Krzysztof Udycz

Uniwersytetu im. Adama Mickiewicza w Poznaniu
Wydział Biologii

Kierunek studiów: Bioinformatyka **Nr albumu: 394025**

Implementacja metod porównania sekwencji typu „alignment-free”

Implementation of alignment-free sequence comparison methods

Praca licencjacka
wykonana w Zakładzie Biologii Obliczeniowej
Instytutu Biologii Molekularnej i Biotechnologii
Uniwersytetu im. Adama Mickiewicza w Poznaniu
pod kierunkiem dr Andrzeja Zielezińskiego

Poznań, 2016

Spis treści

Streszczenie

Motywacja: Metody porównywania sekwencji w oparciu o ich dopasowanie okazały się fundamentalne w wyjaśnieniu funkcji tysięcy różnych rodzin białkowych lub genowych. Jednak zdarzenia takie jak: rekombinacje, tasowanie egzonów i domen białkowych czy istnienie rodzin genów lub białek wykazujących niski stopień identyczności sekwencji powodują, że metody te bardzo często okazują się nieskuteczne. W rezultacie opracowano alternatywną metodę porównywania sekwencji, która nie wymaga tworzenia dopasowania sekwencji (ang. *alignment-free approach*). Jest to stosunkowo nowe podejście, w związku z czym implementacje tych metod są nadal rzadkością, jednak jeśli już istnieją to: (i) nie są zebrane w jednym programie lub serwisie internetowym, (ii) korzystają z zewnętrznych programów, a (iii) ich obsługa jest wyjątkowo trudna i nieintuicyjna dla przeciętnego użytkownika.

Wyniki: W niniejszej pracy zaimplementowano czternaście metod porównywania sekwencji typu *alignment-free*, które wchodzą w skład internetowego meta-serwisu „Alfree", rozwijanego w Zakładzie Biologii Obliczeniowej Instytutu Biologii Molekularnej i Biotechnologii Uniwersytetu im. Adama Mickiewicza w Poznaniu. Serwis umożliwia obliczenie i wizualizację ewolucyjnych zależności pomiędzy sekwencjami nukleotydów lub białek. Utworzone oprogramowanie służące do porównań sekwencji typu *alignment-free* zostało również udostępnione, w formie skryptów i biblioteki napisanych w języku programowania Python, pod adresem: https://github.com/krzyszto9/alignment-free. Ponadto w ramach tej pracy przeprowadzono ocenę jakości przewidywań utworzonego oprogramowania poprzez porównanie ich z referencyjnym zestawem rodzin białkowych zdeponowanym w bazie struktur białkowych SCOPe, oraz poprzez analizę krzywych ROC. Metody typu *alignment-free*, które uzyskały najwyższą czułość i specyficzność przewidywań to odległość euklidesowa oraz kwadratowa odległość euklidesowa (wartość wskaźnika AUC = 0.666). Natomiast metodą, która otrzymała najniższe wartości obu parametrów jest znormalizowany dystans kompresji (AUC = 0.555). W porównaniu do metod *alignment-free* algorytm Smitha-Watermana uzyskał najlepsze wartości wskaźnika AUC dla trzech spośród czterech analizowanych grup sekwencji. Analiza czasu działania programów wykazała, że zaimplementowane w tej pracy metody są ponad tysiąc razy szybsze niż algorytm Smitha-Watermana.

1. Wstęp

Porównywanie sekwencji jest jednym z zasadniczych elementów analizy bioinformatycznej. Stanowi ono najprostszy sposób poznania struktury, funkcji i drogi ewolucyjnej nieopisanych sekwencji poprzez porównanie ich z sekwencjami istniejącymi już w bazach danych. Najbardziej popularną metodą w tego typu porównaniach jest przyrównanie sekwencji (ang. *sequence alignment*), nazywane często dopasowaniem sekwencji. Zdecydowana większość tych metod opiera się na wyszukaniu w sekwencjach serii pojedynczych znaków lub motywów, które występują w tej samej kolejności w sekwencjach. Przyrównanie sekwencji parami znajduje zastosowanie m.in. w przeszukiwaniu baz danych oraz w kształtowaniu dopasowania wielu sekwencji (ang. *multiple sequence alignment)* [1].

Obecnie algorytmy dopasowania bazują na jednej z dwóch głównych technik: metodzie kropkowej (ang. *dot-matrix)* i metodzie programowania dynamicznego [1, 2]. Pierwszy algorytm polega na wizualnej identyfikacji podobnych fragmentów, jednak dla dużych zestawów danych jest czasochłonny i uniemożliwia szczegółowe uwidocznienie dopasowanych fragmentów. Programowanie dynamiczne jest z kolei wyczerpującym algorytmem otrzymywania optymalnego dopasowania sekwencji poprzez maksymalizację wyniku ścieżki, która go wyprodukowała. Procedura ta przeprowadzana jest w trzech krokach. Tworzona jest dwuwymiarowa macierz sekwencja-sekwencja, a następnie wpisywane są do niej odpowiednie wartości, które zależą od przyjętego systemu punktacji (dopasowanie/niedopasowanie reszt, kary za przerwy). Ostatnim krokiem jest odpowiednie przejście po macierzy (ang. *traceback*) – w kierunku odwrotnym niż kolejność zapisu ocen – w celu znalezienia najlepszego dopasowania.

Wyróżnia się dwa rodzaje przyrównania sekwencji w oparciu o programowanie dynamiczne: dopasowanie globalne [3] i dopasowanie lokalne [4]. Dopasowanie globalne realizowane jest najczęściej przez algorytm Needlemana-Wunscha i wykorzystywane jest przy dopasowaniu sekwencji blisko spokrewnionych o podobnej długości. Natomiast w dopasowaniu lokalnym sekwencje nie są przyrównywane na całej długości, w związku z czym można wykorzystać sekwencje różnej długości pochodzące z daleko spokrewnionych organizmów. Metoda ta zaproponowana przez Smitha-Watermana znajduje lokalne regiony o najwyższym wyniku dopasowania i pomija te odcinki sekwencji, których nie można do siebie dopasować [1].

1.1. Problemy metod wyznaczających dopasowanie sekwencji

Metody porównywania sekwencji oparte o ich dopasowanie (ang. *alignment-based*) wiążą się z czterema głównymi problemami.

Po pierwsze, wszystkie techniki przyrównywania sekwencji zakładają, że spokrewnione sekwencje zawierają mniej lub bardziej zachowane fragmenty ułożone w tej samej kolejności (kolinearność). Innymi słowy, metody te biorą pod uwagę jedynie liniowe ułożenie struktury pierwszorzędowej badanych cząsteczek, czyli sekwencję aminokwasów lub nukleotydów. Nie uwzględniają natomiast one struktur wyższych rzędów, w obrębie których dochodzi do wielu oddziaływań między nukleotydami/aminokwasami znajdującymi się w różnych częściach sekwencji. Dodatkowo przyrównanie sekwencji parami pomija daleko dystansowe podobieństwa pomiędzy sekwencjami powstałe poprzez: rekombinacje z tasowaniem konserwatywnych regionów [5] lub zdarzenia związane z tasowaniem egzonów i domen białkowych (ang. *exon/domain shuffling*), a także (retro)transpozycje i retropozycje. Mechanizmy te powodują mutacje, zmianę liniowego układu sekwencji DNA, a nawet zmianę ilości materiału genetycznego w komórce.

Drugim problemem stosowania metod dopasowania sekwencji jest obciążenie obliczeniowe tych algorytmów, wyrażone jako funkcja potęgowa długości sekwencji. Powoduje ono, że przeszukiwanie dużych baz danych z ich użyciem, staje się niewykonalne [6]. Przykładowo liczba możliwych dopasowań z przerwami dla dwóch sekwencji długości 15 nukleotydów/aminokwasów wynosi ponad 155 milionów.

Trzecie utrudnienie metod dopasowania sekwencji związane jest z koniecznością założenia *a priori* systemu punktacji, który obejmuje wartość punktacji związaną z: dopasowaniem / niedopasowaniem, karą za otwarcie przerwy czy karą za wydłużenie przerwy. Zaproponowano wiele systemów punktowania sekwencji białkowych, między innymi macierze oparte na modelu ewolucyjnym akceptowanych mutacji punktowych PAM [7] oraz matryce BLOSUM oparte na częstościach substytucji aminokwasowych w rodzinach białkowych [8]. Jednak takie podejście heurystyczne sprawia, że trudniej jest ocenić istotność statystyczną uzyskanych wyników, co znacznie utrudnia np. oszacowanie przedziałów ufności dla homologii [6].

Czwarty problem programów dopasowania sekwencji dotyczy spokrewnionych rodzin białkowych, które zachowały bardzo wysokie podobieństwo strukturalne i funkcjonalne

pomimo niemal całkowitej utraty podobieństwa sekwencji. Istnieje wiele spokrewnionych rodzin białkowych lub genowych, których podobieństwo sekwencji mieści się w przedziale od 20% do 30%. Przedział ten nazywany jest "strefą cienia" (ang. *twilight zone*) [1], gdzie następuje pomieszanie sekwencji spokrewnionych jak i niespokrewnionych, których podobieństwo jest całkowicie przypadkowe. Natomiast, gdy stopień identyczności sekwencji spada poniżej 20%, przechodząc do tzw. "strefy ciemności" (ang. *midnight zone*) [1], wiarygodne określenie relacji homologicznych analizowanych sekwencji - korzystając z przeszukań baz danych algorytmami BLAST [9] lub FASTA [10] - jest niemożliwe [30].

Na podstawie opisanych powyżej czterech problemów związanych z metodami przyrównywania sekwencji, konieczne stało się utworzenie alternatywnych strategii porównywania sekwencji biologicznych, które nie wykorzystują dopasowania sekwencji (ang. *alignment-free approach*) [6, 11]. Jest to stosunkowo nowe podejście, w związku z czym implementacje tych metod są rzadkością, jednak jeśli już istnieją to nie są zebrane w jednym programie lub serwisie internetowym, korzystają z zewnętrznych programów i specjalistycznych bibliotek, a ich obsługa jest wyjątkowo trudna i nieintuicyjna dla przeciętnego użytkownika.

1.2. Metody „alignment-free”

Metody *alignment-free* nie posiadają typowych ograniczeń dla metod bazujących na przyrównaniach par sekwencji, m.in.: nie wymagają zachowania konserwatywnych regionów pomiędzy porównywanymi sekwencjami, są mniej wymagające obliczeniowo i pamięciowo, można łatwo określić istotność statystyczną uzyskanych wyników, które nie zależą od żadnych modeli ewolucyjnych. Metody te są mniej wrażliwe na losowe zmiany sekwencji, rekombinacje, transfer genetyczny i sekwencje o różnej długości [12].

Wśród metod *alignment-free* możemy wyróżnić dwie główne kategorie: metody bazujące na częstości występowania słów („k”-krotki, fragmenty sekwencji o danej długości, ang. *l-tuples*) oraz metody niewykorzystujące słów o stałej długości, lecz badające ogólną zawartość informacyjną sekwencji [6]. Mimo, że oba podejścia mają różne podstawy teoretyczne, to są one cały czas rozwijane i znajdują zastosowania w wielu dziedzinach biologii molekularnej i bioinformatyki (podrozdział 1.3.).

1.2.1. Metody oparte na częstości występowania słów w sekwencji

Pierwszym etapem analizy, przy użyciu metod bazujących na częstości występowania słów jest zilustrowanie sekwencji za pomocą wektorów reprezentowanych przez zliczenia lub częstości występowania każdej „k"- krotki o stałej długości. Taka transformacja sekwencji w liczbowy wektor umożliwia zastosowanie narzędzi wykorzystywanych przez algebrę liniową i statystykę [6]. U podstaw tych metod leży fakt, że podobne sekwencje homologiczne składają się z podobnych podsekwencji o mniejszej długości. Najpopularniejsze oraz zaimplementowane metody należące do tej kategorii zostały przedstawione w tabeli 1 (podrozdział 3.1.).

Fakt, że stosujemy słowa o stałej długości może być uznany za odstępstwo od koncepcji metod *alignment-free*, ponieważ jest to równoznaczne z dopasowaniem sekwencji pomiędzy identycznymi segmentami. Jednakże poprzez analizę wszystkich możliwych słów o stałej długości pozbyto się ograniczeń typowych dla metod *alignment-based* [6].

1.2.2. Metody oparte na zawartości informacyjnej sekwencji

Metody oparte na zawartości informacyjnej sekwencji są całkowicie obojętne na założenie, że istnieją regiony konserwatywne. Wynik taki uzyskano stosując dwa alternatywne sposoby. Pierwsze podejście skupia się na reprezentowaniu sekwencji przez nią samą, poprzez użycie funkcji iteracyjnych [6]. Podejście to, nazwane *Chaos Game Representation* [13] zostało po raz pierwszy zastosowane do reprezentacji sekwencji DNA. Następnie rozszerzono je o każdy możliwy alfabet, umożliwiając badanie dowolnej sekwencji i przemianowano na *Universal Sequence Maps* (USM) [14]. Ciekawą cechą UMS jest zdolność reprezentowania i podsumowania każdej sekwencji w wielowymiarowej przestrzeni. Natomiast drugie podejście używa kompresji sekwencji, jako narzędzia do mierzenia złożoności sekwencji. Opiera się ono na założeniach teorii informacji, w szczególności na teorii złożoności Kołmogorowa [6]. Najpopularniejsze oraz zaimplementowane metody należące do tej kategorii zostały przedstawione w tabeli 2 (podrozdział 3.1.).

1.3. Uzasadnienie wyboru tematu

Biorąc pod uwagę fakt, że metody *alignment-free* są stosunkowo nowym podejściem to już znalazły zastosowanie w wielu dziedzinach biologii molekularnej i bioinformatyki, m.in.: w filogenetyce molekularnej [15], przy wykrywaniu rekombinacji oraz zmienności pomiędzy sekwencjami białkowymi lub nukleotydowymi [16], w analizie danych otrzymanych z sekwencjonowania nowej generacji (ang. *next-generation sequencing*) [17]. Ponadto są z powodzeniem wykorzystywane w metagenomice [17], epigenetyce [18] oraz genetyce populacyjnej [16].

Metody *alignment-free* zostały pomyślnie zastosowane w wielu analizach filogenetycznych na podstawie sekwencji całych genomów. Przykładem takiego zastosowania jest wykorzystanie siedmiu różnych metod *alignment-free* do obliczenia dystansów, a następnie wizualizacji drzew filogenetycznych gatunków pomiędzy mitochondrialnymi genomami naczelnych. Uzyskane topologie drzew filogenetycznych w dużej mierze zgadzały się z wcześniej wygenerowanymi topologiami przy pomocy oprogramowania ClustalW. Jedynie w przypadku dwóch z siedmiu metod uzyskano różniące się topologie, co autorzy badania uzasadniali wrażliwością na dane wejściowe [15].

W innym teście zastosowano metodę *alignment-free* wykorzystującą najkrótsze unikalne podsekwencje do wykrywania rekombinacji w pięćdziesięciu ośmiu genomach bakterii *Escherichia coli*. Najsilniejszy sygnał rekombinacyjny otrzymano od genomu zawierającego gen (o długości 125 kb) będący wynikiem horyzontalnego transferu genów [16].

Podjęcie tematyki związanej z metodami typu *alignment-free* wydało mi się ciekawym wyzwaniem ze względu na stosunkowo niewielką liczbę ogólnodostępnych publikacji na ten temat oraz z uwagi na fakt, że istniejące, nieliczne implementacje tych metod są wyjątkowo trudne w obsłudze, nieintuicyjne, korzystające z zewnętrznych programów oraz rozproszone w wielu miejscach w sieci. Brak jest jednego miejsca, skryptu czy biblioteki, które zebrałoby wszystkie te metody oraz udostępniłby do użytku w sposób przyjazny dla użytkownika nieposiadającego specjalistycznej wiedzy na dany temat. Implementacja tych metod oraz użycie ich do utworzenia serwisu internetowego oferuje użytkownikom atrakcyjną alternatywę dla tradycyjnych metod opartych o dopasowanie sekwencji.

2. Cel pracy

Celem niniejszej pracy licencjackiej jest implementacja popularnych metod porównywania sekwencji typu *alignment-free* w formie intuicyjnych dla przeciętnego użytkownika, ogólnodostępnych skryptów i modułów języka programowania Python.

Dodatkowym celem mojej pracy licencjackiej jest ocena jakości wyników utworzonego oprogramowania poprzez ich porównanie z referencyjnym zestawem rodzin białkowych zdeponowanym w bazie struktur białkowych SCOPe.

3. Materiały i metody

3.1. Implementacja metod typu „alignment-free”

Wszystkie metody porównywania sekwencji typu *alignment-free* zostały zaimplementowane z użyciem języka programowania Python 2.7.6 (https://www.python.org/) wraz z naukowym pakietem NumPy v1.8.2 (http://www.numpy.scipy.org) służącym do prowadzania zaawansowanych obliczeń matematycznych. Jedna z zaimplementowanych w tej pracy metod - znormalizowany dystans kompresji (ang. *normalized compression distance*) - używa modułu lzma v0.5.3, który jest wykorzystywany do bezstratnej kompresji danych. Skrypty mogą zostać uruchomione pod systemem operacyjnym Windows, Mac OS X oraz Linux z zainstalowanym interpreterem języka Python. W tabeli 1 i 2 przedstawiono zaimplementowane w tej pracy metody oparte na częstości występowania słów w sekwencji (podrozdział 1.2.1.) oraz metody oparte na zawartości informacyjnej sekwencji (podrozdział 1.2.2.).

Dystanse d pomiędzy sekwencjami X i Y są wyrażone wzorami, gdzie:

- $c_{L,i}^X = (c_{L,1}^X, \dots, c_{L,K}^X)$, $c_{L,i}^Y = (c_{L,1}^Y, \dots, c_{L,K}^Y)$ - wektory reprezentujące zliczenia słów;
- $f_{L,i}^X = (f_{L,1}^X, , \dots, f_{L,K}^X)$, $f_{L,i}^Y = (f_{L,1}^Y, , \dots, f_{L,K}^Y)$ - wektory reprezentujące częstości występowania słów;
- W_K – zbiór wszystkich możliwych „k”-krotek zawierający K elementów;
- K - liczba różnych „k”-krotek o długości od L do n;
- L, n – długości słów („k”-krotek);
- s - macierz kowariancji pomiędzy odpowiednimi wektorami sekwencji X i Y;
- e – wykładnik potęgowy;
- ρ_i - waga odpowiadająca i-temu słowu;
- $\cos\theta_{XY} = \sum_{i=1}^{K} c_{L,i}^X * c_{L,i}^Y / [\sqrt[2]{\sum_{i=1}^{K}(c_{L,i}^X)^2} * \sqrt[2]{\sum_{j=1}^{K}(c_{L,j}^Y)^2}]$;
- $K(\ )$ - rozmiar danej sekwencji po kompresji;
- a i b - reprezentacja dwóch dowolnych symboli sekwencji X i Y w formie współrzędnych *USM*.

Przykład:

Dla sekwencji X = ATGTGTT, gdzie długość słowa wynosi trzy (L = 3), istnieją cztery (K = 4) różne „k”-krotki. Wektory W_K, $c_{L,i}^X$ oraz $f_{L,i}^X$ przedstawione są jako:

- $W_4 = \{\texttt{ATG, TGT, GTG, GTT}\}$;
- $c_3^X = (1;\ 2;\ 1;\ 1)$;
- $f_3^X = (0{,}2;\ 0{,}4;\ 0{,}2;\ 0{,}2)$.

Tabela 1. Zestawienie zaimplementowanych metod bazujących na częstości występowania słów w sekwencji.

Pełna nazwa metody	**Nazwa funkcji w utworzonym oprogramowaniu**	**Wzór**	**Referencja literaturowa**
Odległość Euklidesowa (ang. *eucliudean distance*)	euclidean	$d_L^E(X,Y) = \sqrt[2]{\sum_{i=1}^{K}(c_{L,i}^X - c_{L,i}^Y)^2}$	[6]
Kwadratowa odległość euklidesowa (ang. *squared eucliudean distance*)	squaredeuclidean	$d_L^{E^2}(X,Y) = \sum_{i=1}^{K}(c_{L,i}^X - c_{L,i}^Y)^2$	[6, 19, 20]
Standaryzowana odległość Euklidesowa (ang. *standardized eucliudean distance*)	seuclidean	$d_L^{SE}(X,Y) = \sum_{i=1}^{K}\frac{(c_{L,i}^X - c_{L,i}^Y)^2}{s_{ij}}$	[21]
Standaryzowana odległość Euklidesowa dla słów o różnej długości (ang. *standardized eucliudean distance with different resolutions)*	r_seuclidean	$d^{SE*} = \sum_{L=l}^{n} d_L^{SE}$	[21]
Ważona odległość Euklidesowa (ang. *weighted eucliudean distance*)	weuclidean	$d^2(X,Y) = \sum_{i=1}^{K}\rho_i(c_{L,i}^X - c_{L,i}^Y)^2$	[22]
Ważona odległość Euklidesowa dla słów o różnej długości (ang. *weighted eucliudean distance with different resolutions)*	r_weuclidean	$d^{2*} = \sum_{L=l}^{n} d_L^2$	[6]
Odległość Minkowskiego (ang. *minkowski distance*)	minkowski	$d_L^M(X,Y) = \sqrt[e]{\sum_{i=1}^{K}(c_{L,i}^X - c_{L,i}^Y)^e}$	

11:1803195269

Współczynnik korelacji Pearsona (ang. *Pearson product-moment correlation coefficient*)	lcc	$d_L^{LCC}(X,Y) = \left[K\sum_{i=1}^{K} f_{L,i}^{X} * f_{L,i}^{Y} - \sum_{i=1}^{K} f_{L,i}^{X} * \sum_{i=1}^{K} f_{L,i}^{Y}\right] / \left[\left[K\sum_{i=1}^{K}\left(f_{L,i}^{X}\right)^2 - \left(\sum_{i=1}^{K} f_{L,i}^{X}\right)^2\right]^{\frac{1}{2}} * \left[K\sum_{i=1}^{K}\left(f_{L,i}^{Y}\right)^2 - \left(\sum_{i=1}^{K} f_{L,i}^{Y}\right)^2\right]^{\frac{1}{2}}\right]$	[6]
Dywergencja Kullbacka-Leiblera (ang. *Kullback–Leibler divergence*)	kl	$d_L^{KL}(X,Y) = \sum_{i=1}^{K} f_{L,i}^{X} * log_2\left(\frac{f_{L,i}^{X}}{f_{L,i}^{Y}}\right)$	[23]
Odległość kosinusowa (ang. *cosine distance*)	cosine	$d_L^{Cos}(X,Y) = 1 - \cos\theta_{XY}$	[24, 25]
Odległość ewolucyjna (ang. *evolutionary distance*)	evol	$d_L^{Evol}(X,Y) = -\ln[(1+\cos\theta_{XY})/2]$	[24, 25]

Wszystkie wymienione wyżej metody mogą posłużyć do obliczenia dystansów ewolucyjnych pomiędzy sekwencjami aminokwasowymi lub nukleotydowymi. Warto zwrócić uwagę, że dywergencja Kullbacka-Leiblera - jako jedyna metoda - wymaga zastosowania wektorów, które reprezentują częstości występowania słów w sekwencji. Natomiast pozostałe metody mogą korzystać z dwóch typów wektorów.

Tabela 2. Zestawienie zaimplementowanych metod opartych na zawartości informacyjnej sekwencji.

Pełna nazwa metody	**Nazwa funkcji w utworzonym oprogramowaniu**	**Wzór**	**Referencja literaturowa**
Uniwersalne Mapy Sekwencji, USM (ang. *Universal Sequence Maps*)	usm	d^{USM} (a, b) = $-log_2(max\lvert a_i - b_i\rvert)$	[14]
Znormalizowany dystans kompresji (ang. *normalized compression distance*)	ncd_zlib, ncd_lzma	d^{NCD} (X, Y) = $\frac{K(X) + K(Y)}{K(XY)}$	[26]

Podobnie jak w przypadku metod opartych na częstości występowania słów w sekwencji, wszystkie wymienione wyżej metody mogą posłużyć do obliczenia dystansów ewolucyjnych pomiędzy każdym typem sekwencji.

3.2. Sekwencje źródłowe

Sekwencje źródłowe, które posłużyły do oceny jakości zaimplementowanych metod, pochodzą z bazy danych SCOPe (ang. *Structural Classification of Proteins – extended, http://scop.berkeley.edu*) w wersji 2.06 zawierającej 77 439 rekordów [27]. Baza SCOP (ang. *Structural Classification of Proteins*) - rozszerzona przez bazę SCOPe - zawiera wyniki klasyfikacji białek (dla których znane są struktury przestrzenne) przeprowadzonej na podstawie badania zależności ewolucyjnych i strukturalnych [28]. Białka znajdujące się w tej bazie grupowane są hierarchicznie ze względu na rodziny (ang. *family*), nadrodziny (ang. *superfamily*), zwoje (ang. *fold*) i klasy (ang. *class*). Granice między różnymi kategoriami w hierarchii mogą się zacierać, jednak charakterystyczną cechą przyjętego systemu klasyfikacji jest fakt, że im wyższy poziom w hierarchii, tym bardziej wyraźne podobieństwo strukturalne między białkami w tej samej grupie:

- Rodziny – zawierają białka związane ze sobą czytelnymi relacjami ewolucyjnymi o identyczności sekwencji większej lub równej 30%;
- Nadrodziny – obejmują grupy białek o niskiej identyczności sekwencji, których wspólne pochodzenie ewolucyjne przekłada się na podobieństwo w strukturze i funkcji;
- Zwoje – zawierają nadrodziny o jednakowym rdzeniu strukturalnym (tj. grupy białek, w których główne elementy struktury drugorzędowej są rozmieszczone i połączone ze sobą w taki sam sposób);
- Klasy – zawierają zwoje z podobnym rdzeniem.

Baza danych ASTRAL dostarcza narzędzi służących do analizy struktur i sekwencji białkowych zawartych w bazie SCOPe (m.in. możliwość filtrowania zbiorów sekwencji, gdzie dwa wybrane białka mają mniej niż wybrany procent identyczności pomiędzy sobą). Sekwencje źródłowe użyte w niniejszej pracy to zbiór referencyjnych sekwencji bazy ASTRAL zawierający wszystkie możliwe białka, które mają mniej niż 40% identyczności pomiędzy sobą (ASTRAL40). Znając ich powiązania ewolucyjne stały się jednym z najwiarygodniejszych źródeł testowania i oceny jakości wyników metod wykrywających relacje między daleko spokrewnionymi sekwencjami.

Na wzór już przeprowadzanych badań [29] ograniczono oryginalny zbiór danych bazy ASTRAL40 poprzez wykluczenie sekwencji zawierających nieznane aminokwasy oraz wyłączenie rodzin, które zawierały mniej niż 5 sekwencji białkowych. Dodatkowo analizie zostały poddane tylko cztery główne klasy bazy danych SCOPe:

- klasa all-α – białka, których struktura jest zasadniczo utworzona przez a-helisy;
- klasa all-β – ich struktura jest zasadniczo utworzona przez beta-kartki;
- klasa α/β – zawierają a-helisy i beta-kartki;
- klasa α + β –białka zawierające a-helisy i beta-kartki, które są w dużym stopniu rozdzielone.

W tabeli 3 przedstawiono podsumowanie oryginalnego (ASTRAL40) i zredukowanego (ASTRAL40b) zbioru sekwencji źródłowych.

Tabela 3. Dla każdego zbioru sekwencji oraz każdej klasy (cl) przedstawiono liczbę sekwencji (sn), zwojów (cf), nadrodzin (sf) oraz rodzin (fa).

Klasa	all-α				all-β			
Poziom	sn	cf	sf	fa	sn	cf	sf	fa
Oryginalny zbiór	2439	289	513	984	2795	117	365	890
Zredukowany zbiór	1074	46	57	89	1469	44	62	112
Klasa	α/β				α + β			
Poziom	sn	cf	sf	fa	sn	cf	sf	fa
Oryginalny zbiór	3970	149	247	940	3346	385	561	1217
Przycięty zbiór	2490	59	75	159	1577	70	88	144

3.3. Ewaluacja metod typu „alignment-free”

W celu oceny jakości działania programów typu „alignment-free” posłużono się analizą krzywych ROC (ang. *Receiver operating characteristic)*. Krzywe te służą do graficznego przedstawienia jakości klasyfikacji oraz pokazują dla różnych wartości progowych, zależności pomiędzy wskaźnikami TPR (czułość, ang. *True Positive Rate*) oraz FPR (1 – specyficzność, ang. *False Positive Rate*). Wskaźniki te wyrażone są wzorami:

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

gdzie:

- TP (ang. *true positive*) – liczb par sekwencji, które zostały poprawnie zaklasyfikowane jako spokrewnione;
- FP (ang. *false positive*) - liczb par sekwencji, które zostały zaklasyfikowane jako spokrewnione, lecz według bazy SCOPe nie są spokrewnione;
- TN (ang. *true negative*) – liczb par sekwencji, które zostały poprawnie zaklasyfikowane jako niespokrewnione;
- FN (ang. *false negative*) - liczb par sekwencji, które zostały zaklasyfikowane jako niespokrewnione, lecz według bazy SCOPe są spokrewnione.

Jakość klasyfikacji została określona przy użyciu wskaźnika AUC (ang. *Area Under ROC Curve*, pole powierzchni pod krzywą ROC). Wskaźnik AUC przyjmuje wartości z przedziału <0;1>, gdzie wartość:

- 1 - oznacza, że metoda zwraca przewidywania identyczne z wynikami bazy danych SCOPe;
- 0,5 – oznacza, że metoda zwraca przewidywania w 50% zgodne z wynikami bazy danych SCOPe;
- > 0,5 – oznacza, że metoda zwraca przewidywania w ponad 50% zgodne z wynikami bazy danych SCOPe.

Dla wybranych metod obliczono dystanse ewolucyjne pomiędzy każdą parą sekwencji białkowych zawartych w zestawie ASTRAL40b. Dodatkowo każdej parze sekwencji przypisano odpowiednią wartość, gdzie:

- 1 - oznacza, że sekwencje dzielą ze sobą tę samą grupę;
- 0 - oznacza, że sekwencje nie dzielą ze sobą tej samej grupy.

Grupa odnosi się do jednego z poziomów hierarchii bazy danych SCOP – może to być rodzina, nadrodzina, zwój lub klasa.

Do obliczenia krzywych ROC oraz wartości wskaźnika AUC wykorzystano pakiet ROCR (https://rocr.bioinf.mpi-sb.mpg.de/ROCR.pdf) języka programowania R (https://www.r-project.org/).

4. Wyniki i dyskusja

4.1. Programy *alignment-free*

Utworzone w tej pracy programy *alignment-free* dostępne są w postaci dwóch skryptów języka programowania Python. Pierwszy skrypt *ltuple.py* (w dalszej części oznaczany numerem „1") zawiera jedenaście zaimplementowanych metod porównywania sekwencji, które bazują na „k"-krotkach (podrozdział 1.2.1.). Natomiast drugi skrypt *infotheory.py* (w dalszej części oznaczany numerem „2") implementuje trzy metody badające ogólną zawartość informacyjną sekwencji (podrozdział 1.2.2.). Zadaniem skryptów jest obliczenie dystansów ewolucyjnych pomiędzy sekwencjami białek lub kwasów nukleinowych. Otrzymane macierze dystansów mogą z kolei zostać bezpośrednio wczytane przez większość programów filogenetycznych (np. PHYLIP [31], MEGA [32]) w celu utworzenia drzewa filogenetycznego.

4.1.1. Wersja konsolowa

4.1.1.1. Parametry wejściowe

Oba skrypty posiadają następujące, wymagane parametry wejściowe (Tabela 4).

Tabela 4. Opis wymaganych parametrów wejściowych programów *alignment-free*.

Nazwa	Skrót	Argument	Skrypt	Opis
--file	-f	Nazwa pliku	1, 2	Plik zawierający sekwencje biologiczne przeznaczone do analizy.
--method	-m	Nazwa metod(y)	1, 2	Nazwa jednej lub wielu funkcji (Tabela 1), które są oddzielone spacjami. Istnieje możliwość wyboru wszystkich zaimplementowanych metod poprzez wpisanie jako argumentu słowa „all".
--length	-l	Długość słowa	1	Długość słowa, dla którego zostaną obliczone dystanse ewolucyjne pomiędzy sekwencjami.

Skrypt 1 posiada następujące, wymagane przez niektóre metody, parametry wejściowe (Tabela 5).

Tabela 5. Opis, wymaganych przez niektóre metody, parametrów wejściowych programów *alignment-free.*

Nazwa	Skrót	Argument	Skrypt	Opis
--maximum_length	-ml	Maksymalna długość słowa	1	Maksymalna długość słów, które posłużą do obliczania dystansów ewolucyjnych. Wartość parametru musi być większa niż parametr --length/-l. Wymagany przy używaniu metod bazujących na słowach o różnej długości.
--type	-t	Rodzaj sekwencji	1	Rodzaj analizowanej sekwencji. Dostępne opcje to: „p", „prot", „n", „nucl". Wymagany przy używaniu metod bazujących na ważonym dystansie euklidesowym.
--exponent	-e	Wykładnik potęgowy	1	Wykładnik potęgowy, parametr wymagany przy obliczaniu odległości Minkowskiego. Domyślnie przyjmuje wartość 2.

Oba skrypty posiadają następujące, opcjonalne parametry wejściowe (Tabela 5).

Tabela 6. Opis opcjonalnych parametrów wejściowych programów *alignment-free.*

Nazwa	Skrót	Argument	Skrypt	Opis
--calculations	-c	Typ wektora	1	Każda metoda korzysta z wektorów domyślnych dla niej (Tabela 1). Ustawiając parametr wejściowy --calculations/-c określamy typ wektorów, które posłużą do wszystkich obliczeń. Dostępne opcje to: „o", „occurrences", „f", „frequencies".
--help	-h	-	1, 2	Wypisanie na ekran instrukcji obsługi skryptu wraz z opisem wszystkich dostępnych opcji i metod.
--output	-o	- / Nazwa pliku	1, 2	Zapisanie wyniku do pliku w formacie macierzy dystansów (Rycina 2). Domyślnie plik zapisywany jest pod nazwą w formacie: "NazwaPlikuWejsciowego_NazwaMetody_Typ Wektora.mat".
--pairwise	-pw	-	1, 2	Zapisanie wyniku do pliku w postaci listy dystansów pomiędzy wszystkimi możliwymi parami sekwencji (Rycina 4). Opcja dostępna jedynie z opcją --output. Do nazwy pliku wynikowego dodawany jest człon „_pw".
--quiet	-q	-	1, 2	Brak wypisywania informacji na ekran. Opcja dostępna jedynie z opcją --output.

4.1.1.2. Przykłady użycia utworzonego oprogramowania w formie skryptów

Poniżej przedstawiono trzy przykłady użycia utworzonego oprogramowania.

Przykład 1:

```
python ltuple.py --file primates.fa --length 5 --method squaredeuclidean
```

Opis słowny: Obliczono dystanse ewolucyjne pomiędzy wszystkimi sekwencjami nukleotydowymi znajdującymi się w pliku *primates.fa*. Plik ten zawiera genomy mitochondrialne ośmiu współczesnych ludzi, człowieka neandertalskiego oraz trójki naczelnych (szympans karłowaty, zwyczajny i goryl). Do obliczeń wykorzystano kwadratową odległość euklidesową, gdzie długość słowa wynosi 5 nukleotydów. Powstałą macierz dystansów wyświetlono na ekran (Rycina 1).

Wynik:

```
squared eucliudean distance
# Neanderthal Bonobo Chimp Gorilla Human_African_Youruba Human_North_American_Indian
Human_African_Mbenzele Human_China Human_Germany Human_England Human_Asian_Indian Human_Aborigine
 [[    0.  12430.  12292.  12657.   1658.   1800.   1886.   1841.   1783.   1801.   1713.   1875.]
 [ 12430.      0.   6298.  13061.  12354.  12450.  13032.  12949.  12857.   12767.  12709.  12705.]
 [ 12292.   6298.      0.  13393.  12172.  12276.  12682.  12225.  12251.   12225.  12295.  12071.]
 [ 12657.  13061.  13393.      0.  12435.  12669.  12607.  12756.  12920.   12762.  12580.  12676.]
 [  1658.  12354.  12172.  12435.      0.    392.    752.    509.    373.     355.    347.    425.]
 [  1800.  12450.  12276.  12669.    392.      0.    776.    483.    349.     281.    447.    519.]
 [  1886.  13032.  12682.  12607.    752.    776.      0.    873.    739.     667.    711.    793.]
 [  1841.  12949.  12225.  12756.    509.    483.    873.      0.    310.     342.    408.    524.]
 [  1783.  12857.  12251.  12920.    373.    349.    739.    310.      0.     184.    316.    380.]
 [  1801.  12767.  12225.  12762.    355.    281.    667.    342.    184.       0.    272.    352.]
 [  1713.  12709.  12295.  12580.    347.    447.    711.    408.    316.     272.      0.    360.]
 [  1875.  12705.  12071.  12676.    425.    519.    793.    524.    380.     352.    360.      0.]]
```

Rycina 1. Wynik przykładowego użycia skryptu *ltuple.py,* przedstawiający uzyskaną macierz dystansów ewolucyjnych.

Interpretacja: Im uzyskany dystans jest większy, tym sekwencje są od siebie bardziej oddalone ewolucyjnie (podobieństwo sekwencji spada). Wykluczając pary identycznych sekwencji, gdzie dystans ewolucyjny wynosi 0, najbliżej spokrewnionymi są sekwencje o identyfikatorach *Human_England* oraz *Human_Germany* – zostały one wyizolowane od współczesnych ludzi żyjących na terenie Anglii i Niemiec. Dystans wyniósł 184 jednostki. Natomiast sekwencjami najbardziej oddalonymi ewolucyjnie od siebie są te o identyfikatorach *Chimp* oraz *Gorilla* – pochodzą odpowiednio od szympansa zwyczajnego oraz goryla. W tym przypadku dystans wyniósł 13393 jednostek. Różne rasy człowieka należące do gatunku *Homo Sapiens* wykazują bardzo duże podobieństwo sekwencji, co prawdopodobnie spowodowane jest stosunkowo niedawnym czasem pojawienia się tego gatunku (około 190 tys. lat temu), w związku z czym procesy prowadzące do zmienności sekwencji nie zaszły w takim stopniu, jak w przypadku pozostałych naczelnych. Otrzymana macierz dystansów została dodatkowo wykorzystana przez meta-serwis „Alfree" do utworzenia drzewa filogenetycznego (podrozdział 4.1.3., Rycina 8).

Przykład 2:

```
python ltuple.py -f primates.fa -l 2 –m minkowski euclidean –e 3 –o --quiet
```

Opis słowny: Obliczono dystanse ewolucyjne pomiędzy wszystkimi sekwencjami nukleotydowymi znajdującymi się w pliku *primates.fa*. Do obliczeń wykorzystano odległość euklidesową oraz odległość Minkowskiego, gdzie wykładnik potęgowy równa się 3. W obu przypadkach długość słowa wynosi 3 nukleotydy. Powstałe macierz dystansów zapisano do plików *primates_euclidean_o.fa* oraz *primates_minkowski_o.mat* (Rycina 2). Na ekranie nie wyświetlono żadnych informacji.

Wynik:

```
# Neanderthal Bonobo Chimp Gorilla Human_African_Youruba  …
0.0000000000 70.0433744786 61.2941095566 56.9435164279 16.5438076766 …
70.0433744786 0.0000000000 32.8057148480 61.4693702544 73.1141019140 …
61.2941095566 32.8057148480 0.0000000000 57.0917778664 64.1772430609 …
56.9435164279 61.4693702544 57.0917778664 0.0000000000 57.2413079613 …
16.5438076766 73.1141019140 64.1772430609 57.2413079613 0.0000000000 …
```

Rycina 2. Fragment pliku wynikowego *primates_minkowski_o.mat.* Pierwszy wiersz zawiera identyfikatory sekwencji. Kolejne wiersze przedstawiają uzyskaną macierz dystansów ewolucyjnych pomiędzy sekwencjami.

Przykład 3:

```
python infotheory.py -f primates.fa –m ncd_zlib –o -pw
```

Opis słowny: Obliczono dystanse ewolucyjne pomiędzy wszystkimi sekwencjami nukleotydowymi znajdującymi się w pliku *primates.fa*. Do obliczeń wykorzystano znormalizowany dystans kompresji wykorzystujący zlib, jako algorytm kompresji. Powstałą macierz dystansów wyświetlono na ekran (Rycina 3). Wynik zapisano do pliku *primates_ncd_zlib_pw.mat,* w postaci listy dystansów pomiędzy wszystkimi możliwymi parami sekwencji (Rycina 4).

Wynik:

```
normalized compression distance using zlib
# Neanderthal Bonobo Chimp Gorilla Human_African_Youruba …
[[ 0.03877088  0.63516574  0.64044481  0.71210559  0.1592173   … ]
 [ 0.63681285  0.03850113  0.38768787  0.6998147   0.64237183  … ]
 [ 0.64126853  0.38933498  0.03871499  0.70819605  0.64497529  … ]
 [ 0.70942462  0.69899115  0.70304778  0.03912233  0.70854789  … ]
 [ 0.15942327  0.6376364   0.64085667  0.71452111  0.03892894  … ]
 …                                                               ]
File primates_ncd_zlib_pw.mat has been saved successfully
```

Rycina 3. Fragment wyniku przykładowego użycia skryptu *infotheory.py.*

```
#Neanderthal Bonobo Chimp Gorilla Human_African_Youruba Human_North_
American Indian Human African Mbenzele Human China Human Germany
Human_England Human_Asian_Indian Human_Aborigine
1 1   0.0387708806
1 2   0.6351657402
1 3   0.6404448105
1 4   0.7121055888
1 5   0.1592173018
1 6   0.1629919637
1 7   0.1662898253
1 8   0.1667353669
1 9   0.1610378913
1 10  0.1588708016
1 11  0.1579055865
1 12  0.1639175258
2 1   0.6368128474
```

Rycina 4. Fragment pliku wynikowego *primates_ncd_zlib_pw.mat.* Pierwszy wiersz zawiera identyfikatory sekwencji. Kolejne wiersze zawierają uzyskane dystanse (kolumna 3) pomiędzy odpowiednimi parami sekwencji (kolumna 1 i 2).

4.1.2. Moduły programów *alignment-free*

W celu bardziej zaawansowanego użycia zaimplementowanych metod oraz zapewnienia możliwości wykorzystania ich w ramach własnych projektów, aplikacje zostały napisane w formie modułów języka Python. W tym celu w kodzie języka Python należy zaimportować odpowiedni plik zawierający definicje funkcji oraz ich instrukcje (przykładowe użycie znajduje się w podrozdziale 4.1.2.2.).

4.1.2.1. Funkcje programów *alignment-free*

Najważniejsze funkcje zdefiniowane przez moduły *ltuple* oraz *infotheory* (Tabela 7, 8).

Tabela 7. Opis funkcji zdefiniowanych w module *ltuple*.

Definicja	Argumenty	Typ zwracanego obiektu	Opis
calculate_occurrences(length, input_file)	Długość słowa, nazwa pliku	numpy.ndarray	Zwraca tablicę wektorów, które reprezentują zliczenia słów każdej sekwencji znajdującej się w pliku.
calculate_frequencies(occurrences_list, seqs_number)	Tablica wektorów reprezentująca zliczenia słów, liczba sekwencji	numpy.ndarray	Zwraca tablicę wektorów, które reprezentują częstości występowania słów w każdej sekwencji. Jest ona obliczana na podstawie tablicy wektorów reprezentujących zliczenia słów.

calculate_weights(type_, seqs_number,input_file, length)	Rodzaj sekwencji, liczba sekwencji, nazwa pliku, długość słowa	numpy.ndarray	Zwraca tablicę wektorów, które reprezentują wagi słów. Obliczane są dla odpowiedniego typu sekwencji znajdujących się w pliku wejściowym.
squaredeuclidean(list_, seqs_number)	Tablica wektorów, liczba sekwencji	numpy.ndarray	Zwraca macierz dystansów ewolucyjnych pomiędzy sekwencjami, która jest obliczana za pomocą kwadratowej odległości euklidesowej, na podstawie danych zawartych w wektorach.
euclidean(list_, seqs_number)	Tablica wektorów, liczba sekwencji	numpy.ndarray	Zwraca macierz dystansów ewolucyjnych pomiędzy sekwencjami, która jest obliczana za pomocą odległości euklidesowej, na podstawie danych zawartych w wektorach.
seuclidean(list_, seqs_number)	Tablica wektorów, liczba sekwencji	numpy.ndarray	Zwraca macierz dystansów ewolucyjnych pomiędzy sekwencjami, która jest obliczana za pomocą standaryzowanej odległości euklidesowej, na podstawie danych zawartych w wektorach.
weuclidean(list_,w_list, seqs_number)	Tablica wektorów, liczba sekwencji	numpy.ndarray	Zwraca macierz dystansów ewolucyjnych pomiędzy sekwencjami, która jest obliczana za pomocą ważonej odległości euklidesowej, na podstawie danych zawartych w wektorach.
r_seuclidean(length, maximum_length, input_file,seqs_number, string)	Długość słowa, maksymalna długość słowa, nazwa pliku, liczba sekwencji, rodzaj wektorów	numpy.ndarray	Zwraca macierz dystansów ewolucyjnych pomiędzy sekwencjami, która jest obliczana za pomocą standaryzowanej odległości euklidesowej dla słów o różnej długości, na podstawie danych zawartych w wektorach.
r_weuclidean(length, maximum_length, input_file, seqs_number, type_,string)	Długość słowa, maksymalna długość słowa, nazwa pliku, liczba sekwencji, typ sekwencji, rodzaj wektorów	numpy.ndarray	Zwraca macierz dystansów ewolucyjnych pomiędzy sekwencjami, która jest obliczana za pomocą ważonej odległości euklidesowej dla słów o różnej długości, na podstawie danych zawartych w wektorach.
minkowski(list_, seqs_number, exponent)	Tablica wektorów, liczba sekwencji, wykładnik potęgowy	numpy.ndarray	Zwraca macierz dystansów ewolucyjnych pomiędzy sekwencjami, która jest obliczana za pomocą odległości Minkowskiego, na podstawie danych zawartych w wektorach.
lcc(list_, seqs_number)	Tablica wektorów, liczba sekwencji	numpy.ndarray	Zwraca macierz dystansów ewolucyjnych pomiędzy sekwencjami, która jest obliczana za pomocą współczynnika korelacji Pearsona, na podstawie danych zawartych w wektorach.
kl(list_, seqs_number)	Tablica wektorów, liczba sekwencji	numpy.ndarray	Zwraca macierz dystansów ewolucyjnych pomiędzy sekwencjami, która jest obliczana za pomocą dywergencji Kullbacka-Leiblera, na podstawie danych zawartych w wektorach. W związku z tym, że metoda ta zwraca wartości z przedziału <-1; 1>, a dystans pomiędzy sekwencjami nie może być ujemny, to zaimplementowana wersja zwraca wynik znormalizowany z przedziału <0, 1>.

cosine(list_, seqs_number)	Tablica wektorów, liczba sekwencji	numpy.ndarray	Zwraca macierz dystansów ewolucyjnych pomiędzy sekwencjami, która jest obliczana za pomocą odległości kosinusowej, na podstawie danych zawartych w wektorach.
evol(list_, seqs_number)	Tablica wektorów, liczba sekwencji	numpy.ndarray	Zwraca macierz dystansów ewolucyjnych pomiędzy sekwencjami, która jest obliczana za pomocą odległości ewolucyjnej, na podstawie danych zawartych w wektorach.

Tabela 8. Opis funkcji zdefiniowanych w module *infotheory*.

Definicja	**Argumenty**	**Typ zwracanego obiektu**	**Opis**
get_sequences(input_file)	Nazwa pliku	lista	Zwraca listę sekwencji, które znajdują się w pliku wejściowym.
ncd_lzma(seqs_list)	Lista sekwencji	numpy.ndarray	Zwraca macierz dystansów ewolucyjnych pomiędzy sekwencjami, która jest obliczana za pomocą metody ncd. Do kompresji danych wykorzystywany jest algorytm lzma (ang. Lempel–Ziv–Markov chain algorithm).
ncd_zlib(seqs_list)	Lista sekwencji	numpy.ndarray	Zwraca macierz dystansów ewolucyjnych pomiędzy sekwencjami, która jest obliczana za pomocą metody ncd. Do kompresji danych wykorzystywana jest biblioteka zlib.
usm(seqs_list)	Lista sekwencji	numpy.ndarray	Zwraca macierz dystansów ewolucyjnych pomiędzy sekwencjami, która jest obliczana za pomocą metody usm. Dystans pomiędzy dwoma sekwencjami jest średnią arytmetyczną dystansów uzyskanych pomiędzy każdą parą aminokwasów / nukleotydów różnych sekwencji.

Wynik większości funkcji zwracany jest jako obiekt pakietu NumPy – numpy.ndarray. Reprezentuje on wielowymiarową tablicę elementów o stałym rozmiarze.

4.1.2.2. Przykłady użycia utworzonego oprogramowania w formie modułów

Poniżej przedstawiono trzy przykłady użycia utworzonego oprogramowania w formie biblioteki.

Przykład 1:

```
import ltuple as lt
occurrences_list = lt.calculate_occurrences(4, 'primates.fa')
weights_list =lt.calculate_weights('nucl, 12, primates.fa', 4)
distance_matrix = lt.weuclidean(occurrences_list, weights_list, 12)
print distance_matrix
```

Opis słowny: Moduł *ltuple* zostaje zaimportowany pod nazwą *lt.* Do zmiennej *occurrences_list* zostaje przypisany obiekt zwracany przez funkcję *calculate_occurrences,* czyli tablica wektorów, które reprezentują zliczenia słów (o długości 4) każdej sekwencji znajdującej się w pliku *primates.fa.* Podobnie zostaje obliczona tablica wektorów, które reprezentują wagi poszczególnych słów. Na samym końcu obliczona - poprzez przesłanie do funkcji *weuclidean* odpowiednich argumentów - macierz dystansów ewolucyjnych jest wyświetlona na ekran (Rycina 5).

Wynik:

```
[[  0.          47.6171875   42.1484375   42.04296875   4.7734375  … ]
 [ 47.6171875    0.          19.2734375   40.18359375  47.328125   … ]
 [ 42.1484375   19.2734375    0.          44.66796875  42.828125   … ]
 [ 42.04296875  40.18359375  44.66796875   0.          40.19921875 … ]
 [  4.7734375   47.328125    42.828125    40.19921875   0.         … ]
 …                                                                   ]
```

Rycina 5. Fragment macierzy dystansów ewolucyjnych, która została obliczona przy użyciu modułu *infotheory.*

Przykład 2:

```
import infotheory as it
distance_matrix = it.usm(it.get_sequences('primates.fa'))
print type(distance_matrix)
```

Opis słowny: Moduł *infotheory* zostaje zaimportowany pod nazwą *it.* Do zmiennej *distance_matrix* zostaje przypisany obiekt zwracany przez funkcję *usm,* czyli macierz dystansów ewolucyjnych pomiędzy sekwencjami znajdującymi się w pliku *primates.fa.* Na końcu zostaje wyświetlony typ zwracanego obiektu - jest to wielowymiarowa tablica elementów o stałym rozmiarze pakietu NumPy (Rycina 6).

Wynik:

```
<type 'numpy.ndarray'>
```

Rycina 6. Typ zwracanego obiektu przez funkcję *usm* modułu *infotheory*.

Przykład 3:

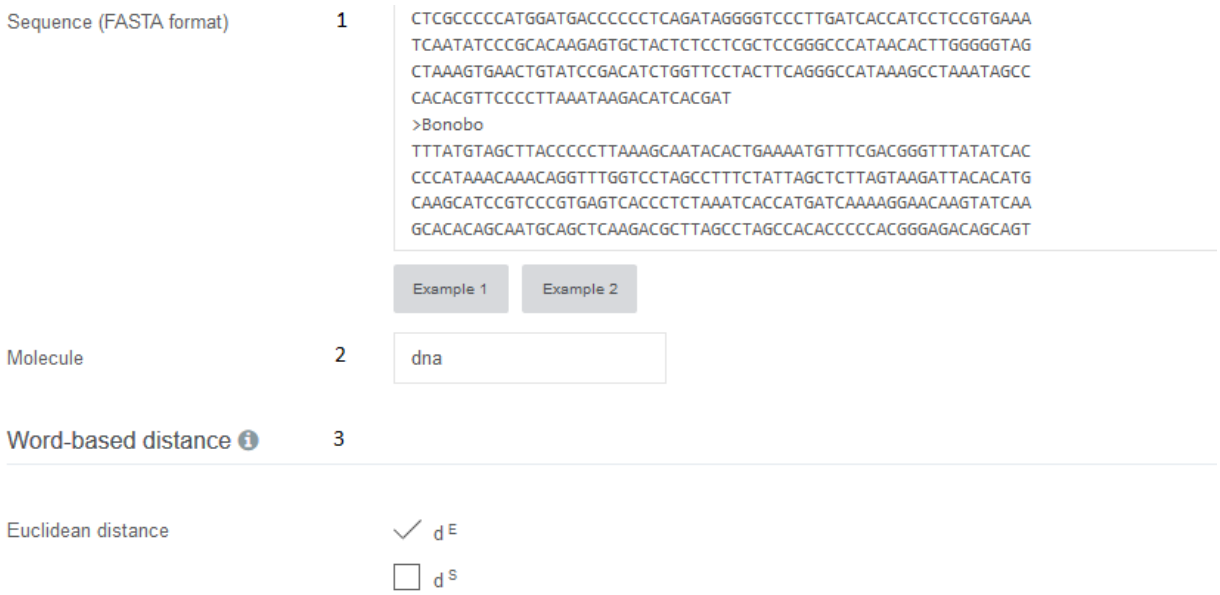
```
import ltuple as lt
distance_matrix = lt.weuclidean(lt.calculate_occurrences(4, 'primates.fa'),
lt.calculate_weights('nucl, 12, primates.fa', 4), 12)
```

Opis słowny: Przykład identyczny jak przykład nr 1, z tą różnicą, że argumentami nie są gotowe tablice wektorów, tylko wywołania funkcji obliczające potrzebne dane.

4.1.3. Aplikacje internetowe (serwis Alfree)

Zaimplementowane w tej pracy metody zostały włączone w skład internetowego meta-serwisu „Alfree", rozwijanego w Zakładzie Biologii Obliczeniowej Instytutu Biologii Molekularnej i Biotechnologii Uniwersytetu im. Adama Mickiewicza w Poznaniu. Serwis ten udostępnia ponad trzydzieści metod porównywania sekwencji typu *alignment-free*, które służą do obliczenia i wizualizacji ewolucyjnych zależności pomiędzy sekwencjami nukleotydów lub białek. Przedpremierowa wersja tego serwisu znajduje się pod adresem: http://www.combio.pl/alfree.

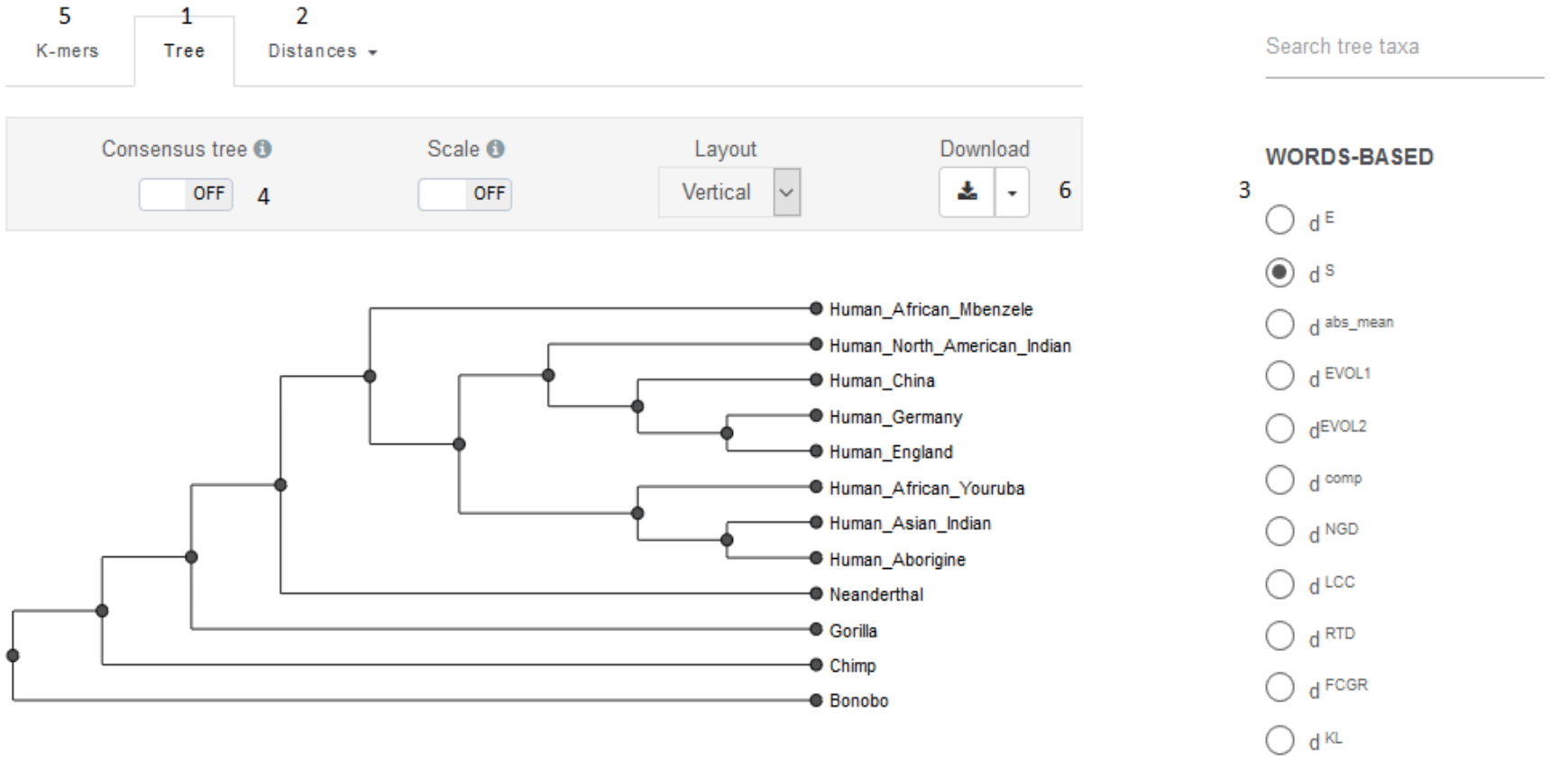
W celu rozpoczęcia analizy należy umieścić przygotowane sekwencje (w formacie FASTA) w polu *Sequence* (Rycina 7.1). Po wprowadzeniu danych serwis automatycznie rozpozna typ analizowanej sekwencji (pole *Molecule,* Rycina 7.2) i umożliwi użytkownikowi zmianę domyślnych parametrów analizy (pole *Advanced options*) oraz wybór odpowiednich metod, które są podzielone na cztery główne kategorie (m.in. pole *Word-based distance,* Rycina 7.3).



Rycina 7. Fragment formularza internetowego meta-serwisu „Alfree", który służy do wprowadzania analizowanych sekwencji oraz wyboru odpowiednich metod.

Po przeprowadzeniu analizy wyniki są bezpośrednio wyświetlane na stronie w formie interaktywnego drzewa filogenetycznego (zakładka *Tree*, Rycina 8.1) i macierzy dystansów (zakładka *Distances*, Rycina 8.2). Użytkownik może wyświetlić otrzymane drzewo filogenetyczne dla każdej metody (Rycina 8.3) lub jedno drzewo konsensusowe (wspólne dla wszystkich wybranych metod) (przycisk *Consensus tree*, Rycina 8.4). Istnieje także możliwość wyświetlenia uzyskanych „k"-krotek (zakładka *K-mers*, Rycina 8.5) oraz pobrania uzyskanych drzew filogenetycznych / macierzy dystansów (przycisk *Download*, Rycina 8.6).

Topologia drzewa filogenetycznego (Rycina 8) uzyskanego na podstawie analizy genomów mitochondrialnych ośmiu współczesnych ludzi, człowieka neandertalskiego oraz trójki naczelnych, potwierdza teorię sugerującą, że gatunek *Homo Sapiens* wywodzi się z Afryki (teoria wyjścia z Afryki). Na podstawie uzyskanej topologii można stwierdzić, że pierwszą rasą człowieka należącą do gatunku *Homo Sapiens* był człowiek z Afryki (*Human_African_Mbenzele*). Po pewnym czasie człowiek z Afryki rozdzielił się na dwie grupy (linie ewolucyjne). Pierwsza grupa udała się do Ameryki, Europy i Północnej Azji (*Human_North_American_Indian, Human_Germany, Human_England, Human_China*), a druga grupa dotarła aż do Australii (*Human_Asian_Indian, Human_Aborigine*).



Rycina 8. Fragment wynikowego formularza meta-serwisu „Alfree", który prezentuje przykładowe - otrzymane za pomocą kwadratowej odległości euklidesowej - drzewo filogenetyczne.

26:3836510880

4.2. Ocena jakości wyników wybranych programów typu „alignment-free”

Do oceny jakości programów typu *alignment-free* wybrano sześć zaimplementowanych metod: odległość euklidesową, kosinusową, ewolucyjną, kwadratową odległość euklidesową, dywergencję Kullbacka-Leiblera oraz znormalizowany dystans kompresji. Dla metod opartych na częstości występowania słów w sekwencji zastosowano cztery różne długości słów. Wybrane metody porównano także z algorytmem Smitha-Watermana, który stanowi jedną z podstawowych metod porównywania sekwencji typu *alignment-based*.

Ocena jakości wybranych metod polegała na porównaniu wyników ich przewidywań z referencyjnym zestawem rodzin białkowych zdeponowanym w bazie SCOPe. Oceny tej dokonano za pomocą analizy krzywych ROC oraz wskaźników AUC, które zostały obliczone dla poszczególnych poziomów hierarchii tej bazy. Dodatkowo dla każdej metody wykonano pomiar czasu wykonywania algorytmu. Zredukowany zbiór sekwencji źródłowych (ASTRAL40) zawiera 21 842 745 różnych par sekwencji białkowych, co łącznie daje liczbę 305 798 430 porównań wykonanych na potrzeby oceny jakości programów typu *alignment-free*.

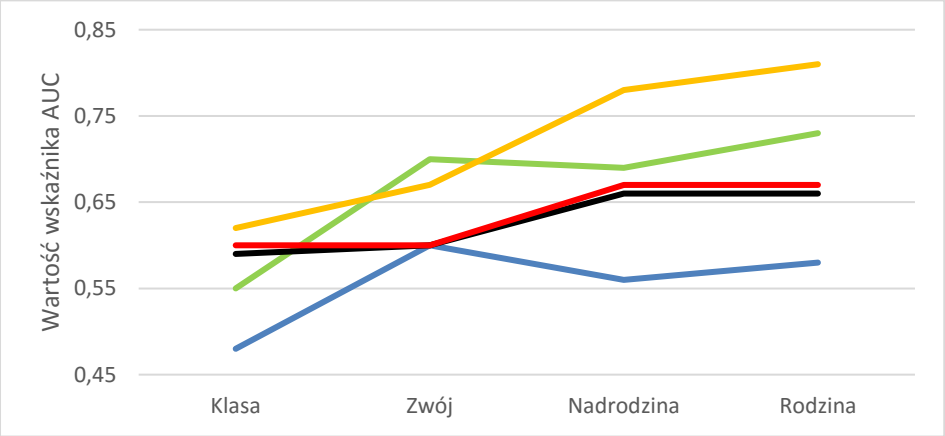
W tabeli 9 przedstawiono uzyskane wartości wskaźnika AUC dla poszczególnych metod, długości słów oraz poziomów hierarchii bazy danych SCOPe. W przypadku odległości euklidesowej i kwadratowej odległości euklidesowej oraz odległości kosinusowej i ewolucyjnej uzyskano identyczne wartości wskaźnika AUC.

Tabela 9. Wartości wskaźnika AUC otrzymanego dla poszczególnych metod, długości słów oraz poziomów hierarchii bazy danych SCOPe.

Nazwa metody	Długość słowa (L)	Wartość wskaźnika AUC dla				Średnia wartość AUC
		Klasy	Zwoju	Nadrodziny	Rodziny	
Odległość euklidesowa/ kwadratowa odległość euklidesowa	1	0.55	0.70	0.69	0.73	0.666
	2	0.48	0.60	0.56	0.59	0.559
	3	0.44	0.55	0.49	0.52	0.499
	4	0.44	0.54	0.48	0.50	0.488
Odległość kosinusowa/ odległość ewolucyjna	1	0.60	0.60	0.67	0.67	0.635
	2	0.61	0.57	0.64	0.63	0.614
	3	0.59	0.59	0.65	0.66	0.621
	4	0.54	0.57	0.61	0.62	0.585
Dywergencja Kullbacka-Leiblera	1	0.59	0.60	0.66	0.66	0.628
	2	0.61	0.54	0.61	0.59	0.587
	3	0.60	0.52	0.59	0.56	0.569
	4	0.51	0.51	0.51	0.51	0.511
Znormalizowany dystans kompresji	-	0.48	0.60	0.56	0.58	0.555
Algorytm Smitha-Watermana	-	0.62	0.67	0.78	0.81	0.720

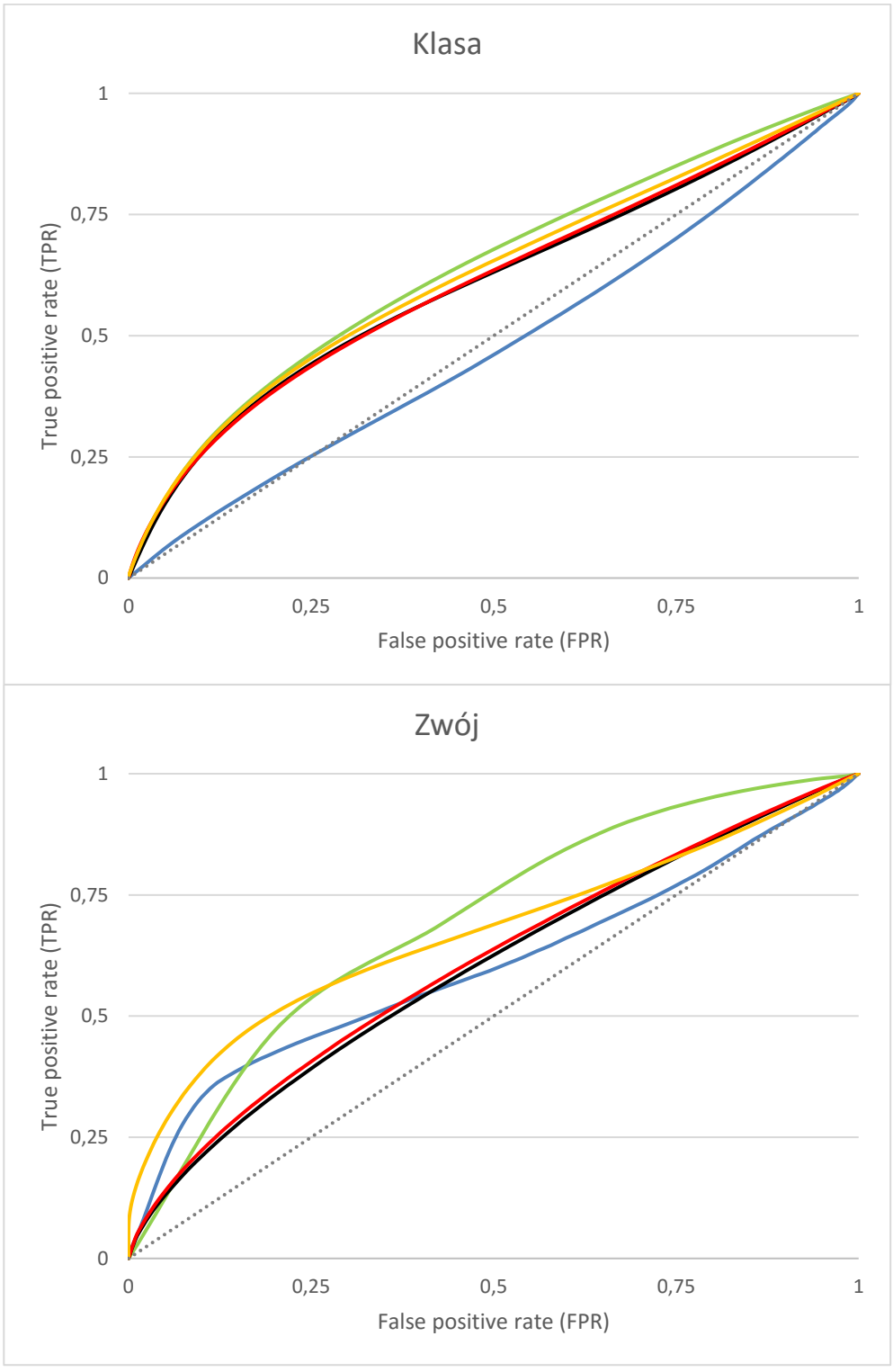
Na podstawie powyższych danych można zauważyć, że wszystkie testowane metody *alignment-free* uzyskały najlepsze wyniki dla długości słowa L = 1, bez względu na poziom hierarchii bazy danych SCOPe. Wraz ze wzrostem długości słowa oraz ze spadkiem podobieństwa między sekwencjami dokładność zaimplementowanych metod spada. Wszystkie programy uzyskały najlepsze wyniki dla sekwencji zgrupowanych w rodziny, czyli dla tych, których podobieństwo zawiera się w przedziale od 30% do 40% (najwyższy przedział podobieństwa dla analizowanych sekwencji; podobieństwo jest ciągle dobrze rozpoznawalne). Wśród metod *alignment-free* najlepszą metodą okazała się odległość euklidesowa/ kwadratowa odległość euklidesowa uzyskując średnią wartość AUC = 0.666, natomiast znormalizowany dystans kompresji uzyskał najniższą średnią wartość AUC = 0.555. Wszystkie testowane metody uzyskały wynik AUC większy niż 0.5, co świadczy o tym, że metody te są lepsze od losowego rozkładu, czyli zwracają przewidywania w ponad 50% zgodne z wynikami bazy danych SCOPe.

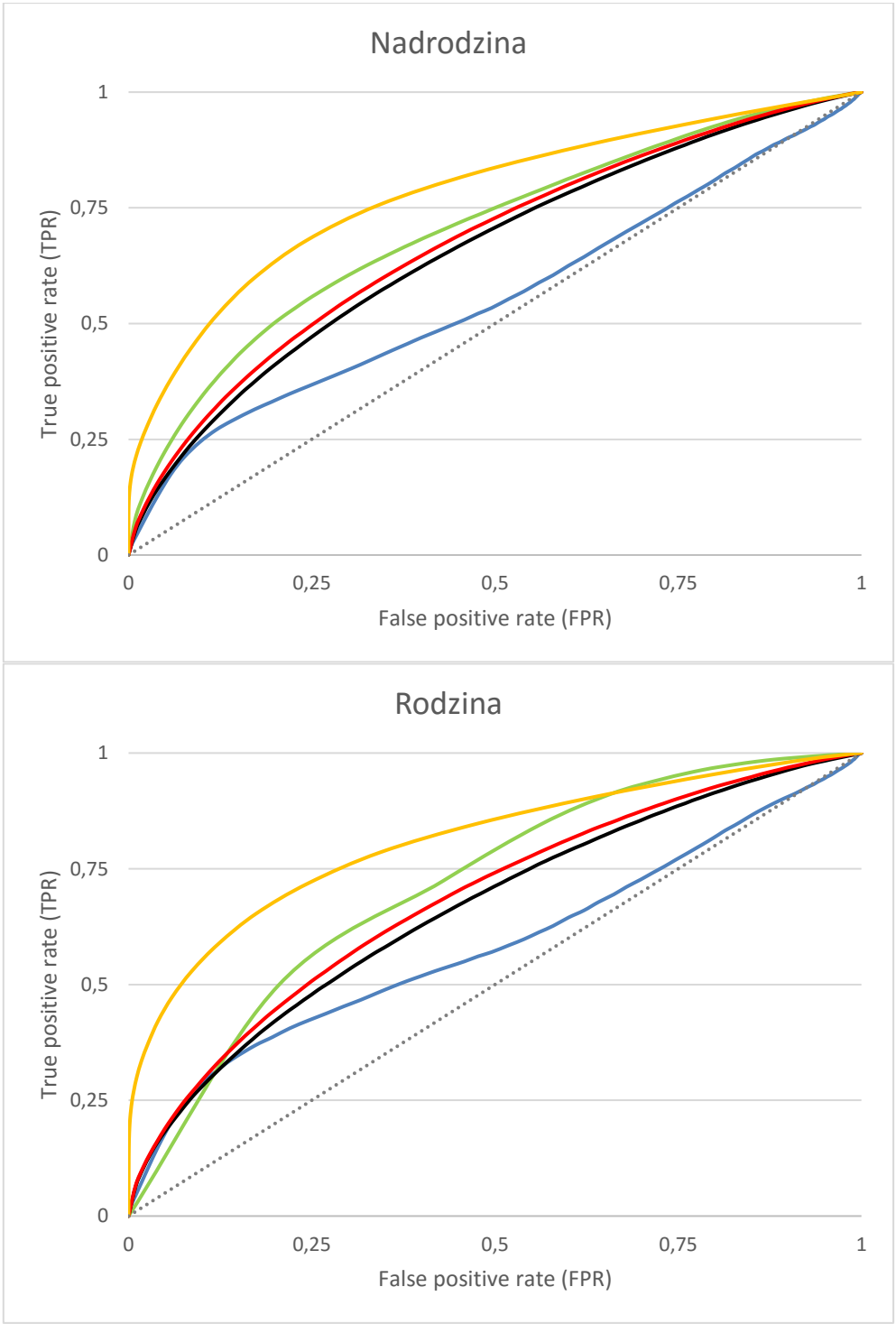
W porównaniu do metod *alignment-free* algorytm Smitha-Watermana uzyskał najlepsze wartości wskaźnika AUC dla trzech grup - z wyjątkiem grupy zwoje (AUC = 0.67), gdzie najlepsza okazała się po raz kolejny odległość euklidesowa/ kwadratowa odległość euklidesowa (AUC = 0.70). Wraz ze spadkiem podobieństwa między sekwencjami dokładność algorytmu spada. Biorąc pod uwagę wszystkie metody, najlepszy okazał się algorytm Smitha-Watermana uzyskując średnią wartość AUC = 0.720.
Na rycinie 9 przedstawiono wartości wskaźnika AUC dla analizowanych metod w obrębie różnych poziomów hierarchii bazy SCOPe.



Rycina 9. Wartości wskaźnika AUC dla analizowanych metod w obrębie różnych poziomów bazy SCOPe. Niebieskim kolorem oznaczono znormalizowany dystans kompresji, czarnym – dywergencję Kullbacka-Leiblera, zielonym - odległości euklidesowe, czerwonym – odległość kosinusową/ewolucyjną, ciemnożółtym - algorytm Smitha-Watermana. Wartość wskaźnika AUC wzrasta wraz z wyższymi poziomami bazy danych SCOPe.

Poniżej przedstawiono krzywe ROC dla analizowanych metod w obrębie różnych poziomów hierarchii bazy SCOPe (Rycina 10).





Rycina 10. Krzywe ROC uzyskane dla zbioru sekwencji ASTRAL40b w obrębie poszczególnych poziomów hierarchii bazy SCOPe. Niebieskim kolorem oznaczono znormalizowany dystans kompresji, czarnym – dywergencję Kullbacka-Leiblera, zielonym - odległości euklidesowe, czerwonym – odległość kosinusową/ewolucyjną, ciemnożółtym – algorytm Smitha-Watermana, natomiast linią przerywaną wartość wskaźnika AUC = 0.5. Algorytm Smitha-Watermana uzyskuje nieznacznie lepszą jakość przewidywań w obrębie rodziny i nadrodziny, natomiast w obrębie zwojów i klasy przewidywania są zbliżone, do tych uzyskanych za pomoca metod opartych na częstości występowania słów w sekwencji. W obrębie wszystkich grup kształ krzywych ROC jest zbliżony, z wyjątkiem krzywych ROC dla znormalizowanego dystansu kompresji.

Dodatkowo dla wybranych metod zmierzono czas wykonania algorytmu, czyli czas potrzebny do obliczenia 21 842 745 porównań par sekwencji. Jako implementację algorytmu Smitha-Watermana wykorzystano narzędzie EMBOSS Water (http://www.ebi.ac.uk/Tools/psa/emboss_water/). Tabela 10 prezentuje uzyskane czasy wykonania algorytmu dla wybranych metod.

Tabela 10. Czasy wykonywania algorytmu uzyskane dla wybranych metod *alignment-free* oraz algorytmu Smitha-Watermana.

Nazwa metody	**Uzyskany czas [HH:MM:SS]**
Kwadratowa odległość euklidesowa	00:02:57
Odległość euklidesowa	00:03:07
Dywergencja Kullbacka-Leiblera	00:05:01
Znormalizowany dystans kompresji	00:05:13
Odległość kosinusowa	00:07:49
Odległość ewolucyjna	00:08:00
Algorytm Smitha-Watermana	72:13:10

Najszybszą metodą *alignment-free* okazała się kwadratowa odległość euklidesowa, która uzyskała czas 2 minut i 57 sekund, a najwolniejszą metodą *alignment-free* została odległość ewolucyjna, z czasem działania programu wynoszącym 8 minut. Natomiast algorytm Smitha-Watermana potrzebował nieco ponad trzech dni do wykonania porównań, co oznacza, że potrzebował 1468 razy więcej czasu, niż najszybsza metoda *alignment-free.*

5. Podsumowanie

W ramach niniejszej pracy licencjackiej zrealizowano następujące cele:

1. Zaimplementowano czternaście metod porównywania sekwencji typu *alignment-free* w formie intuicyjnych dla użytkownika, ogólnodostępnych skryptów i modułów języka programowania Python. Do zaawansowanych obliczeń matematycznych wykorzystano pakiet NumPy, którego wyróżniającą cechą jest szybkość porównywalna z szybkością języka programowania C. Metody zaimplementowane z użyciem tego pakietu okazały się ponad tysiąc razy szybsze niż implementacja algorytmu Smitha-Watermana.

2. Oceniono jakość wyników utworzonego oprogramowania poprzez porównanie ich z referencyjnym zestawem rodzin białkowych zdeponowanym w bazie struktur białkowych SCOPe. Metody typu *alignment-free*, które uzyskały najwyższą czułość i specyficzność przewidywań to odległość euklidesowa oraz kwadratowa odległość euklidesowa (średnia wartość wskaźnika AUC = 0.666). Natomiast metodą, która otrzymała najniższe wartości obu parametrów jest znormalizowany dystans kompresji (AUC = 0.555). W porównaniu do metod *alignment-free,* algorytm Smitha-Watermana uzyskał najlepsze wartości wskaźnika AUC dla trzech grup - z wyjątkiem grupy zwoje, gdzie najlepsze okazały się odległości euklidesowe.

3. Zaimplementowane metody włączono w skład internetowego meta-serwisu „Alfree”, rozwijanego w Zakładzie Biologii Obliczeniowej Instytutu Biologii Molekularnej i Biotechnologii Uniwersytetu im. Adama Mickiewicza w Poznaniu.

6. Bibliografia

1. Xiong, J. (2006) Podstawy bioinformatyki. Wydawnictwo Uniwersytetu Warszawskiego, 39-50.
2. Mount, D. M. (2004) Bioinformatics: Sequence and Genome Analysis (2nd ed.). Cold Spring Harbor Laboratory Press: Cold Spring Harbor, NY.
3. Needleman, S. B., Wunsch, C. D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol., 48, 443–453.
4. Smith, T. F., Waterman, M. S. (1981) Identification of common molecular subsequences. J. Mol. Biol., 147, 195–197.
5. Zhang, Y. X., Perry, K., Vinci, V. A., Powell, K., Stemmer, W. P., del Cardayre, S. B. (2002) Genome shuffling leads to rapid phenotypic improvement in bacteria. Nature, 415, 644–646.
6. Vinga, S., Almeida, J. (Mar 1, 2003) Alignment-free sequence comparison-a review. Bioinformatics 19 (4): 513–23.
7. Dayhoff, M. O., Schwartz, R., Orcutt, B. (1978) A model of evolutionary change in proteins. In Dayhoff,M.O. (ed.), Atlas of protein sequence and structure, National Biomedical Research Foundation, Vol. 5, supplement 3,Washington, DC, pp. 345–352.
8. Henikoff, S., Henikoff, J. G. (1992) Amino acid substitution matrices from protein blocks. Proc. Natl Acad. Sci. USA, 89, 10915–10919.
9. Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., Lipman, D. J. (1997) Gapped BLAST and PSIBLAST: a new generation of protein database search programs. Nucleic Acids Res., 25, 3389–3402.
10. Pearson, W. R. (1990) Rapid and sensitive sequence comparison with FASTP and FASTA. Methods Enzymol., 183, 63–98.
11. Blaisdell, B. E. (1986) A measure of the similarity of sets of sequences not requiring sequence alignment. Proc. Natl Acad. Sci. USA, 83, 5155–5159.
12. Chan, C. X., Ragan, M. A (Jan 22, 2013) Next-generation phylogenomics. Biology direct, 8: 3.
13. Jeffrey, H. J. (1990) Chaos game representation of gene structure. Nucleid Acids Res., 18, 2163–2170.
14. Almeida, J. S., Vinga, S. (2002) Universal sequence map (USM) of arbitrary discrete sequences. BMC Bioinformatics, 3, 6.

15. Haubold, B. (2014) Alignment-free phylogenetics and population genetics. Bioinformatics 15, 407–418.
16. Haubold, B., Krause, L., Horn, T., Pfaffelhuber, P. (2013) An alignment-free test for recombination. Bioinformatics 15, 3121-7.
17. Song, K., Ren, J., Reinert, G., Deng, M., Waterman, M. S., Sun, F. (Nov 26, 2013) New developments of alignment-free sequence comparison: measures, statistics and next-generation sequencing. Bioinformatics 15, 343–353.
18. Pinello, L., Lo Bosco, G., Yuan, G. C. (Nov 6, 2013) Applications of alignment-free methods in epigenomics. Bioinformatics 15, 419–430.
19. Blaisdell, B. E. (1986) A measure of the similarity of sets of sequences not requiring sequence alignment. Proc. Natl Acad. Sci. USA, 83, 5155–5159.
20. Blaisdell, B. E. (1989) Effectiveness of measures requiring and not requiring prior sequence alignment for estimating the dissimilarity of natural sequences. J. Mol. Evol., 29, 526–537.
21. Wu, T. J., Burke, J. P., Davison, D. B. (1997) A measure of DNA sequence dissimilarity based on Mahalanobis distance between frequencies of words. Biometrics, 53, 1431–1439.
22. Torney, D. C., Burks, C., Davison, D., Sirotkin, K. M. (1990) Computation of d2: a measure of sequence dissimilarity. In George, I., Bell, T. G. M. (eds), Computers and DNA : the proceedings of the Interface between Computation Science and Nucleic Acid Sequencing Workshop, held December 12 to 16, 1988 in Santa Fe, New Mexico. Addison-Wesley, Redwood City, CA, pp. 109–125.
23. Wu, T. J., Hsieh, Y. C., Li, L. A. (2001) Statistical measures of DNA sequence dissimilarity under Markov chain models of base composition. Biometrics, 57, 441–443.
24. Stuart, G. W., Moffett, K., Baker, S. (2002a) Integrated gene and species phylogenies from unaligned whole genome protein sequences. Bioinformatics, 18, 100–108.
25. Stuart, G. W., Moffett, K., Leader, J. J. (2002b) A comprehensive vertebrate phylogeny using vector representations of protein sequences from whole genomes. Mol. Biol. Evol., 19, 554–562.
26. Li, M., Badger, J. H., Chen, X., Kwong, S., Kearney, P., Zhang, H. (2001) An information-based sequence distance and its application to whole mitochondrial genome phylogeny. Bioinformatics, 17, 149–154.

27. Fox, N. K., Brenner, S. E., Chandonia, J. M. (2014) SCOPe: Structural Classification of Proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. Nucleic Acids Research 42, 304-309.
28. Murzin, A. G., Brenner, S. E., Hubbard, T. J. P., Chothia, C. (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. Journal of Molecular Biology 247, 536-540.
29. Vinga, S., Gouveia-Oliveira, R., Almeida, J. S. (2004) Comparative evaluation of word composition distances for the recognition of SCOP relationships. Bioinformatics 20, 206–215.
30. Rost B. (2009) Twilight zone of protein sequence alignments. Protein Eng. 12, 85–94.
31. Felsenstein, J. (2005) PHYLIP (Phylogeny Inference Package) version 3.6. Distributed by the author. Department of Genome Sciences, University of Washington, Seattle.
32. Tamura, K., Dudley, J., Nei, M., Kumar, S., (2007) MEGA4: Molecular Evolutionary Genetics Analysis (MEGA) software version 4.0. Molecular Biology and Evolution 24, 1596-1599.