałaby na ich znajdowanie. Taka sytuacja ma miejsce

w przypadku metody DALI i jej publicznie dostępnej implementacji DaliLite[32, 31]. Niekiedy rozważa się wprowadzanie "zawiasów", aby umożliwić porównywanie struktur, pomiędzy którymi występuje odkształcenie (FATCAT[71], FlexProt[62]). Pełniejsze przedstawienie aktualnego stanu wiedzy w tej dziedzinie można znaleźć w pracy[29].

Naturalnym rozszerzeniem problemu znajdowania podobieństw pomiędzy dwiema strukturami jest problem poszukiwania tzw. multi-uliniowień. Multi-uliniowienie można definiować na dwa sposoby: jako znajdowanie podstruktury występującej we wszystkich porównywanych białkach bądź znajdowanie wszystkich podobieństw z zastrzeżeniem, że zidentyfikowane odpowiedniości pomiędzy aminokwasami muszą być jednoznaczne. Istniejące metody znajdowania multi-uliniowień strukturalnych często są rozszerzeniem algorytmów obliczających uliniowienia par. Na podstawie podobieństwa wszystkich par porównywanych struktur budowane jest wtedy drzewo binarne, którego liściom przypisane są rozważane struktury. Następnie węzłom drzewa przypisuje się uliniowienia struktur bądź multi-uliniowień występujących w potomkach, które są obliczane w sposób analogiczny do uliniawiania par struktur. Ostatecznie w korzeniu drzewa znajduje się multi-uliniowienie wszystkich struktur (MUSTANG[40], POSA[72]). Stosuje się również strategię analogiczną do klasteryzacji hierarchicznej polegającą na scalaniu w każdym kroku iteracji pary najbardziej podobnych multi-uliniowień (Matt[50]). Istnieją również metody rozważające wszystkie struktury jednocześnie. Należy do nich algorytm MASS[20], który polega na znajdowaniu maksymalnych odpowiedniości pomiędzy elementami struktury drugorzędowej przy założeniu sztywnych, globalnych nałożeń. Natomiast algorytm MultiProt[63] polega na wyróżnieniu jednej struktury (pełniącej rolę sworznia) i uliniawianiu jej z pozostałymi. Rozważane są wszystkie wybory sworznia, a ostatecznym wynikiem jest maksymalne multi-uliniowienie.

Ponieważ nie istnieje uniwersalna funkcja miary podobieństwa struktur, obiektywna ocena jakości uliniowienia jest dość trudna. W przypadku projektowania testów metod porównywania struktur ważny jest również dobór testowego zestawu białek. W tej rozprawie będziemy wykorzystywać zbiory testowe i wzorcowe uliniowienia wykorzystane w pracach [49, 9]. Jakość obliczonego uliniowienia będziemy oceniać porównując je z wzorcowym i zliczając jednakowo uliniowione pary aminokwasów.

1.4. Kilka uwag o terminologii

W tej pracy często będzie pojawiać się pojęcie uliniowienia (ang. *alignment*) i superpozycji[3] (ang. *superposition*). Tradycyjnie uliniowienie odnosi się do odpowiedniości aminokwasów bądź nukleotydów w porównywanych sekwencjach, natomiast superpozycja jest przestrzennym nałożeniem trójwymiarowych struktur. Niemniej jednak nie jest nam znana metoda znajdowania superpozycji bez wskazania na którymś etapie obliczeń odpowiedniości pomiędzy aminokwasami. Dlatego w niniejszej rozprawie termin uliniowienie będzie się odnosił do odpowiedniości pomiędzy aminokwasami w białku niezależnie od sposobu jej obliczenia i zastosowania. Natomiast pojęcie superpozycji występować będzie wyłącznie w kontekście znajdowania przekształceń minimalizujących odległości pomiędzy punktami w przestrzeni.

Często zakłada się, że uliniowienie nie może zawierać przestawień aminokwasów. W tej rozprawie będziemy rozważać uliniowienia nie obarczone tym ograniczeniem. Zatem o ile z kontekstu nie będzie wynikało coś przeciwnego, pod pojęciem uliniowienia będziemy rozumieć dowolną różnowartościową funkcję częściową z jednego zbioru aminokwasów w drugi.

Wielokrotnie będziemy odwoływać się do bliskości i ciągłości sekwencyjnej. Aminokwasy są bliskie sekwencyjnie, jeżeli ich odległość w sekwencji białka jest niewielka, co implikuje również ich bliskość przestrzenną[4]. Wycinek struktury białka jest ciągły sekwencyjnie, jeżeli wszystkie aminokwasy, za wyjątkiem dwóch końcowych, tworzą dwa wiązania peptydowe. Bliskości sekwencyjnej można przeciwstawić bliskość strukturalną. Ponieważ w większości przypadków łańcuch peptydowy zawiera zakręty, aminokwasy odległe sekwencyjnie mogą znajdować się blisko w przestrzeni.

[3]Właściwie w języku polskim powinno się używać pojęcia nałożenie, lecz kalka językowa z angielskiego wydaje się w tym przypadku wygodniejsza.

[4]Odległość pomiędzy węglami C_α amikokwasów połączonych wiązaniem peptydowym wynosi ok. 3.8Å.

Rozdział 2

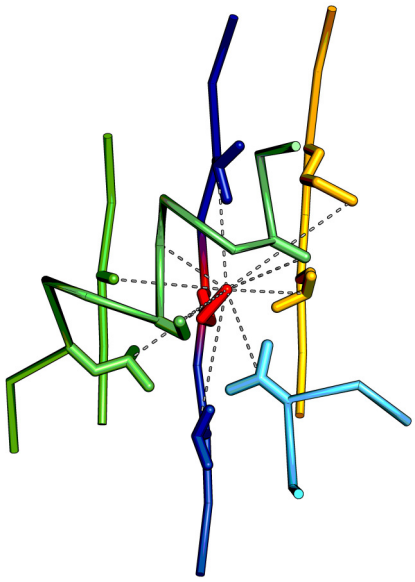
Lokalne deskryptory struktury białek

Jak powiedziano we wstępie, istnieje wiele metod analizy struktury białek, które zakładają podział struktury na mniejsze rozpatrywane osobno elementy. Większość tego typu metod wykorzystuje jednosegmentowe (ciągłe) wycinki struktury. Można zatem przyjąć, że jest to analiza posługująca się kryterium lokalności opartym o bliskość aminokwasów w sekwencji białka. Takie podejście jest atrakcyjne ze względu na niewielki stopień skomplikowania, ale stosowane nieostrożnie może prowadzić do wielu trudności. Zauważmy, że w przypadku typowych białek większość aminokwasów należy do regionów o uporządkowanej strukturze drugorzędowej. Można przyjąć, że niemal każde dwa wycinki o strukturze drugorzędowej tego samego typu są do siebie bardzo podobne. Zatem niewłaściwie stosowana metoda segmentów sekwencyjnych będzie miała trudność z odróżnieniem struktur posiadających elementy struktury drugorzędowej o podobnej długości i rodzaju. Naturalnym remedium tego problemu jest zaproponowanie opisu struktury przy użyciu elementów, które nie będą ograniczone do pojedynczych sekwencyjnie ciągłych wycinków. Jednym z takich formalizmów są rozważane w niniejszej rozprawie Lokalne Deskryptory Struktury Białek (*Local Descriptors of Protein Structure*).

2.1. Zarys metody

Lokalny Deskryptor Struktury[1] jest niewielkim fragmentem struktury białka, który może być rozumiany jako opis lokalnego otoczenia przestrzennego danego aminokwasu. W zasadzie można go zbudować dla każdego aminokwasu rozważanego białka. Aby to

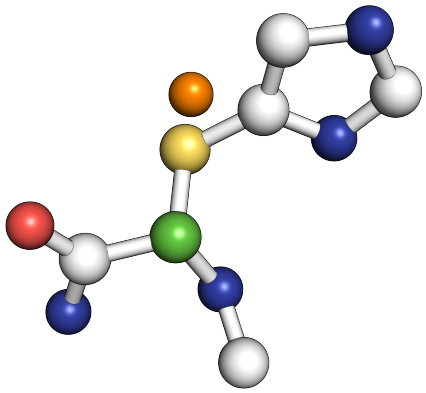
[1]W dalszych rozważaniach, o ile nie będzie to budziło wątpliwości, będziemy pisali w skrócie – deskryptor.



Rysunek 2.1: Przykładowy deskryptor zbudowany wokół aminokwasu 70 domeny białkowej `d1lg7a_`. Deskryptor `d1lg7a_#70` obejmuje 9 kontaktów (linie przerywane) pomiędzy aminokwasem centralnym (kolor czerwony), a aminokwasami będącymi środkami elementów. Niektóre pięcioaminokwasowe elementy nakładają się tworząc dłuższe segmenty (dwa odcinki wstęgi β i helisa α).

uczynić należy zidentyfikować aminokwasy, z którymi aminokwas założycielski jest *w kontakcie* (oddziałuje fizycznie). Następnie wokół wybranych w ten sposób aminokwasów budowane są elementy poprzez dołączenie dwóch aminokwasów poprzedzających i następujących na łańcuchu białka. Nakładające się elementy łączone są w segmenty (rys. 2.1). Promień deskryptora jest tym samym przybliżoną miarą zasięgu oddziaływań między aminokwasami. Natomiast sam deskryptor może być traktowany jako wycinek struktury znajdujący się wewnątrz nieregularnej powierzchni opowiadającej praktycznemu zasięgowi wpływu pojedynczego aminokwasu na resztę struktury. Deskryptor zatem w odróżnieniu od tradycyjnych fragmentów struktury opisuje sąsiedztwo w sensie przestrzennym, a nie sekwencyjnym. Kształt (zawartość) deskryptora zależy od interpretacji pojęcia kontaktu, czyli rodzaju i zasięgu oddziaływań, które uznawane są w danym przypadku za istotne[2].

[2] Należy zwrócić uwagę, że deskryptory stosowane np. w analizie inhibitorowych właściwości molekuł są skalarnymi lub wektorowymi, mierzalnymi lub obliczalnymi funkcjami określanymi na "przestrzeni struktur i/lub pól molekularnych". Tradycyjna nazwa deskryptorów struktury zastosowana w tej pracy odbiega więc od tej ogólnie stosowanego rozumienia tego pojęcia.



Rysunek 2.2: Histydyna z zaznaczonym punktem C^{β_x} (kolor pomarańczowy); pozostałe atomy: C^{α} – zielony, C^{β} – żółty, tlen – czerwony, azot – niebieski, niewymienione atomy węgla – biały.

2.2. Podstawowe definicje

2.2.1. Sekwencja i struktura białka

Niech Σ_P będzie zbiorem biogennych reszt peptydowych. Wtedy zbiór $\mathbb{A} = \Sigma_P \times \mathbb{R}^3 \times \mathbb{R}^3$nazwiemy **zbiorem aminokwasów**, zaś dowolny skończony różnowartościowy ciąg nad zbiorem aminokwasów **strukturą**.

Niech $a = \langle s, C^{\alpha}, C^{\beta} \rangle$ będzie aminokwasem. Kolejne elementy trójki a będziemy interpretować jako: rodzaj aminokwasu, współrzędne atomu C^{α}, współrzędne atomu C^{β}. Będziemy również rozważać punkt $C^{\beta_x} = C^{\alpha} + \left(C^{\beta} - C^{\alpha}\right) \frac{\left\|C^{\beta} - C^{\alpha}\right\| + 1\text{Å}}{\left\|C^{\beta} - C^{\alpha}\right\|}$ leżący na przedłużeniu wektora $\overrightarrow{C^{\beta} - C^{\alpha}}$ o 1Å (rys. 2.2).[3].

Niech $a^{(i)}$ będzie i-tym aminokwasem struktury S[4]. Pozycję aminokwasu a w strukturze S oznaczymy $n_S(a)$. Współrzędne atomów C^{α} i C^{β} oraz punktów C^{β_x} i R będziemy oznaczać odpowiednio $C^{\alpha}_{a^{(i)}}$, $C^{\beta}_{a^{(i)}}$, $C^{\beta_x}_{a^{(i)}}$ lub, o ile nie będzie to budzić wątpliwości, C^{α}_i, C^{β}_i, $C^{\beta_x}_i$. Funkcję $RMSD$ zdefiniowaną w rozdziale 1.2 rozszerzymy na ciągi aminokwasów:

$$RMSD(a_1 \dots a_n, b_1 \dots b_n) = RMSD(C^{\alpha}_{a_1} \dots C^{\alpha}_{a_n} C^{\beta_x}_{a_1} \dots C^{\beta_x}_{a_n}, C^{\alpha}_{b_1} \dots C^{\alpha}_{b_n} C^{\beta_x}_{b_1} \dots C^{\beta_x}_{b_n})$$

Zdefiniujemy również oznaczoną symbolem $\sim_S$ **relację sąsiedztwa sekwencyj-**

[3]Współrzędne punktu C^{β_x} można również obliczyć bez znajomości C^{β} na podstawie współrzędnych atomów C^{α}, C i N. Tak postępujemy w przypadku glicyny, która nie ma atomu C^{β}

[4]W dalszych rozważaniach jedynie wartości umieszczone w nawiasie w indeksie górnym oznaczają pozycję aminokwasu w strukturze.

nego w strukturze S:

$$b \sim_S c \Leftrightarrow n_S(b) + 1 = n_S(c)$$

Zbiór aminokwasów struktury S będziemy oznaczać symbolem $S^{\mathbb{A}}$.

Reprezentacja struktury białka wynikająca z powyższej definicji jest uproszczona. Pomijane są w niej współrzędne atomów reszt bocznych aminokwasów. Zabieg ten jest konieczny, aby możliwe było porównywanie konformacji przestrzennej struktur różniących się sekwencją. Należy jednak zauważyć, że model ten jest jedynie pozornie zdegenerowany, gdyż przy założeniu znajomości współrzędnych atomów C^α i C^β jest możliwa dość precyzyjna rekonstrukcja pozostałych współrzędnych przy użyciu tzw. bibliotek rotamerów.

Należy również pamiętać, że jedynie niektóre struktury z powyższej definicji spełniają warunki wynikające z fizycznych i chemicznych właściwości cząsteczek białek i tym samym mają szanse odpowiadać rzeczywistym strukturom białek.

2.2.2. Kontakty pomiędzy aminokwasami

Symetryczną relację $C \subseteq \mathbb{A} \times \mathbb{A}$ będziemy nazywać **kryterium kontaktu**. **Zbiorem kontaktów** w strukturze S nazwiemy obcięcie relacji C do zbioru $S^{\mathbb{A}} \times S^{\mathbb{A}}$: $C_S = C \cap \left(S^{\mathbb{A}} \times S^{\mathbb{A}}\right)$. Powiemy, że aminokwasy a_1 i a_2 **są w kontakcie** (przy kryterium kontaktu C), jeżeli $\langle a_1, a_2 \rangle \in C$.

W praktyce relacja C powinna być dobrana w sposób możliwie dobrze odzwierciedlający rzeczywiste oddziaływania fizykochemiczne aminokwasów. Aby zdefiniować kryterium kontaktu wykorzystywane w rzeczywistych obliczeniach, wprowadzimy pojęcie **kryterium α-kontaktu**. Kryterium α-kontaktu dla progu T_α nazwiemy relację:

$$C_\alpha\left(T_\alpha\right) = \{\langle a_1, a_2 \rangle \in \mathbb{A} \times \mathbb{A} | d_\alpha(a_1, a_2) \leq T_\alpha\}$$

gdzie $d_\alpha(a_1, a_2) = \left\|\mathrm{C}^\alpha_{a_1} - \mathrm{C}^\alpha_{a_2}\right\|$. **Kryterium β-kontaktu** dla progów T_β, T_Δ nazwiemy relację:

$$C_\beta\left(T_\beta, T_\Delta\right) = \{\langle a_1, a_2 \rangle \in \mathbb{A} \times \mathbb{A} | d_{\beta_x}(a_1, a_2) \leq T_\beta \wedge d_\alpha(a_1, a_2) - d_{\beta_x}(a_1, a_2) \geq T_\Delta\}$$

gdzie $d_{\beta_x}(a_1, a_2) = \|\mathrm{R}_{a_1} - \mathrm{R}_{a_2}\|$, zaś sposób wyznaczenia R zależy od rodzaju aminokwasu:

$$\mathrm{R}_a = \begin{cases} \mathrm{C}^\alpha_a & a \text{ jest glicyną} \\ \mathrm{C}^\beta_a & a \text{ jest alaniną} \\ \mathrm{C}^{\beta_x}_a & \text{w p. p.} \end{cases}$$

Fakt 1. *Kryteria α-kontaktu i kryterium β-kontaktu dla dowolnych rzeczywistych dodatnich progów T_α, T_β i T_Δ są relacjami symetrycznymi.*

W dalszych rozważaniach będziemy posługiwać się kryterium kontaktu będącym teoriomnogościową sumą kryteriów α-kontaktu i β-kontaktu:

$$C\left(T_\alpha, T_\beta, T_\Delta\right) = C_\alpha\left(T_\alpha\right) \cup C_\beta\left(T_\beta, T_\Delta\right)$$

2.2.3. Deskryptory

Niech S będzie strukturą, zaś C pewnym kryterium kontaktu. W poniższych rozważaniach będziemy przyjmować, że o ile nie wskazano inaczej, wszystkie przytaczane aminokwasy należą do S. Zanim podamy formalną definicję deskryptora, zdefiniujemy kilka pomocniczych pojęć. **Elementem deskryptorowym** aminokwasu $a^{(i)}$ $(2 \leq i < |S| - 2)$ w strukturze S nazywać będziemy podciąg $a^{(i-2)}a^{(i-1)}a^{(i)}a^{(i+1)}a^{(i+2)}$ i będziemy oznaczać go symbolem $El\left(a^{(i)}\right)$. Zauważmy, że istnieją w strukturze aminokwasy, dla których element deskryptorowy jest niezdefiniowany (np. $a^{(0)}$)[5]. Zbiór aminokwasów elementu $El(a)$ będziemy oznaczać $El^{\mathbb{A}}(a)$. Operacja **scalania** elementów deskryptorowych polega na ich konkatenacji, jeżeli ich zbiory aminokwasów mają puste przecięcie, lub w przeciwnym przypadku wybraniu z S podciągu zawierającego aminokwasy należące do sumy zbiorów aminokwasów scalanych elementów. Będziemy ją oznaczać symbolem $\dot{\cup}$[6]. Niech a i $A = \{b_1, \ldots, b_n\}$ będą odpowiednio aminokwasem i zbiorem aminokwasów. Powiemy, że A jest **wzorcem kontaktów** aminokwasu a, jeżeli $\forall_{b \in A} \langle a, b\rangle \in C$. Wreszcie powiemy, że wzorzec kontaktów jest **deskryptorowo dopuszczalny**, jeżeli dla wszystkich zawartych w nim aminokwasów oraz aminokwasu a element deskryptorowy jest określony.

Definicja 1. Deskryptorem niewłaściwym aminokwasu a w strukturze S przy kryterium kontaktu C nazwiemy trójkę uporządkowaną $\langle a, C_a, R\rangle$, gdzie C_a jest deskryptorowo dopuszczalnym wzorcem kontaktów a, zaś R zbiorem aminokwasów należących do elementów deskryptorowych aminokwasów należących do C_a i a:

$$R = \bigcup_{b \in C_a \cup \{a\}} El^{\mathbb{A}}\left(b\right)$$

[5]Dla uproszczenia w przedstawianym modelu pomijamy sytuację występowania nieciągłości w rozważanej strukturze. Mimo że taka sytuacja w obrębie pojedynczego łańcucha polipeptydowego jest niemożliwa w naturze, dość często występuje ona w danych pochodzących z eksperymentalnego pomiaru struktury białka i jest spowodowana niedoskonałością stosowanych metod. Przedstawiony formalizm łatwo rozszerzyć, aby uwzględniał takie przypadki.

[6]Operacja scalania nie jest przemienna.

Aminokwas a nazywać będziemy **aminokwasem centralnym** deskryptora, zaś jego element deskryptorowy **elementem centralnym**.

Zbiorem segmentów deskryptora D nazwiemy zbiór podciągów S indukowanych klasami abstrakcji relacji $\sim_{\overline{S}}$, która jest domknięciem relacji $\sim_S$ do relacji równoważności nad zbiorem R. Zbiór segmentów deskryptora D będziemy oznaczać przez $seg(D)$. Mniej formalnie, segmentem nazwiemy maksymalny podciąg S aminokwasów należących do R taki, że każde jego dwa kolejne aminokwasy sąsiadują ze sobą w S.

Łatwo zauważyć, że możliwe jest zdefiniowanie relacji zawierania się dla deskryptorów niewłaściwych. Niech $D_1 = \langle a_1, C_{a_1}, R_1\rangle$ i $D_2 = \langle a_2, C_{a_2}, R_2\rangle$ będą deskryptorami niewłaściwymi. Powiemy, że deskryptor D_1 **jest zawarty** w D_2, jeżeli $a_1 = a_2$ oraz wzorzec kontaktów $C_{a_1} \subseteq C_{a_2}$. Relację zawierania się deskryptorów jest częściowym porządkiem. Będziemy ją oznaczać symbolem $\sqsubseteq$.

Fakt 2. *Jeżeli $D_1 \sqsubseteq D_2$, to $R_1 \subseteq R_2$.*

Twierdzenie 1. *Jeżeli dla ustalonej struktury S, kryterium kontaktu C i aminokwasu a istnieje co najmniej jeden deskryptor niewłaściwy D, to istnieje największy w sensie relacji $\sqsubseteq$ deskryptor niewłaściwy zawierający deskryptor D.*

Dowód. Niech $C_{\max}$ będzie zbiorem wszystkich aminokwasów będących w kontakcie z A, dla których element deskryptorowy jest określony.

$$C_{\max} = \{b \in S | \langle a, b\rangle \in C \wedge El(b) \text{ jest określony}\}$$

Jeżeli istnieje jakikolwiek deskryptor niewłaściwy, $C_{\max}$ jest zbiorem niepustym. Zatem istnieje deskryptor niewłaściwy $D_{\max} = \langle a, C_{\max}, R_{\max}\rangle$, który zawiera wszystkie deskryptory niewłaściwe aminokwasu a. □

Definicja 2. Deskryptorem aminokwasu a w strukturze S przy kryterium kontaktu C nazwiemy największy w sensie relacji $\sqsubseteq$ deskryptor niewłaściwy aminokwasu a.

Fakt 3. *Deskryptor aminokwasu a jest określony, jeżeli istnieje co najmniej jeden deskryptor niewłaściwy a.*

Deskryptor aminokwasu a będziemy oznaczać symbolem $D(a, S, C)$ lub w skrócie D_a, zaś zbiór wszystkich deskryptorów struktury S symbolem $\mathcal{D}_{S,C}$ lub w skrócie $\mathcal{D}_S$.

2.3. Podobieństwo deskryptorów

Niech $D_1 = \langle a_1, C_1, R_1 \rangle$ i $D_2 = \langle a_2, C_2, R_2 \rangle$ będą dowolnymi, ustalonymi na potrzeby tego podrozdziału deskryptorami niewłaściwymi. **Parowaniem wzorców kontaktów** będziemy nazywali różnowartościową funkcję częściową $\varphi\colon C_1 \to C_2$. Powiemy, że parowanie wzorców kontaktów φ jest **deskryptorowo dopuszczalne**, jeżeli φ można rozszerzyć na zbiory R_1 i R_2 z zachowaniem ciągłości elementów deskryptorowych. Uściślając, φ jest deskryptorowo dopuszczalne, jeżeli istnieje różnowartościowa funkcja częściowa $\psi\colon R_1 \to R_2$, taka że jeżeli $b^{(j)} = \overline{\varphi}(a^{(i)})$, to:

$$\psi(El(a^{(i)})) = \psi(a^{(i-2)})\psi(a^{(i-1)})\psi(a^{(i)})\psi(a^{(i+1)})\psi(a^{(i+2)}) = \\ = b^{(j-2)}b^{(j-1)}b^{(j)}b^{(j+1)}b^{(j+2)} = El(\overline{\varphi}(a^{(i)}))$$

gdzie:

$$\overline{\varphi}\colon C_1 \cup \{a_1\} \to C_2 \cup \{a_2\}\text{, t. że } \overline{\varphi}(a) = \begin{cases} a_2 & a = a_1 \\ \varphi(a) & \text{w p. p.} \end{cases}$$

Funkcję ψ będziemy nazywali **rozszerzeniem deskryptorowym** φ, będziemy się również posługiwać symbolem $\widetilde{\varphi} = \psi$.

Uliniowieniem deskryptorów D_1 i D_2 będziemy nazywali deskryptorowo dopuszczalne parowanie ich wzorców kontaktów. Niech C_1' i C_2' będą odpowiednio dziedziną i obrazem parowania φ, a $D_1' \sqsubseteq D_1$ i $D_2' \sqsubseteq D_2$ deskryptorami niewłaściwymi dla wzorców C_1' i C_2'. Zauważmy, że ψ jest izomorfizmem zachowującym relację C oraz relację sąsiedztwa sekwencyjnego. Powiemy, że deskryptory D_1' i D_2' są **obcięte** przez uliniowienie φ, co będziemy oznaczać symbolem $D_i' = D_i|_\varphi$.

Uliniowienie deskryptorów jest **dopuszczalne strukturalnie** dla progu T, jeżeli odległość średniokwadratowa pomiędzy parowanymi aminokwasami jest niewiększa od T_{RMSD} oraz, że jest **dopuszczalne sekwencyjnie** dla progów $\langle T_{nAA}, T_{nel}, T_{nseg} \rangle$, jeżeli spełnione są następujące nierówności:

(1) $\frac{|R_1'|}{|R_1|} \geq T_{nAA}, \frac{|R_2'|}{|R_2|} \geq T_{nAA}$,

(2) $\frac{|C_1'|+1}{|C_1|+1} \geq T_{nel}, \frac{|C_2'|+1}{|C_2|+1} \geq T_{nel}$,

(3) $\frac{|seg(D_1')|}{|seg(D_1)|} \geq T_{nseg}, \frac{|seg(D_2')|}{|seg(D_2)|} \geq T_{nseg}$

Definicja 3. Dla ustalonych progów $\langle T_{RMSD}, T_{nAA}, T_{nel}, T_{nseg} \rangle$ dwa deskryptory D_1 i D_2 są **podobne**, jeżeli istnieje ich uliniowienie, które jest dopuszczalne sekwencyjnie i strukturalnie.

W następnym podrozdziale wykażemy, że problem rozstrzygania, czy dwa deskryptory są podobne, jest NP-zupełny. Potem podamy praktyczny algorytm rozwiązujący problem nieco uproszczony. W tym celu zdefiniujemy pojęcie zrównoważonej dopuszczalności strukturalnej.

Uliniowienie deskryptorów jest **zrównoważenie dopuszczalne strukturalnie** dla progów $\langle T_{0el}, T_{el}, T_{pair}, T_{RMSD}\rangle$, jeżeli jest dopuszczalne strukturalnie dla progu T_{RMSD} oraz spełnione są następujące warunki:

(1) $RMSD\left(El(a_1), \psi\left(El(a_1)\right)\right) \leq T_{0el}$,

(2) $\bigwedge_{a\in C_1'} RMSD\left(El(a), \psi\left(El(a)\right)\right) \leq T_{el}$,

(3) $\bigwedge_{a\in C_1'} RMSD\left(El(a_1) \dot{\cup} El(a), \psi\left(El(a_1) \dot{\cup} El(a)\right)\right) \leq T_{pair}$.

Zauważmy, że zrównoważona dopuszczalność strukturalna nakłada dodatkowe ograniczenie na wartość RMSD pewnych fragmentów porównywanych deskryptorów. Zatem zaburzenia kształtu polegające na wzajemnym przesunięciu elementów są preferowane w stosunku do odkształceń samych elementów. Jeżeli przyjąć, że każda para ze zbiorów $\{a_1\} \times C_1'$ i $\{a_2\} \times C_2'$ odpowiada oddziaływaniu fizykochemicznemu aminokwasów, za naturę tego oddziaływania najbardziej odpowiadają te aminokwasy i ich elementy deskryptorowe. Zatem uliniowienie deskryptorów zachowujące konformację tychże najprawdopodobniej najlepiej odzwierciedla podobieństwo pod względem fizykochemicznym i biologicznym.

2.4. NP-zupełność problemu znajdowania najlepszego uliniowienia deskryptorów

Zdefiniujemy problem najlepszego uliniowienia w następujący sposób:

Definicja 4. Niech parametry $\langle T_{RMSD}, T_{nAA}, T_{nel}, T_{nseg}\rangle \in \mathbb{R}^4$ będą ustalone. Dla danych deskryptorów D_1 i D_2 **problem najlepszych dopuszczalnych uliniowień** polega na znalezieniu parowania wzorców kontaktów φ będącego uliniowieniem dopuszczalnym strukturalnie i sekwencyjnie, którego rozszerzenie deskryptorowe ψ będzie maksymalne w sensie liczności dziedziny.

Dla ustalonej liczby $m \in \mathbb{N}$ decyzyjny **problem optymalnego dopuszczalnego uliniowienia** (PODU) brzmi: Czy dla danych deskryptorów D_1 i D_2 istnieje uliniowienie dopuszczalne strukturalnie i sekwencyjnie φ mające rozszerzenie deskryptorowe ψ takie, że $|Dom(\psi)| \geq m$?

Analogicznie zdefiniujemy **problem najlepszych zrównoważonych uliniowień** i decyzyjny **problem optymalnego zrównoważonego uliniowienia** (POZU), ograniczając problemy dopuszczalnych uliniowień do uliniowień zrównoważenie dopuszczalnych strukturalnie z ustalonymi parametrami $\langle T_{0el}, T_{el}, T_{pair}, T_{RMSD} \rangle$.

Zanim dowiedziemy NP-zupełności problemu optymalnego dopuszczalnego uliniowienia, pokażemy NP-zupełność prostszego problemu optymalnego uliniowienia.

Definicja 5. Dla ustalonej liczby $m \in \mathbb{N}$ decyzyjny **problem optymalnego uliniowienia** (POU) brzmi: czy dla danych deskryptorów D_1 i D_2 istnieje uliniowienie φ mające rozszerzenie deskryptorowe ψ takie, że $|Dom(\psi)| \geq m$.

Zauważmy, że POU jest intuicyjnie równie trudny jak PODU dla odpowiednio wysokiej wartości progu T_{RMSD} oraz zerowych progów T_{AA}, T_{el}, T_{seg}. Formalny dowód tego faktu przeprowadzimy pod koniec tego podrozdziału.

Lemat 1. *Problem POU jest NP-zupełny.*

Dowód. Łatwo zauważyć, że $POU \in NP$. Niedeterministyczny algorytm potrzebuje odgadnąć uliniowienie φ. Sprawdzenie, czy φ jest uliniowieniem, obliczenie ψ i sprawdzenie, czy $|Dom(\psi)| \geq m$, jest możliwe w czasie wielomianowym.

Pokażemy, że POU jest NP-trudny. W tym celu dokonamy redukcji znanego problemu 3-PARTITION[24, problem SP15] do POU.

Definicja 6. Dla danego zbioru A liczącego $3m$ elementów, liczby $B \in \mathbb{Z}^+$ i funkcji $s\colon A \to \mathbb{Z}^+$ takiej, że:

$$\bigwedge_{a \in A} \frac{1}{4}B < s(a) < \frac{1}{2}B,$$

$$\sum_{a \in A} s(a) = mB,$$

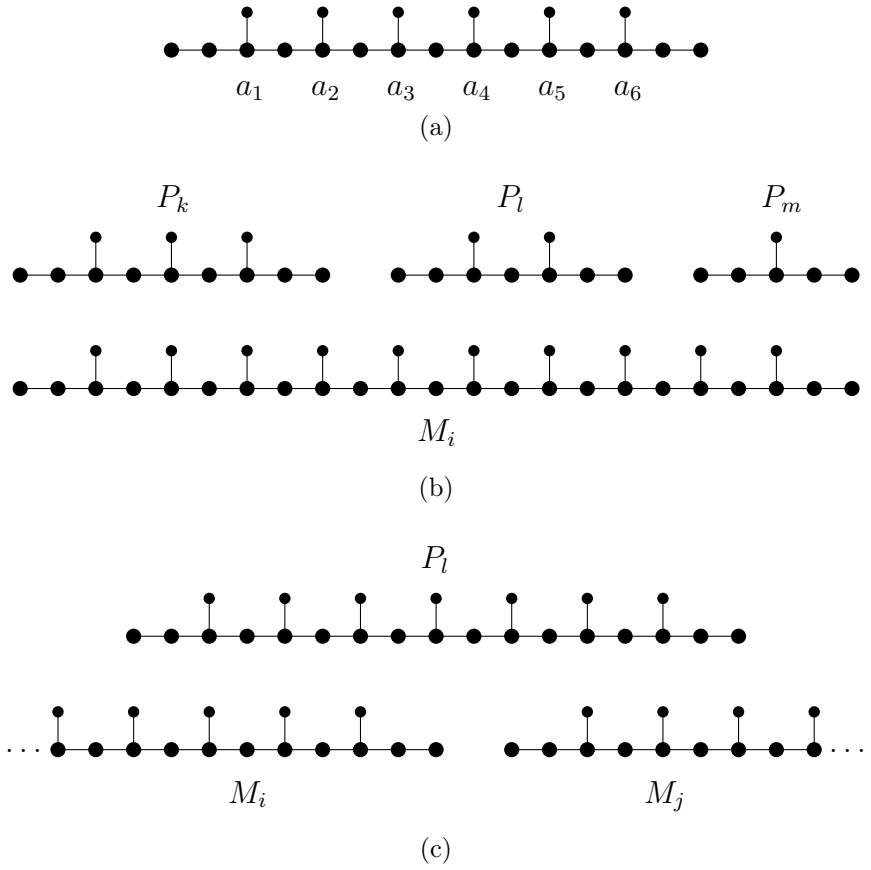
problem **3-PARTITION** polega na rozstrzygnięciu, czy istnieje podział A na m rozłącznych podzbiorów $A_1, A_2, \ldots, A_m$ taki, że:

$$\bigwedge_{1 \leq i \leq m} \sum_{a \in A_i} s(a) = B$$

Fakt 4. *Każdy zbiór A_i w rozwiązaniu 3-PARTITION liczy dokładnie 3 elementy.*

Grzebieniem długości k będziemy nazywać k-elementowy wzorzec kontaktów $C = \{a_1, a_2, \ldots, a_k\}$ taki, że kolejne jego aminokwasy są w strukturze oddzielone dokładnie jednym aminokwasem (rys. 2.3a):

$$\bigwedge_{1 \leq i < k} \bigvee_{b \in S} a_i \sim_S b \wedge b \sim_S a_{i+1}$$



Rysunek 2.3: (a) Grzebień długości 6 wraz z rozszerzeniem deskryptorowym. (b) Przykładowe przyporządkowanie trzech grzebieni z P do M_i. (c) Przy przyporządkowaniu grzebienia P_l do dwóch grzebieni z M jeden aminokwas z P_l pozostanie nieuliniowiony.

zaś (p, q)**-wycinkiem** grzebienia G nazwiemy podciąg $w_{p,q}(G) = \{a_p, a_{p+1}, \ldots, a_q\}$. Liczność rozszerzenia deskryptorowego grzebienia wynosi:

$$\left| \bigcup_{1 \leq i \leq k} El^{\mathbb{A}}(a_i) \right| = 2k + 3$$

Niech $\langle A, s, B \rangle$ będzie instancją problemu 3-PARTITION o liczności zbioru A równej $3m$. Elementy zbioru A będziemy oznaczać przez a_i $(1 \leq i \leq 3m)$. Niech $P = \{P_1, P_2, \ldots, P_{3m}\}$ będzie zbiorem rozłącznych grzebieni należących do struktury S_1 takim, że długość grzebienia P_i jest równa $s(a_i)$. Wreszcie niech $M = \{M_1, M_2, \ldots, M_m\}$ będzie zbiorem rozłącznych grzebieni o długości $B + 4$ należących do struktury S_2. Dobranie kryterium kontaktu i centralnych aminokwasów[7] tak, aby istniały deskryptory $D_1 = \langle c_1, \bigcup P, R_1 \rangle$ i $D_2 = \langle c_2, \bigcup M, R_2 \rangle$ jest trywialne. Pokażemy, że podział A

[7]Centralne aminokwasy są dobrane tak, aby $El(a_1) \cap \bigcup \vec{El}(P) = El(a_2) \cap \bigcup \vec{El}(M) = \emptyset$.

na podzbiory zgodnie z definicją problemu 3-PARTITION istnieje wtedy i tylko wtedy, gdy istnieje uliniowienie deskryptorów D_1 i D_2 o liczności rozszerzenia deskryptorowego wynoszącej $m(2B+9)+5$.

(1) Niech $A_1, A_2, \ldots, A_m$ będzie rozwiązaniem problemu 3-PARTITION. Przypuśćmy, że dla pewnego i $A_i = \{a_{k_1}, a_{k_2}, a_{k_3}\}$. Weźmy funkcję $\varphi_{k_1} : P_{k_1} \to M_i$, która jest deskryptorowo dopuszczalnym parowaniem wzorców kontaktów i przeprowadza grzebień P_{k_1} na $(1, s(a_{k_1}))$-wycinek grzebienia M_i. Analogicznie weźmy deskryptorowo dopuszczalne parowania wzorców kontaktów $\varphi_{k_2} : P_{k_2} \to M_i$ i $\varphi_{k_3} : P_{k_3} \to M_i$ takie, że(rys. 2.3b):

$$\vec{\varphi_{k_2}}(P_{k_2}) = w_{s(a_{k_1})+3, s(a_{k_1})+s(a_{k_2})+2}(M_i)$$
$$\vec{\varphi_{k_3}}(P_{k_3}) = w_{s(a_{k_1})+s(a_{k_2})+5, s(a_{k_1})+s(a_{k_2})+s(a_{k_3})+4}(M_i)$$

Ostatni z wycinków jest dobrze określony, a zbiory wartości i tym samym obrazy funkcji wypełniają cały grzebień M_i, ponieważ $s(a_{k_1}) + s(a_{k_2}) + s(a_{k_3}) + 4 = B + 4$. Zauważmy również, że dziedziny i obrazy funkcji φ_{k_j} ani ich rozszerzenia deskryptorowe się nie przecinają. Zatem funkcja $\varphi_i : \bigcup_{1 \le j \le 3} P_{k_j} \to M_i$ określona wzorem:

$$\varphi_i(c) = \begin{cases} \varphi_{k_1}(c) & c \in P_{k_1} \\ \varphi_{k_2}(c) & c \in P_{k_2} \\ \varphi_{k_3}(c) & c \in P_{k_3} \end{cases}$$

jest deskryptorowo dopuszczalnym parowaniem. W podobny sposób korzystając z wzajemnej rozłączności grzebieni P_i i M_i definiujemy funkcję φ:

$$\varphi(c) = \begin{cases} \varphi_i(c) & c \in P_i \\ a_2 & c = a_1 \end{cases}$$

Funkcja φ jest uliniowieniem deskryptorów D_1 i D_2. Pozostaje obliczyć liczność dziedziny jej rozszerzenia deskryptorowego ψ. Zauważmy, że:

$$Dom(\psi) = El^{\mathbb{A}}(a_1) \cup \bigcup_{c \in P} El^{\mathbb{A}}(c) = El^{\mathbb{A}}(a_1) \cup \bigcup_{1 \le i \le 3m} \bigcup_{c \in P_i} El^{\mathbb{A}}(c)$$

zaś z rozłączności grzebieni P_i i elementu deskryptorowego a_1:

$$|Dom(\psi)| = \left|El^{\mathbb{A}}(a_1)\right| + \sum_{1 \le i \le 3m} \left|\bigcup_{c \in P_i} El^{\mathbb{A}}(c)\right| =$$
$$= 5 + \sum_{1 \le i \le 3m} 2s(a_i) + 3 = 5 + 2mB + 9m = m(2B+9) + 5$$

(2) Niech φ będzie uliniowieniem deskryptorów D_1 i D_2 takim, że liczność jego rozszerzenia deskryptorowego ψ wynosi $m(2B + 9) + 5$. Ponieważ zbiór R_1 liczy $m(2B+9)+5$ aminokwasów, ψ jest funkcją pełną (nie częściową), zatem wszystkie kontakty grzebieni P zostały przyporządkowane kontaktom grzebieni M. Wystarczy pokazać, że $\forall_{P_i \in P} \exists_{M_j \in M} \vec{\varphi}(P_i) \subseteq M_j$. Załóżmy, że jest przeciwnie. Niech P_i będzie grzebieniem, którego obraz ma niepuste przecięcie z grzebieniami M_p i M_q, oraz k i l $(k < l)$ będą odpowiednio największym indeksem aminokwasu o obrazie w M_p i najmniejszym indeksem aminokwasu o obrazie w M_q. Skoro φ jest deskryptorowo dopuszczalna, zachodzi $l - k > 4$, zatem co najmniej jeden aminokwas w P_i nie może należeć do dziedziny ψ, co prowadzi do sprzeczności (rys. 2.3c). Niech $A_i = \{a_j | \vec{\varphi}(P_j) \subseteq M_i\}$, wtedy:

$$\sum_{a_j \in A_i} s(a) = \sum |P_j| \leq B$$

Zatem ponieważ rodzina zbiorów A_i liczy m elementów, a wszystkie grzebienie P_i zostały uliniowione, na mocy własności funkcji s:

$$\sum_{a_j \in A_i} s(a) = \sum |P_j| = B$$

Ostatnią częścią dowodu NP-zupełności jest pokazanie, że proponowana konwersja jest wielomianowa. Problem 3-PARTITION jest silnie NP-zupełny, zatem zachowuje tę właściwość również przy kodowaniu w systemie jedynkowym. Rozmiar instancji problemu POU po konwersji jest liniowy względem rozmiaru instancji problemu 3-PARTITION w kodowaniu jedynkowym. Zatem przedstawiona konwersja ma złożoność wielomianową. □

Twierdzenie 2. *Problemy PODU i POZU są NP-zupełne.*

Dowód. Obydwa problemy są w klasie NP. Aby udowodnić ich NP-zupełność wystarczy zauważyć, że każda instancja POU jest instancją PODU dla parametrów $T_{RMSD} = \infty$[8] oraz zerowych progów $T_{AA} = 0$, $T_{el} = 0$, $T_{seg} = 0$, ponieważ każde uliniowienie jest dla tych parametrów dopuszczalne sekwencyjnie i strukturalnie. Analogicznie, każde uliniowienie jest zrównoważenie dopuszczalne strukturalnie dla parametrów $T_{0el} = \infty$, $T_{el} = \infty$, $T_{pair} = \infty$. □

[8]Formalnie należałoby powiedzieć, że istnieje wartość T_{RMSD}, która jest górnym ograniczeniem funkcji $RMSD(S, \varphi)$ dla $S \subseteq S_1$ i $\varphi\colon S \to S_2$. Taka wartość jest proporcjonalna do maksymalnej odległości między aminokwasami w S_1 i S_2.

Powyższe dowody NP-zupełności wymagają komentarza. Otóż w ich konstrukcji nie uwzględniliśmy biologicznej wiedzy o strukturze białek i rzeczywistych rozmiarach deskryptorów. Jest dosyć naturalne, że przestrzenny układ obiektów o niezerowym rozmiarze, które podlegają ograniczeniom fizycznym, przy rzeczywistym kryterium kontaktu nie może zazwyczaj przyjąć konformacji, jaka byłaby konieczna w zaproponowanej konwersji problemu 3-PARTITION do POU. W szczególności pojedynczy aminokwas nie może być w kontakcie z większą liczbą aminokwasów niż możliwa do zmieszczenia liczba atomów węgla w sferze o promieniu 8.5Å. Niemniej jednak przedstawiony dowód ma wartość poznawczą, gdyż demonstruje trudności, jakie można napotkać podczas konstruowania algorytmu znajdowania uliniowienia dopuszczalnego sekwencyjnie.

2.5. Algorytm znajdowania najlepszego uliniowienia deskryptorów

Mimo, że problem znajdowania najlepszego zrównoważonego uliniowienia deskryptorów jest NP-zupełny, w praktyce możliwe jest zaproponowanie wydajnego algorytmu porównywania deskryptorów. Wynika to głównie z faktu, że wzorce kontaktów wstępujących w strukturach białkowych deskryptorów nie liczą więcej niż 20 aminokwasów. Poniżej przedstawimy algorytm, który został zaimplementowany na potrzeby przedstawionych w rozprawie badań. Zawiera on pewne uproszczenia, które zostaną uzupełnione w podrozdziale 2.6. Ze względu na stopień skomplikowania pominiemy również szczegóły techniczne, które nie wpływają na wynik i mogą się zmieniać w zależności od implementacji.

Niech $D_1 = \langle a_1, C_1, R_1 \rangle$ i $D_2 = \langle a_2, C_2, R_2 \rangle$ będą porównywanymi deskryptorami. Obliczenie można podzielić na następujące etapy (na każdym etapie niespełnienie warunku lub pusty wynik oznacza, że nie istnieje zrównoważone uliniowienie):

(1) RMSD centralnych elementów nie może być większe niż T_{0el}:

$$RMSD(El(a_1), El(a_2)) \leq T_{0el}$$

(2) Parowane mogą być wyłącznie kontakty, które w połączeniu z parą $\langle a_1, a_2 \rangle$ dadzą parowanie deskryptorowo dopuszczalne:

$$P_0 = \left\{ \langle c_1, c_2 \rangle \in C_1 \times C_2 \;\middle|\; \begin{array}{l} El^{\mathbb{A}}(a_1) \cap El^{\mathbb{A}}(c_1) = El^{\mathbb{A}}(a_2) \cap El^{\mathbb{A}}(c_2) = \emptyset \vee \\ \vee\, n_S(a_1) - n_S(c_1) = n_S(a_2) - n_S(c_2) \end{array} \right\}$$

(3) Parowane mogą być wyłącznie kontakty, dla których RMSD ich elementów deskryptorowych jest nie większe niż T_{el}:

$$P_1 = \{\langle c_1, c_2\rangle \in P_0 | RMSD(El(c_1), El(c_2)) \leq T_{el}\}$$

(4) Spośród par kontaktów w P_1 należy odrzucić takie, których RMSD w połączeniu z centralnymi elementami deskryptorów jest większe niż T_{pair}:

$$P_2 = \{\langle c_1, c_2\rangle \in P_1 | RMSD(El(a_1) \dot{\cup} El(c_1), El(a_2) \dot{\cup} El(c_2)) \leq T_{pair}\}$$

Powiemy, że dwie pary kontaktów $\langle f_1, f_2\rangle$ i $\langle g_1, g_2\rangle$ są **niesprzeczne**, jeżeli parowanie $\varphi\colon \{f_1, g_1\} \to \{f_2, g_2\}$ dane wzorem:

$$\varphi(f) = \begin{cases} g_1 & f = f_1 \\ g_2 & f = f_2 \end{cases}$$

jest deskryptorowo dopuszczalne. Niesprzeczność par kontaktów będziemy oznaczać symbolem $\backsim$.

(5) Niech $G = \langle V, E\rangle$ będzie grafem nieskierowanym o zbiorze wierzchołków $V = P_1$ i zbiorze krawędzi $E = \{\langle p_1, p_2\rangle \in P_1 \times P_1 | p_1 \backsim p_2\}$. Należy obliczyć zbiór $\mathcal{C}$ maksymalnych w sensie zawierania klik grafu G. Każda klika w $\mathcal{C}$ zawiera wzajemnie niesprzeczne pary kontaktów i może być utożsamiana z deskryptorowo dopuszczalnym parowaniem kontaktów[9] oraz po uzupełnieniu o parę $\langle a_1, a_2\rangle$ traktowana jak uliniowienie. Ostatecznie otrzymujemy zbiór Φ_0 uliniowień.

(6) Uliniowienia w zbiorze Φ_0 mogą nie być dopuszczalne strukturalnie[10]. Jeżeli tak się zdarzy, należy usunąć pewną możliwie małą liczbę par z kliki odpowiadającej takiemu uliniowieniu, tak aby stało się ono strukturalnie dopuszczalne, a jego rozszerzenie deskryptorowe możliwie mało się zmniejszyło. Problem ten sam w sobie jest NP-zupełny (dowód tego faktu pominiemy), ale możliwe jest zaproponowanie wydajnego algorytmu heurystycznego, który daje zadowalające rozwiązanie. Ostatecznie otrzymujemy zbiór Φ_1 uliniowień dopuszczalnych strukturalnie.

(7) Ostatnim krokiem jest odrzucenie uliniowień niedopuszczalnych sekwencyjnie:

$$\Phi_2 = \{\varphi \in \Phi_1 | \varphi \text{ jest dopuszczalne sekwencyjnie}\}$$

[9] W sensie teoriomnogościowym kliki z $\mathcal{C}$ są podzbiorami $C_1 \times C_2$, czyli mają sens relacji. Relacja zaś po określeniu dziedziny i zbioru wartości może być interpretowana jako funkcja częściowa.

[10] Zauważmy, że uliniowienia które są dopuszczalne strukturalnie, są również zrównoważenie dopuszczalne strukturalnie (wynika to bezpośrednio z poprzednich kroków algorytmu).

Koszt obliczeniowy 5. kroku powyższego algorytmu jest wykładniczy ze względu na liczność zbioru P_1. W zastosowaniu praktycznym okazuje się jednak, że jest on nieznaczny wobec wcześniejszych kroków o złożoności wielomianowej.

2.6. Udoskonalenia metody porównywania deskryptorów

2.6.1. Reprezentacja deskryptorów przy użyciu zbiorów przybliżonych

Dobór kryterium kontaktu, aby odzwierciedlało występowanie rzeczywistych oddziaływań pomiędzy aminokwasami, jest kluczowy by deskryptory miały sens biologiczny. Należy sobie jednak zdawać sprawę, że dla każdego zestawu stałych można skonstruować odpowiednio "złośliwy" przypadek. Niech $a_1 \in S_1^{\mathbb{A}}$ i $a_2 \in S_2^{\mathbb{A}}$ będą aminokwasami, $\underline{C}$ i $\overline{C}$ kryteriami kontaktów takimi, że $\underline{C} \subset \overline{C}$, a $\underline{D}_1 = \langle a_1, \underline{C}_1, \underline{R}_1 \rangle$, $\underline{D}_2 = \langle a_2, \underline{C}_2, \underline{R}_2 \rangle$ i $\overline{D}_1 = \langle a_1, \overline{C}_1, \overline{R}_1 \rangle$, $\overline{D}_2 = \langle a_2, \overline{C}_2, \overline{R}_2 \rangle$ deskryptorami aminokwasów a_1 i a_2 dla kryteriów kontaktów $\underline{C}$ i $\overline{C}$. Łatwo zauważyć, że:

$$\underline{C}_1 \subseteq \overline{C}_1, \underline{R}_1 \subseteq \overline{R}_1,$$

$$\underline{C}_2 \subseteq \overline{C}_2, \underline{R}_2 \subseteq \overline{R}_2$$

Przypuśćmy, że $\overline{D}_1$ i $\overline{D}_2$ są podobne, a uliniowienie φ jest świadkiem tego podobieństwa oraz że $\underline{C}$ i $\overline{C}$ są tak dobrane, że $\underline{D}_2 = \overline{D}_2$, a $\frac{\left|\overline{C}_1 \smallsetminus \underline{C}_1\right|}{\left|\overline{C}_2\right|} > T_{nel}$. Przy takich założeniach $\underline{D}_1$ nie jest podobny do $\overline{D}_2$. Dobranie struktur i kryteriów kontaktu dla tego przykładu nie jest szczególnie skomplikowane. S_1 i S_2 mogą być niemal identyczne, z minimalnie zaburzonymi odległościami pomiędzy a_1 i aminokwasami z $\overline{C}_1$, zaś progi T_α i T_β w kryteriach kontaktu odpowiednio mieszczące się pomiędzy tymi odległościami. Posługując się tym przykładem można powiedzieć, że porównywanie deskryptorów jest źle uwarunkowane, gdyż drobne zaburzenia odległości pomiędzy aminokwasami i progów w kryterium kontaktu mogą wpływać na to, czy deskryptory są uznawane za podobne. Aby rozwiązać ten problem, posłużymy się formalizmem zbiorów przybliżonych.

Niech tak jak w przykładzie $\underline{C}$ i $\overline{C}$ będą kryteriami kontaktu. **Przybliżonym deskryptorem** aminokwasu a nazwiemy parę $\langle \underline{D}, \overline{D} \rangle$ taką, że $\underline{D}$ jest deskryptorem a przy kryterium kontaktu $\underline{C}$, zaś $\overline{D}$ deskryptorem przy kryterium $\overline{C}$. Rozważmy parę

deskyptorów przybliżonych $D_1 = \langle \underline{D}_1, \overline{D}_1 \rangle$ i $D_2 = \langle \underline{D}_2, \overline{D}_2 \rangle$. Jeżeli $\overline{\varphi}$ jest uliniowieniem $\overline{D}_1$ i $\overline{D}_2$, jego obcięcie $\underline{\varphi} = \overline{\varphi}|_{\underline{C}_1}$ jest uliniowieniem $\underline{D}_1$ i $\underline{D}_2$. Ponadto, jeżeli $\overline{\varphi}$ jest zrównoważenie dopuszczalne strukturalnie, zaś $\underline{\varphi}$ dopuszczalne strukturalnie, to $\underline{\varphi}$ również jest zrównoważenie dopuszczalne strukturalnie[11].

Definicja 7. Deskryptory przybliżone $D_1 = \langle \underline{D}_1, \overline{D}_1 \rangle$ i $D_2 = \langle \underline{D}_2, \overline{D}_2 \rangle$ są **podobne**, jeżeli istnieje zrównoważenie dopuszczalne strukturalnie uliniowienie $\overline{\varphi}$ deskryptorów $\overline{D}_1$ i $\overline{D}_2$ oraz $\overline{\varphi}$ jest sekwencyjnie dopuszczalnym uliniowieniem deskryptorów uzyskanych w wyniku scalenia $\overline{D}_1|_{\varphi}$ z $\underline{D}_1$ i $\overline{D}_2|_{\varphi}$ z $\underline{D}_2$[12].

Przedstawiony formalizm będziemy interpretować następująco. Powiemy, że $\underline{D}$ jest częścią deskryptora, do istotności której nie mamy wątpliwości. Natomiast zewnętrzna część $(\overline{R} \setminus \underline{R})$ jest wątpliwa. Przy porównywaniu dwóch deskryptorów żądamy, aby pewne części były uliniowione możliwie w całości, natomiast elementy wątpliwe uznajemy za należące do deskryptora wtedy i tylko wtedy, gdy należą do uliniowienia.

2.6.2. Liczność zbioru segmentów

W podrozdziale 2.2.3 zdefiniowaliśmy zbiór segmentów deskryptora, jako zbiór ciągłych w sensie sąsiedztwa sekwencyjnego podciągów struktury zawartych w rozszerzeniu deskryptorowym wzorca kontaktów. Taka definicja może rodzić problemy przy rozstrzyganiu o dopuszczalności sekwencyjnej uliniowienia. Przypuśćmy, że próg T_{seg} ma wartość $\frac{2}{3}$ i porównywane są deskryptory liczące po 3 segmenty, spośród których możliwe jest uliniowienie dwóch. W takim przypadku warunek minimalnej liczby uliniowionych segmentów będzie spełniony i deskryptory będą uznane za podobne. Rozważmy teraz podobny przypadek, w którym uliniowione segmenty z poprzedniego przykładu są ze sobą połączone pętlą. W takiej sytuacji uliniowienie będzie liczyło jeden segment i nie będzie dopuszczalne sekwencyjnie. Nie jest to korzystne, aby niewielka różnica (dodanie jednego kontaktu) w taki sposób zaburzała wynik. Ponadto intuicyjnie postrzegamy segmenty deskryptora jako niezależne strukturalnie elementy struktury drugorzędowej i fakt, że pętla je łącząca jest na tyle krótka, że należy w całości do deskryptora, nie powinien mieć tutaj istotnego znaczenia. Konieczne jest zatem zaproponowanie metody rozróżniania, czy ciągły wycinek struktury zawarty w deskryptorze jest pojedynczym

[11] Założenie o dopuszczalności strukturalnej $\underline{\varphi}$ jest konieczne, ze względu na niemonotoniczność funkcji RMSD.

[12] Scalenie rozumiemy jako wzięcie deskryptora, którego wzorzec kontaktów jest sumą wzorców kontaktów scalanych deskryptorów.

"prostym" segmentem, czy też łamaną złożoną z kilku takich segmentów. Opracowanie dobrego algorytmu znajdowania punktów rozcięcia wymagałoby niestety głębokiej analizy uwzględniającej występowanie w strukturach białek różnego rodzaju odstępstw od kanonicznej struktury drugorzędowej (np. *β-bulge*). Szczęśliwie, aby ulepszyć sposób rozstrzygania o dopuszczalności sekwencyjnej, wystarczy zmodyfikować sposób zliczania segmentów, bez wskazywania konkretnych punktów rozcięcia.

Strukturalną długość segmentu $s = a^{(p)}a^{(p+1)}\dots a^{(q)}$ obliczamy następująco:

$$L(s) = \sum_{p \le i < q} \|M_{i+1} - M_i\|$$

gdzie M_i jest środkiem geometrycznym trzech kolejnych atomów C^α:

$$M_i = \frac{1}{3}\left(C^\alpha_{i-1} + C^\alpha_i + C^\alpha_{i+1}\right)$$

Skorygowaną licznością segmentu będziemy nazywali liczbę określoną wzorem:

$$N(s) = \left\lceil \frac{L(s)}{18.0\text{Å}} \right\rceil$$

Miara ta opiera się na obserwacji, że strukturalna długość względnie prostego (z dokładnością do struktury drugorzędowej) segmentu jest ograniczona maksymalną odległością pomiędzy aminokwasami wynikającą z kryterium kontaktu. Jeżeli długość segmentu przekracza pewną wartość, na mocy warunku trójkąta nie jest możliwe, aby mógł być prosty, a aminokwasy przy jego końcach były w kontakcie z aminokwasem centralnym. Pojęcie strukturalnej długości służy ujednoliceniu podejścia dla różnych rodzajów struktury drugorzędowej[13].

Ostatecznie warunek (3) w definicji dopuszczalności sekwencyjnej uliniowienia przyjmuje postać:

$$\frac{\min\left(N(D_i|_\varphi), N(D_i)\right)}{N(D_i)} \ge T_{nseg}, \text{ dla } i = 1, 2$$

gdzie:

$$N(D) = \sum_{s \in seg(D)} N(s)$$

Wzięcie minimum z $N(D_i|_\varphi)$ i $N(D_i)$ zabezpiecza przed sytuacją, gdy na skutek niedoskonałości proponowanego oszacowania $N(D_i|_\varphi)$ byłoby większe od $N(D_i)$.

[13]Odległość pomiędzy punktami będącymi rzutami kolejnych aminokwasów na oś helisy α wynosi ok. 1.5Å, zaś w przypadku wstęgi β ok. 3Å.

Rozdział 3

Porównywanie struktur białek

Każdy deskryptor osadzony jest w pewnej strukturze, zatem opisana w poprzednim rozdziale metoda porównywania deskryptorów może zostać wykorzystana do wykrywania niewielkich lokalnych podobieństw pomiędzy dwiema strukturami białek. W tym rozdziale zajmiemy się problemem, jak z informacji zawartych w zbiorze takich lokalnych podobieństw zbudować możliwie pełny obraz globalnego podobieństwa rozważanych struktur. Aby tego dokonać zdefiniujemy formalnie podobieństwo struktur białkowych oraz przedstawimy algorytmy wybierania lokalnych podobieństw, które są zawarte w pewnym maksymalnym (lub bliskim maksymalnemu) globalnym podobieństwie. Wielokrotnie będziemy posługiwać się pojęciami podobieństwa i uliniowienia struktur. Zanim przedstawimy formalne definicje, zaproponujemy czytelnikowi wyrobienie sobie pewnych intuicji związanych z tymi pojęciami. Przez uliniowienie struktur będziemy rozumieć pewne częściowe odwzorowanie pomiędzy zbiorami ich aminokwasów, które w założeniu ma być "izomorfizmem" w sensie pewnych biologicznych relacji występujących pomiędzy aminokwasami w obrębie rozważanych struktur. Stopień podobieństwa struktur określa miara maksymalnego uliniowienia.

Relacje, o których mowa, mogą obejmować różne cechy związane z funkcją pełnioną przez rozważane białka (np. miejsca wiązania podobnych cząsteczek), ich strukturą (np. kluczowe oddziaływania odpowiadające za proces zwijania białka lub stabilizujące strukturą) oraz inne cechy szczególnie istotne dla danej pary białek. Ich poprawna identyfikacja wymaga szerokiej wiedzy biochemicznej, niejednokrotnie popartej danymi doświadczalnymi i stanowi przypuszczalnie niedościgniony wzór dla metod maszynowych.

Samo pojęcie uliniowienia jest tłumaczeniem angielskiego słowa *alignment* i po raz pierwszy w bioinformatyce zostało zastosowane w kontekście porównywania sekwencji

(białkowych, DNA, bądź RNA). Tradycyjnie maksymalne uliniowienia sekwencji obliczane były przy użyciu algorytmów opartych na programowaniu dynamicznym[53], co przy pewnych założeniach jest równoważne znajdowaniu minimalnej odległości edycyjnej pomiędzy rozważanymi sekwencjami[1]. Podejście tego typu nie pozwala rozważać zamiany kolejności fragmentów, co może mieć miejsce wskutek procesów ewolucyjnych takich jak np. duplikacja genu lub zmian zachodzących w strukturze białka podczas jego zwijania[26, 68] . Próba uogólnienia metod porównywania sekwencji, aby dopuszczały przestawienia kolejności, wydaje się być trudna lub wręcz niemożliwa ze względu na złożoność obliczeniową oraz na fakt, iż usunięcie ograniczenia wymuszającego zachowanie kolejności uliniowionych aminokwasów poszerzyłoby przestrzeń dopuszczalnych rozwiązań do tego stopnia, że znajdowane byłby uliniowienia niemające sensu biologicznego. W odróżnieniu od sekwencji struktura białka niesie znacznie większą ilość informacji i w związku z tym można zrezygnować ze wspomnianego ograniczenia. Dlatego zwracamy uwagę na fakt, że w poniższych rozważaniach nasze rozumienie pojęcia uliniowienia odbiega od tradycyjnego.

Zaproponujemy metodę określania podobieństwa struktur znajdującą maksymalne uliniowienia pomiędzy strukturami, które cechują się lokalnym podobieństwem i zbliżonymi wzorcami kontaktów pomiędzy aminokwasami, niezależnie od kolejności występowania uliniowionych fragmentów w sekwencji białka oraz globalnych odkształceń.

3.1. Podstawowe definicje

Na potrzeby tego i następnych podrozdziałów przyjmijmy, że S_1 i S_2 są strukturami białkowymi.

Parowaniem struktur S_1 i S_2 nazwiemy różnowartościową funkcję częściową $\xi\colon S_1^{\mathbb{A}} \to S_2^{\mathbb{A}}$. Niech Φ będzie pewnym zbiorem uliniowień deskryptorów z $\mathcal{D}_{S_1}$ i $\mathcal{D}_{S_2}$. **Wsparciem deskryptorowym** $Supp(\xi, \Phi)$ parowania struktur ξ w zbiorze Φ nazwiemy taki podzbiór Φ, że:

$$Supp(\xi, \Phi) = \left\{ \varphi \in \Phi \middle| \bigwedge_{a \in Dom(\widetilde{\varphi})} \widetilde{\varphi}(a) = \xi(a) \right\}$$

Zawieranie się uliniowienia deskryptorów φ (ściśle rozszerzenia $\widetilde{\varphi}$) w parowaniu struktur ξ będziemy oznaczać symbolem $\varphi \sqsubset \xi$. Będziemy również posługiwać się pojęciem

[1] Odległość edycyjna to minimalna liczba operacji typu wstawienie, usunięcie bądź zmiana symbolu konieczna, aby jedną sekwencję przekształcić w drugą[60].

zawierania się parowań struktur:

$$\xi_1 \subseteq \xi_2 \Leftrightarrow \bigwedge_{a \in Dom(\xi_1)} \xi_1(a) = \xi_2(a)$$

Łatwo zauważyć, że jeżeli $\xi_1 \subseteq \xi_2$ to $Dom(\xi_1) \subseteq Dom(\xi_2)$.

Definicja 8. Uliniowieniem struktur S_1 i S_2 przy wsparciu w Φ nazwiemy parowanie, dla którego istnieje wsparcie deskryptorowe w Φ, które pokrywa całą dziedzinę ξ. Powiemy również, że Φ **pokrywa** ξ, co będziemy oznaczać $\xi \Subset \Phi$.

W szczególności rozszerzenie deskryptorowe dowolnego uliniowienia deskryptorów jest uliniowieniem struktur. Możemy również zdefiniować pojęcie analogiczne do sumy teoriomnogościowej. **Sumą zbioru uliniowień deskryptorowych** $\Phi = \{\varphi_1, \ldots, \varphi_k\}$ nazwiemy uliniowienie strukturalne ξ o wsparciu Φ:

$$\xi = \bigsqcup \Phi = \varphi_1 \sqcup \varphi_2 \sqcup \ldots \sqcup \varphi_k$$

Liczność dziedziny ξ nazywać będziemy **wielkością uliniowienia** i oznaczać przez $|\xi|$.

Koncepcja uliniowienia struktur i wsparcia deskryptorowego jest analogiczna do deskryptorowej dopuszczalności parowania kontaktów z poprzedniego rozdziału. W przypadku uliniowień deskryptorów musiało istnieć odwzorowanie pomiędzy aminokwasami zachowujące ciągłość elementów. W przypadku uliniowień struktur każda para odpowiadających sobie aminokwasów musi należeć do pewnego uliniowienia deskryptorów, które całe zawiera się w rozważanym uliniowieniu struktur.

Powiemy, że parowanie bądź uliniowienie ξ jest **proste**, jeżeli odwzorowuje struktury zachowując kolejność aminokwasów, czyli spełnia warunek:

$$\bigwedge_{a^{(i)}, a^{(j)} \in Dom(\xi)} b^{(k)} = \xi(a^{(i)}) \wedge b^{(l)} = \xi(a^{(j)}) \wedge i \leq j \Rightarrow k \leq l$$

Uliniowienie proste odpowiada standardowemu rozumieniu tego pojęcia w kontekście minimalizowania odległości edycyjnej, czy też algorytmów opartych na programowaniu dynamicznym (patrz rozdział 1).

Wreszcie, powiemy, że uliniowienie ξ jest **spójne**, jeżeli dla każdych dwóch uliniowień deskryptorowych φ_1, φ_2 należących do jego wsparcia istnieje zawarty w nim ciąg $\varphi_1\chi_1 \ldots \chi_k\varphi_2$, którego kolejne elementy mają niepuste przecięcie dziedziny. Relację przecinania się dziedziny nazwiemy **nakładaniem** i będziemy ją oznaczali symbolem $\bowtie$. Nieformalnie, uliniowienie jest spójne, jeżeli uliniowienia deskryptorowe, z których się ono składa, tworzą ciągły przestrzennie szkielet.

Zbiór wszystkich uliniowień struktur S_1, S_2 przy wsparciu Φ będziemy oznaczali przez $Al(S_1, S_2, \Phi)$. Czasami nie będziemy podawać zbioru Φ, a Al będzie zbiorem wszystkich uliniowień, dla których przy ustalonych parametrach sekwencyjnej i strukturalnej dopuszczalności uliniowień deskryptorowych istnieje wsparcie.

3.2. NP-zupełność problemu znajdowania maksymalnego uliniowienia struktur

Definicja 9. Dla danego zbioru uliniowień deskryptorów Φ struktur S_1 i S_2 **problem najlepszych uliniowień** polega na znalezieniu uliniowienia struktur S_1 i S_2 przy wsparciu Φ o maksymalnej wielkości. Dla danego zbioru uliniowień deskryptorów Φ struktur S_1 i S_2 i liczby $n \in \mathbb{N}$ decyzyjny **problem optymalnego uliniowienia struktur** (POUS) polega na rozstrzygnięciu, czy istnieje uliniowienie S_1 i S_2 przy wsparciu Φ o wielkości nie mniejszej niż n.

Twierdzenie 3. *Problem optymalnego uliniowienia struktur jest NP-zupełny.*

Dowód. Łatwo sprawdzić, że $POUS \in NP$. Algorytm sprawdzający, czy dane uliniowienie ma wsparcie deskryptorowe ma oczywiście wielomianową złożoność obliczeniową, zaś obliczenie wielkości uliniowienia jest trywialne.

Przypomnijmy znany z literatury NP-zupełny problem 3-DIMENSIONAL MATCHING (3DM)[24, problem SP16].

Definicja 10. Dla danego zbioru $M \subseteq W \times X \times Y$, gdzie W, X i Y są rozłącznymi zbiorami liczności q, **problem 3-DIMENSIONAL MATCHING** polega na określeniu, czy istnieje podzdbiór $M' \subseteq M$ taki, że $|M'| = q$ oraz elementy M' są rozłączne.

Istnieje również wielomianowy wariant problemu 3DM – 2-DIMENSIONAL MATCHING (2DM), gdzie $M \subseteq X \times Y$ zwany problemem kojarzenia małżeństw. Zaczniemy od pokazania, że jeżeli w 2DM wprowadzimy możliwość wzajemnego wymuszania się pewnych par, stanie się on NP-zupełny.

Definicja 11. Dla danych zbiorów $M \subseteq X \times Y$ oraz $G \subseteq P(M)$[2], gdzie X i Y są rozłącznymi zbiorami liczności q, **problem RESTRICTED 2-DIMENSIONAL MATCHING** (R2DM) polega na rozstrzygnięciu, czy istnieje podzbiór $M' \subseteq M$ taki, że $|M'| = q$, elementy M' są rozłączne oraz istnieje zbiór $G' \subseteq G$ taki, że $\bigcup G' = M'$.

[2]Symbol $P(M)$ oznacza zbiór potęgowy zbioru M, czyli rodzinę wszystkich podzbiorów M.

Lemat 2. *Problem RESTRICTED 2-DIMENSIONAL MATCHING jest NP-zupełny.*

Dowód. Dowód, że $R2DM \in NP$ jest łatwy. Wystarczy zauważyć, że sprawdzenie, czy dla każdego $m \in M'$ istnieje $G_i \in G$ takie, że $m \in G_i$ i $G_i \subseteq M'$ jest wykonalne w czasie wielomianowym. Aby udowodnić NP-zupełność, przeprowadzimy redukcję problemu 3DM do R2DM. Niech $M \subseteq W \times X \times Y$ ($|W| = |X| = |Y| = q$) będzie instancją problemu 3DM. Zdefiniujemy pomocniczy zbiór X' rozłączny ze zbiorami W, X, Y o liczności q zawierający elementy odpowiadające elementom zbioru X. Dla każdej należącej do M trójki $m_i = \langle w_{l(i)}, x_{m(i)}, y_{n(i)} \rangle$, gdzie $l(i)$, $m(i)$, $n(i)$ oznaczają indeksy elementów ze zbiorów W, X, Y w i-tej trójce, zdefiniujemy następujący zestaw par:

$$A_i = \left\{ \langle w_{l(i)}, x_{m(i)} \rangle, \langle x'_{m(i)}, y_{n(i)} \rangle \right\}$$

Możemy teraz opisać zbiór par $A \subseteq P \times Q$ konstruowanej instancji problemu R2DM:

$$P = W \cup X'$$

$$Q = X \cup Y$$

$$A = \bigcup_{1 \leq i \leq |M|} A_i$$

Aby zakończyć konstrukcję musimy jeszcze określić zbiór G. Posłuży on do tego, żeby zagwarantować, że jeżeli jedna para ze zbioru A_i należy do rozwiązania problemu, druga para z A_i również należy to tego rozwiązania:

$$G = \{A_i | 1 \leq i \leq |M|\}$$

Zauważmy, że zbiory zawarte w G są przy tej konstrukcji rozłączne. P i Q liczą po $2q$ elementów, zbiór A zawiera $2|M|$ par, zaś G zawiera $|M|$ dwuelementowych zbiorów, co dowodzi, że konstrukcja ma złożoność wielomianową.

Przykład 1. Niech $M \subseteq W \times X \times Y$ będzie następującą instancją problemu 3DM:

$$\begin{aligned} W &= \{\mathrm{a}, \mathrm{b}, \mathrm{c}\} \\ X &= \{\mathrm{A}, \mathrm{B}, \mathrm{C}\} \\ Y &= \{1, 2, 3\} \\ M &= \{\langle \mathrm{a}, \mathrm{A}, 1 \rangle, \langle \mathrm{b}, \mathrm{B}, 2 \rangle, \langle \mathrm{c}, \mathrm{C}, 3 \rangle, \langle \mathrm{a}, \mathrm{B}, 3 \rangle\} \end{aligned}$$

Dla tak dobranych zbiorów istnieje zbiór $M' = \{\langle \mathrm{a}, \mathrm{A}, 1 \rangle, \langle \mathrm{b}, \mathrm{B}, 2 \rangle, \langle \mathrm{c}, \mathrm{C}, 3 \rangle\}$, który jest rozwiązaniem problemu 3DM. Odpowiadająca instancja problemu R2DM będzie miała postać:

$$
\begin{aligned}
P &= \{\mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{A}', \mathrm{B}', \mathrm{C}'\} \\
Q &= \{\mathrm{A}, \mathrm{B}, \mathrm{C}, 1, 2, 3\} \\
A &= \left\{ \begin{array}{ll} \langle \mathrm{a}, \mathrm{A} \rangle, & \langle \mathrm{A}', 1 \rangle, \\ \langle \mathrm{b}, \mathrm{B} \rangle, & \langle \mathrm{B}', 2 \rangle, \\ \langle \mathrm{c}, \mathrm{C} \rangle, & \langle \mathrm{C}', 3 \rangle, \\ \langle \mathrm{a}, \mathrm{B} \rangle, & \langle \mathrm{B}', 3 \rangle \end{array} \right\} \\
G &= \left\{ \begin{array}{l} \{\langle \mathrm{a}, \mathrm{A} \rangle, \langle \mathrm{A}', 1 \rangle\}, \\ \{\langle \mathrm{b}, \mathrm{B} \rangle, \langle \mathrm{B}', 2 \rangle\}, \\ \{\langle \mathrm{c}, \mathrm{C} \rangle, \langle \mathrm{C}', 3 \rangle\}, \\ \{\langle \mathrm{a}, \mathrm{B} \rangle, \langle \mathrm{B}', 3 \rangle\} \end{array} \right\}
\end{aligned}
$$

Jak łatwo zauważyć zbiór:

$$
A' = \left\{ \begin{array}{ll} \langle \mathrm{a}, \mathrm{A} \rangle, & \langle \mathrm{A}', 1 \rangle, \\ \langle \mathrm{b}, \mathrm{B} \rangle, & \langle \mathrm{B}', 2 \rangle, \\ \langle \mathrm{c}, \mathrm{C} \rangle, & \langle \mathrm{C}', 3 \rangle \end{array} \right\}
$$

spełnia warunki wymagane dla rozwiązania problemu R2DM, co kończy przykład.

Załóżmy, że istnieje podzbiór $M' \subseteq M$ będący rozwiązaniem rozważanej instancji 3DM. Dla uproszczenia notacji niech $c(\cdot)$ oznacza indeksy trójek z M występujących w zbiorach M'. Weźmy zbiór $A' \subseteq A$ określony wzorem:

$$
A' = \bigcup_{1 \leq i \leq q} A_{c(i)} = \bigcup_{1 \leq i \leq q} \left\{ \left\langle w_{l(c(i))}, x_{m(c(i))} \right\rangle, \left\langle x'_{m(c(i))}, y_{n(c(i))} \right\rangle \right\}
$$

Jak łatwo zauważyć A' zawiera pary kodujące trójki należące do M'. Udowodnimy, że jest to rozwiązanie odpowiadającej rozwiązaniu instancji problemu R2DM. Zaczniemy od zbadania, czy A' pokrywa zbiory P i Q. Zauważmy, że A' zawiera $2q$ par. Wszystkie elementy zbiorów P i Q należą do jednej z par w A', ponieważ:

(1) W, X, Y – dla każdego elementu z tych zbiorów istnieje trójka w M', która go zawiera;

(2) X' – dla każdego j zbiór A_i, który pokrywa x_j, pokrywa również x'_j.

Z konstrukcji A' i G wynika natychmiast, że spełniony jest warunek o zawieraniu się elementów G w A'. Z liczności $|A'| = 2q$ i faktu, że A' pokrywa zbiory P i Q wynika, że żadne dwa jego elementy nie są równe na żadnej z pozycji.

Załóżmy teraz, że A' jest rozwiązaniem skonstruowanej instancji R2DM. Z definicji R2DM wiemy, że istnieje $G' \subseteq G$ taki, że $A' = \bigcup G'$, zaś na podstawie konstrukcji G wnioskujemy, że:

$$A' = \bigcup_{1 \leq i \leq q} A_{e(i)}$$

gdzie $e(\cdot)$ jest pewnym ciągiem indeksów. Niech M' będzie określone wzorem:

$$M' = \left\{ \left\langle w_{l(e(i))}, x_{m(e(i))}, y_{n(e(i))} \right\rangle \middle| 1 \leq i \leq q \right\}$$

Łatwo sprawdzić, że M' jest rozwiązaniem problemu 3DM. Pokrywanie zbiorów W, X, Y wynika z pokrywania przez A' zbioru P i Q. Wzajemna rozłączność trójek wynika z konstrukcji zbiorów A_i. □

Przedstawimy teraz redukcję instancji $M \subseteq X \times Y$, $G \subseteq P(M)$, $|X| = |Y| = q$, $|G| = r$ problemu R2DM do POUS. Czytelnik na pewno łatwo zauważy, że struktura problemu POUS jest znacznie bardziej skomplikowana. Dlatego w naszej konstrukcji będziemy posługiwać się nieco uproszczoną notacją. Niech $S_1 = \{a_1, \ldots, a_{5q}\}$ i $S_2 = \{b_1, \ldots, b_{5q}\}$ będą strukturami liczącymi po $5q$ aminokwasów, zaś $e_i = El(a_{5i-2})$, $f_i = El(b_{5i-2})$ (dla $1 \leq i \leq q$) ciągami elementów deskryptorowych. Widać, że tak określone elementy są rozłączne i pokrywają całe struktury S_1 i S_2. Ciągi $E = \{e_i\}$ i $F = \{f_i\}$ będą w naszej konstrukcji odpowiadały elementom zbiorów X i Y.

Zauważmy, że zbiór par $\left\{ \left\langle e_{k(1)}, f_{l(1)} \right\rangle, \ldots, \left\langle e_{k(n)}, f_{l(n)} \right\rangle \right\}$ określa parę deskryptorów D_1 i D_2 oraz pewne uliniowienie tych deskryptorów φ dane wzorami:

$$\begin{aligned} D_1 &= \left\langle e_{k(1)}, \left\{ e_{k(2)}, \ldots, e_{k(n)} \right\}, R_1 \right\rangle \\ D_2 &= \left\langle f_{l(1)}, \left\{ f_{l(2)}, \ldots, f_{l(n)} \right\}, R_2 \right\rangle \\ \varphi(e_{k(i)}) &= f_{l(i)} \end{aligned}$$

Niech $\Phi = \{\varphi_1, \ldots, \varphi_r\}$ będzie zbiorem uliniowień, a $\mathcal{D}_1$ i $\mathcal{D}_2$ zbiorami deskryptorów zbudowanych w powyższy sposób dla elementów zbioru G. Otrzymaliśmy w ten sposób pełną instancję problemu POUS. Pokażemy, że zbiór M' będący rozwiązaniem R2DM istnieje wtedy i tylko wtedy, gdy struktury S_1 i S_2 mają przy wsparciu Φ uliniowienie liczące $5q$ aminokwasów.

Załóżmy, że $M' = \left\{ m_{g(1)}, \ldots, m_{g(q)} \right\}$ jest rozwiązaniem problemu R2DM. Z definicji R2DM wynika, że istnieje pewna rodzina zbiorów $G' = \left\{ G_{h(1)}, \ldots, G_{h(r')} \right\}$ zawarta w G taka, że $\bigcup G' = M'$. Pokażemy, że istnieje uliniowienie ξ o wsparciu $\Phi' = \left\{ \varphi_{h(1)}, \ldots, \varphi_{h(r')} \right\} \subseteq \Phi$. Wynika to z faktu, że dla każdego $x_i \in X$ istnieje dokładnie jeden element $y_j \in Y$ taki, że $\langle x_i, y_j \rangle \in M'$, skąd wniosek, że:

$$\bigwedge_{\varphi \in \Phi'} e_i \in Dom(\varphi) \Rightarrow \varphi(e_i) = f_j$$

Ponadto z faktu, że elementy G' pokrywają M' natychmiast wynika, że $Dom(\xi) = E$ i $|\xi| = 5q$.

Aby dowieść implikacji odwrotnej, załóżmy teraz, że ξ jest uliniowieniem struktur S_1 i S_2 o wsparciu $\Phi' = \{\varphi_{h(1)}, \ldots, \varphi_{h(r')}\} \subseteq \Phi$ i liczności $5q$. Pokażemy, że zbiór $M' = \{\langle x_i, y_j\rangle \in X \times Y | \xi(e_i) = f_j\}$ jest rozwiązaniem R2DM. Ponieważ ξ jest bijekcją, pary w M' są rozłączne i pokrywają zbiory X i Y. Wystarczy zatem pokazać, że istnieje zbiór G' taki, że $\bigcup G' = M'$. Oczywiście zbiór $G' = \{G_{h(1)}, \ldots, G_{h(r')}\}$ złożony z elementów G odpowiadających uliniowieniom z Φ' jest odpowiedni. □

3.3. Szczególne przypadki problemu znajdowania maksymalnego uliniowienia struktur i ich złożoność

Podstawową wadą przedstawionego w poprzednim podrozdziale twierdzenia jest jego ogólność. Z poprzedniego rozdziału wiemy wszakże, że w rzeczywistych zastosowaniach liczba segmentów w deskryptorze jest niewielka. Dlatego praktyczne znaczenie miałoby zbadanie, jaką złożoność ma problem znajdowania maksymalnego uliniowienia przy ograniczeniu liczby segmentów w deskryptorze. Złożoność problemu może również zależeć od maksymalnej liczby przestawień dopuszczalnych w uliniowieniu.

Definicja 12. Dla danego zbioru uliniowień deskryptorów Φ struktur S_1 i S_2 oraz liczb $s \in \mathbb{N}$ i $n \in \mathbb{N}$ takich, że:

$$\bigwedge_{\varphi \in \Phi} |seg(D_{1,2}|_\varphi)| \leq s, \text{ gdzie } D_1 \text{ i } D_2 \text{ są deskryptorami uliniawianymi przez } \varphi$$

decyzyjny **s-ograniczony problem optymalnego uliniowienia struktur** (POUS-s) polega na rozstrzygnięciu, czy istnieje uliniowienie S_1 i S_2 przy wsparciu Φ o wielkości nie mniejszej niż n, zaś decyzyjny **s-ograniczony problem optymalnego prostego uliniowienia struktur** (POPUS-s) polega na rozstrzygnięciu, czy istnieje proste uliniowienie spełniające powyższe warunki.

Przedstawimy teraz serię twierdzeń o złożoności problemów POUS-s i POPUS-s dla różnych wartości s.

Twierdzenie 4. *Problem POPUS-1 jest rozwiązywalny w czasie wielomianowym.*

Dowód. Tak postawiony problem jest rozwiązywalny przy użyciu metody programowania dynamicznego, a konkretnie pewnej modyfikacji algorytmu Needlemana–Wunscha.

W standardowej wersji tego algorytmu wypełniana jest macierz $F \in \mathbb{R}^{n \times m}$, gdzie m i n są długościami uliniawianych sekwencji. Dla ustalonej funkcji wartościującej dopasowanie danych elementów $s\colon \{1,\ldots,n\} \times \{1,\ldots,m\} \to \mathbb{R}$ oraz braku kary za wstawienie spacji, rekurencyjna definicja macierzy F ma postać:

$$\begin{aligned} F_{0j} &= 0 \\ F_{i0} &= 0 \\ F_{ij} &= \max\left(F_{i(j-1)}, F_{(i-1)j}, F_{(i-1)(j-1)} + s_{ij}\right) \end{aligned}$$

Jeżeli S jest zbiorem odpowiadających sobie par segmentów, reprezentowanych w postaci krotek $\langle \langle p_1, k_1 \rangle, \langle p_2, k_2 \rangle, v \rangle$, gdzie $p_{1,2}$ i $k_{1,2}$ są odpowiednio początkiem i końcem segmentu w pierwszej i drugiej strukturze, a v wartością dopasowania tych segmentów, algorytm ulega następującej modyfikacji:

$$\begin{aligned} G_{ij} &= \max_{\substack{\langle \langle p_1,k_1 \rangle, \langle p_2,k_2 \rangle, v \rangle \in S \\ k_1 = i \wedge k_2 = j}} F_{p_1 p_2} + v \\ F_{ij} &= \max\left(F_{i(j-1)}, F_{(i-1)j}, G_{ij}\right) \end{aligned}$$

□

Twierdzenie 5. *Problem POUS-2 jest NP-zupełny.*

Dowód. Dowód twierdzenia 3 jest prawdziwy również dla problemu POUS-2, ponieważ lemat 2 jest spełniony przy założeniu, że zbiory rodziny G liczą nie więcej niż dwa elementy. □

Twierdzenie 6. *Problem POPUS-3 jest NP-zupełny.*

Dowód. Fakt przynależności problemu do klasy NP jest oczywisty i nie wymaga komentarza. NP-zupełność udowodnimy wykonując redukcję znanego problemu 3SAT w wariancie zakładającym, że każda zmienna występuje dokładnie w trzech klauzulach[24, problem LO01].

Definicja 13. Dla ustalonego zbioru zmiennych U oraz zbioru klauzul logicznych C nad zbiorem U takich, że każda klauzula $c \in C$ liczy dokładnie trzy literały, **problem 3SAT** polega na rozstrzygnięciu, czy istnieje wartościowanie zmiennych ze zbioru U takie, że wszystkie klauzule z C są prawdziwe.

Niech $U = \{u_1,\ldots,u_k\}$, $C = \{C_1,\ldots,C_l\}$ będzie instancją problemu 3SAT. Weźmy dwie struktury $S_1 = a^{(1)}a^{(2)}\ldots a^{(5(k+l))}$ i $S_2 = b^{(1)}b^{(2)}\ldots b^{(6(k+l))}$ liczące odpowiednio $5(k+l)$ i $6(k+l)$ aminokwasów. Niech $E = \{e_i\}_{i=1}^{k+l}$, gdzie $e_i = El(a^{(5i-2)})$, będzie zbiorem rozłącznych elementów struktury S_1, a $F = \{f_i\}_{i=1}^{k+l}$, $\overline{F} = \left\{\overline{f}_i\right\}_{i=1}^{k+l}$, gdzie $f_i =$

$El(b^{(6i-3)})$, $\overline{f}_i = El(b^{(6i-2)})$, będą zbiorami elementów struktury S_2. Zauważmy, że dla $i \neq j$ elementy f_i i f_j oraz f_i i $\overline{f}_j$ są rozłączne, oraz że tak zdefiniowane zbiory liczą po $k + l$ elementów. Elementy o indeksach w przedziale $[1, k]$ będą w naszej redukcji odpowiadać zmiennym, a elementy o indeksach w przedziale $[k + 1, k + l]$ klauzulom. Niech $P_i, N_i \subseteq C$ będą odpowiednio zbiorami klauzul, w których zmienna u_i występuje pozytywnie i negatywnie. Skonstruujemy zbiory uliniowień deskryptorów Φ i $\overline{\Phi}$ odpowiadające pozytywnym i negatywnym wystąpieniom zmiennych. Niech φ_i będące uliniowieniem związanym z pozytywnymi wystąpieniami u_i będzie zdefiniowane następująco:

$$\varphi_i(e_p) = \begin{cases} f_p & p = i \\ f_p & C_{p-k} \in P_i \end{cases}$$

Analogicznie zdefiniujemy $\overline{\varphi}_i$, które będzie odpowiadało negatywnym wystąpieniom u_i:

$$\overline{\varphi}_i(e_p) = \begin{cases} \overline{f}_p & p = i \\ f_p & C_{p-k} \in N_i \end{cases}$$

W powyższych definicjach pominęliśmy łatwe do uzupełnienia szczegóły związane z określeniem uliniawianych deskryptorów i wyróżnieniem ich centralnych elementów. Łatwo zauważyć, że φ_i i $\overline{\varphi_j}$ nie mogą należeć do wsparcia tego samego uliniowienia wtedy i tylko wtedy, gdy $i = j$, gdyż różnią się wartością dla e_i, oraz że dla dowolnych φ_i i φ_j oraz $\overline{\varphi}_i$ i $\overline{\varphi_j}$ istnieje uliniowienie, do wsparcia którego należą (por. definicja 14). Niech Φ będzie zbiorem uliniowień zdefiniowanych w powyższy sposób:

$$\Phi = \{\varphi_i | 1 \leq i \leq k\} \cup \{\overline{\varphi}_i | 1 \leq i \leq k\}$$

Przykład 2. Niech C będzie zbiorem klauzul nad zbiorem zmiennych $U = \{u_1, u_2, u_3\}$:

$$C = \{\{\neg u_1, u_2, u_3\}, \{u_1, \neg u_2, u_3\}, \{u_1, u_2, \neg u_3\}\}$$

odpowiadającym formule logicznej:

$$(\neg u_1 \vee u_2 \vee u_3) \wedge (u_1 \vee \neg u_2 \vee u_3) \wedge (u_1 \vee u_2 \vee \neg u_3)$$

Instancji problemu 3SAT dla powyższej formuły odpowiada następująca instancja POPUS-3 ($\varphi(a) = \perp$ oznacza, że $a \notin Dom(\varphi)$):

$$S_1 = \overbrace{a^{(1)}\, a^{(2)}\, a^{(3)}\, a^{(4)}\, a^{(5)}}^{e_1} \overbrace{a^{(6)}\, a^{(7)}\, a^{(8)}\, a^{(9)} a^{(10)}}^{e_2} \overbrace{a^{(11)} a^{(12)} a^{(13)} a^{(14)} a^{(15)}}^{e_3}$$
$$\overbrace{a^{(16)} a^{(17)} a^{(18)} a^{(19)} a^{(20)}}^{e_4} \overbrace{a^{(21)} a^{(22)} a^{(23)} a^{(24)} a^{(25)}}^{e_5} \overbrace{a^{(26)} a^{(27)} a^{(28)} a^{(29)} a^{(30)}}^{e_6}$$

$$S_2 = \overbrace{b^{(1)}\,\underbrace{b^{(2)}\;b^{(3)}\;b^{(4)}\;b^{(5)}}_{\overline{f}_1}\;b^{(6)}}^{f_1}\;\overbrace{b^{(7)}\,\underbrace{b^{(8)}\;b^{(9)}\,b^{(10)}b^{(11)}}_{\overline{f}_2}\,b^{(12)}}^{f_2}\;\overbrace{b^{(13)}\underbrace{b^{(14)}b^{(15)}b^{(16)}b^{(17)}}_{\overline{f}_3}\,b^{(18)}}^{f_3}$$
$$\overbrace{b^{(19)}\underbrace{b^{(20)}b^{(21)}b^{(22)}b^{(23)}}_{\overline{f}_4}\,b^{(24)}}^{f_4}\;\overbrace{b^{(25)}\underbrace{b^{(26)}b^{(27)}b^{(28)}b^{(29)}}_{\overline{f}_5}\,b^{(30)}}^{f_5}\;\overbrace{b^{(31)}\underbrace{b^{(32)}b^{(33)}b^{(34)}b^{(35)}}_{\overline{f}_6}\,b^{(36)}}^{f_6}$$

$$\vec{\varphi}_1(S_1) = \overbrace{\underbrace{b^{(1)}\;b^{(2)}\;b^{(3)}\;b^{(4)}\;b^{(5)}}_{f_1}}^{e_1}\;\overbrace{\perp\;\perp\;\perp\;\perp\;\perp}^{e_2}\;\overbrace{\perp\;\perp\;\perp\;\perp\;\perp}^{e_3}$$
$$\overbrace{\perp\;\perp\;\perp\;\perp\;\perp}^{e_4}\;\overbrace{\underbrace{b^{(25)}b^{(26)}b^{(27)}b^{(28)}b^{(29)}}_{f_5}}^{e_5}\;\overbrace{\underbrace{b^{(31)}b^{(32)}b^{(33)}b^{(34)}b^{(35)}}_{f_6}}^{e_6}$$

$$\overrightarrow{\varphi}_1(S_1) = \overbrace{\underbrace{b^{(2)}\;b^{(3)}\;b^{(4)}\;b^{(5)}\;b^{(6)}}_{\overline{f}_1}}^{e_1}\;\overbrace{\perp\;\perp\;\perp\;\perp\;\perp}^{e_2}\;\overbrace{\perp\;\perp\;\perp\;\perp\;\perp}^{e_3}$$
$$\overbrace{\underbrace{b^{(19)}b^{(20)}b^{(21)}b^{(22)}b^{(23)}}_{f_4}}^{e_4}\;\overbrace{\perp\;\perp\;\perp\;\perp\;\perp}^{e_5}\;\overbrace{\perp\;\perp\;\perp\;\perp\;\perp}^{e_6}$$

$$\vec{\varphi}_2(S_1) = \overbrace{\perp\;\perp\;\perp\;\perp\;\perp}^{e_1}\;\overbrace{\underbrace{b^{(7)}\;b^{(8)}\;b^{(9)}\,b^{(10)}b^{(11)}}_{f_2}}^{e_2}\;\overbrace{\perp\;\perp\;\perp\;\perp\;\perp}^{e_3}$$
$$\overbrace{\underbrace{b^{(19)}b^{(20)}b^{(21)}b^{(22)}b^{(23)}}_{f_4}}^{e_4}\;\overbrace{\perp\;\perp\;\perp\;\perp\;\perp}^{e_5}\;\overbrace{\underbrace{b^{(31)}b^{(32)}b^{(33)}b^{(34)}b^{(35)}}_{f_6}}^{e_6}$$

$$\overrightarrow{\varphi}_2(S_1) = \overbrace{\perp\;\perp\;\perp\;\perp\;\perp}^{e_1}\;\overbrace{\underbrace{b^{(7)}\;b^{(8)}\;b^{(9)}\,b^{(10)}b^{(11)}}_{\overline{f}_2}}^{e_2}\;\overbrace{\perp\;\perp\;\perp\;\perp\;\perp}^{e_3}$$
$$\overbrace{\perp\;\perp\;\perp\;\perp\;\perp}^{e_4}\;\overbrace{\underbrace{b^{(25)}b^{(26)}b^{(27)}b^{(28)}b^{(29)}}_{f_5}}^{e_5}\;\overbrace{\perp\;\perp\;\perp\;\perp\;\perp}^{e_6}$$

$$\vec{\varphi}_3(S_1) = \overbrace{\perp\;\perp\;\perp\;\perp\;\perp}^{e_1}\;\overbrace{\perp\;\perp\;\perp\;\perp\;\perp}^{e_2}\;\overbrace{\underbrace{b^{(13)}b^{(14)}b^{(15)}b^{(16)}b^{(17)}}_{f_2}}^{e_3}$$
$$\overbrace{\underbrace{b^{(19)}b^{(20)}b^{(21)}b^{(22)}b^{(23)}}_{f_4}}^{e_4}\;\overbrace{\underbrace{b^{(25)}b^{(26)}b^{(27)}b^{(28)}b^{(29)}}_{f_5}}^{e_5}\;\overbrace{\perp\;\perp\;\perp\;\perp\;\perp}^{e_6}$$

$$\overrightarrow{\overline{\varphi}_3}(S_1) = \overbrace{\underbrace{\bot\ \bot\ \bot\ \bot\ \bot}_{e_1}\ \underbrace{\bot\ \bot\ \bot\ \bot\ \bot}_{e_2}\ \underbrace{b^{(13)}b^{(14)}b^{(15)}b^{(16)}b^{(17)}}_{e_3}}^{\overline{f}_2}$$
$$\overbrace{\underbrace{\bot\ \bot\ \bot\ \bot\ \bot}_{e_4}\ \underbrace{\bot\ \bot\ \bot\ \bot\ \bot}_{e_5}\ \underbrace{b^{(31)}b^{(32)}b^{(33)}b^{(34)}b^{(35)}}_{e_6}}^{f_6}$$

Rozważana formuła jest spełniona między innymi dla wartościowania:

$$u_1 \to 0, u_2 \to 1, u_3 \to 1$$

Zatem istnieje uliniowienie o wsparciu $\{\overline{\varphi}_1, \varphi_2, \varphi_3\}$:

$$\vec{\xi}(S_1) = \overbrace{\underbrace{b^{(2)}\ b^{(3)}\ b^{(4)}\ b^{(5)}\ b^{(6)}}_{e_1}\ \underbrace{b^{(7)}\ b^{(8)}\ b^{(9)}\ b^{(10)}b^{(11)}}_{e_2}\ \underbrace{b^{(13)}b^{(14)}b^{(15)}b^{(16)}b^{(17)}}_{e_3}}^{\overline{f}_1\quad\quad f_2\quad\quad f_2}$$
$$\overbrace{\underbrace{b^{(19)}b^{(20)}b^{(21)}b^{(22)}b^{(23)}}_{e_4}\ \underbrace{b^{(25)}b^{(26)}b^{(27)}b^{(28)}b^{(29)}}_{e_5}\ \underbrace{b^{(31)}b^{(32)}b^{(33)}b^{(34)}b^{(35)}}_{e_6}}^{f_4\quad\quad f_5\quad\quad f_6}$$

ξ jest uliniowieniem prostym o wielkości 30, co kończy przykład.

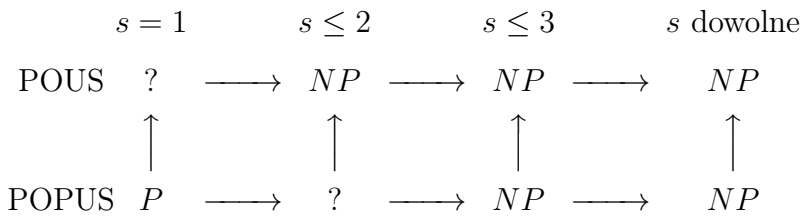
Udowodnimy, że struktury S_1 i S_2 mają uliniowienie proste wielkości $5(k+l)$ przy wsparciu Φ wtedy i tylko wtedy, gdy zbiór klauzul C jest spełnialny.

Przypuśćmy, że istnieje wartościowanie zmiennych $\sigma\colon U \to \{0,1\}$, przy którym wszystkie klauzule z C mają wartość logiczną 1. Weźmy zbiór $\Phi' \subseteq \Phi$ zawierający uliniowienia odpowiadające wartościowaniu σ:

$$\Phi' = \{\varphi_i \in \Phi | \sigma(u_i) = 1\} \cup \{\overline{\varphi}_i \in \Phi | \sigma(u_i) = 0\}$$

Tak określony podzbiór uliniowień jest wsparciem uliniowienia prostego struktur o wielkości $5(k+l)$.

Załóżmy teraz, że istnieje uliniowienie proste ξ o wielkości $5(k+l)$ o wsparciu w Φ. Zauważmy, że zbiór $\Phi' \subseteq \Phi$ będący wsparciem ξ musi dla każdego i zawierać φ_i lub $\overline{\varphi}_i$ (w przeciwnym wypadku ξ nie pokrywałoby wszystkich elementów $e_{1,\ldots,k}$). Równocześnie pokryte są wszystkie elementy $e_{k+1,\ldots,k+l}$ zatem dla każdej klauzuli istnieje należący do niej literał, który odpowiada pewnemu uliniowieniu w Φ'. Ponieważ wykazaliśmy, że do Φ' nie mogą należeć równocześnie uliniowienia odpowiadające literałowi



Rysunek 3.1: Złożoność problemów POUS i POPUS w zależności od maksymalnej liczby segmentów w deskryptorze s.

pozytywnemu i negatywnemu jednej zmiennej, Φ' jednoznacznie określa wartościowanie spełniające C:

$$\sigma(u_i) = \begin{cases} 1 & \varphi_i \in \Phi' \\ 0 & \overline{\varphi}_i \in \Phi' \end{cases}$$

Przedstawiona redukcja ma złożoność wielomianową, co kończy dowód. □

Podsumowanie rozważań, które przeprowadziliśmy w tym podrozdziale, ilustruje rysunek 3.1. Udowodniliśmy, że problem znajdowania uliniowień jest NP-zupełny nawet przy założeniu, że deskryptory mają nie więcej niż 2 segmenty. Radykalne uproszczenie problemu polegające na ograniczeniu zbioru rozwiązań do uliniowień prostych ma niewielki wpływ na złożoność, gdyż taki problem jest NP-zupełny dla deskryptorów mających nie więcej niż 3 segmenty. Znaki zapytania na rysunku oznaczają warianty problemu, których nie rozważaliśmy.

3.4. Grafowa reprezentacja problemu znajdowania maksymalnego uliniowienia struktur

W poprzednim rozdziale wykazaliśmy NP-zupełność problemu znajdowania optymalnego uliniowienia struktur. Wynika stąd, że uprawnione jest poszukiwanie analogii pomiędzy tym problemem a dobrze zbadanym problemem wyszukiwania klik w grafach. W tym podrozdziale przedstawimy grafową reprezentację problemu optymalnego uliniowienia oraz zdefiniujemy klasę funkcji wartościujących uliniowienia. Przyjmijmy na potrzeby poniższych rozważań, że S_1 i S_2 są strukturami, a Φ pewnym zbiorem uliniowień deskryptorów należących do tych struktur.

Definicja 14. Uliniowienia φ_1 i φ_2 są **niesprzeczne**, jeżeli istnieje uliniowienie struktur $\xi = \varphi_1 \sqcup \varphi_2$. Niesprzeczność uliniowień deskryptorowych oznaczać będziemy symbolem $\backsim$.

Bezpośrednią konsekwencją definicji jest fakt:

Fakt 5. *Dwa uliniowienia deskryptorów φ_1 i φ_2 są **niesprzeczne** wtedy i tylko wtedy, gdy ich rozszerzenia deskryptorowe spełniają warunek:*

$$\bigwedge_{a \in Dom(\widetilde{\varphi}_1) \cap Dom(\widetilde{\varphi}_2)} \widetilde{\varphi}_1(a) = \widetilde{\varphi}_2(a)$$

Relacja niesprzeczności uliniowień deskryptorów w naturalny sposób uogólnia się na uliniowienia struktur. W dalszych rozważaniach będziemy posługiwać się nią w obu kontekstach. Mając relację możemy zdefiniować graf niesprzeczności.

Definicja 15. Grafem niesprzeczności zbioru uliniowień Φ nazwiemy graf $G = \langle V, E \rangle$, którego zbiory wierzchołków i krawędzi są dane wzorami:

$$V = \Phi$$
$$E = \{\langle \varphi_i, \varphi_j \rangle \in \Phi \times \Phi | \varphi_i \backsim \varphi_j\}$$

Fakt 6. *Jeżeli $C \subseteq V$ jest pewnym podzbiorem wierzchołków G, uliniowienie struktur $\xi = \bigsqcup C$ istnieje wtedy i tylko wtedy, gdy C jest kliką w G.*

Dowód. Prawdziwości faktu dowodzimy przez indukcję zauważając, że dla trzech uliniowień $\xi_{1,2,3}$ zachodzi:

$$\xi_1 \backsim \xi_2 \wedge \xi_2 \backsim \xi_3 \wedge \xi_1 \backsim \xi_3 \Leftrightarrow (\xi_1 \sqcup \xi_2) \backsim \xi_3$$

□

Do tego momentu w naszych rozważaniach jedyną właściwością uliniowienia struktur, którą optymalizowaliśmy była jego wielkość. W praktyce mogą występować inne cechy, które również mogłyby podlegać optymalizacji. Rozważmy funkcję $s\colon Al \to \mathbb{R}$. Powiemy, że jest ona **monotoniczną miarą podobieństwa** jeżeli:

$$\bigwedge_{\xi_{1,2} \in Al} \xi_1 \subseteq \xi_2 \Rightarrow s(\xi_1) \leq s(\xi_2)$$

Zauważmy, że maksymalne w sensie zawierania kliki G są związane z lokalnymi maksimami s.

Fakt 7. *Jeżeli $s(\xi)$ jest globalnym maksimum s nad zbiorem $Al(S_1, S_2, \Phi)$, to odpowiadająca mu klika C jest maksymalna pod względem zawierania.*

W ten sposób powiązaliśmy problem znajdowania uliniowień maksymalizujących zadaną funkcję z dobrze zbadanym problemem znajdowania maksymalnych klik. W dalszej części tego rozdziału przedstawimy algorytmy zaimplementowane na potrzeby tego problemu.

3.5. Algorytmy znajdowania maksymalnego uliniowienia struktur

Zaimplementowaliśmy trzy rodzaje algorytmów poszukujących uliniowienia struktur o maksymalnej mierze. Algorytmy dokładne (TS - Tree Search i CTS - Continuous Tree Search) oparte są o analizę drzewa decyzyjnego z wykorzystaniem odcięć. Mają one wykładniczą pesymistyczną złożoność obliczeniową, ale gwarantują znalezienie uliniowienia maksymalizującego miarę podobieństwa przy założeniu, że jest ona monotoniczna. Algorytm probabilistyczny (REMC) oparty na metodzie Monte-Carlo z wymianą replik może maksymalizować dowolną miarę podobieństwa. Wreszcie zaproponowany został algorytm przybliżony (MS) oparty na twierdzeniu Motzkina-Strausa o związku kliki o maksymalnej liczności z maksimami pewnej formy kwadratowej, co pozwala znajdować najliczniejszą klikę w grafie niesprzeczności z nadzieją, że maksymalizuje ona miarę podobieństwa.

3.5.1. Algorytmy dokładne – TS i CTS

Zgodnie z faktem 7, jeżeli nie można skorzystać z dodatkowych właściwości miary podobieństwa, aby znaleźć uliniowienie dwóch struktur, które ją maksymalizuje, należy obliczyć jej wartość dla wszystkich maksymalnych klik. Zaprezentujemy algorytm oparty na metodzie podziałów i ograniczeń (ang. *branch and bound*). Zdefiniujmy najpierw pewne pomocnicze operacje.

(1) Niech $ext(C)$ będzie zbiorem węzłów, o które można rozszerzyć klikę C:

$$ext(C) = \left\{ v \in V \setminus C \middle| \bigwedge_{c \in C} v \backsim c \right\}$$

(2) Niech $contr(C)$ będzie zbiorem węzłów sprzecznych z co najmniej jednym elementem C:

$$contr(C) = \left\{ v \in V \setminus C \middle| \bigvee_{c \in C} v \not\backsim c \right\} = V \setminus (C \cup ext(C))$$

(3) Niech $overlap(C)$ będzie podzbiorem węzłów z $ext(C)$, które nakładają się na co najmniej jeden węzeł z C:

$$overlap(C) = \left\{ v \in V \setminus C \middle| \bigwedge_{c \in C} v \backsim c \wedge \bigvee_{c \in C} c \bowtie v \right\}$$

(4) Niech $cont(C)$ będzie analogicznym zbiorem dla domknięcia przechodniego relacji nakładania:

$$cont(C) = \left\{ v \in V \setminus C \middle| \bigwedge_{c \in C} v \backsim c \wedge \bigvee_{c \in C} c \bowtie^{+} v \right\}$$

Zacznijmy nasze rozważania od rekurencyjnej procedury TS-NAÏVE-STEP(proc. 3.1), która obchodzi wszystkie maksymalne kliki grafu. Jej parametrami są klika, która ma być zawarta w znajdowanych klikach C_i oraz parametr pomocniczy będący zbiorem zawierającym rozważone wierzchołki. Procedura wywoływana jest z parametrami będącymi pustymi zbiorami: TS-NAÏVE-STEP$(\emptyset, \emptyset)$.

TS-NAÏVE-STEP(C_i, B_i)

```
 1  Cand ← ext(C_i) \ B_i        ▷ węzły, o które można rozszerzyć klikę C_i
 2  if ∃_{v∈B_i\C_i} ∀_{u∈C_i∪Cand} v ⋍ u
 3      then ▷ Istnieje zakazany element, który należy do wszystkich
 4           ▷ maksymalnych klik zawierających C_i.
 5           return
 6  if Cand ≠ ∅
 7      then p_{i+1} ← pewien element ze zbioru Cand
 8           B_{i+1} ← B_i
 9           for each v ∈ {p_{i+1}} ∪ (contr(p_{i+1}) ∩ Cand)
10                do C_{i+1} ← C_i ∪ {v}
11                   B_{i+1} ← B_{i+1} ∪ {v}
12                   TS-STEP(C_{i+1}, B_{i+1})
13      else  ▷ C_i jest maksymalną kliką.
14           return C_i
```

Procedura 3.1: TS-NAÏVE-STEP

Twierdzenie 7. *Wywołanie procedury* TS-NAÏVE-STEP$(\emptyset, \emptyset)$ *obchodzi wszystkie maksymalne kliki w G dokładnie raz.*

Dowód.

Lemat 3. *Jeżeli warunek w linii 2 procedury* TS-NAÏVE-STEP *jest spełniony, nie istnieje klika zawarta w $C_i \cup (V \setminus B_i)$, która jest maksymalna w G.*

Dowód. Niech $C \subseteq C_i \cup (V \setminus B_i)$ będzie pewną kliką zawierającą C_i. Pokażemy, że jeżeli spełniony jest warunek:

$$\exists_{v \in B_i \setminus C_i} \forall_{u \in C_i \cup (ext(C_i) \setminus B_i)} v \backsim u$$

C nie może być maksymalna. Niech v będzie węzłem G, który spełnia powyższy warunek. Z definicji C wynika, że $v \notin C$. Równocześnie $C \subseteq C_i \cup (ext(C_i) \setminus B_i)$. Zatem $C \cup \{v\}$ jest kliką. □

Lemat 4. *Linia 13 procedury* TS-NAÏVE-STEP *jest wykonywana wyłącznie dla maksymalnych klik w G.*

Dowód. Istotnie tak jest, bowiem gdyby klika C_i nie była maksymalna, zbiór $ext(C_i)$ byłby niepusty. W takiej sytuacji albo zbiór *Cand* musiałby być niepusty i linia 13 nie mogłaby być wykonana, albo $ext(C_i)$ musiałby być zawarty w całości w B_i. Przeanalizujmy ten przypadek. W takiej sytuacji warunek podany w linii 2 zawiera zdanie:

$$\exists_{v \in ext(C_i)} \forall_{u \in C_i} v \backsim u$$

które jest prawdziwe dla niepustego $ext(C_i)$. Zatem na mocy lematu 3 C_i nie może być maksymalna w G. □

Lemat 5. *Procedura* TS-NAÏVE-STEP *obchodzi wszystkie maksymalne kliki w G zawierające C_i i nie zawierające B_i co najwyżej raz.*

Dowód. Zauważmy, że dla pewnego p_{i+1} należącego do *Cand* każda klika zawierająca C_i musi zawierać p_{i+1} lub jeden z elementów $contr(p_{i+1}) \cap ext(C_i)$. Jeżeli $u_1, \ldots, u_k$ są elementami $contr(p_{i+1}) \cap ext(C_i)$, możemy kliki zawierające C_i podzielić na następujące zbiory:

$$\begin{aligned}
D_0 &= \{C \supseteq C_i | p_{i+1} \in C\} \\
D_1 &= \{C \supseteq C_i | u_1 \in C\} \\
D_2 &= \{C \supseteq C_i | u_2 \in C \wedge C \cap \{u_1\} = \emptyset\} \\
D_3 &= \{C \supseteq C_i | u_3 \in C \wedge C \cap \{u_1, u_2\} = \emptyset\} \\
&\ldots \\
D_k &= \{C \supseteq C_i | u_k \in C \wedge C \cap \{u_1, \ldots, u_{k-1}\} = \emptyset\}
\end{aligned}$$

Zbiory te są generowane przez kolejne rekurencyjne wywołania procedury TS-NAÏVE-STEP w linii 12. Powyższe rozumowanie jest krokiem w dowodzie indukcyjnym. Warunek początkowy indukcji wynika z lematu 4 i faktu, że V jest skończony, więc dla każdej kliki G istnieje maksymalna klika ją zawierająca. □

Na mocy lematów 4 i 5 twierdzenie jest prawdziwe. □

Przedstawiony algorytm został nazwany "naiwnym", ponieważ przebiega wszystkie maksymalne kliki w G, podczas gdy nietrudno zauważyć, że dla pewnych miar podobieństwa (np. dla wielkości uliniowienia) można rozpoznać, że kliki zawierające rozważane rozwiązanie nie mogą dawać globalnego maksimum miary. Zdefiniujemy następujący związek pomiędzy miarą podobieństwa, a wielkością uliniowienia. Monotoniczna miara podobieństwa s jest **liniowo ograniczona** ze stałą k, jeżeli:

$$\bigwedge_{\xi \in Al} s(\xi) \leq k|\xi|$$

Zauważmy, że dla danego zbioru uliniowień deskryptorów Φ i pewnego uliniowienia struktur ξ o wsparciu w Φ górne ograniczenie wielkości uliniowienia o wsparciu w Φ zawierającego ξ dane jest nierównością:

$$\xi \subseteq \eta \Subset \Phi \Rightarrow |\eta| \leq |\xi| + \min\left(\left|\bigcup_{\substack{\varphi \in \Phi \\ \varphi \frown \xi}} Dom(\varphi) \setminus Dom(\xi)\right|, \left|\bigcup_{\substack{\varphi \in \Phi \\ \varphi \frown \xi}} im(\varphi) \setminus im(\xi)\right|\right)$$

Niech $extSize\colon Al(S_1, S_2, \Phi) \to \mathbb{N}$ dana wzorem:

$$extSize(\xi) = \min\left(\left|\bigcup_{\substack{\varphi \in \Phi \\ \varphi \frown \xi}} Dom(\varphi) \setminus Dom(\xi)\right|, \left|\bigcup_{\substack{\varphi \in \Phi \\ \varphi \frown \xi}} im(\varphi) \setminus im(\xi)\right|\right)$$

będzie górnym ograniczeniem różnicy między wielkością uliniowienia o wsparciu w Φ zawierającego ξ, a wielkością ξ.

Procedura TS-Step(proc. 3.2) jest rozwinięciem procedury TS-Naïve-Step, która dla ustalonej monotonicznej subaddytywnej ograniczonej liniowo ze stałą k miary podobieństwa s znajduje uliniowienie ją maksymalizujące. Dodatkowy parametr oznacza minimalną akceptowalną wartość miary uliniowienia.

Możliwa jest również modyfikacja procedury TS-Step, aby uliniowienia odpowiadające znajdowanym klikom były spójne. W odróżnieniu od procedur TS-Naïve-Step i TS-Step, które będąc rekurencyjne w sposób niejawny wykorzystują strukturę stosu, w procedurze CTS-Naïve-Step(proc. 3.3) wykorzystaliśmy kolejkę. Ponadto w każdym kroku dodawane uliniowienie musi nakładać się na przynajmniej jedno z uliniowień kliki. Kolejka w odróżnieniu od stosu pozwala, aby kolejność rozważania węzłów w każdej gałęzi drzewa mogła być inna. Dowód poprawności tej procedury jest analogiczny do dowodu twierdzenia 7. Oczywiście rozważana procedura może zostać uzupełniona o weryfikowanie górnego ograniczenia miary maksymalnego uliniowienia.

TS-STEP$(C_i, B_i, minSize)$

```
1  Cand ← ext(C_i) \ B_i        ▷ węzły, o które można rozszerzyć klikę C_i
2  if ∃_{v∈B_i\C_i} ∀_{u∈C_i∪Cand} v ⋬ u
3      then ▷ Istnieje zakazany element, który należy do wszystkich
4           ▷ maksymalnych klik zawierających C_i.
5           return ∅
6  if s(⨆ C_i) + k ( extSize(⨆ C_i)) < minSize
7      then ▷ Nie ma szans na osiągnięcie uliniowienia o s(ξ) ≥ minSize.
8           return ∅
9  if Cand ≠ ∅
10     then p_{i+1} ← pewien element ze zbioru Cand
11          B_{i+1} ← B_i
12          C_max ← ∅
13          for each v ∈ {p_{i+1}} ∪ (contr(p_{i+1}) ∩ Cand)
14               do C_{i+1} ← C_i ∪ {v}
15                  B_{i+1} ← B_{i+1} ∪ {v}
16                  C ← TS-STEP(C_{i+1}, B_{i+1}, max(minSize, s(⨆ C_max)))
17                  if s(C) > s(C_max)
18                     then C_max ← C
19     else  ▷ C_i jest maksymalną kliką.
20          return C_i
```

Procedura 3.2: TS-STEP

```
CTS-Naïve-Step(C_i, B_i, minSize)
 1  Q ← Empty-Queue
 2  Push(Q, ⟨∅, ∅⟩)
 3  while ⟨C, B⟩ ← Pop(Q)
 4        do if ∃_{v∈cont(C)∩(B\C)} ∀_{u∈C∪cont(C)∩(ext(C)\B)} v ∽ u
 5              then ▷ Istnieje w B węzeł, który należy do wszystkich
 6                   ▷ maksymalnych klik zawierających C.
 7              else
 8                   Cand ← (ext(C) ∩ overlap(C)) \ B
 9                   if Cand = ∅
10                     then if s(C') > s(C_max)
11                                then C_max ← C'
12                     else  for v ∈ Cand
13                                do B ← B ∪ {v}
14                                   C' ← C ∪ {v}
15                                   Push(⟨C', B⟩)
```

Procedura 3.3: CTS-Naïve-Step

3.5.2. Algorytm probabilistyczny – REMC

Opisane w poprzednim podrozdziale algorytmy mają złożoność wykładniczą ze względu na liczbę wierzchołków w grafie. Może się zatem zdarzyć, że uliniowienie przy ich pomocy pewnych struktur będzie technicznie niemożliwe lub nieopłacalne. Dlatego zaprezentujemy algorytm znajdowania uliniowień oparty na metodzie Monte-Carlo z wymianą replik.

Szeroko stosowany do próbkowania rozmaitych rozkładów algorytm Metropolisa polega na zbudowaniu łańcucha Markowa, w którym stan x_{t+1} jest konstruowany z poprzedniego stanu x_t przez wylosowanie nowego stanu z pewnego rozkładu zależnego od x_t i jego akceptację z prawdopodobieństwem $\min\left(1, \frac{\pi(x_{t+1})}{\pi(x_t)}\right)$, gdzie π jest pewnym rozkładem prawdopodobieństwa określonym na przestrzeni stanów. Algorytm Metropolisa został pierwotnie opracowany z myślą o symulowaniu układów termodynamicznych w fizyce. Dlatego zazwyczaj przyjmuje się, że rozkład π ma postać:

$$\pi(x) = e^{\frac{-E(x)}{k_B T}}$$

gdzie $E(x)$ jest energią układu w stanie x, T temperaturą, a k_B stałą Boltzmanna. Jeżeli rozkład z którego losowane są kolejne stany, jest dobrany tak, aby rozważany łańcuch Markowa był ergodyczny, rozkład prawdopodobieństwa stanów jest stacjonarny i ma postać:

$$P(x) = \frac{e^{\frac{-E(x)}{k_B T}}}{\sum_{y \in X} e^{\frac{-E(y)}{k_B T}}}$$

gdzie X jest przestrzenią stanów. W tym rozkładzie największe prawdopodobieństwo wystąpienia mają stany o najniższej energii. Zatem algorytm może zostać wykorzystany do rozwiązywania problemów optymalizacyjnych. Temperatura (czynnik $k_B T$) powinna być dobrana tak, aby zapewnić równowagę pomiędzy dążeniem do najbliższego minimum (dla $T = 0$ akceptowane są wyłącznie przejścia do stanu o niższej energii), a losowym błądzeniem nie ograniczonym kryterium energii (dla $T = \infty$ akceptowane jest każde przejście).

Istnieje również schemat stosowania algorytmu Metropolisa, w którym rozważanych jest równolegle kilka łańcuchów Markowa dla różnych temperatur, przy czym co określoną liczbę kroków temperatury stanów są zamieniane z prawdopodobieństwem:

$$\rho(x_i, x_j) = \min\left(1, e^{(E(x_i)-E(x_j))\left(\frac{1}{kT_i}-\frac{1}{kT_j}\right)}\right)$$

Metodę tę nazywa się Monte-Carlo z wymianą replik (ang. *Replica Exchange Monte-Carlo*). Pozwala ona na łatwiejsze unikanie lokalnych minimów, w których replika mo-

głaby utknąć w zbyt niskiej temperaturze i przyspiesza zbieżność do stanu o minimalnej energii.

Aby zastosować metodę REMC do problemu optymalizacyjnego należy określić sposób losowania kolejnych stanów, funkcję energii, która będzie minimalizowana oraz zbiór temperatur. W przypadku uliniawiania struktur funkcją energii jest zanegowana wartość miary podobieństwa dla uliniowienia reprezentowanego przez rozważany stan. Niech C_i będzie kliką w i-tym kroku symulacji. Następna klika jest obliczana następująco:

(1) $v \leftarrow$węzeł wylosowany z rozkładu jednostajnego nad $V \setminus C_i$

(2) $C_{i+1} \leftarrow \{v\} \cup C_i \setminus contr(v)$

Należy zwrócić uwagę na fakt, że algorytm REMC może zostać zastosowany dla dowolnej (w tym niemonotonicznej) miary podobieństwa.

3.5.3. Algorytm przybliżony – MS

W tym podrozdziale zaprezentujemy przybliżony algorytm znajdowania najliczniejszej kliki w G. Można się spodziewać, że najliczniejsza klika odpowiada uliniowieniu o największej mierze podobieństwa, ale oczywiście nie musi to być regułą. W szczególności postulat ten może nie być spełniony dla nietypowych miar podobieństwa. Doświadczenie wskazuje jednak, że zazwyczaj wsparcie optymalnego uliniowienia jest liczniejsze niż innych suboptymalnych, nawet jeżeli ich miary podobieństwa niewiele się różnią. Dlatego będziemy twierdzić, że uliniowienie związane z najliczniejszą kliką w grafie niesprzeczności jest bliskie optymalnemu.

Wektorem charakterystycznym podzbioru $S \subseteq V$ będziemy nazywać następujący wektor $\mathbf{u} \in \mathbb{R}^{|V|}$:

$$u_i = \begin{cases} \frac{1}{|S|} & v_i \in S \\ 0 & \text{w p.p.} \end{cases}$$

Macierzą sąsiedztwa G będziemy nazywać macierz $A \in \mathbb{R}^{|V|\times|V|}$ o elementach:

$$a_{ij} = \begin{cases} 1 & \langle v_i, v_j \rangle \in E \\ 0 & \text{w p.p.} \end{cases}$$

Niech $\Delta^{|V|-1}$ będzie jednostkowym $(|V|-1)$-wymiarowym sympleksem:

$$\Delta^{|V|-1} = \left\{ \left(t_1, \cdots, t_{|V|}\right) \in \mathbb{R}^{|V|} \middle| \sum_{i=1}^{|V|} t_i = 1 \wedge \bigwedge_{i=1}^{|V|} t_i \geq 0 \right\}$$

Twierdzenie 8 (Motzkin-Straus[51]). *Niech $\alpha = f(\mathbf{u}^*)$ będzie globalnym maksimum formy kwadratowej:*

$$f(\mathbf{u}) = \frac{1}{2}\mathbf{u}^T A\mathbf{u}$$

dla $\mathbf{u} \in \Delta^{|V|-1}$. Wówczas klika C o maksymalnej liczności w grafie G o macierzy sąsiedztwa A ma $k = 1/(1-2\alpha)$ węzłów.

Powyższe twierdzenie jest dosyć interesujące, gdyż charakteryzuje związek pomiędzy zasadniczo odmiennymi problemami. Z jednej strony występuje dyskretny problem kombinatoryczny, z drugiej zaś ciągły problem numeryczny. Korzystając z twierdzenia Motzkina-Strausa można w najlepszym przypadku (jeżeli uda się znaleźć globalne maksimum f) obliczyć rozmiar kliki o maksymalnej liczności. Można jednak udowodnić mocniejsze twierdzenie:

Twierdzenie 9. *Podzbiór węzłów C jest kliką o maksymalnej liczności wtedy i tylko wtedy, gdy f osiąga globalne maksimum dla wektora charakterystycznego C.*

Jeżeli zatem znalezione maksimum ma postać wektora charakterystycznego można odczytać z niego zbiór węzłów należących do najliczniejszej kliki. Oczywiście podane twierdzenia według bieżącego stanu wiedzy nie pozwalają rozwiązać problemu NP-zupełnego, jakim jest poszukiwanie najliczniejszej kliki w czasie wielomianowym. Funkcja f może osiągać globalne maksimum również dla wektorów, które nie mają pożądanej postaci i w takiej sytuacji poprawienie takiego wektora może być kosztowne obliczeniowo. Szczegółowe rozważania na temat lokalnych maksimów f i unikaniu rozwiązań niemających właściwej postaci można znaleźć w literaturze[57, 58, 14].

Do znajdowania maksimum rozważanej formy kwadratowej posłużymy się twierdzeniem Bauma-Eagona:

Twierdzenie 10 (Baum-Eagon[8]). *Niech $P(\mathbf{u})$ będzie wielomianem o nieujemnych współczynnikach, zaś $\mathbf{u}$ niech należy do sympleksu Δ^n. Niech funkcja $\mathbf{z} = \mathcal{M}(\mathbf{u})$ będzie określona następująco:*

$$z_i = \frac{u_i \frac{\partial P(\mathbf{u})}{\partial u_i}}{\sum_{j=1}^{n} u_j \frac{\partial P(\mathbf{u})}{\partial u_j}} \quad dla\ i = 1 \ldots n$$

Wtedy $P(\mathcal{M}(\mathbf{u})) > P(\mathbf{u})$, chyba że $\mathcal{M}(\mathbf{u}) = \mathbf{u}$.

Wynika z niego natychmiast, że zdefiniowany rekurencyjnie ciąg $\mathbf{u}^{(t)}$:

$$\mathbf{u}^{(0)} = \left(|V|^{-1}, \ldots, |V|^{-1}\right)$$

$$u_{i+1}^{(t)} = u_i^{(t)} \frac{A\mathbf{u}^{(t)}}{\left[\mathbf{u}^{(t)}\right]^T A\mathbf{u}^{(t)}}$$

jest zbieżny do pewnego wektora $\mathbf{u}^*$, dla którego f osiąga lokalne maksimum i co za tym idzie, który potencjalnie może mieć postać wektora charakterystycznego pewnej maksymalnej kliki w G. W przeprowadzonych eksperymentach numerycznych nigdy nie była to klika inna od najliczniejszej. Jeżeli jest wiele klik o maksymalnej liczności wektor $\mathbf{u}^*$ jest pewną średnią ważoną wektorów charakterystycznych tych klik. Ponieważ w badanych przypadkach różniły się one co najwyżej kilkoma węzłami, zaproponowanie algorytmu znajdującego jedną z nich nie stanowi trudności.

Jedną z zalet tego algorytmu jest łatwość wydajnej implementacji na procesorach graficznych w technologii CUDA lub OpenCL. Na relatywnie mało wydajnym urządzeniu nVidia GT 430 wyposażonym w 96 rdzeni obliczeniowych uzyskaliśmy ponad dziesięciokrotny wzrost wydajności w stosunku do procesora AMD Opteron 2354 (2.2 GHz).

3.6. Wybrane aspekty implementacji

3.6.1. Przestawienia sekwencyjne

Definicja 16. Pozycją przestawienia sekwencyjnego w uliniowieniu struktur ξ nazwiemy parę aminokwasów $\left\langle a^{(l)}, b^{(n)} \right\rangle$ taką, że $\xi(a^{(l)}) = b^{(n)}$ oraz dla $a^{(k)}$ będącego aminokwasem o maksymalnym indeksie w S_1 mniejszym od l i należącym do dziedziny ξ:

$$k = \sup \left\{ i < l \middle| a^{(i)} \in Dom(\xi) \right\}$$

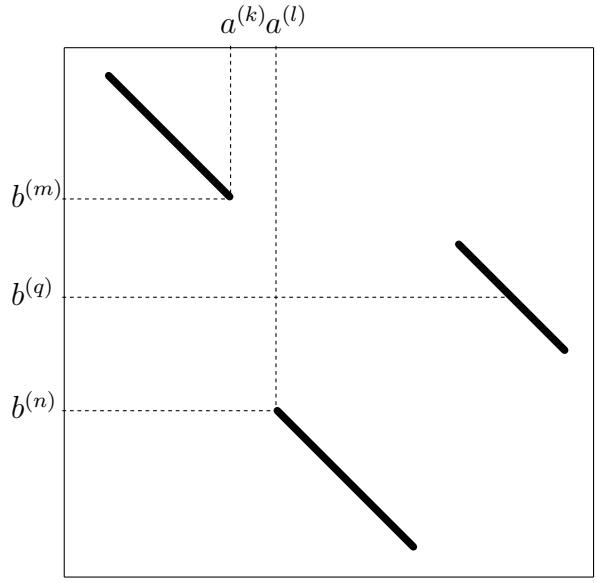
i $b^{(m)} = \xi(a^{(k)})$ istnieje w strukturze S_2 aminokwas $b^{(q)}$ należący do obrazu ξ o indeksie zawartym pomiędzy m i n (rys. 3.2).

Miara podobieństwa deskryptorów może zawierać ograniczenie na liczbę przestawień sekwencyjnych, będąc równą wielkości maksymalnego uliniowienia zawartego w rozważanym o liczbie przestawień sekwencyjnych nie przekraczającej zadanej. Taka miara jest oczywiście monotoniczna oraz liniowo ograniczona i może zostać zastosowana między innymi do wykrywania permutacji cyrkularnych.

3.6.2. Miara lokalnej jakości uliniowienia

Niech C będzie pewnym kryterium kontaktu[3].

[3]Nie musi to być kryterium kontaktu użyte do wygenerowania deskryptorów, których uliniowienia są wsparciem ξ, ale w praktyce tak być powinno.



Rysunek 3.2: Przykładowa pozycja przestawienia sekwencyjnego (oznaczenia jak w definicji 16).

Definicja 17. Uliniowionym kontaktem w uliniowieniu ξ przy kryterium kontaktu C nazwiemy parę $\left\langle \left\langle a^{(k)}, b^{(m)} \right\rangle, \left\langle a^{(l)}, b^{(n)} \right\rangle \right\rangle$ taką, że:

$$b^{(m)} = \xi(a^{(k)}) \wedge b^{(n)} = \xi(a^{(l)})$$

oraz

$$\left\langle a^{(k)}, a^{(l)} \right\rangle \in C \vee \left\langle b^{(m)}, b^{(n)} \right\rangle \in C$$

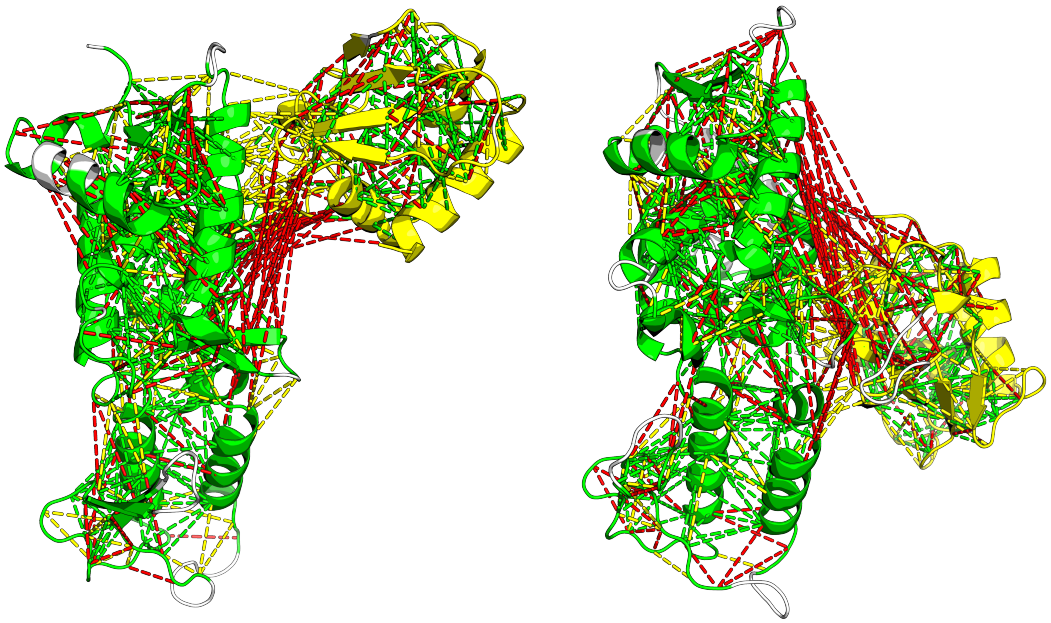
Ponadto uliniowiony kontakt nazwiemy **właściwym**, jeżeli uliniowione aminokwasy są w kontakcie w obydwu strukturach:

$$\left\langle a^{(k)}, a^{(l)} \right\rangle \in C \wedge \left\langle b^{(m)}, b^{(n)} \right\rangle \in C$$

Rysunek 3.3 przedstawia uliniowione kontakty dla pewnej pary struktur. Kolorami wyróżniono kontakty właściwe. **Lokalnym naprężeniem** uliniowionego kontaktu $\left\langle \left\langle a^{(k)}, b^{(m)} \right\rangle, \left\langle a^{(l)}, b^{(n)} \right\rangle \right\rangle$ będzie odległość RMSD pomiędzy elementami deskryptorowymi:

$$tens(\left\langle \left\langle a^{(k)}, b^{(m)} \right\rangle, \left\langle a^{(l)}, b^{(n)} \right\rangle \right\rangle) = RMSD(El(a^{(k)}) \dot{\cup} El(a^{(l)}), El(b^{(m)}) \dot{\cup} El(b^{(n)}))$$

Przez **naprężenie** uliniowienia będziemy rozumieć średnią kwadratową naprężeń jego kontaktów obliczaną dwuetapowo: dla każdego aminokwasu osobno, a następnie



(a) (b)

Rysunek 3.3: Podobne struktury (kody ASTRAL (a) `d1d5fa_` i (b) `d1nd7a_`) składają się z dwóch poddomen, które są różnie względem siebie zorientowane. Kontakty uliniowione właściwe zaznaczono zielonymi przerywanymi liniami. Żółte linie oznaczają kontakty, które nie mają odpowiedników w drugiej strukturze. Czerwonymi liniami połączono aminokwasy, które są w kontakcie w drugiej ze struktur. Aby nałożyć struktury na siebie należałoby rozciągnąć sprężyny związane z żółtymi kontaktami do długości odpowiadających im czerwonych linii.

po wszystkich aminokwasach:

$$tens(\xi) = \sqrt{\sum_{a^{(i)} \in Dom(\xi)} \frac{\sum_{a^{(j)} \in T_{a^{(i)}}} \frac{\left[tens(\left\langle \left\langle a^{(i)}, \xi(a^{(i)}) \right\rangle, \left\langle a^{(j)}, \xi(a^{(j)}) \right\rangle \right\rangle)\right]^2}{|T_{a^{(j)}}|}}{|\,Dom(\xi)|}}$$

gdzie $a^{(j)} \in T_{a^{(i)}}$, jeżeli $\left\langle \left\langle a^{(i)}, \xi(a^{(i)}) \right\rangle, \left\langle a^{(j)}, \xi(a^{(j)}) \right\rangle \right\rangle$ jest uliniowionym kontaktem w ξ. Tak zdefiniowana miara pozwala ocenić geometryczną jakość uliniowienia bez uciekania się do miar globalnych typu RMSD. Wszelkiego rodzaju lokalne odkształcenia skutkują niewielkim średnim naprężeniem. Również podobieństwo dwóch odmiennie zorientowanych względem siebie regionów, o ile nie są one w kontakcie w żadnej ze struktur, nie zwiększa naprężenia. Dlatego ta miara dobrze nadaje się do oceniania podobieństwa, jeżeli chcemy dopuszczać odkształcenia mało zaburzające fizyko-chemiczne oddziaływania aminokwasów. Nazwa "naprężenie" ma przywodzić na myśl analogię do modeli, które oddziaływania fizyko-chemiczne (w naszym przypadku kontakty), reprezentują przy pomocy sprężyn. W takim modelu nałożenie na siebie struktur wymagałoby pewnego odkształcenia sprężyn, tym mniejszego im bardziej są one podobne.

Miara podobieństwa struktur zastosowana w wynikach przedstawionych w następnym podrozdziale określona jest wzorem:

$$s(\xi) = |\xi| - [tens(\xi)]^2$$

Wzór ten został wybrany eksperymentalnie. Daje on najlepszą proporcję pomiędzy niemianowaną liczbą uliniowionych aminokwasów a naprężeniem wyrażonym w Å^2.

3.7. Zastosowania

3.7.1. Implementacja

Zaimplemetowaliśmy opisane algorytmy w języku C na platformie GNU/Linux. Aby przyspieszyć obliczenia i zmaksymalizować szanse uzyskania wyników biologicznie istotnych podzieliliśmy proces uliniawiania struktur na trzy etapy:

(1) Identyfikacja par podobnych deskryptorów i budowa zbioru uliniowień deskryptorowych.

(2) Znajdowanie optymalnych uliniowień o wsparciu w zbiorze uliniowień liczących co najmniej 3 segmenty algorytmem TS lub CTS.

(3) Rozszerzenie uliniowień znalezionych w poprzednim etapie o pozostałe uliniowienia deskryptorowe algorytmem CTS.

Takie podejście gwarantuje, że każda para uliniowionych aminokwasów należy do 3-segmentowego uliniowienia deskryptorowego lub we wsparciu istnieje 3-segmentowe uliniowienie, które jest w domkniętej przechodnio relacji nakładania z uliniowieniem, do którego ta para należy oraz gwarantuje znalezienie dużej liczby kontaktów właściwych. Kombinacje algorytmów TS lub CTS w pierwszym etapie i CTS w drugim, będziemy nazywać TS+CTS i CTS+CTS.

Przeciętny czas pojedynczego porównania pary struktur przy użyciu algorytmów TS i CTS mieści się w przedziale od kilkunastu sekund do kilku minut (na pojedynczym rdzeniu procesora AMD Opteron 2.6GHz) i zależy od liczby rozważanych uliniowień deskryptorowych. W pewnych przypadkach, kiedy struktury zbudowane są w wielu podobnych poddomen (np. *beta-propeller*), czas działania może wydłużyć się do kilku godzin. Wyodrębniliśmy 14 najbardziej złożonych obliczeniowo przypadków i wykorzystaliśmy je do kalibracji algorytmu REMC. Na ich podstawie określiliśmy właściwą liczbę replik, częstotliwość ich zamiany, temperatury oraz liczbę iteracji gwarantującą zbieżność do optymalnego rozwiązania. Czas działania algorytmu REMC zależy głównie od liczby replik i iteracji, zatem każda para struktur może zostać uliniowiona w ciągu kilku minut. W opisanych poniżej eksperymentach algorytm REMC został użyty jako rezerwowy w przypadkach, gdy obliczenie algorytmami dokładnymi zajmowało ponad 120 sekund.

Program i metoda uliniawiania par struktur przy pomocy deskryptorów otrzymały nazwę DEDAL (*DEscriptor Defined ALignment*)[17].

3.7.2. Zbiory testowe

Skuteczność metod porównywania struktury jest często oceniania przez wielkość i RMSD obliczonych uliniowień. Takie podejście jest użyteczne w przypadku metod, które optymalizują te parametry, jednak wielokrotnie może prowadzić do faworyzowania metod, które dopuszczają błędy w uliniowieniu wynikające z przestrzennej bliskości aminokwasów, zamiast kierować się rzeczywistą rolą jaką pełnią rozważane aminokwasy oraz "architekturą" cząsteczki białka. To z kolei może prowadzić do błędnych ocen skuteczności, zwłaszcza w przypadkach, gdy podobieństwo strukturalne jest niewielkie i trudne do wykrycia. Dlatego w naszych rozważaniach wykorzystamy zweryfikowaną przez ekspertów bazę danych zawierającą nietrywialne podobieństwa strukturalne i bę-

dziemy oceniać stopień podobieństwa obliczonych uliniowień do referencyjnych. Posłużymy się w tym celu liczbą będącą stosunkiem liczby par aminokwasów uliniowionych zgodnie z uliniowieniem referencyjnym do rozmiaru uliniowienia referencyjnego.

Testy przeprowadziliśmy na trzech zestawach uliniowień:

(a) baza SISYPHUS[4];

(b) zbiór SISY będący podzbiorem bazy SISYPHUS o konstrukcji opisanej w pracy [49];

(c) zbiór RIPC również zaprezentowany w [49], zawierający trudne uliniowienia wybrane z kompendium ASTRAL[15].

Jedną z zalet zbiorów SISY i RIPC jest możliwość bezpośredniego skonfrontowania wyników z analizą zaprezentowaną w pracy [49].

Baza SISYPHUS zawiera ręcznie opracowane nietrywialne uliniowienia struktur białkowych, które są podzielone na trzy kategorie (fragmenty, sekwencje homologiczne, foldy). Podobieństwa w dwóch ostatnich kategoriach są zazwyczaj wystarczająco duże, aby obejmować znaczącą część uliniawianych struktur i mogą zostać wykorzystane do oceniania metod porównywania. Każde multi-uliniowienie zawiera co najmniej dwie struktury o wyróżnionej części wspólnej. Często się zdarza, że struktury te są niemal identyczne. Dlatego programem LGA[73] zidentyfikowaliśmy pary, w których co najmniej 80% aminokwasów było nakładalne z maksymalną odległością pomiędzy odpowiadającymi sobie aminokwasami nie przekraczającą 2Å i odfiltrowaliśmy je prostym algorytmem zachłannym. Pozostałe 119 multi-uliniowień przypisaliśmy do następujących kategorii:

(a) SCOP – multi-uliniowienia zawierające wyłącznie struktury odpowiadające domenom w bazie SCOP;

(b) MD – multi-uliniowienia zawierające struktury składające się z wielu domen białkowych;

(c) MC – multi-uliniowienia zawierające struktury mające więcej niż jeden łańcuch polipeptydowy.

Ponieważ struktury w bazie PDB zazwyczaj zawierają wiele łańcuchów polipeptydowych upakowanych w pojedynczą komórkę kryształu, w przypadku ostatniej kategorii, aby uniknąć niepożądanej redundancji wszędzie, gdzie to było możliwe, zamiast całej

struktury użyliśmy jednej ze zdefiniowanych w PDB "jednostek biologicznych" (ang. *biological unit*).

Zbiór SISY zawiera 69 nieredundantnych par wybranych z bazy SISYPHUS. Dla każdego multi-uliniowienia wybrano parę struktur o najmniejszym podobieństwie sekwencyjnym. Pary o identyczności sekwencyjnej przekraczającej 40% i wielołańcuchowe struktury pominięto.

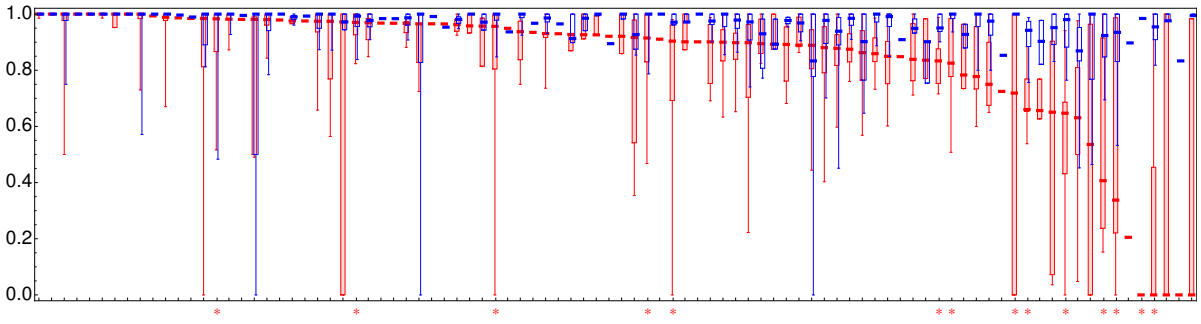
Zbiór RIPC zawiera 40 par domen z bazy ASTRAL. Są one podobne strukturalnie, ale trudne do uliniowienia ze względu na występowanie powtórzeń, rozległych insercji lub delecji, permutacji cyrkularnych oraz odkształceń przestrzennych. Dla 23 par autorzy podają referencyjne uliniowienia wynikające z wiedzy o ewolucyjnej lub funkcjonalnej odpowiedniości aminokwasów.

3.7.3. Rekonstrukcja uliniowień z bazy SISYPHUS

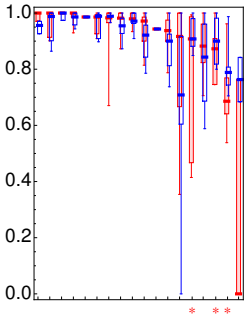
Uliniowiliśmy wszystkie pary struktur z oczyszczonej bazy SISYPHUS przy użyciu algorytmów TS+CTS i CTS+CTS, obliczając dla każdej pary co najwyżej pięć największych znacząco różniących się uliniowień i wybierając spośród nich jedno najbardziej podobne do referencyjnego[4]. Dla struktur zawierających nie więcej niż jeden łańcuch polipeptydowy powtórzyliśmy ten eksperyment programem DaliLite implementującym metodę DALI[31] przy domyślnych ustawieniach. DALI jest uniwersalną metodą porównywania struktur systematycznie zajmującą czołowe miejsca w rankingach. W szczególności została również uznana za najlepszą w pracy [49].

Ostatecznie dla badanych algorytmów i rozważanych par struktur otrzymaliśmy po jednej liczbie odpowiadającej procentowi aminokwasów uliniowionych zgodnie ze wzorcem (rys. 3.4, 3.5). Obydwie metody mają zbliżoną skuteczność w przypadku łatwych podobieństw. Natomiast problematyczne dla DALI podobieństwa (prawa strona wykresów pudełkowych) są dobrze rozpoznawane przez DEDAL. Średnia skuteczność metody DEDAL na uliniowieniach z bazy SISYPHUS wynosi 90% (przy medianie 95%), zaś DALI 90% (przy medianie 97%). Porównując DEDAL i DALI należy zauważyć, że DALI używa mniejszych niż deskryptory fragmentów struktury i w związku z tym rzadko pozostawia pojedyncze nieuliniowione aminokwasy. Uliniowienia obliczone przez DEDAL często mogłyby być bez straty jakości uzupełnione, gdyby nie to, że w zbiorze

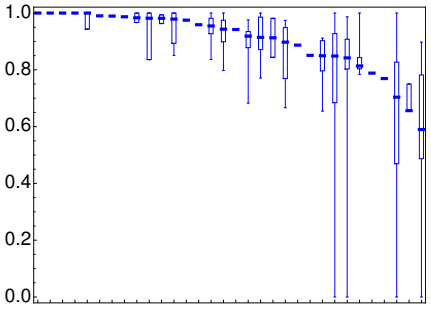
[4]Zabieg z wybieraniem najbardziej podobnego uliniowienia z pięciu jest podyktowany faktem, że uliniowienie referencyjne nie zawsze jest optymalne. Dzieje się tak na przykład, gdy struktura zawiera powtarzający się motyw i autorzy bazy uznali, że uliniowienie alternatywne jest z jakiegoś powodu istotniejsze. Ta cecha bazy SISYPHUS została również odnotowana w pracy [49].



(a)

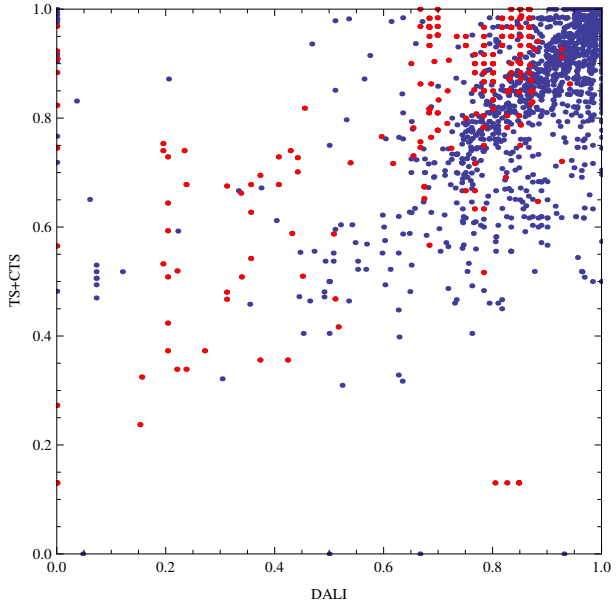


(b)

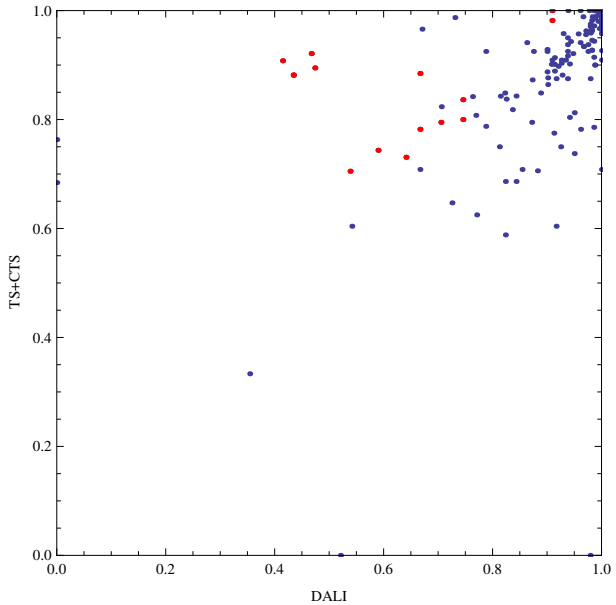


(c)

Rysunek 3.4: Jakość odtwarzania uliniowień z bazy SISYPHUS przez algorytm TS+CTS (niebieski) i DALI (czerwony) dla podzbiorów (a) SCOP, (b) MD i (c) MC. Kolumny na wykresie pudełkowym odpowiadają rozkładowi jakości uliniowień w poszczególnych multi-uliniowieniach w bazie. Wyniki są posortowane malejąco według średniej jakości metody DALI (dla zbiorów SCOP i MD) i TS+CTS (dla zbioru MC). Gwiazdki oznaczają multi-uliniowienia zawierające przestawienia segmentów lub permutacje cyrkularne. Podczas gdy DALI działa nieco lepiej niż TS+CTS dla łatwych przypadków, TS+CTS daje lepsze rezultaty dla przypadków trudnych i uliniowień struktur wielodomenowych.

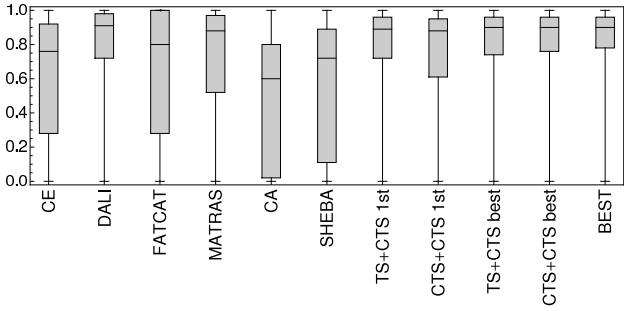


(a)

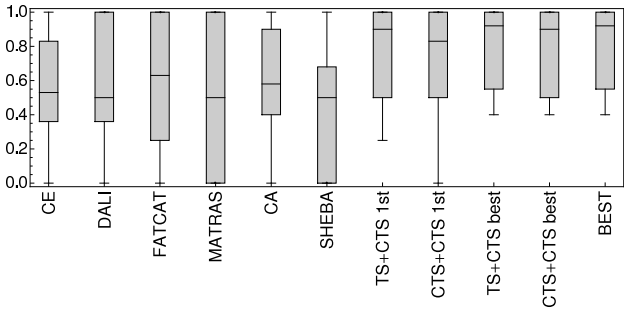


(b)

Rysunek 3.5: Porównanie jakości odtwarzania uliniowień z bazy SISYPHUS ((a) zbiór SCOP; (b) zbiór MD) przez metody DEDAL i DALI. Czerwone punkty oznaczają uliniowienia zawierające zamiany segmentów lub permutacje cyrkularne.



(a)



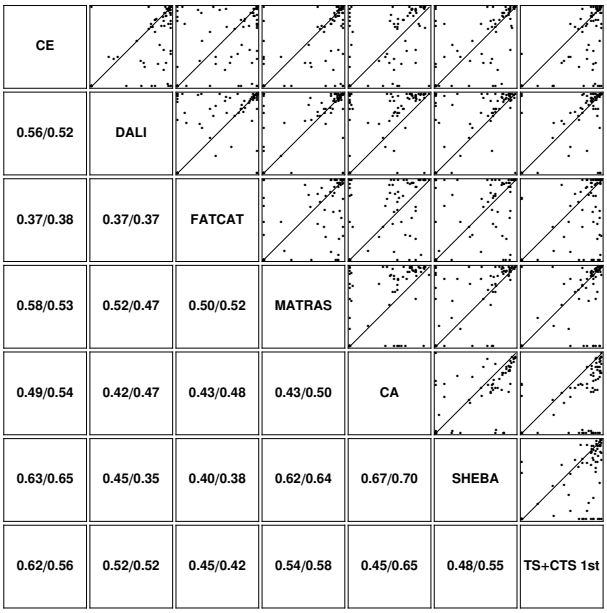
(b)

Rysunek 3.6: Jakość, z jaką odtwarzane są uliniowienia ze zbiorów (a) SISY i (b) RIPC. Wykresy pudełkowe prezentują rozkłady jakości uliniowień odtworzonych przez badane metody. Wyniki algorytmów TS+CTS i CTS+CTS prezentowane są dwojako: najwyższa jakość pięciu obliczonych uliniowień i jakość największego uliniowienia. Kolumna BEST zawiera maksimum z jakości algorytmów TS+CTS i CTS+CTS wzięte z osobna dla każdej badanej pary struktur. Wyniki pozostałych metod pochodzą z pracy [49]. Tabele 3.1 i 3.2 zawierają wyniki analizy statystycznej istotności różnic.

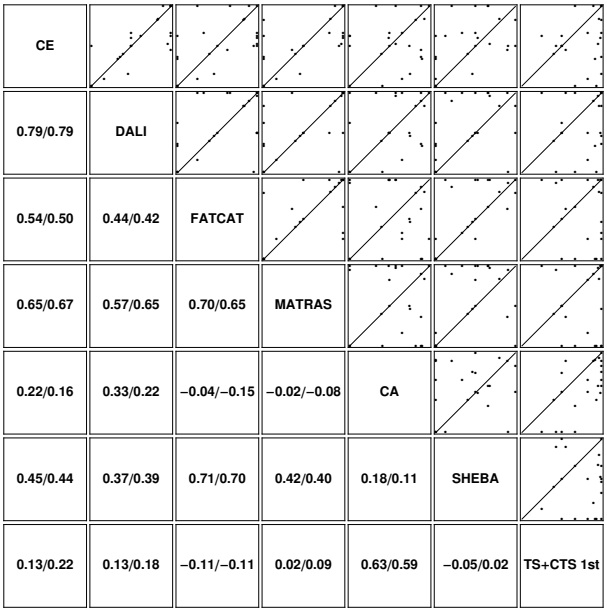
uliniowień deskryptorowych nie istnieją niesprzeczne uliniowienia, które mogłyby do tego posłużyć. Należy również zwrócić uwagę, że DEDAL może uliniawiać struktury wielodomenowe (rys. 3.4c) i wielołańcuchowe (rys. 3.4b), podczas gdy DALI nie pozwala uliniawiać struktur mających więcej niż jeden łańcuch i nie działa właściwie, jeżeli wielodomenowe struktury różnią się wzajemnym położeniem domen.

3.7.4. Rekonstrukcja uliniowień ze zbiorów SISY i RIPC

Zastosowaliśmy metodę opisaną w poprzednim podrozdziale do odtworzenia uliniowień ze zbiorów SISY i RIPC. Porównaliśmy wyniki metody DEDAL z wynikami metod CE, DALI, FATCAT, MATRAS, CA i SHEBA obliczonymi w pracy [49] (rys. 3.6, 3.7). W cytowanej pracy wykorzystywano co najwyżej po jednym uliniowieniu dla każdej



(a)



(b)

Rysunek 3.7: Korelacja pomiędzy jakością odtwarzania uliniowień referencyjnych ze zbiorów SISY (a) i RIPC (b) przez badane metody. W prawym górnym trójkącie umieszczono wykresy punktowe, zaś lewy dolny zawiera współczynniki korelacji Pearsona i Spearmana. Wyniki dla metod innych niż DEDAL pochodzą z pracy [49].

	DALI	FATCAT	MATRAS	CA	SHEBA	TS+CTS
CE	$3.7 \cdot 10^{-5}$	$2.7 \cdot 10^{-1}$	$1.5 \cdot 10^{-2}$	$6.6 \cdot 10^{-2}$	$2.5 \cdot 10^{-1}$	$1.0 \cdot 10^{-4}$
DALI		$1.4 \cdot 10^{-2}$	$1.2 \cdot 10^{-2}$	$2.2 \cdot 10^{-8}$	$9.5 \cdot 10^{-7}$	$1.3 \cdot 10^{-1}$
FATCAT			$3.6 \cdot 10^{-1}$	$9.3 \cdot 10^{-3}$	$1.4 \cdot 10^{-1}$	$3.5 \cdot 10^{-2}$
MATRAS				$5.5 \cdot 10^{-5}$	$6.9 \cdot 10^{-4}$	$4.2 \cdot 10^{-1}$
CA					$9.2 \cdot 10^{-3}$	$4.8 \cdot 10^{-9}$
SHEBA						$2.0 \cdot 10^{-6}$

Tabela 3.1: Wyniki testu istotności Wilcoxona dla jakości porównywanych metod na zbiorze SISY

	DALI	FATCAT	MATRAS	CA	SHEBA	TS+CTS
CE	$1.9 \cdot 10^{-1}$	$3.3 \cdot 10^{-1}$	$3.6 \cdot 10^{-1}$	$4.8 \cdot 10^{-1}$	$8.4 \cdot 10^{-2}$	$3.9 \cdot 10^{-3}$
DALI		$3.7 \cdot 10^{-1}$	$2.9 \cdot 10^{-1}$	$3.4 \cdot 10^{-1}$	$2.2 \cdot 10^{-2}$	$2.9 \cdot 10^{-2}$
FATCAT			$3.5 \cdot 10^{-1}$	$3.4 \cdot 10^{-1}$	$2.1 \cdot 10^{-2}$	$3.3 \cdot 10^{-2}$
MATRAS				$4.8 \cdot 10^{-1}$	$8.4 \cdot 10^{-2}$	$2.9 \cdot 10^{-2}$
CA					$9.8 \cdot 10^{-2}$	$5.9 \cdot 10^{-4}$
SHEBA						$1.2 \cdot 10^{-3}$

Tabela 3.2: Wyniki testu istotności Wilcoxona dla jakości porównywanych metod na zbiorze RIPC

pary struktur i metody. Dlatego prezentujemy również wyniki uwzględniające wyłącznie pierwsze uliniowienie obliczone metodą DEDAL. W szczególności, aby zachować spójność, tych wyników użyliśmy do analizy istotności statystycznej. Wykresy pudełkowe pokazują, że DEDAL jest co najmniej tak samo skuteczny jak DALI i MATRAS (rys. 3.6a). Średnia dokładność uzyskana na zbiorze SISY wynosi 76% (mediana wynosi 89%). Dla porównania DALI osiąga średnią dokładność 75% (mediana 91%), zaś MATRAS – 67% (mediana 88%). Różnica pomiędzy metodami jest większa w przypadku zbioru RIPC (rys. 3.6b), gdzie dolny kwartyl jakości uliniowień obliczonych algorytmem TS+CTS jest porównywalny z medianą innych metod. Średnia dokładność wynosi 77% (mediana 90%) podczas, gdy DALI osiąga średnią jakość 60% (mediana 50%).

Porównaliśmy rozkłady jakości uliniowień dla poszczególnych metod oceniając statystyczną istotność hipotezy, że są różne, dwustronym testem Wilcoxona[69] dla par obserwacji (tab. 3.1, 3.2). Na zbiorze SISY algorytm TS+CTS działa istotnie lepiej niż CE, CA i SHEBA (p-wartości nie przekraczające 1×10^{-4}). Działa również le-

piej niż FATCAT (p-wartość 3.5×10^{-2}) i MATRAS (w tym przypadku różnica nie jest statystycznie istotna) oraz porównywalnie z DALI. Na trudniejszym zbiorze, jakim jest RIPC, DEDAL działa istotnie lepiej niż wszystkie pozostałe testowane metody (wszystkie p-wartości są większe, ponieważ RIPC jest mniej liczny niż SISY).

3.7.5. Wybrane przykłady

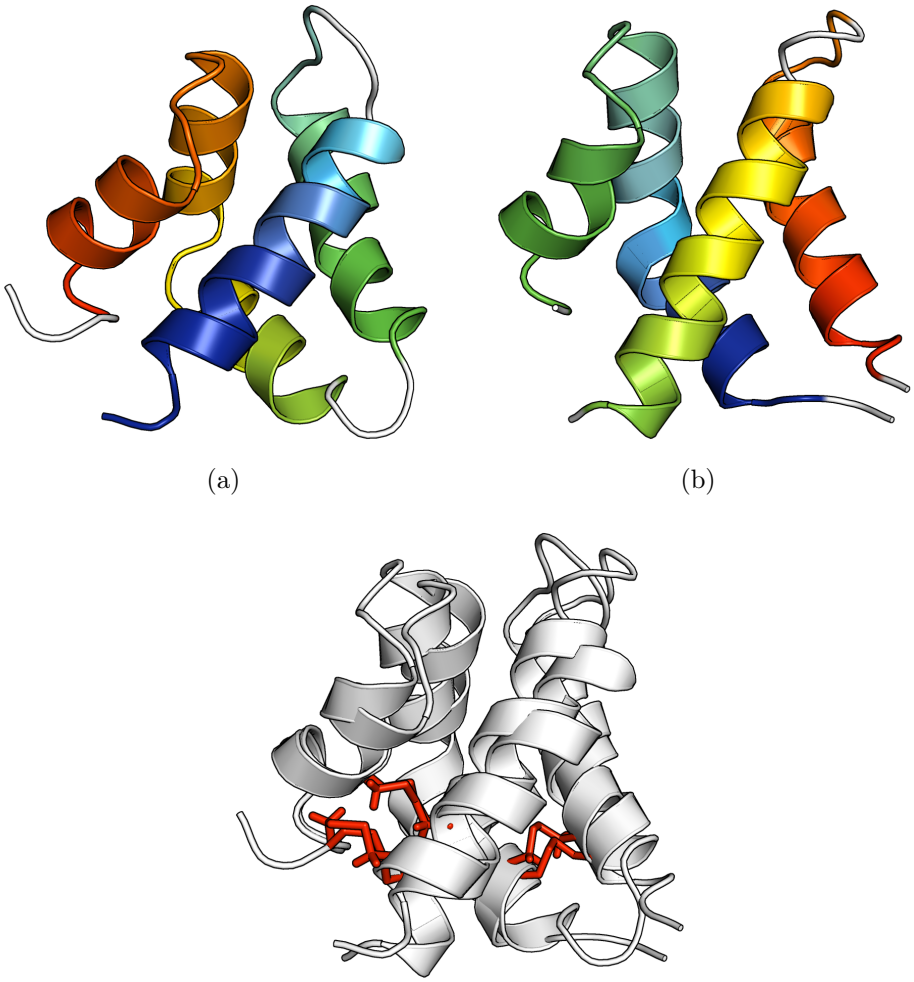
Korzyść ze stosowania deskryptorowej metody porównywania struktur dobrze oddają konkretne, mające znaczenie biologiczne, przykłady podobieństwa struktur. Przedstawimy trzy przykłady uliniowień, które zawierają permutacje cyrkularne lub odkształcenie.

Sapozyny

Jednym z pierwszych odkryć związanych z występowaniem permutacji cyrkularnych w strukturach białek były spermutowane odpowiedniki sapozyny. Hipoteza o ich istnieniu została sformułowana na podstawie analizy sekwencji[59] i zweryfikowana, gdy struktury krystaliczne tych białek stały się dostępne. NK-lizyna (domena ASTRAL `d1nkla_`) jest zbudowana z pięciu helis α ułożonych w kształt "zwiniętego liścia" (ang. *folded leaf*) (rys. 3.8a)[43]. Odpowiadająca jej spermutowana domena (`d1qdma1`) proteazy asparaginianowej (ang. *aspartic proteinase prophytepsin*) ma tę samą architekturę, ale jej helisy występują w innej kolejności (rys. 3.8b)[39]. Nie przeszkadza to większości standardowych metod porównywania struktur uliniawiać helis zgodnie z kolejnością ich ułożenia w sekwencji, co skutkuje niezbyt dobrym nałożeniem struktur. Problematyczne jest również poprawne uliniowienie cystein, które tworzą mostki dwusiarczkowe stabilizujące strukturę (rys. 3.8c).

GTPazy

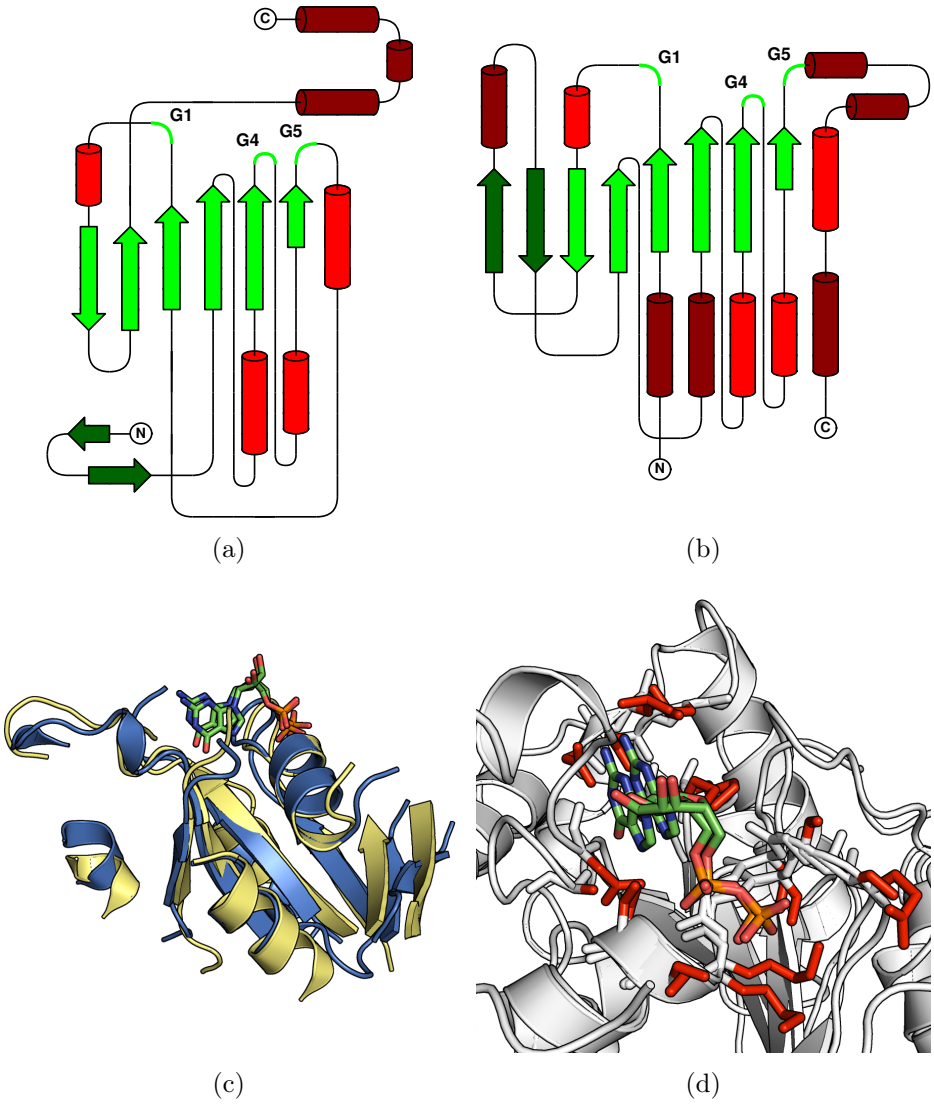
Białka wiążące fosforany guanozyny (białka G) odpowiadają za regulację wielu procesów komórkowych. Można powiedzieć, że działają jak przełączniki binarne, których stan odpowiada przyłączeniu cząsteczki GTP lub GDP. W związku z tym białka G zawierają domenę GTPazy, która odpowiada za wiązanie GTP/GDP. Badania pokazały, że aktywność GTPazy zależy od pięciu konserwowanych motywów sekwencyjnych[54]. Istnieje również alternatywna struktura GTPazy (cpGTPaza), w której występuje permutacja cyrkularna[64], wprawdzie zawierająca wspomniane pięć motywów, lecz występujących w innej kolejności (rys. 3.9a, 3.9b). Mimo, że cpGTPaza ma inną topologię,



(a) (b)

(c)

Rysunek 3.8: (a) Domena sapozynowa NK-lizyny (kod ASTRAL `d1nkla_`) i (b) odpowiadająca jej spermutowana domena proteazy asparaginianowej (`d1qdma1`). Kolor odpowiada położeniu aminokwasu w sekwencji białka (koniec-N – niebieski, koniec-C – czerwony). Pomimo różnej topologii obydwie domeny mają tę samą architekturę i układ mostków dwusiarczkowych. (c) DEDAL poprawnie znajduje optymalną superpozycję i uliniawia mostki dwusiarczkowe (identyczność sekwencyjna porównywanych białek wynosi 14.5%).

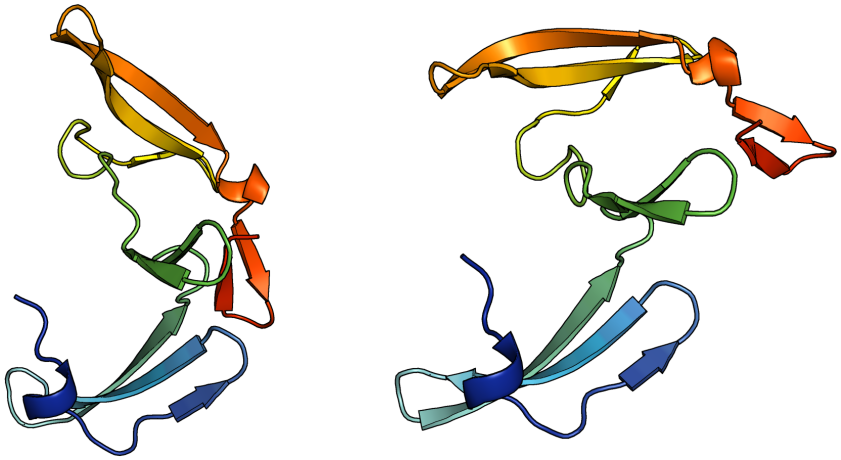


Rysunek 3.9: Topologie domen GTPazy z (a) Dynaminy A (`d1jwyb_`) i (b) białka YjeQ (`d1u0la2`) (spermutowana). Uliniowione elementy struktury drugorzędowej oznaczone są jaśniejszym kolorem. (c) Superpozycja domen GTPazy (żółty) i cpGTPazy (błękitny) obliczone algorytmem TS+CTS (dla przejrzystości pokazano tylko uliniowione fragmenty). (d) Superpozycja miejsc wiążących GTP/GDP (czerwony) z zadokowaną cząsteczką GDP. Pomimo znaczącej różnicy w topologii struktur DEDAL skutecznie uliniawia wszystkie możliwe elementy struktury drugorzędowej i precyzyjnie nakłada miejsca aktywne. Identyczność sekwencyjna uliniowionych fragmentów wynosi 24.2%.

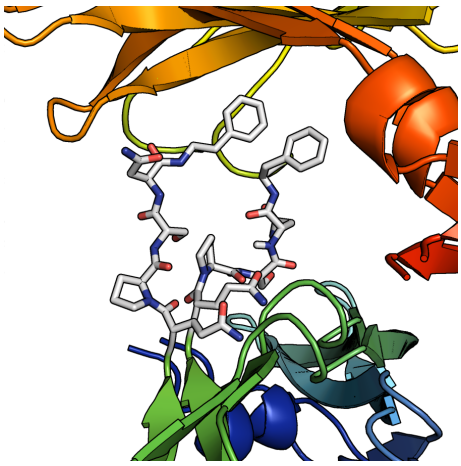
zachowuje architekturę i aktywność GTPazy. Pomimo silnego podobieństwa sekwencyjnego kluczowych motywów[3], większość metod porównywania struktur niepoprawnie uliniawia aminokwasy tworzące miejsce wiążące GTP/GDP. Metoda deskryptorowa oblicza poprawną superpozycję tych regionów (rys. 3.9c, 3.9d).

Cyjanowiryna-N

Cyjanowiryna-N jest białkiem hamującym rozwój wirusa HIV. Występuje ona w formie monomerycznej oraz dimerycznej z zamienionymi domenami. Mimo, że forma monomeryczna dominuje w roztworze i jej struktura przestrzenna została określona najwcześniej[11], metastabilna forma dimeryczna również występuje w postaci rozpuszczonej. Z kolei w stanie krystalicznym stabilniejszy jest dimer[70], aczkolwiek struktura formy występującej w roztworze również została określona techniką NMR[7]. Struktura krystaliczna (`d1l5ba_`) i NMR (`d1l5ea_`) dimeru różnią się wzajemnym położeniem poddomen (rys. 3.10a, 3.10b). Poza rejonem "zawiasu" (PRO51-ASN53, rys. 3.10c) lokalna konformacja pozostałych aminokwasów jest identyczna. Mimo to, metody traktujące strukturę białka w sposób sztywny nie są w stanie poprawnie wykryć tego podobieństwa.



(a) (b)



(c)

Rysunek 3.10: Konformacja formy dimerowej cyjanowiryny-N zależy od środowiska molekularnego. (a) Struktura krystalograficzna (`d1l5ba_`) i (b) NMR (`d1l5ea_`) różnią się konformacją "zawiasu" (PRO51-ASN53) (c). Aby w pełni zaobserwować podobieństwo tych struktur, należy zastosować metodę dopuszczającą odkształcenia. Subdomeny po obu stronach "zawiasu" muszą być nakładane osobno. DEDAL dokonuje uliniowienia przez identyfikację lokalnych podobieństw obydwu regionów i wskazuje na "zawias" jako jedyny nieuliniowiony fragment.

Rozdział 4

Uliniowienia wielu struktur białek

Omówiony w poprzednim rozdziale problem znajdowania uliniowienia pary struktur można uogólnić. W tym rozdziale zdefiniujemy problem znajdowania uliniowień wielu struktur (multi-uliniowień) i omówimy istotne aspekty tego problemu oraz przedstawimy algorytm ewolucyjny, który może być wykorzystany do jego rozwiązania.

Pojęcie *multi-uliniowienia* jest kalką językową z angielskiego *multi-alignment* (*multiple alignments*) i oznacza uliniowienie więcej niż dwóch struktur lub sekwencji. W naszych rozważaniach będziemy rozumieli je szerzej niż jako operację wstawienia spacji w uliniawiane ciągi, aby zmaksymalizować pewną funkcję dopasowania – tak jak w poprzednim rozdziale będziemy dopuszczali przestawienia kolejności.

Dosyć istotnym aspektem jest określenie miary podobieństwa. W przypadku problemu multi-uliniowienia sekwencji stosuje się jedną z trzech strategii:

- suma par (ang. *sum-of-pairs score*, *SP-score*),
- uliniowienie gwiaździste (ang. *star alignment*),
- uliniowienie przy zadanym drzewie filogenetycznym (ang. *tree alignment*).

Każde z tych podejść ma nieco inne właściwości i zastosowania. W szczególności uliniowienie gwiaździste, które polega na znalezieniu sekwencji najbardziej podobnej do uliniawianych, będącej niejako ich uśrednieniem, można w kontekście porównywania struktur rozumieć jako poszukiwanie rdzenia wspólnego dla wszystkich porównywanych struktur, natomiast maksymalizację sumy podobieństwa wszystkich par jako znajdowanie sumy wszystkich podobieństw. Znajdowanie uliniowienia przy zadanym drzewie filogenetycznym jest podejściem pośrednim i ma zastosowanie tylko w sytuacji, gdy na podstawie dodatkowej wiedzy można postawić hipotezę o pokrewieństwie ewolucyj-

nym. Należy pamiętać, że niezależnie od przyjętej strategii problem multi-uliniowienia sekwencji dla większości miar podobieństwa symboli jest NP-trudny [21].

W przypadku porównywania struktur białek wykrycie wspólnego rdzenia, o ile takowy istnieje i jest dobrze określony, jest łatwiejsze od znalezienia wszystkich podobieństw. Mimo tego w dalszych rozważaniach będziemy posługiwać się strategią sumy par, jako dającą pełniejszy obraz podobieństwa struktur. Przyjmiemy, że multi-uliniowienie można opisać zbiorem uliniowień wszystkich par struktur, które do niego należą. Ponieważ nie każdy zbiór uliniowień par opisuje pewne multi-uliniowienie, określimy i będziemy badać warunek konieczny i dostateczny, aby tak było.

4.1. Podstawowe pojęcia

Niech $\mathcal{S} = \{S_1, S_2, \ldots, S_N\}$ będzie pewnym zbiorem struktur, które nie posiadają wspólnych aminokwasów[1]. **Multi-parowaniem zbioru struktur** $\mathcal{S}$ nazwiemy zwrotną i symetryczną relację $Z \subseteq \bigcup S_i^{\mathbb{A}} \times \bigcup S_i^{\mathbb{A}}$, której domknięcie przechodnie Z^+ spełnia następujący warunek:

$$\bigwedge_{A \in \left(\bigcup S_i^{\mathbb{A}}\right)/Z^+} \bigwedge_{1 \leq i \leq N} |A \cap S_i^{\mathbb{A}}| \leq 1$$

Innymi słowy, nie istnieje ciąg aminokwasów, z których każde dwa kolejne są w relacji Z, który ma dwa różne wyrazy należące do jednej ze struktur. Zaś relacja Z^+ po obcięciu do $S_i^{\mathbb{A}} \times S_j^{\mathbb{A}}$ jest parowaniem tych struktur. Będziemy czasem stosować uproszczoną notację zastępując w operacji obcięcia multi-parowania iloczyn kartezjański listą struktur (np. $Z|_{(S_i^{\mathbb{A}} \cup S_j^{\mathbb{A}} \cup S_k^{\mathbb{A}}) \times (S_i^{\mathbb{A}} \cup S_j^{\mathbb{A}} \cup S_k^{\mathbb{A}})} = Z|_{S_i, S_j, S_k}$). Z powyższej definicji wynika, że jeżeli aminokwasy $a^{(p)}, a^{(q)} \in S_i^{\mathbb{A}}$ są w relacji Z^+, to p równa się q. Zatem $Z^+|_{S_i}$ jest pewnym podzbiorem trywialnego parowania. W dalszych rozważaniach będziemy pomijali parowania aminokwasów należących do tej samej struktury.

Definicja 18. Multi-uliniowieniem Z zbioru struktur $\mathcal{S} = \{S_1, S_2, \ldots, S_N\}$ przy wsparciu Φ nazwiemy multi-parowanie, dla którego istnieje zbiór uliniowień $\{\xi_{ij}\}_{i \neq j}$ ($\xi_{ij} \colon S_i^{\mathbb{A}} \to S_j^{\mathbb{A}}$) o wsparciu w Φ, taki że $\xi_{ij} = Z|_{S_i, S_j}$ (funkcja częściowa jest szczególnym przypadkiem relacji). Powiemy, że zbiór $\{\xi_{ij}\}$ **indukuje** multi-uliniowienie Z.

[1] Wymóg rozłączności rozważanych struktur jest czysto techniczny i podyktowany tym, żeby suma $\bigcup S_i^{\mathbb{A}}$ nie prowadziła do "sklejenia" aminokwasów pochodzących z różnych struktur. Łatwo zauważyć, że nie prowadzi on do utraty ogólności rozważań. Gdyby zdarzyło się, że jakieś dwie struktury mają wspólny aminokwas, wystarczy odpowiednio przesunąć w $\mathbb{R}^3$ aminokwasy jednej z nich.

Zauważmy, że dla każdego multi-uliniowienia istnieje dokładnie jeden zbiór uliniowień je indukujący. Powiemy, że zbiór uliniowień jest **zgodny**, jeżeli indukuje multi-uliniowienie. Oczywiście nie każdy zbiór uliniowień jest zgodny (por. rys. 4.2 i 4.3).

Zdefiniowane w 3.1 pojęcia wzajemnego **zawierania** parowań struktur, **pokrywania** uliniowienia przez zbiór uliniowień deskryptorowych i **sumy zbioru uliniowień deskryptorowych** w naturalny sposób uogólniają się na przypadek multi-uliniowień. Średnią wielkość uliniowień indukujących multi-uliniowienie będziemy nazywali **wielkością multi-uliniowienia**:

$$|Z| = \frac{1}{(N-1)N} \sum_{i \neq j} |\xi_{ij}|$$

Niech $s\colon Al \to \mathbb{R}$ będzie pewną miarą podobieństwa. Funkcję $\overline{s}\colon MAl \to \mathbb{R}$ daną wzorem:

$$\overline{s}(Z) = \frac{1}{(N-1)N} \sum_{i \neq j} s(\xi_{ij})$$

nazwiemy **miarą podobieństwa indukowaną** przez s (MAl jest zbiorem wszystkich multi-uliniowień).

4.2. NP-zupełnosć problemu znajdowania maksymalnego multi-uliniowienia

Definicja 19. Dla ustalonego zbioru uliniowień deskryptorów Φ struktur ze zbioru $\mathcal{S} = \{S_1, S_2, \ldots, S_N\}$ **problem najlepszych multi-uliniowień** polega na znalezieniu multi-uliniowienia zbioru $\mathcal{S}$ o wsparciu w Φ i maksymalnej wielkości. Dla ustalonego zbioru uliniowień deskryptorów Φ struktur ze zbioru $\mathcal{S}$ i liczby $r \in \mathbb{R}$ decyzyjny **problem optymalnego multi-uliniowienia struktur** (POMUS) polega na rozstrzygnięciu, czy istnieje multi-uliniowienie zbioru $\mathcal{S}$ przy wsparciu Φ o wielkości nie mniejszej niż r.

Twierdzenie 11. *Problem optymalnego multi-uliniowienia jest NP-zupełny.*

Dowód. Przynależność problemu do klasy NP jest oczywista. Sprawdzenie czy dana relacja jest poprawnym multi-uliniowieniem i obliczenie jej wielkości w czasie wielomianowym jest łatwe. POMUS zawiera NP-zupełny problem POUS (por. def. 9 i tw. 3). □

Powyższa definicja i twierdzenie w żaden sposób nie poszerzają naszej wiedzy na temat złożoności problemu multi-uliniowień. W odróżnieniu od tradycyjnie formułowanych problemów uliniowień sekwencyjnych, gdzie znajdowanie optymalnego uliniowienia dwóch sekwencji ma złożoność wielomianową, podczas gdy znajdowanie multi-uliniowień jest NP-zupełne [21], w rozważanym przypadku uliniowienie pary struktur już jest NP-zupełne. Niemniej jednak, jak pokazaliśmy w poprzednim rozdziale, istnieją wydajne algorytmy heurystyczne pozwalające na znajdowanie uliniowienia dwóch struktur. Uprawnione jest więc pytanie o złożoność problemu POMUS przy założeniu, że umiemy wydajnie (np. przy użyciu wyroczni) rozwiązywać POUS. Udowodnimy, że nawet w sytuacji, gdy liczba elementów Φ odpowiadających każdej parze struktur jest ograniczona stałą (POUS przy tym założeniu jest rozwiązywalny w czasie stałym), POMUS pozostaje NP-zupełny.

Definicja 20. Decyzyjny **ograniczony problem optymalnego multi-uliniowienia struktur** (OPOMUS) ze stałą $k \in \mathbb{N}$ jest szczególnym przypadkiem POMUS, w którym liczność zbiorów Φ_{ij} będących zbiorami uliniowień deskryptorów ze struktur S_i i S_j ($\Phi = \bigcup_{i \neq j} \Phi_{ij}$) jest nie większa niż k.

Twierdzenie 12. *Ograniczony problem optymalnego multi-uliniowienia jest NP-zupełny.*

Dowód. Przynależność OPOMUS do NP wynika z twierdzenia 11. Aby udowodnić NP-zupełność przeprowadzimy redukcję problemu 3SAT do OPOMUS (por. def. 13).

Niech $U = \{u_1, \ldots, u_k\}$, $C = \{C_1, \ldots, C_l\}$ będzie instancją problemu 3SAT. Skonstruujemy trzy zbiory struktur odpowiadające: zmiennym (zbiór V), klauzulom (zbiór K) oraz wartościowaniu zmiennych (zbiór L):

(1) $V = \{V_1, \ldots, V_k\}$, gdzie $V_i = a_1^{V_i} a_2^{V_i} \ldots a_{19}^{V_i} a_{20}^{V_i}$[2]

(2) $K = \{K_1, \ldots, K_l\}$, gdzie $K_i = a_1^{K_i} a_2^{K_i} \ldots a_{14}^{K_i} a_{15}^{K_i}$

(3) $L = \{L_0\}$, gdzie $L_0 = a_1^{L_0} a_2^{L_0} \ldots a_{20}^{L_0} a_{21}^{L_0}$

Dla uproszczenia dalszych rozważań wyróżnimy następujące elementy deskryptorowe:

(1) $v_i = El(a_3^{V_i})$, $v_i^1 = El(a_8^{V_i})$, $v_i^2 = El(a_{13}^{V_i})$, $v_i^3 = El(a_{18}^{V_i})$

(2) $k_{i1} = El(a_3^{K_i})$, $k_{i2} = El(a_8^{K_i})$, $k_{i3} = El(a_{13}^{K_i})$

[2]Ze względów notacyjnych w dowodzie odstąpimy od konwencji podawania pozycji aminokwasu w indeksie górnym.

(3) $t = El(a_3^{L_0})$, $f = El(a_4^{L_0})$, $l^1 = El(a_9^{L_0})$, $l^2 = El(a_{14}^{L_0})$, $l^3 = El(a_{19}^{L_0})$

W podobny sposób określimy cztery zbiory uliniowień deskryptorowych odpowiadających: wartościowaniu zmiennych (zbiory Φ^T i Φ^F), występowaniu zmiennych w klauzulach (zbiór Φ^K) oraz rodzajom literałów występujących w klauzulach (zbiór Φ^L):

(1) $\Phi^T = \left\{\varphi_i^T \colon V_i \to L_0 \middle| 1 \leq i \leq k \wedge \varphi_i^T(v_i) = t \wedge \varphi_i^T(v_i^j) = l^j \text{ dla } j = 1,2,3\right\}$

(2) $\Phi^F = \left\{\varphi_i^F \colon V_i \to L_0 \middle| 1 \leq i \leq k \wedge \varphi_i^F(v_i) = f \wedge \varphi_i^F(v_i^j) = l^j \text{ dla } j = 1,2,3\right\}$

(3) $\Phi^K = \left\{ \varphi_{ij}^K \colon K_i \to V_p \middle| \begin{array}{l} 1 \leq i \leq l \wedge 1 \leq j \leq 3 \wedge 1 \leq p \leq k \wedge \varphi_{ij}^K(k_{ij}) = v_p \wedge \\ \wedge\, u_p \text{ lub } \neg u_p \text{występuje na } j\text{-tym miejscu w klauzuli } C_i \end{array} \right\}$

(4) $\Phi^L = \left\{ \varphi_{ij}^L \colon K_i \to L_0 \middle| \begin{array}{l} 1 \leq i \leq l \wedge 1 \leq j \leq 3 \wedge \\ \\ \wedge\, \varphi_{ij}^L(k_{ij}) = \begin{cases} t & \text{na } j\text{-tym miejscu w klauzuli } C_i \\ & \text{występuje literał pozytywny} \\ f & \text{na } j\text{-tym miejscu w klauzuli } C_i \\ & \text{występuje literał negatywny} \end{cases} \end{array} \right\}$

Ostatecznie zbiór uliniowień deskryptorowych, który będzie wsparciem rozważanych uliniowień, przyjmuje postać $\Phi = (\Phi^T \cup \Phi^F \cup \Phi^K \cup \Phi^L) \cup (\Phi^T \cup \Phi^F \cup \Phi^K \cup \Phi^L)^{-\mathbf{1}}$ (gdzie symbol $\Phi^{-\mathbf{1}}$ oznacza zbiór uliniowień deskryptorowych odwrotnych do Φ). Pokażemy, że wartościowanie zmiennych, przy którym klauzule są spełnione, istnieje wtedy i tylko wtedy, gdy istnieje uliniowienie zbioru $\{L_0, V_1, \ldots, V_k, K_0, \ldots, K_l\}$ przy wsparciu zawartym w Φ o wielkości $\frac{2(20k+10l)}{(k+l)(k+l+1)}$. Ponadto, jeżeli Φ' jest wsparciem takiego uliniowienia, to dla każdego i $\varphi_i^T \in \Phi'$ albo $\varphi_i^F \in \Phi'$, a wartościowanie:

$$u_i \to \begin{cases} 1 & \varphi_i^T \in \Phi' \\ 0 & \varphi_i^F \in \Phi' \end{cases}$$

jest rozwiązaniem problemu 3SAT.

Przykład 3. Niech C będzie zbiorem klauzul nad zbiorem zmiennych $U = \{u_1, u_2, u_3\}$:

$$C = \{\{\neg u_1, u_2, u_3\}, \{u_1, \neg u_2, u_3\}, \{u_1, u_2, \neg u_3\}\}$$

odpowiadającym formule logicznej:

$$(\neg u_1 \vee u_2 \vee u_3) \wedge (u_1 \vee \neg u_2 \vee u_3) \wedge (u_1 \vee u_2 \vee \neg u_3)$$

Instancji problemu 3SAT dla powyższej formuły odpowiada następująca instancja OPOMUS:

$$L_0 = \overbrace{a_1^{L_0}\underbrace{a_2^{L_0}a_3^{L_0}a_4^{L_0}a_5^{L_0}}_{f}a_6^{L_0}}^{t}\overbrace{a_7^{L_0}a_8^{L_0}a_9^{L_0}a_{10}^{L_0}a_{11}^{L_0}}^{l^1}\overbrace{a_{12}^{L_0}a_{13}^{L_0}a_{14}^{L_0}a_{15}^{L_0}a_{16}^{L_0}}^{l^2}\overbrace{a_{17}^{L_0}a_{18}^{L_0}a_{19}^{L_0}a_{20}^{L_0}a_{21}^{L_0}}^{l^3}$$

$$V_1 = \overbrace{a_1^{V_1}a_2^{V_1}a_3^{V_1}a_4^{V_1}a_5^{V_1}}^{v_1}\overbrace{a_6^{V_1}a_7^{V_1}a_8^{V_1}a_9^{V_1}a_{10}^{V_1}}^{v_1^1}\overbrace{a_{11}^{V_1}a_{12}^{V_1}a_{13}^{V_1}a_{14}^{V_1}a_{15}^{V_1}}^{v_1^2}\overbrace{a_{16}^{V_1}a_{17}^{V_1}a_{18}^{V_1}a_{19}^{V_1}a_{20}^{V_1}}^{v_1^3}$$

$$V_2 = \overbrace{a_1^{V_2}a_2^{V_2}a_3^{V_2}a_4^{V_2}a_5^{V_2}}^{v_2}\overbrace{a_6^{V_2}a_7^{V_2}a_8^{V_2}a_9^{V_2}a_{10}^{V_2}}^{v_2^1}\overbrace{a_{11}^{V_2}a_{12}^{V_2}a_{13}^{V_2}a_{14}^{V_2}a_{15}^{V_2}}^{v_2^2}\overbrace{a_{16}^{V_2}a_{17}^{V_2}a_{18}^{V_2}a_{19}^{V_2}a_{20}^{V_2}}^{v_2^3}$$

$$V_3 = \overbrace{a_1^{V_3}a_2^{V_3}a_3^{V_3}a_4^{V_3}a_5^{V_3}}^{v_3}\overbrace{a_6^{V_3}a_7^{V_3}a_8^{V_3}a_9^{V_3}a_{10}^{V_3}}^{v_3^1}\overbrace{a_{11}^{V_3}a_{12}^{V_3}a_{13}^{V_3}a_{14}^{V_3}a_{15}^{V_3}}^{v_3^2}\overbrace{a_{16}^{V_3}a_{17}^{V_3}a_{18}^{V_3}a_{19}^{V_3}a_{20}^{V_3}}^{v_3^3}$$

$$K_1 = \overbrace{a_1^{K_1}a_2^{K_1}a_3^{K_1}a_4^{K_1}a_5^{K_1}}^{k_{1,1}}\overbrace{a_6^{K_1}a_7^{K_1}a_8^{K_1}a_9^{K_1}a_{10}^{K_1}}^{k_{1,2}}\overbrace{a_{11}^{K_1}a_{12}^{K_1}a_{13}^{K_1}a_{14}^{K_1}a_{15}^{K_1}}^{k_{1,3}}$$

$$K_2 = \overbrace{a_1^{K_2}a_2^{K_2}a_3^{K_2}a_4^{K_2}a_5^{K_2}}^{k_{2,1}}\overbrace{a_6^{K_2}a_7^{K_2}a_8^{K_2}a_9^{K_2}a_{10}^{K_2}}^{k_{2,2}}\overbrace{a_{11}^{K_2}a_{12}^{K_2}a_{13}^{K_2}a_{14}^{K_2}a_{15}^{K_2}}^{k_{2,3}}$$

$$K_3 = \overbrace{a_1^{K_3}a_2^{K_3}a_3^{K_3}a_4^{K_3}a_5^{K_3}}^{k_{3,1}}\overbrace{a_6^{K_3}a_7^{K_3}a_8^{K_3}a_9^{K_3}a_{10}^{K_3}}^{k_{3,2}}\overbrace{a_{11}^{K_3}a_{12}^{K_3}a_{13}^{K_3}a_{14}^{K_3}a_{15}^{K_3}}^{k_{3,3}}$$

$$\varphi_1^T(e) = \begin{cases} t & e = v_1 \\ l^i & e = v_1^i \end{cases} \qquad \varphi_2^T(e) = \begin{cases} t & e = v_2 \\ l^i & e = v_2^i \end{cases} \qquad \varphi_3^T(e) = \begin{cases} t & e = v_3 \\ l^i & e = v_3^i \end{cases}$$

$$\varphi_1^F(e) = \begin{cases} f & e = v_1 \\ l^i & e = v_1^i \end{cases} \qquad \varphi_2^F(e) = \begin{cases} f & e = v_2 \\ l^i & e = v_2^i \end{cases} \qquad \varphi_3^F(e) = \begin{cases} f & e = v_3 \\ l^i & e = v_3^i \end{cases}$$

$$\varphi_{1,1}^K(e) = \begin{cases} v_1 & e = k_{1,1} \end{cases} \qquad \varphi_{1,2}^K(e) = \begin{cases} v_2 & e = k_{1,2} \end{cases} \qquad \varphi_{1,3}^K(e) = \begin{cases} v_3 & e = k_{1,3} \end{cases}$$

$$\varphi_{2,1}^K(e) = \begin{cases} v_1 & e = k_{2,1} \end{cases} \qquad \varphi_{2,2}^K(e) = \begin{cases} v_2 & e = k_{2,2} \end{cases} \qquad \varphi_{2,3}^K(e) = \begin{cases} v_3 & e = k_{2,3} \end{cases}$$

$$\varphi_{3,1}^K(e) = \begin{cases} v_1 & e = k_{3,1} \end{cases} \qquad \varphi_{3,2}^K(e) = \begin{cases} v_2 & e = k_{3,2} \end{cases} \qquad \varphi_{3,3}^K(e) = \begin{cases} v_3 & e = k_{3,3} \end{cases}$$

$$\varphi_{1,1}^L(e) = \begin{cases} f & e = k_{1,1} \end{cases} \qquad \varphi_{1,2}^L(e) = \begin{cases} t & e = k_{1,2} \end{cases} \qquad \varphi_{1,3}^L(e) = \begin{cases} t & e = k_{1,3} \end{cases}$$

$$\varphi_{2,1}^L(e) = \begin{cases} t & e = k_{2,1} \end{cases} \qquad \varphi_{2,2}^L(e) = \begin{cases} f & e = k_{2,2} \end{cases} \qquad \varphi_{2,3}^L(e) = \begin{cases} t & e = k_{2,3} \end{cases}$$

$$\varphi_{3,1}^L(e) = \begin{cases} t & e = k_{3,1} \end{cases} \qquad \varphi_{3,2}^L(e) = \begin{cases} t & e = k_{3,2} \end{cases} \qquad \varphi_{3,3}^L(e) = \begin{cases} f & e = k_{3,3} \end{cases}$$

Rozważana formuła jest spełniona między innymi dla wartościowania:

$$u_1 \to 0, u_2 \to 1, u_3 \to 1$$

Zatem istnieje multi-uliniowienie o wsparciu zawierającym $\left\{\varphi_1^F, \varphi_2^T, \varphi_3^T\right\}$ i wielkości $\frac{180}{21}$. Takim wsparciem jest $\left\{\varphi_1^F, \varphi_2^T, \varphi_3^T, \varphi_{1,1}^K, \varphi_{2,3}^K, \varphi_{3,2}^K, \varphi_{1,1}^L, \varphi_{2,3}^L, \varphi_{3,2}^L\right\}$, a uliniowienie ma postać (por. rys. 4.1) ($\varphi(a) = \perp$ oznacza, że $a \notin Dom(\varphi)$):

$$\vec{\xi}_{V_1L_0}(V_1) = \overbrace{\underbrace{a_2^{L_0}a_3^{L_0}a_4^{L_0}a_5^{L_0}a_6^{L_0}}_{f}}^{v_1}\overbrace{a_7^{L_0}a_8^{L_0}a_9^{L_0}a_{10}^{L_0}a_{11}^{L_0}}^{v_1^1}\overbrace{a_{12}^{L_0}a_{13}^{L_0}a_{14}^{L_0}a_{15}^{L_0}a_{16}^{L_0}}^{v_1^2}\overbrace{a_{17}^{L_0}a_{18}^{L_0}a_{19}^{L_0}a_{20}^{L_0}a_{21}^{L_0}}^{v_1^3}$$

$$\vec{\xi}_{V_2L_0}(V_2) = \overbrace{\underbrace{a_1^{L_0}a_2^{L_0}a_3^{L_0}a_4^{L_0}a_5^{L_0}}_{t}}^{v_2}\overbrace{a_7^{L_0}a_8^{L_0}a_9^{L_0}a_{10}^{L_0}a_{11}^{L_0}}^{v_2^1}\overbrace{a_{12}^{L_0}a_{13}^{L_0}a_{14}^{L_0}a_{15}^{L_0}a_{16}^{L_0}}^{v_2^2}\overbrace{a_{17}^{L_0}a_{18}^{L_0}a_{19}^{L_0}a_{20}^{L_0}a_{21}^{L_0}}^{v_2^3}$$

$$\vec{\xi}_{V_3L_0}(V_3) = \overbrace{\underbrace{a_1^{L_0}a_2^{L_0}a_3^{L_0}a_4^{L_0}a_5^{L_0}}_{t}}^{v_3}\overbrace{a_7^{L_0}a_8^{L_0}a_9^{L_0}a_{10}^{L_0}a_{11}^{L_0}}^{v_3^1}\overbrace{a_{12}^{L_0}a_{13}^{L_0}a_{14}^{L_0}a_{15}^{L_0}a_{16}^{L_0}}^{v_3^2}\overbrace{a_{17}^{L_0}a_{18}^{L_0}a_{19}^{L_0}a_{20}^{L_0}a_{21}^{L_0}}^{v_3^3}$$

$$\vec{\xi}_{K_1V_1}(K_1) = \overbrace{\underbrace{a_1^{V_1}a_2^{V_1}a_3^{V_1}a_4^{V_1}a_5^{V_1}}_{v_1}}^{k_{1,1}}\overbrace{\perp\perp\perp\perp\perp}^{k_{1,2}}\overbrace{\perp\perp\perp\perp\perp}^{k_{1,3}}$$

$$\vec{\xi}_{K_2V_3}(K_2) = \overbrace{\perp\perp\perp\perp\perp}^{k_{2,1}}\overbrace{\perp\perp\perp\perp\perp}^{k_{2,2}}\overbrace{\underbrace{a_1^{V_3}a_2^{V_3}a_3^{V_3}a_4^{V_3}a_5^{V_3}}_{v_3}}^{k_{2,3}}$$

$$\vec{\xi}_{K_3V_2}(K_3) = \overbrace{\perp\perp\perp\perp\perp}^{k_{3,1}}\overbrace{\underbrace{a_1^{V_2}a_2^{V_2}a_3^{V_2}a_4^{V_2}a_5^{V_2}}_{v_2}}^{k_{3,2}}\overbrace{\perp\perp\perp\perp\perp}^{k_{3,3}}$$

$$\vec{\xi}_{K_1L_0}(K_1) = \overbrace{\underbrace{a_2^{L_0}a_3^{L_0}a_4^{L_0}a_5^{L_0}a_6^{L_0}}_{f}}^{k_{1,1}}\overbrace{\perp\perp\perp\perp\perp}^{k_{1,2}}\overbrace{\perp\perp\perp\perp\perp}^{k_{1,3}}$$

$$\vec{\xi}_{K_2L_0}(K_2) = \overbrace{\perp\perp\perp\perp\perp}^{k_{2,1}}\overbrace{\perp\perp\perp\perp\perp}^{k_{2,2}}\overbrace{\underbrace{a_1^{L_0}a_2^{L_0}a_3^{L_0}a_4^{L_0}a_5^{L_0}}_{t}}^{k_{2,3}}$$

$$\vec{\xi}_{K_3L_0}(K_3) = \overbrace{\perp\perp\perp\perp\perp}^{k_{3,1}}\overbrace{\underbrace{a_1^{L_0}a_2^{L_0}a_3^{L_0}a_4^{L_0}a_5^{L_0}}_{t}}^{k_{3,2}}\overbrace{\perp\perp\perp\perp\perp}^{k_{3,3}}$$

Zaczniemy od pokazania, że jeżeli w skonstruowanej w powyższy sposób instancji OPOMUS istnieje multi-uliniowienie wielkości $\frac{2(20k+10l)}{(k+l)(k+l+1)}$, to fomuła z problemu 3SAT jest spełnialna.

Lemat 6. *Wsparcie maksymalnych (pod względem wielkości) multi-uliniowień w powyższej konstrukcji zawiera φ_i^T lub φ_i^F dla wszystkich $1 \leq i \leq k$.*

$$
\begin{array}{ll}
L_0 & \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ a_1^{L_0}\, a_2^{L_0}\, a_3^{L_0}\, a_4^{L_0}\, a_5^{L_0}\, a_6^{L_0} \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \ldots \\
V_1 & \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ a_1^{V_1}\, a_2^{V_1}\, a_3^{V_1}\, a_4^{V_1}\, a_5^{V_1} \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \ldots \\
V_2 & \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ a_1^{V_2}\, a_2^{V_2}\, a_3^{V_2}\, a_4^{V_2}\, a_5^{V_2} \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \ldots \\
V_3 & \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ a_1^{V_3}\, a_2^{V_3}\, a_3^{V_3}\, a_4^{V_3}\, a_5^{V_3} \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \ldots \\
K_1 & \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ a_1^{K_1} a_2^{K_1} a_3^{K_1} a_4^{K_1} a_5^{K_1} a_6^{K_1} a_7^{K_1} a_8^{K_1} a_9^{K_1} a_{10}^{K_1} a_{11}^{K_1} a_{12}^{K_1} a_{13}^{K_1} a_{14}^{K_1} a_{15}^{K_1} \ldots \\
K_2 & a_1^{K_2} a_2^{K_2} a_3^{K_2} a_4^{K_2} a_5^{K_2} a_6^{K_2} a_7^{K_2} a_8^{K_2} a_9^{K_2} a_{10}^{K_2} \bot\ \bot\ \bot\ \bot\ \bot\ a_{11}^{K_2} a_{12}^{K_2} a_{13}^{K_2} a_{14}^{K_2} a_{15}^{K_2} \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \ldots \\
K_3 & \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ a_1^{K_3} a_2^{K_3} a_3^{K_3} a_4^{K_3} a_5^{K_3} a_6^{K_3} a_7^{K_3} a_8^{K_3} a_9^{K_3} a_{10}^{K_3} \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \ldots
\end{array}
$$

$$
\begin{array}{ll}
L_0 & \ldots \bot\ \bot\ \bot\ \bot\ \bot\ a_7^{L_0}\, a_8^{L_0}\, a_9^{L_0}\, a_{10}^{L_0}\, a_{11}^{L_0}\, a_{12}^{L_0}\, a_{13}^{L_0}\, a_{14}^{L_0}\, a_{15}^{L_0}\, a_{16}^{L_0}\, a_{17}^{L_0}\, a_{18}^{L_0}\, a_{19}^{L_0}\, a_{20}^{L_0}\, a_{21}^{L_0} \\
V_1 & \ldots \bot\ \bot\ \bot\ \bot\ \bot\ a_6^{V_1}\, a_7^{V_1}\, a_8^{V_1}\, a_9^{V_1}\, a_{10}^{V_1}\, a_{11}^{V_1}\, a_{12}^{V_1}\, a_{13}^{V_1}\, a_{14}^{V_1}\, a_{15}^{V_1}\, a_{16}^{V_1}\, a_{17}^{V_1}\, a_{18}^{V_1}\, a_{19}^{V_1}\, a_{20}^{V_1} \\
V_2 & \ldots \bot\ \bot\ \bot\ \bot\ \bot\ a_6^{V_2}\, a_7^{V_2}\, a_8^{V_2}\, a_9^{V_2}\, a_{10}^{V_2}\, a_{11}^{V_2}\, a_{12}^{V_2}\, a_{13}^{V_2}\, a_{14}^{V_2}\, a_{15}^{V_2}\, a_{16}^{V_2}\, a_{17}^{V_2}\, a_{18}^{V_2}\, a_{19}^{V_2}\, a_{20}^{V_2} \\
V_3 & \ldots \bot\ \bot\ \bot\ \bot\ \bot\ a_6^{V_3}\, a_7^{V_3}\, a_8^{V_3}\, a_9^{V_3}\, a_{10}^{V_3}\, a_{11}^{V_3}\, a_{12}^{V_3}\, a_{13}^{V_3}\, a_{14}^{V_3}\, a_{15}^{V_3}\, a_{16}^{V_3}\, a_{17}^{V_3}\, a_{18}^{V_3}\, a_{19}^{V_3}\, a_{20}^{V_3} \\
K_1 & \ldots \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot \\
K_2 & \ldots \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot \\
K_3 & \ldots\ a_{11}^{K_3} a_{12}^{K_3} a_{13}^{K_3} a_{14}^{K_3} a_{15}^{K_3} \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot\ \bot
\end{array}
$$

Rysunek 4.1: Multi-uliniowienie struktur odpowiadające rozwiązaniu problemu 3SAT z przykładu 3

Dowód. Załóżmy, że tak nie jest. Istnieje zatem maksymalne multi-uliniowienie o wsparciu Φ' oraz i, takie że $\varphi_i^T \notin \Phi'$ i $\varphi_i^F \notin \Phi'$. Niech C_p, C_q i C_r będą klauzulami, w których występuje zmienna u_i. Tym wystąpieniom odpowiadają uliniowienia deskryptorowe φ_{pf}^K, φ_{qg}^K, φ_{rh}^K (gdzie f, g, h są pozycjami, na których u_i występuje). Zauważmy, że łączny wkład φ_{pf}^K, φ_{qg}^K, φ_{rh}^K do wielkości multi-uliniowienia wynosi $\frac{30}{(k+l)(k+l+1)}$, podczas gdy wkład φ_i^T lub φ_i^F lub wynosi $\frac{40}{(k+l)(k+l+1)}$. Zatem zmodyfikowane multi-uliniowienie, w którego wsparciu φ_{pf}^K, φ_{qg}^K, φ_{rh}^K zostaną zastąpione przez φ_i^T lub φ_i^F jest większe, co kończy dowód lematu. □

Lemat 7. *Wsparcie maksymalnych (pod względem wielkości) multi-uliniowień w powyższej konstrukcji zawiera dla wszystkich* $1 \leq i \leq l$ *co najwyżej jedną z trzech par uliniowień* $\{\varphi_{i1}^K, \varphi_{i1}^L\}$, $\{\varphi_{i2}^K, \varphi_{i2}^L\}$, $\{\varphi_{i3}^K, \varphi_{i3}^L\}$.

Dowód. Niech u_p, u_q i u_r będą zmiennymi występującymi w klauzuli C_i. Na mocy poprzedniego lematu wiemy, że wsparcie multi-uliniowienia zawiera uliniowienie struktur V_p, V_q i V_r z L_0. Zatem φ_{ij}^K przechodnio uliniawia K_i z L_0. Z definicji multi-uliniowienia wynika, że jeżeli φ_{ij}^K lub φ_{ij}^L należy do wsparcia, nie mogą do niego należeć $\varphi_{ij'}^K$ i $\varphi_{ij'}^L$ dla $j' \neq j$, ponieważ w przeciwnym wypadku każdy z aminokwasów $a_2^{L_0}, \ldots, a_4^{L_0}$ byłby uliniowiony z dwoma aminokwasami z K_i. □

Z powyższych lematów wynika, że do wsparcia maksymalnego multi-uliniowienia należą:

- dla każdej zmiennej jedno z uliniowień φ_i^T lub φ_i^F o wkładzie do wielkości multi-uliniowienia $\frac{40}{(k+l)(k+l+1)}$;
- dla każdej klauzuli co najwyżej jedna z par $\{\varphi_{i1}^K, \varphi_{i1}^L\}$, $\{\varphi_{i2}^K, \varphi_{i2}^L\}$, $\{\varphi_{i3}^K, \varphi_{i3}^L\}$ o wkładzie do wielkości multi-uliniowienia $\frac{20}{(k+l)(k+l+1)}$.

Zatem maksymalne uliniowienie ma wielkość co najwyżej $\frac{2(20k+10l)}{(k+l)(k+l+1)}$. Wnioskujemy stąd, że jeżeli istnieje multi-uliniowienie o tej wielkości, musi ono zawierać wymienione uliniowienia deskryptorowe. Łatwo zauważyć, że w takiej sytuacji dla wartościowania:

$$u_i \to \begin{cases} 1 & \varphi_i^T \in \Phi' \\ 0 & \varphi_i^F \in \Phi' \end{cases}$$

wszystkie klauzule są spełnione. Dla każdej klauzuli istnieją bowiem uliniowienia φ_{ij}^K, φ_{ij}^L należące do Φ'. Ponadto, jeżeli u_p jest zmienną występującą na j-tym miejscu w klauzuli C_i i $\varphi_p^T \in \Phi'$, obrazem elementu deskryptorowego k_{ij} w φ_{ij}^L musi być t (z

definicji multi-uliniowienia). Zatem literał związany ze zmienną u_i w klauzuli C_i jest pozytywny, a klauzula spełniona. Analogiczne rozumowanie można przeprowadzić dla przypadku przeciwnego. Zatem istnienie rozwiązania problemu OPOMUS implikuje istnienie rozwiązania 3SAT. Opisane uliniowienie da się zawsze skonstruować, jeżeli formuła jest spełnialna, co dowodzi implikacji odwrotnej. Redukcja ma złożoność wielomianową, co kończy dowód. □

4.3. Multi-uliniowienie pary multi-uliniowień

Udowodniliśmy, że nie istnieje algorytm znajdujący optymalne multi-uliniowienie o złożoności obliczeniowej lepszej niż wykładnicza ze względu na liczbę struktur (o ile $P \neq NP$). Zatem, skoro znajdowanie optymalnego multi-uliniowienia w drodze pojedynczej optymalizacji jest trudne, będziemy rozważali algorytmy oparte na metodzie "dziel i zwyciężaj". Szczególnym przypadkiem problemu optymalnego multi-uliniowienia jest problem optymalnego uliniowienia pary multi-uliniowień.

Definicja 21. Dla ustalonego zbioru uliniowień deskryptorów Φ struktur ze zbioru $\mathcal{S} = \{S_1, S_2, \ldots, S_N\}$ oraz multi-uliniowień Z_1 zbioru $\mathcal{S}_1 = \{S_1, \ldots, S_l\}$ i Z_2 zbioru $\mathcal{S}_2 = \{S_{l+1}, \ldots, S_N\}$ **problem najlepszych uliniowień pary muli-uliniowień** polega na znalezieniu multi-uliniowienia zbioru $\mathcal{S}$ o wsparciu w Φ i maksymalnej wielkości, które obcięte do struktur za zbiorów $\mathcal{S}_1$ i $\mathcal{S}_2$ jest równe odpowiednio Z_1 i Z_2. Analogicznie, decyzyjny **problem optymalnego uliniowienia pary multi-uliniowień** (POUPMU) przy ustalonej liczbie $r \in \mathbb{R}$ polega na rozstrzygnięciu, czy takie uliniowienie o wielkości nie mniejszej niż r istnieje.

Problem ten jest NP-zupełny, ponieważ zawiera zbadany w poprzednim rozdziale problem POUS. Istotne jest, że pozostaje on NP-zupełny nawet, gdybyśmy dysponowali wyrocznią zwracającą optymalne uliniowienie dla danej pary struktur.

Twierdzenie 13. *Jeżeli dane wejściowe problemu POUPMU zostaną uzupełnione o optymalne uliniowienia wszystkich par struktur ze zbioru $\mathcal{S}_1 \times \mathcal{S}_2$, tak sformułowany problem pozostaje NP-zupełny.*

Dowód. Ponownie posłużymy się redukcją problemu 3SAT. Aby uniknąć zagubienia idei w formalnych szczegółach, będziemy się posługiwać uproszczoną notacją oraz poprzestaniemy na łatwym do uzupełnienia szkicu dowodu. Ograniczymy się do określenia uliniowień deskryptorowych pomiędzy pewnymi elementami deskryptorowymi (lub

zbiorami elementów), przy założeniu, że są one rozłączne. Niech $U = \{u_1, \ldots, u_k\}$, $C = \{C_1, \ldots, C_l\}$ będzie instancją problemu 3SAT. Podobnie, jak w poprzednim dowodzie, skonstruujemy zbiory struktur odpowiadających: zmiennym (zbiory V i V'), klauzulom (zbiór K) oraz wartościowaniu zmiennych (zbiory T, F i L) zawierające podane rozłączne elementy deskryptorowe[3]:

(1) $V = \{V_1, \ldots, V_k\}$, gdzie $V_i = \left\{v_i, s_i^{V_1}, s_i^{V_2}\right\}$

(2) $V' = \{V'_1, \ldots, V'_k\}$, gdzie $V'_i = \left\{v'_i, s_i^{V'_1}, x_i^{V'}\right\}$

(3) $K = \{K_1, \ldots, K_l\}$, gdzie $K_i = \left\{k_{i1}, k_{i2}, k_{i3}, s_i^{K_1}, s_i^{K_2}, s_i^{K_3}, x_i^{K_1}, x_i^{K_2}, x_i^{K_3}\right\}$

(4) $T = \{T_1, \ldots, T_k\}$, gdzie $T_i = \left\{t_i, s_i^{T_1}, s_i^{T_2}, s_i^{T_3}, s_i^{T_4}, s_i^{T_5}\right\}$

(5) $F = \{F_1, \ldots, F_k\}$, gdzie $F_i = \left\{f_i, s_i^{F_1}, s_i^{F_2}, s_i^{F_3}, s_i^{F_4}, s_i^{F_5}\right\}$

(6) $L = \{L_1, \ldots, L_k\}$, gdzie $L_i = \left\{t'_i, f'_i, s_i^{L_1}, s_i^{L_2}\right\}$

Struktury te podzielimy na dwa zbiory:

$$\mathcal{S}_1 = T \cup F \cup V'$$
$$\mathcal{S}_2 = L \cup K \cup V$$

Niech multi-uliniowienie Z_1 utożsamia ze sobą dla elementy $x_i^{V'}$ wszystkich wartości i, natomiast Z_2 będzie puste. Zdefiniujemy uliniowienia deskryptorowe odpowiadające za wartościowanie zmiennych:

(1) $\Phi^{TL} = \left\{\varphi_i^{TL} \colon T_i \to L_i \middle| 1 \leq i \leq k \wedge \varphi_i^{TL}(t_i) = t'_i \wedge \varphi_i^{TL}(s_i^{T_1}) = s_i^{L_1}\right\}$

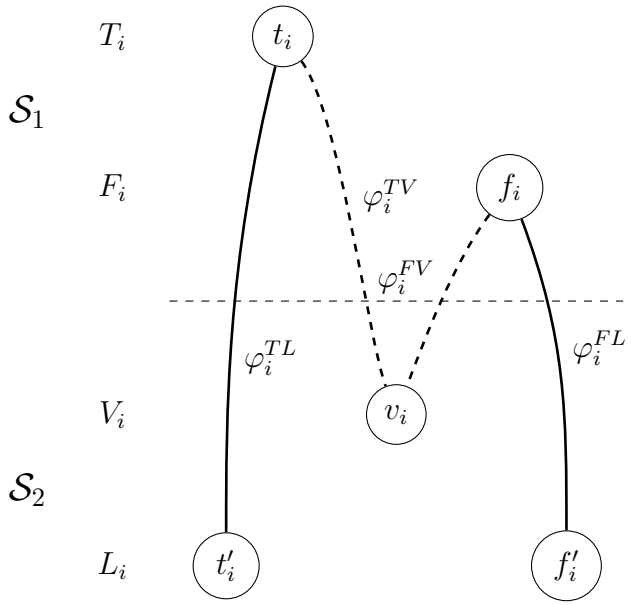
(2) $\Phi^{FL} = \left\{\varphi_i^{FL} \colon F_i \to L_i \middle| 1 \leq i \leq k \wedge \varphi_i^{FL}(f_i) = f'_i \wedge \varphi_i^{FL}(s_i^{F_1}) = s_i^{L_2}\right\}$

(3) $\Phi^{TV} = \left\{\varphi_i^{TV} \colon T_i \to V_i \middle| 1 \leq i \leq k \wedge \varphi_i^{TV}(t_i) = v_i \wedge \varphi_i^{TV}(s_i^{T_2}) = s_i^{V_1}\right\}$

(4) $\Phi^{FV} = \left\{\varphi_i^{FV} \colon F_i \to V_i \middle| 1 \leq i \leq k \wedge \varphi_i^{FV}(f_i) = v_i \wedge \varphi_i^{FV}(s_i^{F_2}) = s_i^{V_1}\right\}$

Symbole $s_i^{(\cdot)}$ oznaczają paczki elementów, które pełnią rolę analogiczną do v_i^j i l_i^j w poprzednim dowodzie. Służą one mianowicie zagwarantowaniu, że zawierające je uliniowienia deskryptorowe będą należały do maksymalnego multi-uliniowienia. Ich długości są tak dobrane, żeby dla ustalonego i wsparcie maksymalnego multi-uliniowienia zawierało φ_i^{TL}, φ_i^{FL} oraz jedno z φ_i^{TV}, φ_i^{FV} (por. rys. 4.2). Wszystkie cztery uliniowienia

[3]Określamy w tym miejscu struktury przez podanie zbiorów elementów deskryptorowych. Pomijamy szczegóły techniczne związane między innymi z kolejnością w jakiej elementy występują.



Rysunek 4.2: Jeżeli uliniowienia φ_i^{TL} i φ_i^{FL} należą do wsparcia, uliniowienia φ_i^{TV} i φ_i^{FV} (linia przerywana) wykluczają się wzajemnie.

nie mogą należeć do wsparcia multi-uliniowienia, ponieważ oznaczałoby to, że elementy t'_i i f'_i należące do struktury L_i są ze sobą uliniowione za pośrednictwem t_i, v_i i f_i, co przeczy definicji multi-uliniowienia.

Podobnie zdefiniujemy uliniowienia deskryptorowe odpowiadające za kodowanie zmiennych i rodzajów literałów występujących w klauzuli:

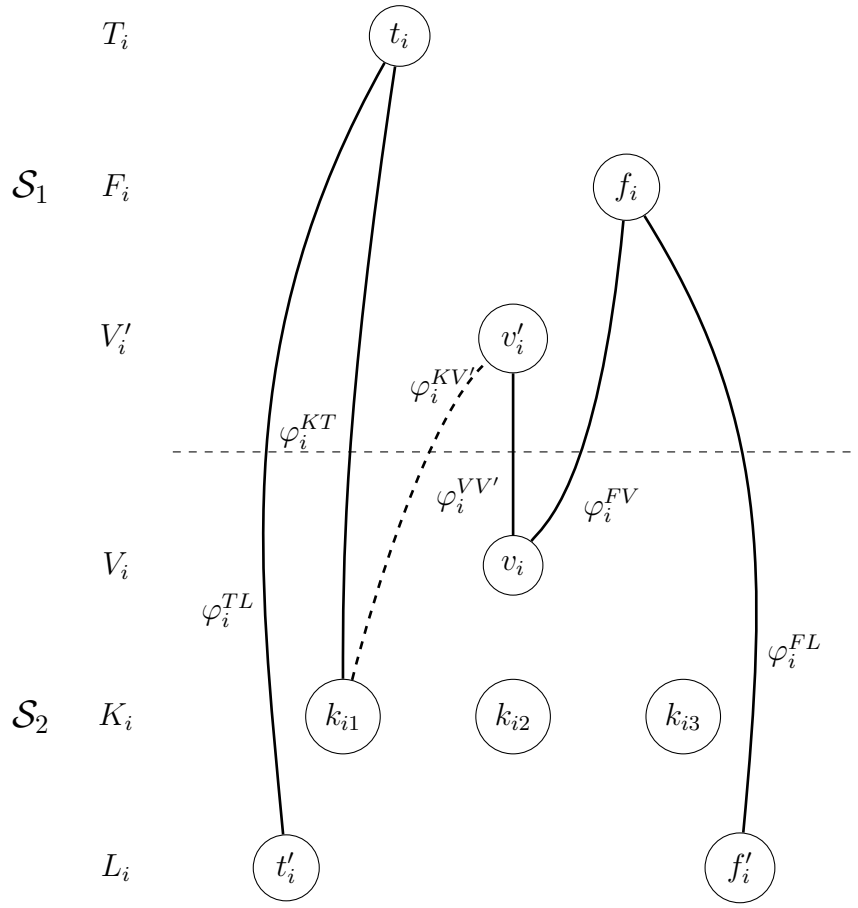
$$(5)\quad \Phi^{KT} = \left\{ \varphi_{ij}^{KT} \colon K_i \to T_p \;\middle|\; \begin{array}{l} 1 \le i \le l \wedge 1 \le j \le 3 \wedge \\ \wedge\, \varphi_{ij}^{KT}(k_{ij}) = t_p \wedge \varphi_{ij}^{KT}(s_i^{K_j}) = s_p^{T_{i+2}} \wedge \\ \wedge\, u_p \text{ występuje na } j\text{-tym miejscu w klauzuli } C_i \end{array} \right\}$$

$$(6)\quad \Phi^{KF} = \left\{ \varphi_{ij}^{KF} \colon K_i \to F_p \;\middle|\; \begin{array}{l} 1 \le i \le l \wedge 1 \le j \le 3 \wedge \\ \wedge\, \varphi_{ij}^{KF}(k_{ij}) = f_p \wedge \varphi_{ij}^{KF}(s_i^{K_j}) = s_p^{F_{i+2}} \wedge \\ \wedge\, \neg u_p \text{ występuje na } j\text{-tym miejscu w klauzuli } C_i \end{array} \right\}$$

$$(7)\quad \Phi^{KV'} = \left\{ \varphi_{ij}^{KV'} \colon K_i \to V'_p \;\middle|\; \begin{array}{l} 1 \le i \le l \wedge 1 \le j \le 3 \wedge \\ \wedge\, \varphi_{ij}^{KV'}(k_{ij}) = v'_p \wedge \varphi_{ij}^{KV'}(x_{K_j}) = x_p^{V'} \wedge \\ \wedge\, u_p \text{ lub } \neg u_p \text{ występuje na } j\text{-tym miejscu} \\ \quad \text{w klauzuli } C_i \end{array} \right\}$$

$$(8)\quad \Phi^{VV'} = \left\{ \varphi_i^{VV'} \colon V_i \to V'_i \;\middle|\; 1 \le i \le k \wedge \varphi_i^{VV'}(v_i) = v'_i \wedge \varphi_i^{VV'}(s_i^{V_2}) = s_i^{V'_1} \right\}$$

Długości paczek elementów $s_i^{\square}$ są tak dobrane, aby do wsparcia maksymalnego multi-uliniowienia należały wszystkie uliniowienia postaci $\varphi_i^{VV'}$. Struktury ze zbioru



Rysunek 4.3: Wartościowanie $u_i \to 0$ reprezentowane przez φ_i^{FV} wyklucza użycie tej zmiennej do potwierdzenia klauzuli, w której występuje literał pozytywny u_i (reprezentowany przez φ_j^{KT}).

V' są konieczne, aby można było uliniawiać elementy odpowiadające zmiennym (v_i) z elementami odpowiadającymi literałom w klauzulach (k_{ij}). Do wsparcia maksymalnego uliniowienia również należeć będą wszystkie uliniowienia ze zbiorów Φ^{KT} i Φ^{FT}, które służą kodowaniu rodzaju literału występującego w klauzuli. Dla każdej klauzuli do wsparcia multi-uliniowienia może należeć co najwyżej jedno uliniowienie ze zbioru $\Phi^{KV'}$, które koduje informację, że dana klauzula jest spełniona dzięki wartościowaniu odpowiadającej uliniowieniu zmiennej. Jest tak, ponieważ wszystkie elementy postaci $x_i^{V'}$ są ze sobą uliniowione, zatem dla ustalonego i co najwyżej jeden element postaci $x_i^{K_j}$ może być im przyporządkowany (por. rys. 4.3). To zabezpieczenie jest konieczne, aby zagwarantować, że jeżeli wsparcie multi-uliniowienia o ustalonej wielkości zawiera l uliniowień ze zbioru $\Phi^{KV'}$, to pokrywają one po jednym literale w każdej z klauzul i, co za tym idzie, formuła 3SAT przy opisanym przez to multi-uliniowienie wartościowaniu jest spełniona.

Zauważmy na koniec, że maksymalne uliniowienia wszystkich par struktur są sumą wszystkich uliniowień deskryptorowych odnoszącej się do danej pary struktur. Pozostałe istotne elementy rozumowania są analogiczne jak w dowodzie twierdzenia 12. □

W dowodzie wykorzystaliśmy fakt, że wzajemna niesprzeczność uliniowień deskryptorowych w problemie znajdowania maksymalnego multi-uliniowienia zależy od kontekstu, czyli innych uliniowień należących do wsparcia rozważanego multi-uliniowienia. Ta właściwość w zasadzie uniemożliwia zastosowanie metody opartej na wyszukiwaniu klik w grafie niesprzeczności. Ponadto, jeżeli dysponowalibyśmy zbiorem optymalnych uliniowień wszystkich par rozważanych struktur, znalezienie optymalnego multi-uliniowienia o wsparciu zawartym w sumie wsparć uliniowień par jest NP-trudne. Z tego powodu zaproponowany został algorytm heurystyczny.

4.4. Ewolucyjny algorytm znajdowania multi-uliniowień

Ten podrozdział zawiera opis heurystycznego algorytmu znajdowania maksymalnych multi-uliniowień. W poprzednim podrozdziale wprawdzie udowodniliśmy, że problem optymalnych uliniowień pary multi-uliniowień jest NP-zupełny tak, jak problem optymalnych multi-uliniowień. Niemniej jednak podstawową trudnością, z jaką należy się uporać, jest zależąca od kontekstu relacja niesprzeczności pomiędzy uliniowieniami deskryptorowymi. Jeżeli uliniawiane są dwa multi-uliniowienia, które są bliskie optymalnym, szansa, że dwa aminokwasy w obrębie jednego z nich zostaną ze sobą uliniowione na dwa różne sposoby za pośrednictwem aminokwasów z drugiego multi-uliniowienia

(por. rys. 4.2 i 4.3), jest niewielka, a sama sprzeczność w praktyce jest zazwyczaj usuwalna niewielkim kosztem. Dlatego algorytm, który opiszemy, polega na dzieleniu budowanego multi-uliniowienia na dwie części i optymalizowaniu uliniowienia pomiędzy nimi. Uzasadnieniem jego skuteczności jest poniższy fakt:

Fakt 8. *Jeżeli multi-uliniowienie zbioru struktur $\mathcal{S}$ ma maksymalną miarę (dla ustalonego zbioru Φ), zawarte w nim uliniowienie dowolnej pary multi-uliniowień zbiorów struktur $\mathcal{S}_1$ i $\mathcal{S}_2$ (t. że $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$ i $\mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset$) jest optymalne.*

Niestety implikacja odwrotna nie jest prawdziwa, co oznacza, że prezentowany algorytm może utknąć w lokalnym maksimum. Dodatkowo należy zauważyć, że trudno jest się spodziewać, że będzie możliwe przebadanie wszystkich możliwych podziałów $\mathcal{S}$. Istotny jest zatem sposób wyboru multi-uliniowień, które będą uliniawiane.

Algorytmy ewolucyjne stanowią popularną klasę heurystyk stosowanych do rozwiązywania problemów optymalizacyjnych. Ich niewątpliwą zaletą jest olbrzymia wszechstronność. Jedynym wymaganiem jest istnienie funkcji celu oraz możliwość zdefiniowania operatorów mutacji i krzyżowania. Ponieważ metoda ta jest szeroko opisana w literaturze[6, 22] ograniczymy się jedynie do skrótowego opisu jej ogólnego zarysu(proc. 4.1).

```
GENETIC({Z_1, ..., Z_n})
 1  P ← {Z_1, ..., Z_n}
 2  P' ← ∅
 3  while max_{Z∈P} score(Z) ≠ max_{Z∈P'} score(Z)
 4        do for i ← 1 to m
 5              do if RAND(0,1) ≤ mutProb
 6                    then p ← losowa wartość całkowita z przedziału [1,n]
 7                         Z'_i ← MUTATE(Z_p)
 8                    else p ← losowa wartość całkowita z przedziału [1,n]
 9                         q ← losowa wartość całkowita z przedziału [1,n]
10                         Z'_i ← CROSSOVER(Z_p, Z_q)
11           P' = P ∪ {Z'_1, ..., Z'_m}
12           P ← SELECTION(P', n)
```

Procedura 4.1: GENETIC

Jak widać algorytm jest nieskomplikowany i polega na generowaniu kolejnych pokoleń rozwiązań dopóki jakość najlepszego rozwiązania się zwiększa. Kolejne rozwiązania generowane są za pomocą jednej z dwóch funkcji MUTATE i CROSSOVER. Funkcja MUTATE, jak sama nazwa wskazuje dokonuje losowej poprawki w rozwiązaniu potencjalnie (choć niekoniecznie) poprawiającej jego jakość. Funkcja CROSSOVER generuje nowe rozwiązanie na bazie dwóch wskazanych. Wreszcie funcja SELECTION odpowiada procesowi biologicznej selekcji naturalnej i zmniejsza liczebność wygenerowanej populacji pozostawiając jedynie najlepsze rozwiązania.

Jak już wpominaliśmy, opisywany algorytm zasadza się na metodzie "dziel i zwyciężaj". Do podziału problemu na części wykorzystamy pojęcie uliniowienia pary multi-uliniowień. Aby rozłożyć multi-uliniowienie na zestaw uliniowień par, zdefiniujemy binarne drzewa rozpinające multi-uliniowienie.

Definicja 22. Drzewem rozpinającym multi-uliniowienie Z zbioru struktur $\mathcal{S} = \{S_1, \ldots, S_N\}$ nazwiemy drzewo binarne mające N liści etykietowanych strukturami S_i, którego wierzchołki etykietowane są uliniowieniami multi-uliniowień przypisanych jego potomkom oraz wszystkie te multi-uliniowienia są zawarte w Z i dodatkowo korzeń jest etykietowany Z.

Nietrudno zauważyć, że dowolne drzewo binarne mające właściwą liczbę liści przy dowolnym ich etykietowaniu strukturami można poetykietować multi-uliniowieniami tak, aby rozpinało dowolne multi-uliniowienie tych struktur. Powiemy, że multi-uliniowienie jest **maksymalne** dla danego drzewa (przy ustalonym zbiorze Φ), jeżeli wierzchołki są poetykietowane optymalnymi uliniowieniami multi-uliniowień zawartych w potomkach. Analogicznie, powiemy, że drzewo rozpinające jest **optymalne** dla danego multi-uliniowienia (przy ustalonej mierze podobieństwa), jeżeli suma wartości miary podobieństwa obliczonej dla multi-uliniowień zawartych w wierzchołkach jest maksymalna.

Obliczanie maksymalnego uliniowienia dla danego drzewa rozpinającego wymaga obliczenia $N-1$ uliniowień par multi-uliniowień. Jest zatem operacją znacznie prostszą niż znajdowanie optymalnego multi-uliniowienia. Natomiast do obliczania optymalnego drzewa rozpinającego zastosowaliśmy algorytm zachłanny w każdym kroku scalający parę wierzchołków o maksymalnej wartości miary multi-uliniowienia.

Inicjalizacja

Przed rozpoczęciem iteracji algorytmu genetycznego konieczne jest stworzenie populacji początkowej. W tym celu obliczamy maksymalne multi-uliniowienia dla losowych drzew rozpinających. Drzewa rozpinające losowane są przy użyciu randomizowanego algorytmu zachłannego wykonującego hierarchiczną klasteryzację struktur na podstawie ich podobieństwa. Randomizacja polega na wybieraniu zbiorów struktur do scalenia z prawdopodobieństwem proporcjonalnym do średniego podobieństwa struktur w nich zawartych.

Mutacja

Celem mutacji w algorytmie genetycznym jest dokonanie pewnej losowej zmiany w rozwiązaniu. Mutacja, w odróżnieniu od krzyżowania, może tworzyć fragmenty rozwiązania, które nie występują w populacji i w ten sposób zabezpiecza obliczenie przed utknięciem w maksimum lokalnym. W opisywanym rozwiązaniu mutacja polega na wylosowaniu wierzchołka w optymalnym drzewie rozpinającym rozważane multi-uliniowienie z prawdopodobieństwem proporcjonalnym do różnicy pomiędzy aktualną miarą podobieństwa multi-uliniowienia w nim zawartego a jej górnym ograniczeniem wynikającym z podobieństwa par struktur. Im ta różnica jest większa, tym większą można mieć nadzieję, że takie multi-uliniowienie można zastąpić bardziej optymalnym. Następnie aktualizowane jest multi-uliniowienie w rodzicu rozważanego wierzchołka i dalej aż do korzenia.

Krzyżowanie

Krzyżowanie jest centralnym aspektem koncepcji algorytmu genetycznego. Celem tej operacji jest stworzenie nowego rozwiązania na bazie dwóch istniejących. W wielu przypadkach stanowi ona o przewadze algorytmu genetycznego nad innymi metodami optymalizacji, których krok polega na poprawianiu pojedynczego rozwiązania. W omawianym przypadku krzyżowanie danych multi-uliniowień Z_1 i Z_2 polega na wylosowaniu pewnego podziału zbioru struktur i obliczeniu optymalnego uliniowienia rozważanych multi-uliniowień obciętych do wylosowanych zbiorów (odpowiednio $Z_1|_{S_{p_1},\ldots,S_{p_k}}$ i $Z_2|_{S_{q_1},\ldots,S_{q_{N-k}}}$).

Selekcja

Spośród przebadanych typowych algorytmów selekcji najlepsze rezultaty dawał algorytm selekcji elitarnej polegający na pozostawieniu do następnej iteracji rozwiązań o najwyższej mierze. Rezultat ten jest o tyle zaskakujący, że jest to algorytm najprostszy, "niebiologiczny" (najsilniejsze osobniki w populacji są nieśmiertelne, a najsłabsze nie mają żadnych szans na przeżycie) i najłatwiej może prowadzić do utknięcia w lokalnym maksimum.

Obliczanie uliniowienia pary multi-uliniowień

Mimo dodatkowych komplikacji, do znajdowania uliniowienia pary multi-uliniowień można z powodzeniem zastosować opisane w poprzednim rozdziale algorytmy uliniowiania pary struktur. Obliczenie składa się z trzech zasadniczych kroków:

(1) Obliczenie relacji niesprzeczności uliniowień deskryptorowych przy uwzględnieniu informacji o aminokwasach utożsamianych w danych multi-uliniowieniach (jeżeli multi-uliniowienia pokrywają te fragmenty struktur, które mają szanse zostać uliniowione, ten krok pozwala uniknąć większości potencjalnych niezgodności).

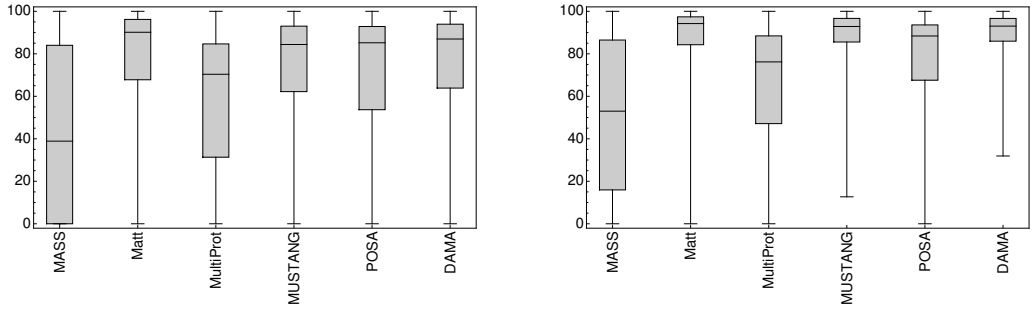
(2) Obliczenie uliniowienia jednym z trzech algorytmów: TS (por. 3.5.1), REMC (por. 3.5.2), MS (por. 3.5.3).

(3) Jeżeli znaleziony w poprzednim kroku zbiór uliniowień deskryptorowych nie jest zgodny, odrzucenie możliwie małej liczby elementów, aby go uzgodnić (jak wynika z tw. 13 jest to problem NP-trudny, dlatego zastosowaliśmy suboptymalny algorytm zachłanny).

4.5. Zastosowania

4.5.1. Implementacja

Wykorzystując moduły opisanego w rozdziale 3.7.1 programu DEDAL zaimplementowaliśmy opisany powyżej algorytm. Ze względu na znacznie większą złożoność problemu zrezygnowaliśmy z dwuetapowego sposobu obliczeń, w którym uliniowienie zbudowane z 3-segmentowych uliniowień deskryptorowych jest następnie rozszerzane o mniejsze uliniowienia deskryptorowe, poprzestając wyłącznie na 3-segmentowych uliniowieniach. Ponadto, ze względu na fakt, że w przypadku uliniawiania par multi-uliniowień graf



Rysunek 4.4: Jakość, z jaką odtwarzane są uliniowienia ze zbioru SISY-multiple. Wykresy pudełkowe prezentują rozkłady jakości uliniowień odtworzonych przez badane metody według miary Q_C (a) i Q_P (b). Wyniki metod innych niż DAMA pochodzą z pracy [9].

niesprzeczności jest zazwyczaj znacznie większy niż dla pary struktur, zastosowaliśmy algorytm MS (por. 3.5.3). W algorytmie genetyczym rozważaliśmy populację liczącą 5 osobników, w każdym pokoleniu tworzyliśmy 10 nowych potomków oraz przyjęliśmy prawdopodobieństwo mutacji równe 0.2. Opisaną implementację nazwaliśmy DAMA (*Descriptor Assisted Multi-Alignment*)[16].

4.5.2. Test na zbiorze SISY-multiple

Zbiór SISY-multiple [9] jest rozwinięciem opisanego w pracy [49] i wykorzystanego przez nas w rozdziale 3.7.4 zbioru SISY. Zawiera on multi-uliniowienia pochodzące z bazy uliniowień SISYPHUS [4], które zostały oczyszczone przez pominięcie nieaktualnych struktur usuniętych z bazy PDB. Również struktury występujące wielokrotnie w multi-uliniowieniu zostały pominięte, aby uniknąć trudnej do uwzględnienia na etapie oceniania jakości wyników wieloznaczności. Ostatecznie spośród 149 multi-uliniowień z bazy SISYPHUS w zbiorze SISY-multiple pozostało 106 liczących co najmniej 3 struktury.

Jakość multi-uliniowień obliczonych programem DAMA podobnie jak w rozdziale 3.7 ocenialiśmy przez porównywanie z uliniowieniami wzorcowymi. Rozważaliśmy dwie miary podobieństwa pomiędzy obliczonym uliniowieniem a wzorcowym. Miara Q_C jest liczbą pełnych kolumn uliniowionych zgodnie ze wzorcem, unormowaną przez liczbę kolumn w uliniowieniu wzorcowym[67]. Miara ta jest stosowalna, jeżeli uliniowienie wzorcowe obejmuje jedynie aminokwasy wspólne dla wszystkich struktur. Mniej restryk-

	MASS	Matt	MultiProt	MUSTANG	POSA	DAMA
Q_C – mediana	38.89	90.14	70.37	84.38	85.19	86.96
Q_C – średnia	43.65	75.83	58.20	71.74	71.33	72.78
Q_P – mediana	53.00	94.23	76.19	92.86	88.39	93.03
Q_P – średnia	51.27	85.09	63.85	85.23	77.11	86.64

Tabela 4.1: Średnia i mediana jakości działania porównywanych metod na zbiorze SISY-multiple

cyjna miara Q_P jest proporcjonalna do liczby poprawnie uliniowionych par aminokwasów[61]. Porównaliśmy jakość uliniowień obliczonych programem DAMA z wynikami dla innych metod przedstawionymi w pracy [9] (tab. 4.1, rys. 4.4). Ze względu na to, że autorom pracy [9] z powodu technicznych usterek testowanych programów nie udało się obliczyć wszystkich multi-uliniowień, w dalszej analizie będziemy rozważać zbiór testowy zawężony do 61 multi-uliniowień, na których żadna z testowanych metod nie zawiodła. DAMA daje wyniki o porównywalnej jakości z metodami Matt i MUSTANG, jest nieco lepsza od POSA oraz istotnie lepsza od metod MASS i MultiProt. Wyniki istotności statystycznej różnic pomiędzy jakością analizowanych metod przedstawiamy w tabeli 4.2. Rysunek 4.5 przedstawia porównanie jakości wyników metody DAMA z pozostałymi. Widać, że DAMA ma przewagę nad pozostałymi metodami w przypadku, gdy wymagane jest wykrycie permutacji[4]. Należy zwrócić uwagę na fakt, że miara Q_P jest korzystniejsza dla metody DAMA. Jest to przypuszczalnie konsekwencja zastosowania miary jakości multi-uliniowienia opartej na sumie jakości składających się na nie uliniowień par, aczkolwiek nie bez znaczenia może być również wybór operatorów mutacji i krzyżowania w algorytmie genetycznym. Niewykluczone, że w sytuacji, gdy wszystkie struktury z wyjątkiem jednej uliniowione są zgodnie ze wzorcem, wykonanie dodatkowej procedury mutacji przy podziale zbioru struktur na singleton zawierający strukturę niedopasowaną oraz pozostałe struktury, doprowadziłoby to wyniku zgodnego z pożądanym.

[4]Jedynie MASS wśród pozostałych metod ma możliwość uliniawiania struktur z przestawieniami.

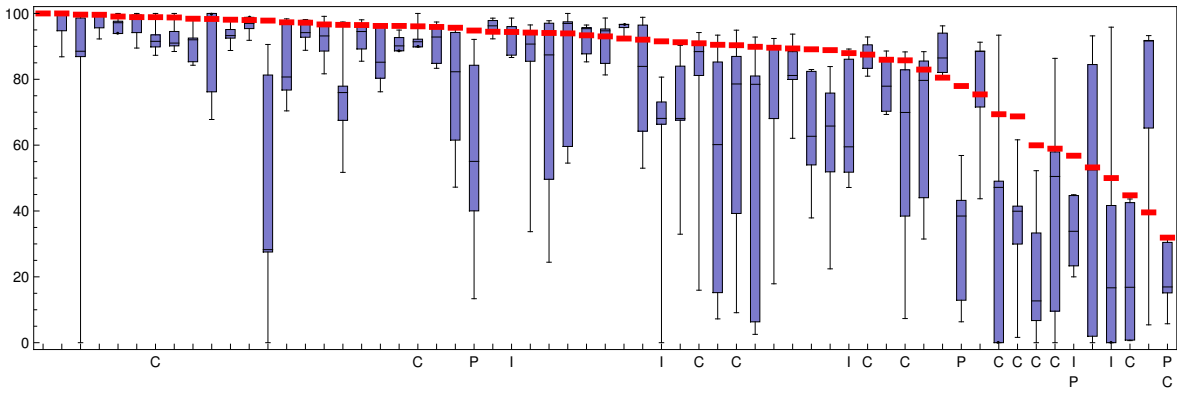
	Matt	MultiProt	MUSTANG	POSA	DAMA
MASS	$4.2 \cdot 10^{-10}$	$1.1 \cdot 10^{-8}$	$6.3 \cdot 10^{-9}$	$9.4 \cdot 10^{-9}$	$5.6 \cdot 10^{-10}$
Matt		$1.2 \cdot 10^{-8}$	$7.5 \cdot 10^{-2}$	$1.9 \cdot 10^{-2}$	$7.7 \cdot 10^{-2}$
MultiProt			$3.3 \cdot 10^{-6}$	$3.1 \cdot 10^{-6}$	$3.6 \cdot 10^{-8}$
MUSTANG				$4.4 \cdot 10^{-1}$	$6.2 \cdot 10^{-1}$
POSA					$3.3 \cdot 10^{-1}$

(a)

	Matt	MultiProt	MUSTANG	POSA	DAMA
MASS	$7.6 \cdot 10^{-11}$	$6.0 \cdot 10^{-7}$	$7.6 \cdot 10^{-11}$	$3.0 \cdot 10^{-9}$	$2.2 \cdot 10^{-11}$
Matt		$4.1 \cdot 10^{-10}$	$6.8 \cdot 10^{-1}$	$3.9 \cdot 10^{-4}$	$5.1 \cdot 10^{-1}$
MultiProt			$3.8 \cdot 10^{-11}$	$8.8 \cdot 10^{-7}$	$3.2 \cdot 10^{-10}$
MUSTANG				$1.3 \cdot 10^{-4}$	$3.4 \cdot 10^{-1}$
POSA					$2.4 \cdot 10^{-4}$

(b)

Tabela 4.2: Wyniki testu istotności Wilcoxona dla jakości porównywanych metod na zbiorze SISY-multiple dla miary Q_C (a) i Q_P (b).



Rysunek 4.5: Jakość odtwarzania uliniowień (miara Q_P) ze zbioru SISY-multiple przez algorytm DAMA (kolor czerwony) i pozostałe metody (wykres pudełkowy). Kolumny na wykresie odpowiadają wynikom dla poszczególnych multi-uliniowień. Litery oznaczają trudności występujące w danym przykładzie (I – insercje/delecje, P – permutacje, C – odkształcenia).

	MASS	Matt	MultiProt	MUSTANG	POSA	DAMA
Q_C – AL10074933	0.00	0.00	73.91	82.61	18.84	95.65
Q_C – AL10050155	0.00	0.00	20.00	–	–	0.00 (31.67)
Q_P – AL10074933	27.54	0.00	81.30	90.58	28.26	97.83
Q_P – AL10050155	0.67	44.99	37.49	–	–	89.11 (93.12)

Tabela 4.3: Jakość rekonstrukcji multi-uliniowienia AL10050155. W nawiasach podane wartości po uwzględnieniu symetrii struktury `1p1dA`.

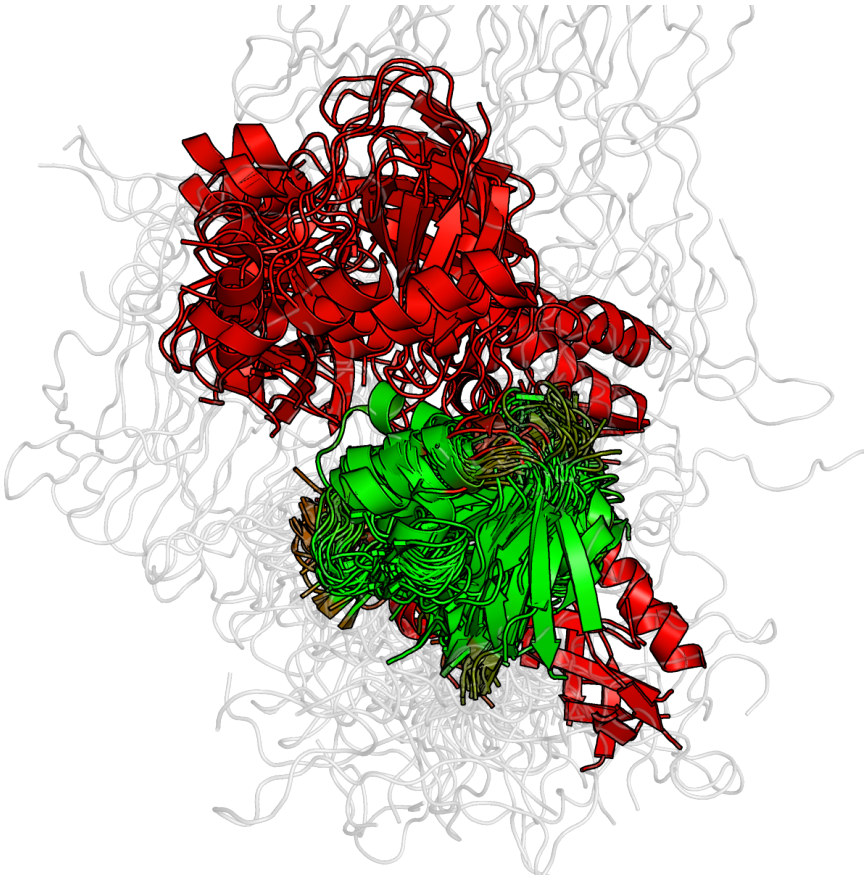
4.5.3. Wybrane przykłady

Domeny PDZ

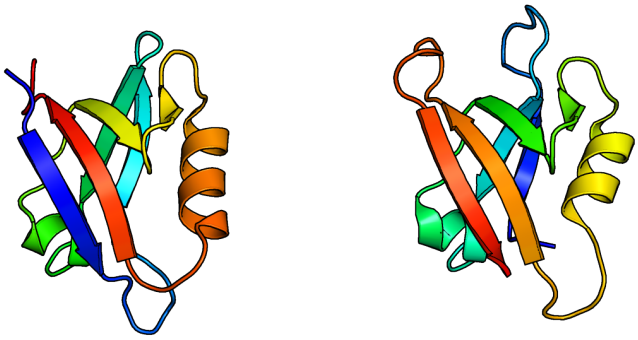
Domeny PDZ pełnią funkcję rozpoznawania fragmentów polipeptydowych. Zostały one odkryte we wszystkich zsekwencjonowanych genomach zwierzęcych. Odgrywają kluczową rolę w procesach przekazywania sygnałów w komórce. Domeny o podobnej strukturze i sekwencji, aczkolwiek z permutacją cyrkularną, zostały również zidentyfikowane w niektórych bakteriach. Ponieważ nie odnaleziono białek o sekwencji homologicznej do PDZ w drożdżach (*Saccharomyces cerevisiae*), przypuszcza się, że te domeny pojawiły się u bakterii w drodze horyzontalnego transferu genowego [28]. W zbiorze SISY-multiple występują trzy multi-uliniowienia zawierające domeny PDZ. Jednak ze względu na techniczne usterki badanych metod tylko jedno z nich (AL10074933) obejmujące 5 bakteryjnych odpowiedników domeny PDZ zostało uwzględnione w przedstawionych powyżej rezultatach. Multi-uliniowienie (AL10050155) zawiera 51 struktur (46 zwierzęcych i 5 bakteryjnych z permutacją cyrkularną). W tabeli 4.3 podajemy wyniki testowanych metod dla wymienionych multi-uliniowień. Jak widać rekonstrukcja podobieństwa domen PDZ jest trudna. DAMA radzi sobie z tym najlepiej. Należy jeszcze zauważyć, że jedna ze struktur w uliniowieniu AL10050155 zawiera dwa niemal identyczne (RMSD 2.74Å) powtórzenia domeny PDZ. DAMA w obliczonym multi-uliniowieniu wybiera inne powtórzenie niż autorzy multi-uliniowienia wzorcowego. Po manualnej zamianie domen rezultat znacząco się poprawia.

Kinazy białkowe

Kinazy białkowe są liczną grupą enzymów odpowiadających za przenoszenie grupy fosforanowej z wysokoenergetycznego związku, jakim jest ATP, na docelową cząsteczkę

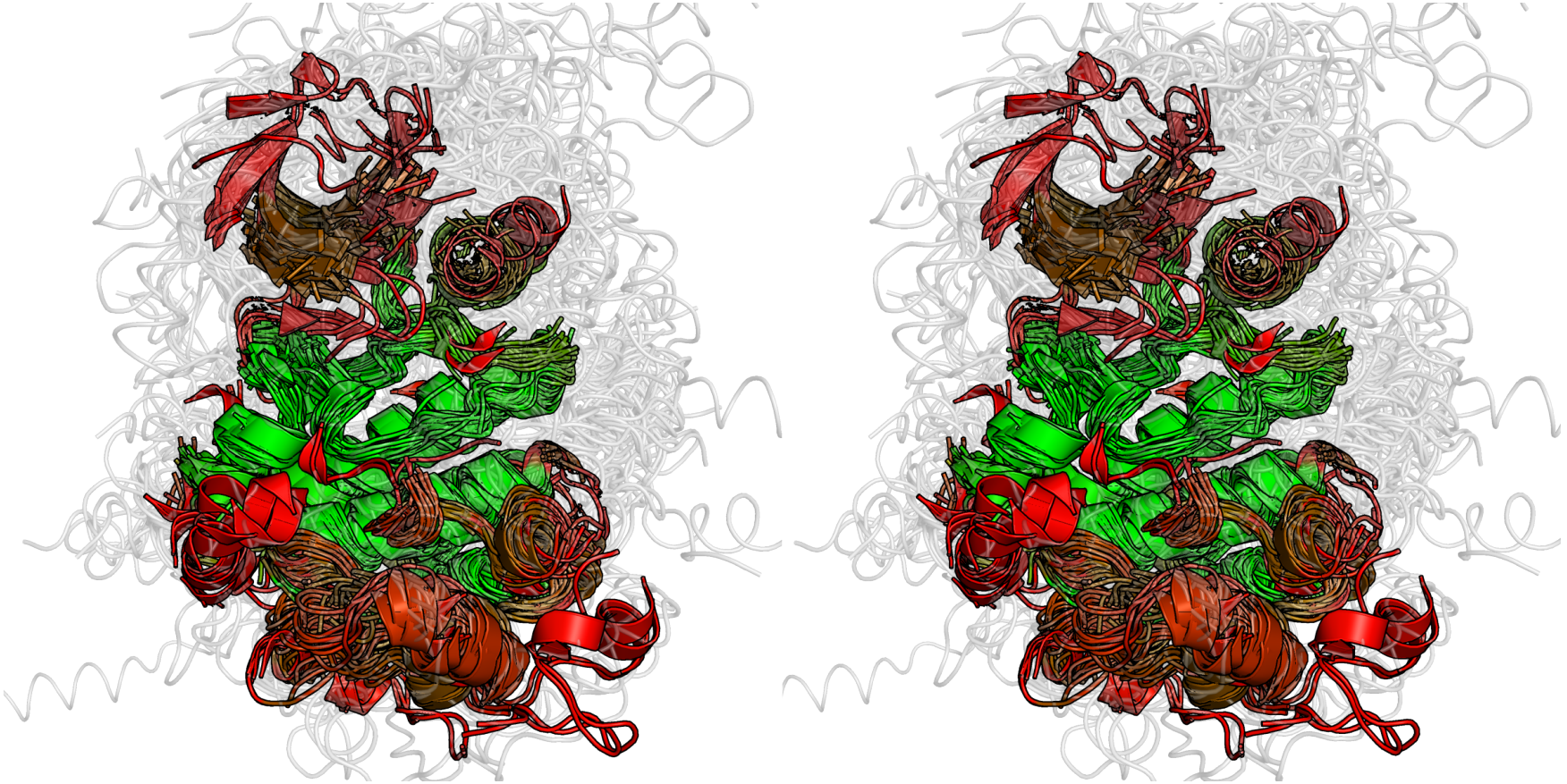


(a)



(b) (c)

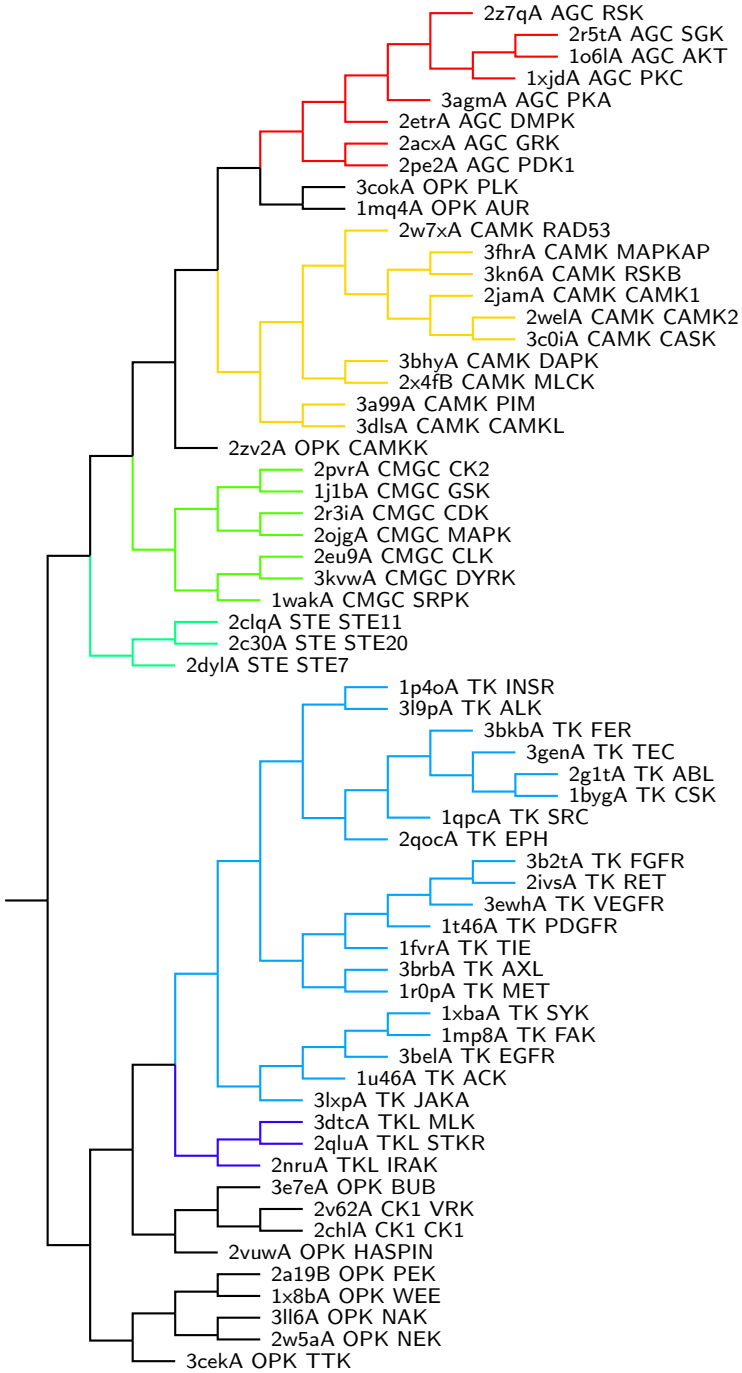
Rysunek 4.6: (a) Superpozycja domen PDZ obliczona programem DAMA. Kolorem zielonym oznaczono aminokwasy wspólne dla wszystkich struktur. Stopniowe przechodzenie koloru w kierunku czerwieni odpowiada zmniejszającej się liczbie aminokwasów uliniowionych z danym. (b) i (c) Superpozycja przykładowej domeny PDZ (`1obzA`)(b) i spermutowanego bakteryjnego odpowiednika domeny PDZ (`1fc9A`)(c). Kolor odpowiada położeniu aminokwasu w sekwencji białka (koniec-N – niebieski, koniec-C – czerwony). Pomimo różnej topologii obydwie domeny mają tę samą architekturę.



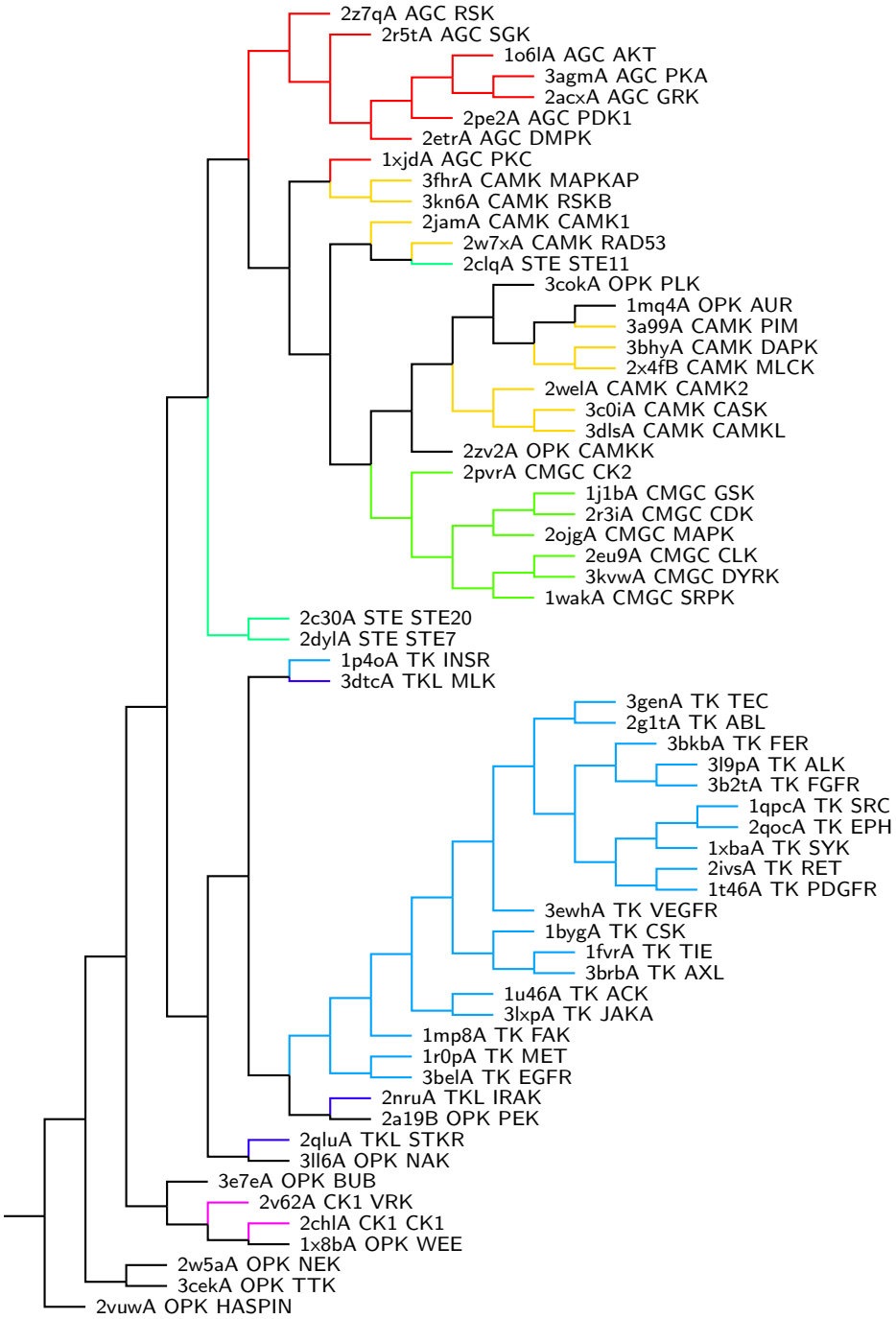
Rysunek 4.7: Obraz stereograficzny superpozycji wybranych kinaz białkowych obliczona programem DAMA. Schemat kolorystyczny taki jak na rys. 4.6a.

białka, co powoduje jej uaktywnienie. Z tego względu kinazy pełnią kluczową rolę w niemal wszystkich szlakach sygnałowych i regulują większość procesów komórkowych. Zaburzenia w działaniu kinaz są częstą przyczyną chorób. Dlatego są one atrakcyjnym celem przy opracowywaniu między innymi terapii nowotworowych. Zidentyfikowano już ponad 500 kinaz białkowych występujących u człowieka, które zostały sklasyfikowane w rodziny i podrodziny [47]. W przedstawionym poniżej eksperymencie wybraliśmy po jednej strukturze reprezentującej 63 rodziny kinaz, a następnie obliczyliśmy ich multi-uliniowienie (rys. 4.7). Zidentyfikowaliśmy 48 aminokwasów wspólnych dla wszystkich rozważanych kinaz oraz obliczyliśmy *quasi*-optymalne drzewo rozpinające multi-uliniowienie. Porównaliśmy obliczone drzewo z klasyfikacją obliczoną na podstawie podobieństwa sekwencyjnego zamieszczoną w pracy [47] przy pomocy następującej procedury. Zidentyfikowaliśmy we wzorcowym drzewie sekwencje odpowiadające porównywanym strukturom i usunęliśmy z niego pozostałe liście. Ponieważ miary podobieństwa, a co za tym idzie wagi, przypisane krawędziom, na podstawie których zbudowano rozważane drzewa, nie są kompatybilne, ograniczyliśmy się do porównania ich topologii. Zastosowaliśmy miarę podobieństwa drzew filogenetycznych zwaną odległością trójkową, która służy do porównywania drzew o jednakowych zbiorach etykiet liści. Polega ona na określeniu liczby trzyelementowych podzbiorów etykiet liści, które się różnią wzajemnym położeniem[5]. Liczba ta jest następnie normowana przez liczbę wszystkich trójek. Pojedyncza wartość tej miary jest trudna do zinterpretowania. Zbadaliśmy zatem rozkład podobieństwa losowych drzew binarnych o liściach etykietowanych porównywanymi strukturami do drzewa wzorcowego generując 1000 losowych drzew i przybliżając uzyskany histogram rozkładem normalnym. Na podstawie tego rozkładu oszacowaliśmy prawdopodobieństwo wylosowania drzewa o podobieństwie do wzorca nie mniejszym niż dla obliczonego drzewa na $1.2 \cdot 10^{-127}$. Analizowane drzewa przedstawiają rysunki 4.8 i 4.9.

[5]Porównywane jest obcięcie obydwu drzew do wybranej trójki etykiet.



Rysunek 4.8: Drzewo filogenetyczne kinaz białkowych pochodzące z pracy [47] obcięte do struktur badanych w rozdziale 4.5.3. Etykiety liści drzewa zawierają kod PDB wraz z oznaczeniem łańcucha, klasę i rodzinę kinazy. Kolor gałęzi odpowiada klasie kinaz. OPK oznacza klasę pozostałych kinaz białkowych (*Other Protein Kinases*).



Rysunek 4.9: Optymalne drzewo rozpinające multi-uliniowienie wybranych kinaz białkowych obliczone metodą DAMA. Oznaczenia i kolorowanie jak na rysunku 4.8.

Rozdział 5

Wnioski

Liczba struktur białek zdeponowanych w bazie PDB przyrasta ostatnio o 7-8 tysięcy rocznie. Ogółem znanych jest 73 tysiące struktur pogrupowanych w ok. 1400 foldów. Tak liczny i coraz bardziej kompletny zbiór danych daje możliwości nowych odkryć i w perspektywie nadzieję na znacznie pełniejsze zrozumienie zależności pomiędzy sekwencją, strukturą i funkcją białek. Aby to osiągnąć konieczne są wszakże coraz dokładniejsze i bardziej wydajne metody analizy tych struktur.

Często stosowaną strategią w badaniu struktur białek jest ich dekompozycja na mniejsze fragmenty. Tradycyjnie w tej roli występują niezbyt długie ciągłe wycinki łańcucha głównego, pary takich wycinków lub elementy struktury drugorzędowej (helisy α i harmonijki β). W tej rozprawie badaliśmy możliwość zastosowania bardziej zaawansowanego opisu przestrzennego otoczenia aminokwasów – tzw. lokalnych deskryptorów struktury. Deskryptory ze względu na swoją zróżnicowaną wielkość i strukturę topologiczną wyznaczoną przez wzorzec kontaktów ich centralnego aminokwasu z otaczającymi aminokwasami w strukturze białka, wymagają dość złożonego aparatu formalnego. Wykazaliśmy również, że problem określania podobieństwa dwóch deskryptorów jest NP-zupełny. Ponieważ rozmiar deskryptora jest ograniczony uwarunkowaniami stereochemicznymi, w toku prezentowanych badań powstała wydajna implementacja algorytmu porównującego deskryptory.

Następnie zaproponowaliśmy formalny opis uliniowień strukturalnych i badaliśmy teoretyczną złożoność obliczeniową problemu znajdowania optymalnych uliniowień zbudowanych z par podobnych fragmentów struktur. W szczególności wykazaliśmy, że problem znajdowania optymalnego uliniowienia jest NP-zupełny dla fragmentów liczących co najmniej dwa segmenty, jeżeli dopuszczalne są uliniowienia zawierające przestawienia sekwencyjne. Natomiast, jeżeli wykluczy się uliniowienia zawierające przestawie-

nia sekwencyjne, problem znajdowania optymalnych uliniowień jest NP-zupełny dla fragmentów trójsegmentowych. Zaimplementowaliśmy dokładny algorytm znajdujący optymalne uliniowienia oraz dwie heurystyki mające zastosowanie, gdy obliczenie algorytmem dokładnym o wykładniczej złożoności obliczeniowej jest zbyt czasochłonne.

Naturalnym uogólnieniem porównywania par struktur jest problem znajdowania multi-uliniowień strukturalnych. Jest on oczywiście NP-zupełny, ponieważ zawiera problem znajdowania optymalnego uliniowienia pary struktur. Bardziej szczegółowa analiza wykazała NP-zupełność przy założeniu, że dane są optymalne uliniowienia wszystkich par struktur podlegających multi-uliniowieniu. Opisaliśmy także NP-zupełny problem uliniawiania dwóch multi-uliniowień, oraz zaproponowaliśmy stosowalny w praktyce algorytm ewolucyjny.

Opisane w pracy metody zostały przetestowane na dostępnych w literaturze zbiorach danych SISYPHUS[4], SISY, RIPC[49], SISY-multiple[9], a ich wyniki skonfrontowane z innymi, popularnymi metodami (CE, DALI, FATCAT, MATRAS, C_α-match, SHEBA, MASS, Matt, MultiProt, MUSTANG, POSA). Dzięki zastosowaniu podziału struktury na relatywnie duże i specyficzne fragmenty, metody obliczania uliniowień wykorzystujące deskryptory lokalnej struktury są skuteczne w tzw. trudnych przypadkach, które obejmują permutacje cyrkularne i inne przestawienia sekwencyjne oraz odkształcenia przestrzenne. Równocześnie w odróżnieniu od metod specjalnie zaprojektowanych z myślą o takowych, bardzo dobrze radzą sobie z przypadkami łatwymi, co wykazało porównanie z metodą DALI w rozdziale 3.7.3.

Projektując testy kierowaliśmy się założeniem, że ocena poprawności uliniowienia struktur nie powinna być wypadkową jego wielkości i jakości rozumianej jako pewna miara związana z superpozycją uliniowionych fragmentów. Znacznie istotniejszy jest biologiczny sens uzyskanego dopasowania. W szczególności aminokwasy pełniące odpowiadające sobie funkcje powinny zostać uliniowione. Tylko takie uliniowienie daje użyteczne przesłanki do wyciągania wniosków o pokrewieństwie funkcjonalnym badanych struktur. Ponadto niejednokrotnie uliniowienia strukturalne stanowią bazę do szacowania prawdopodobieństw mutacji aminokwasów, czyli ich ewolucyjnego podobieństwa[18, 30].

Prezentowana metoda opiera się na relatywnie nieskomplikowanej koncepcji identyfikowania zbioru bazowych, strukturalnych klocków (w tym przypadku par podobnych deskryptorów), określenia sposobu dopasowywania klocków do siebie (relacja niesprzeczności uliniowień deskryptorowych) oraz budowania maksymalnych zespołów pasujących do siebie klocków (wyszukiwania klik). Jednak, mimo prostej koncepcji jako-

ścią wyników przewyższa konkurencyjne metody. Jest to argument przemawiający za tezą, że deskryptory lokalnej struktury są dobrym formalizmem opisu struktury białka. Przypuszczalnie ich główną zaletą jest obejmowanie fragmentów struktury, które mimo bliskości przestrzennej mogą być znacznie oddalone w sekwencji.

Zasadniczą trudnością jaka wiąże się ze składaniem uliniowień z fragmentów jest kombinatoryczna złożoność tego problemu. W przypadku zbyt małych i niewystarczająco specyficznych par podobnych fragmentów jest zbyt wiele, co prowadzi do zbyt wysokich kosztów obliczeń i wymusza uproszczenia. W metodzie deskryptorowej złożoność kombinatoryczna niejako rozkłada się na dwa poziomy obliczeń: identyfikowanie par podobnych deskryptorów i poszukiwanie optymalnego uliniowienia. Dzięki temu, mimo że formalnie oba problemy obarczone są wykładniczą złożonością obliczeniową, są one rozwiązywalne relatywnie niewielkim kosztem.

Zarówno służący do porównywania par struktur progam DEDAL, jak i obliczający multi-uliniowienia program DAMA zostały udostępnione w ramach stworzonego w tym celu serwisu *Essentia Proteomica* (http://bioexploratorium.pl/EP).

O ile metoda DEDAL ma wyraźną przewagę nad konkurencyjnymi metodami i nadaje się do eksploatacji, multi-uliniowienia obliczone metodą DAMA są z konkurencją porównywalne. W obydwu przypadkach przestrzeń przeszukiwanych rozwiązań jest większa niż w tradycyjnych podejściach, gdyż dopuszczalne są przestawienia sekwencyjne. Skutkuje to możliwością znalezienia sztucznego rozwiązania o wysokiej mierze lecz pozbawionego sensu biologicznego. W przypadku uliniowień par jest to utrudnione dzięki wprowadzeniu lokalnej miary jakości uliniowienia (rozdz. 3.6.2). Natomiast algorytm ewolucyjny znajdujący multi-uliniowienia najwyraźniej jest bardziej podatny na utykanie w tego typu lokalnych maksimach (rozdz. 4.5.3). Przypuszczalnie problem ten da się usunąć przez zastosowanie lepiej dobranych procedur mutacji i krzyżowania. Wydaje się, że istotnym udoskonaleniem byłoby wprowadzenie poziomu pośredniego pomiędzy lokalnym podobieństwem deskryptorów i globalnym podobieństwem struktur. Nowy poziom w hierarchii odpowiadałby konserwowanym elementom strukturalnym takim jak poddomeny, które mogą być tylko nieznacznie odkształcone i powinny być uliniowione z ograniczoną liczbą przestawień sekwencyjnych.

Ponieważ lokalne deskryptory struktury okazały się niezwykle użyteczne do porównywania struktur białek, wyniki zachęcają do dalszego prowadzenia badań w tym kierunku. W szczególności użyteczne wydaje się narzędzie do szybkiego przeszukiwania dużych zbiorów struktur pod kątem podobieństwa do struktury zadanej w zapytaniu. Byłoby to narzędzie analogiczne do stosowanych w dziedzinie sekwencji[2]. Przeszuki-

wanie rozpoczynałoby się od zidentyfikowania podobieństw pomiędzy deskryptorami występującymi w zadanej sekwencji a klastrami deskryptorów struktur występujących w przeszukiwanej bazie. Na ich podstawie identyfikowane byłyby struktury potencjalnie podobne. Aplikacją dualną do przedstawionej jest narzędzie wykrywające konserwowane motywy sekwencyjne występujące w badanym zbiorze struktur. Następnie takie motywy sekwencyjne mogłyby być powiązane z funkcją lub innymi cechami białek. Ich wykrycie w nowej strukturze wskazywałoby na występowanie powiązanej z nimi cechy. Interesujące wydaje się również obliczenie multi-uliniowień dużych zbiorów struktur i odtwarzanie na ich podstawie ewolucyjnego pokrewieństwa białek.

Z punktu widzenia użytkowników zaprezentowanych i proponowanych narzędzi istotna jest łatwość ich obsługi. Tradycyjnie tego typu programy udostępnia się w formie serwisów WWW, gdzie dane wejściowe podawane są w formularzu, obliczenie przeprowadzane jest na serwerach obliczeniowych będących własnością twórców programu, a obliczony wynik jest prezentowany w formie strony WWW. Taka forma jest wygodna dla osób korzystających okazjonalnie z udostępnianych usług. Niestety natura serwisów WWW i chronicznie ograniczone zasoby komputerowe grup badawczych praktycznie wykluczają stworzenie usługi bardziej interaktywnej. Dlatego postulujemy opracowanie narzędzia pozwalającego na w pełni interaktywne obliczanie uliniowień i multi-uliniowień. Szczególnie cennymi funkcjami takiego programu byłaby możliwość wskazania konserwowanych zdaniem użytkownika aminokwasów i wymuszanie ich uliniowienia, wprowadzanie ręcznych poprawek do drzewa rozpinającego multi-uliniowienie, czy też umożliwienie pogłębionego przeszukiwania przestrzeni rozwiązań poprzez wykluczenie uliniowień nie spełniających oczekiwań eksperta.

Bibliografia

[1] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990.

[2] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*, 25(17):3389–402, Sep 1997.

[3] B. Anand, S. K. Verma, and B. Prakash. Structural stabilization of GTP-binding domains in circularly permuted GTPases: implications for RNA binding. *Nucleic Acids Res*, 34(8):2196–205, 2006.

[4] A. Andreeva, A. Prlic, T. J. Hubbard, and A. G. Murzin. SISYPHUS–structural alignments for proteins with non-trivial relationships. *Nucleic Acids Res*, 35(Database issue):D253–9, 2007.

[5] O. Bachar, D. Fischer, R. Nussinov, and H. Wolfson. A computer vision based technique for 3-d sequence-independent structural comparison of proteins. *Protein Eng*, 6(3):279–88, 1993.

[6] T. Bäck. *Evolutionary algorithms in theory and practice*. Oxford University Press New York, 1996.

[7] L. G. Barrientos, J. M. Louis, I. Botos, T. Mori, Z. Han, B. R. O'Keefe, M. R. Boyd, A. Wlodawer, and A. M. Gronenborn. The domain-swapped dimer of cyanovirin-N is in a metastable folded state: reconciliation of X-ray and NMR structures. *Structure*, 10(5):673–86, 2002.

[8] L. Baum and J. Eagon. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bull. Amer. Math. Soc*, 73(360-363):212, 1967.

[9] C. Berbalk, C. S. Schwaiger, and P. Lackner. Accuracy analysis of multiple structure alignments. *Protein Sci*, 18(10):2027–35, 2009.

[10] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The Protein Data Bank. *Nucleic Acids Res*, 28(1):235–42, 2000.

[11] C. A. Bewley, K. R. Gustafson, M. R. Boyd, D. G. Covell, A. Bax, G. M. Clore, and A. M. Gronenborn. Solution structure of cyanovirin-N, a potent HIV-inactivating protein. *Nat Struct Biol*, 5(7):571–8, 1998.

[12] P. Björkholm, P. Daniluk, A. Kryshtafovych, K. Fidelis, R. Andersson, and T. R. Hvidsten. Using multi-data hidden Markov models trained on local neighborhoods of protein structure to predict residue-residue contacts. *Bioinformatics*, 25(10):1264–70, 2009.

[13] P. Bork, C. Sander, and A. Valencia. Convergent evolution of similar enzymatic function on different protein folds: the hexokinase, ribokinase, and galactokinase families of sugar kinases. *Protein Sci*, 2(1):31–40, Jan 1993.

[14] S. Busygin. A new trust region technique for the maximum weight clique problem. *Discrete Applied Mathematics*, 154(15):2080–2096, 2006.

[15] J. M. Chandonia, N. S. Walker, L. Lo Conte, P. Koehl, M. Levitt, and S. E. Brenner. ASTRAL compendium enhancements. *Nucleic Acids Res*, 30(1):260–3, 2002.

[16] P. Daniluk and B. Lesyng. DAMA – a novel method for multi-alignment of protein structures. *In preparation for Bioinformatics*.

[17] P. Daniluk and B. Lesyng. A novel method to compare protein structures using local descriptors. *BMC Bioinformatics*, 12(1):344, Aug 2011.

[18] M. Dayhoff and R. Schwartz. A model of evolutionary change in proteins. In *In Atlas of protein sequence and structure*. Citeseer, 1978.

[19] S. Dobbins, V. Lesk, and M. Sternberg. Insights into protein flexibility: The relationship between normal modes and conformational change upon protein–protein docking. *Proceedings of the National Academy of Sciences*, 105(30):10390, 2008.

[20] O. Dror, H. Benyamini, R. Nussinov, and H. Wolfson. MASS: multiple structural alignment by secondary structures. *Bioinformatics*, 19 Suppl 1:i95–104, 2003.

[21] I. Elias. Settling the intractability of multiple alignment. *J Comput Biol*, 13(7):1323–39, Sep 2006.

[22] D. Fogel. *Evolutionary computation: toward a new philosophy of machine intelligence*, volume 1. Wiley-IEEE Press, 2006.

[23] A. Gambin, S. Lasota, R. Szklarczyk, J. Tiuryn, and J. Tyszkiewicz. Contextual alignment of biological sequences (extended abstract). *Bioinformatics*, 18(suppl 2):S116, 2002.

[24] M. R. Garey and D. S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*. A Series of books in the mathematical sciences. W. H. Freeman, San Francisco, 1979.

[25] M. Gerstein and N. Echols. Exploring the range of protein flexibility, from a structural proteomics perspective. *Current opinion in chemical biology*, 8(1):14–19, 2004.

[26] N. V. Grishin. Fold change in evolution of protein structures. *J Struct Biol*, 134(2-3):167–85, 2001.

[27] A. Guerler and E. W. Knapp. Novel protein folds and their nonsequential structural analogs. *Protein Sci*, 17(8):1374–82, 2008.

[28] B. Z. Harris and W. A. Lim. Mechanism and role of PDZ domains in signaling complex assembly. *J Cell Sci*, 114(Pt 18):3219–31, Sep 2001.

[29] H. Hasegawa and L. Holm. Advances and pitfalls of protein structural alignment. *Current opinion in structural biology*, 19(3):341–348, 2009.

[30] S. Henikoff and J. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the United States of America*, 89(22):10915, 1992.

[31] L. Holm and J. Park. DaliLite workbench for protein structure comparison. *Bioinformatics*, 16(6):566–7, 2000.

[32] L. Holm and C. Sander. Protein structure comparison by alignment of distance matrices. *J Mol Biol*, 233(1):123–38, 1993.

[33] T. R. Hvidsten, A. Kryshtafovych, and K. Fidelis. Local descriptors of protein structure: a systematic analysis of the sequence-structure relationship in proteins using short- and long-range interactions. *Proteins*, 75(4):870–84, 2009.

[34] T. R. Hvidsten, A. Kryshtafovych, J. Komorowski, and K. Fidelis. A novel approach to fold recognition using sequence-derived properties from sets of structurally similar local fragments of proteins. *Bioinformatics*, 19 Suppl 2:ii81–91, 2003.

[35] V. A. Ilyin, A. Abyzov, and C. M. Leslin. Structural alignment of proteins by a novel TOPOFIT method, as a superimposition of common volumes at a topomax point. *Protein Sci*, 13(7):1865–74, 2004.

[36] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5):922–923, 1976.

[37] W. Kabsch. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 34(5):827–828, 1978.

[38] T. Kawabata and K. Nishikawa. Protein structure comparison using the markov transition model of evolution. *Proteins*, 41(1):108–22, 2000.

[39] J. Kervinen, G. J. Tobin, J. Costa, D. S. Waugh, A. Wlodawer, and A. Zdanov. Crystal structure of plant aspartic proteinase prophytepsin: inactivation and vacuolar targeting. *EMBO J*, 18(14):3947–55, 1999.

[40] A. S. Konagurthu, J. C. Whisstock, P. J. Stuckey, and A. M. Lesk. MUSTANG: a multiple structural alignment algorithm. *Proteins*, 64(3):559–74, Aug 2006.

[41] A. Kryshtafovych, M. Milostan, L. Szajkowski, P. Daniluk, and K. Fidelis. CASP6 data processing and automatic evaluation at the protein structure prediction center. *Proteins*, 61 Suppl 7:19–23, 2005.

[42] A. Kryshtafovych, A. Prlic, Z. Dmytriv, P. Daniluk, M. Milostan, V. Eyrich, T. Hubbard, and K. Fidelis. New tools and expanded data analysis capabilities at the Protein Structure Prediction Center. *Proteins*, 69 Suppl 8:19–26, 2007.

[43] E. Liepinsh, M. Andersson, J. M. Ruysschaert, and G. Otting. Saposin fold revealed by the NMR structure of NK-lysin. *Nat Struct Biol*, 4(10):793–5, 1997.

[44] Y. Lindqvist and G. Schneider. Circular permutations of natural protein sequences: structural evidence. *Curr Opin Struct Biol*, 7(3):422–7, 1997.

[45] D. Lipman and W. Pearson. Rapid and sensitive protein similarity searches. *Science*, 227(4693):1435, 1985.

[46] T. Madej, J. F. Gibrat, and S. H. Bryant. Threading a database of protein cores. *Proteins*, 23(3):356–69, 1995.

[47] G. Manning, D. B. Whyte, R. Martinez, T. Hunter, and S. Sudarsanam. The protein kinase complement of the human genome. *Science*, 298(5600):1912–34, Dec 2002.

[48] L. Mavridis and D. Ritchie. 3D-blast: 3D protein structure alignment, comparison, and classification using spherical polar fourier correlations. In *Pacific Symposium on Biocomputing*, volume 2010, pages 281–292, 2010.

[49] G. Mayr, F. S. Domingues, and P. Lackner. Comparative analysis of protein structure alignments. *BMC Struct Biol*, 7:50, 2007.

[50] M. Menke, B. Berger, and L. Cowen. Matt: local flexibility aids protein multiple structure alignment. *PLoS Comput Biol*, 4(1):e10, Jan 2008.

[51] T. S. Motzkin and E. G. Straus. Maxima for graphs and a new proof of a theorem of Turán. *Canadian Journal of Mathematics*, 17:533–540, 1965.

[52] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol*, 247(4):536–40, 1995.

[53] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48(3):443–53, 1970.

[54] H. H. Niemann, M. L. Knetsch, A. Scherer, D. J. Manstein, and F. J. Kull. Crystal structure of a dynamin GTPase domain in both nucleotide-free and GDP-bound forms. *EMBO J*, 20(21):5813–21, 2001.

[55] C. A. Orengo, A. D. Michie, S. Jones, D. T. Jones, M. B. Swindells, and J. M. Thornton. CATH–a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–108, 1997.

[56] C. A. Orengo and W. R. Taylor. SSAP: sequential structure alignment program for protein structure comparison. *Methods Enzymol*, 266:617–35, 1996.

[57] M. Pelillo. Relaxation labeling networks for the maximum clique problem. *J. Artif. Neural Netw.*, 2(4):313–328, 1996.

[58] M. Pelillo and A. Jagota. Feasible and infeasible maxima in a quadratic program for maximum clique. *J. Artif. Neural Netw.*, 2(4):411–420, 1996.

[59] C. P. Ponting and R. B. Russell. Swaposins: circular permutations within genes encoding saposin homologues. *Trends Biochem Sci*, 20(5):179–80, 1995.

[60] D. Sankoff. Matching sequences under deletion-insertion constraints. *Proc Natl Acad Sci U S A*, 69(1):4–6, 1972.

[61] J. M. Sauder, J. W. Arthur, and R. L. Dunbrack, Jr. Large-scale comparison of protein sequence alignment algorithms with structure alignments. *Proteins*, 40(1):6–22, Jul 2000.

[62] M. Shatsky, R. Nussinov, and H. J. Wolfson. FlexProt: alignment of flexible protein structures without a predefinition of hinge regions. *J Comput Biol*, 11(1):83–106, 2004.

[63] M. Shatsky, R. Nussinov, and H. J. Wolfson. A method for simultaneous alignment of multiple protein structures. *Proteins*, 56(1):143–56, Jul 2004.

[64] D. H. Shin, Y. Lou, J. Jancarik, H. Yokota, R. Kim, and S. H. Kim. Crystal structure of YjeQ from Thermotoga maritima contains a circularly permuted GTPase domain. *Proc Natl Acad Sci U S A*, 101(36):13198–203, 2004.

[65] I. N. Shindyalov and P. E. Bourne. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng*, 11(9):739–47, 1998.

[66] H. Strömbergsson, P. Daniluk, A. Kryshtafovych, K. Fidelis, J. E. Wikberg, G. J. Kleywegt, and T. R. Hvidsten. Interaction model based on local protein substructures generalizes to the entire structural enzyme-ligand space. *J Chem Inf Model*, 2008.

[67] J. D. Thompson, F. Plewniak, and O. Poch. A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Res*, 27(13):2682–90, Jul 1999.

[68] C. Vogel and V. Morea. Duplication, divergence and formation of novel protein topologies. *Bioessays*, 28(10):973–8, Oct 2006.

[69] F. Wilcoxon. Comparisons by ranking methods. *Biometric Bulletin*, 1:80–82, 1945.

[70] F. Yang, C. A. Bewley, J. M. Louis, K. R. Gustafson, M. R. Boyd, A. M. Gronenborn, G. M. Clore, and A. Wlodawer. Crystal structure of cyanovirin-N, a potent HIV-inactivating protein, shows unexpected domain swapping. *J Mol Biol*, 288(3):403–12, 1999.

[71] Y. Ye and A. Godzik. Flexible structure alignment by chaining aligned fragment pairs allowing twists. *Bioinformatics*, 19 Suppl 2:ii246–55, 2003.

[72] Y. Ye and A. Godzik. Multiple flexible structure alignment using partial order graphs. *Bioinformatics*, 21(10):2362–9, May 2005.

[73] A. Zemla. LGA: A method for finding 3D similarities in protein structures. *Nucleic Acids Res*, 31(13):3370–4, 2003.