Uniwersytet Warszawski
Wydział Matematyki, Informatyki i Mechaniki

Aleksander Mućk
Nr albumu: 382184

Algorytmy do klastrowania duplikacji genomowych

Praca licencjacka
na kierunku BIOINFORMATYKA I BIOLOGIA SYSTEMÓW

Praca wykonana pod kierunkiem
dra hab. Pawła Góreckiego

Sierpień 2019

Oświadczenie kierującego pracą

Potwierdzam, że niniejsza praca została przygotowana pod moim kierunkiem i kwalifikuje się do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data Podpis kierującego pracą

Oświadczenie autora (autorów) pracy

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data Podpis autora pracy

Streszczenie

Niniejsza praca przedstawia propozycje rozwiązań algorytmicznych dla problemów klastrowania duplikacji genomowych w oparciu o scenariusze ewolucyjne. W części pierwszej wprowadzane są podstawowe pojęcia dotyczące drzew genów, gatunków, modeli ich uzgadniania oraz tworzenia scenariuszy ewolucyjnych. Omówiony został również problem przeliczania i klastrowania duplikacji genomowych. W części drugiej opisana została proponowana heurystyka wraz z przykładowymi testami oraz jej implementacją w języku Python.

Słowa kluczowe

duplikacja genu, drzewo genów, drzewo gatunków, analiza filogenetyczna, drzewo uzgadniające, Python, scenariusz ewolucyjny, strata genu, minimalizacja kosztu ewolucyjnego

Dziedzina pracy (kody wg programu Socrates-Erasmus)

11.9 Inne nauki matematyczne i informatyczne

Klasyfikacja tematyczna

Computional biology, Applied computing, Life and medical sciences

Tytuł pracy w języku angielskim

Algorithms for the clustering of genomic duplication

Spis Treści

Wprowadzenie

Uzgadnianie drzew filogenetycznych jest, przez rozmiar danych i coraz bardziej skomplikowane modele, niezwykle złożone zarówno obliczeniowo jak i koncepcyjnie. Badania drzew genów i gatunków, a w szczególności zależności między nimi może odpowiedzieć na pytania w jaki sposób wyodrębniały się gatunki przez pryzmat zmian w ich genomie. Mimo wszystko jednak należy pamiętać, że pokrewieństwo gatunków nie zawsze implikuje pokrewieństwo genów, których drzewo ewolucyjne nie musi pokrywać się z drzewem zawierających je gatunków, które samo w sobie nie jest tak bardzo bardzo zróżnicowane jak drzewo genów. Tworzenie scenariuszy ewolucyjnych, dzięki którym możemy poznać w jaki sposób ewolucja genów wpływała na ewolucję gatunków jest zadaniem nietrywialnym. Potrzebne są narzędzia, które potrafiłyby ocenić scenariusze pod kątem ilości epizodów ewolucyjnych. Epizody, takie jak duplikacje genomowe, mogą być wyznacznikami prawdopodobieństwa danego scenariusza. Zagadnienie to jest jeszcze bardziej wymagające od uzgodnienia pojedynczego drzewa, ponieważ jeden scenariusz zawiera wiele drzew genów co wpływa na poziom złożoności obliczeń. W niniejszej pracy proponowany jest algorytm, który ocenia zbiór scenariuszy tworząc na ich podstawie jeden, którego koszt, liczony jako ilość duplikacji, będzie możliwie najmniejszy.
Praca składa się z czterech rozdziałów i dodatków. W rozdziale 1 przedstawiono podstawowe pojęcia dotyczące drzew genów, drzew gatunków oraz modeli i scenariuszy ewolucyjnych. Rozdział 2 przedstawia propozycję heurystyki wraz z jej testami na rzeczywistych danych. W rozdziale tym opisano również implementację i sposób użycia programu napisanego na podstawie przybliżonej we wcześniejszej sekcji heurystyki. Ostatni rozdział zawiera przemyślenia dotyczące możliwego użycia algorytmu i perspektyw jego rozwoju. W dodatkach umieszczono fragmenty kodu, przykładowe dane wejściowe i wyniki działania algorytmu.

Rozdział 1

Podstawowe pojęcia

W tym rozdziale poruszane są pojęcia i definicje niezbędne do zrozumienia problematyki klastrowania duplikacji genomowych.

1.1. Wstęp biologiczny

Ewolucja biologiczna jest procesem zmian w trakcie których organizmy stopniowo nabywają lub tracą pewne cechy. Jest to element kluczowy dla powstawania nowych gatunków: specjacji. Śledzenie w jaki sposób kształtowały się nowe gatunki i w jaki sposób zachodziły na Ziemi procesy ewolucyjne jest zadaniem niezwykle złożonym i wymagającym specyficznego podejścia. Jednym z możliwych sposobów przedstawienia historii ewolucyjnej gatunków jest drzewo filogenetyczne, które przedstawiają zależności ewolucyjne pomiędzy umieszczonymi na nim gatunkach lub genach.

Początkowo za wyznacznik pokrewieństwa gatunków służyło podobieństwo morfologiczne, jednak obecnie często stosuje się metody polegające na badaniu podobieństwa danych rodzin genów. Można założyć, że im większe podobieństwo genomu danych organizmów tym bliżej są one spokrewnione. Z punktu widzenia tej pracy genom jest niczym więcej jak zbiorem genów obecnym w danym organizmie.

1.2. Drzewa genów i gatunków

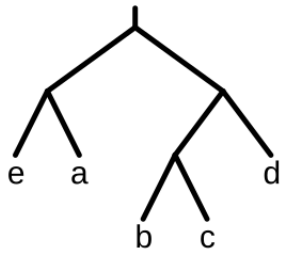
W pracy tej T niech będzie ukorzenionym, binarnym drzewem o zbiorze krawędzi E_T i zbiorze węzłów V_T:

$$\mathrm{T} = <V_T\ ,\ E_T>.$$

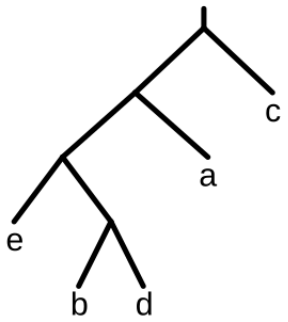
E_T zawiera pary węzłów (v_x , v_y) takie, że $V_T \ni v_x$,v_y. Węzły z których nie wychodzą żadne krawędzie nazywane są liśćmi, a korzeń jest węzłem do którego nie prowadzą żadne krawędzie (nieposiadającym rodzica). Węzeł v_x jest przodkiem węzła v_y jeśli istnieje ścieżka skierowana z węzła v_x do węzła v_y. Liczba krawędzi w ścieżce od węzła v_x do węzła v_y jest nazywana długością. Poddrzewem węzła v_x jest drzewo oznaczone T(v_x) w którym węzeł v_x jest korzeniem.

Definicja 1.2.1 *Związki między gatunkami przedstawia się za pomocą drzewa T, zwanego drzewem gatunków S. W drzewie S każdy z liści reprezentuje inny gatunek Sp, a węzły wewnętrzne są specjacjami.*

Definicja 1.2.2 *Związki między genami w danej rodzinie przedstawia się za pomocą drzewa T, zwanego drzewem genów G. W drzewie G każdy z liści reprezentuje przynależność danego genu do gatunku Sp.*



(a) Przykładowe drzewo genów G



(b) Przykładowe drzewo gatunków S

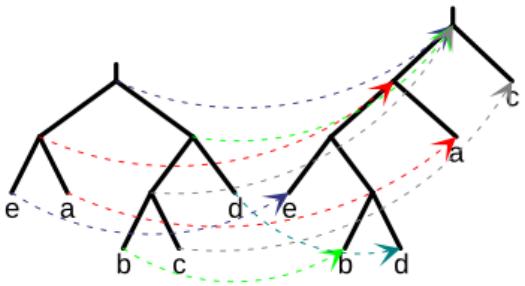
Rysunek 1.1: Przykładowe drzewa T [1]

1.3. Uzgodnienie drzew

Bardzo częste różnice struktury drzewa genów w stosunku do historii ewolucyjnej opisanej drzewem gatunków wymagają mapowania węzłów drzewa G na węzły znajdujące się w drzewie S. Jest to krok niezbędny by zrozumieć w jaki sposób ewolucja gatunków wpływała na strukturę ich genomów.

1.3.1. Mapowanie LCA

Podstawowym algorytmem dla tego typu uzgodnień jest **algorytm LCA** (ang. *Lowest Common Ancestor*; pl. *Najniższy Wspólny Przodek*). Najniższym przodkiem węzłów v_x i v_y jest taki węzeł v_{anc}, który jest przodkiem obu węzłów i którego długość od korzenia drzewa jest największa.



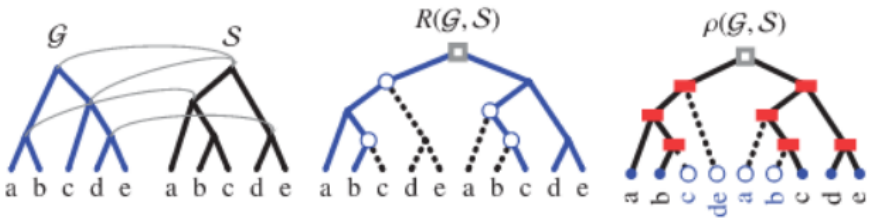
Rysunek 1.2: Uzgodnienie LCA dla przykładowych drzew genów G i gatunków S

Definicja 1.3.1 *Uzgodnienie drzew jest taką funkcją MAP_{LCA}: $G \to S$ gdzie dla każdego węzła v_g z drzewa genów G przyporządkowuje się węzeł v_s z drzewa gatunków S tak, że węzeł*

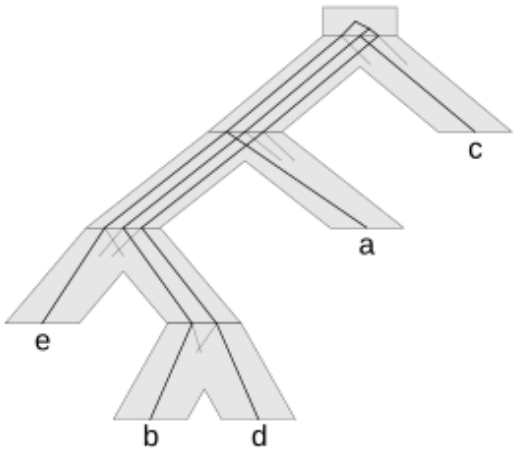
v_s jest węzłem v_{anc} i węzły z poddrzewa $T(v_g)$ są obecne w $T(v_s)$.

1.3.2. Drzewa DLS

Z wykorzystaniem mapowania węzłów drzewa genów do drzewa gatunków można stworzyć drzewo uzgadniające zwane drzewem DLS (*Duplication-Loss Tree*), które jest reprezentacją pojedynczego scenariusza ewolucyjnego. Scenariusz ewolucyjny jest przedstawieniem ewolucji danej rodziny genów, przy uwzględnieniu ewolucji gatunków, które owe geny zawierają. Drzewa te posiadają dwa rodzaje węzłów. Pierwszy rodzaj węzła opisuje zjawisko duplikacji, czyli powielenia tego samego genu do dwóch kopi (oznaczmy jako DUP), zaś drugi jest wyrażeniem zjawiska specjacji, to jest powstania nowych genów (oznaczmy jako SPEC). W drzewie DLS obecne są również dwa różne rodzaje liści, gdzie jeden z nich jest reprezentacją straty genu (oznaczmy jako LOSS), a drugi opisuje jego obecność [**?**]. Przykład tego typu drzewa można znaleźć w pracy [3] , której pochodzi rysunek 1.3.



Rysunek 1.3: Drzewo DLS: p(G,S), które uzgadnia drzewo genów G i drzewo gatunków S. Węzeł oznaczony kwadratem jest węzłem duplikacyjnym. Gałęzie narysowane linią przerywaną, które prowadzą do liści, które są reprezentacją straty genów.



Rysunek 1.4: Wbudowanie

1.3.3. Koszt scenariusza ewolucyjnego

Scenariusze ewolucyjne porównywane są ze sobą za pomocą ich kosztu ewolucyjnego. W pracy tej liczba ta wyraża liczbę duplikacji potrzebnych do uzgodnienia danego drzewa genów

i gatunków, co jest równoznaczne z ilością węzłów duplikacyjnych w drzewie uzgadniającym. Dla przykładu drzew z Rysunku 1.2 koszt ten wynosi 2.

Definicja 1.3.2 *MEscore* = $\sum_{g \in DLS} 1$ *jeśli g jest DUP*

1.4. Modele scenariuszy ewolucyjnych

Drzewo uzgadniające, które opiera się o mapowanie LCA, jest drzewem o możliwie najmniejszym koszcie ewolucyjnym i możliwe najgłębiej położonych węzłach duplikacyjnych. Nie jest to jednak jedyny możliwy scenariusz, których w rzeczywistości jest nieskończenie wiele. Należy jednak pamiętać, że nie powinno się brać pod uwagę przypadków skrajnie nieprawdopodobnych, gdzie gen jest, dla przykładu, wielokrotnie duplikowany i tracony. Po wprowadzeniu tego typu ograniczeń możliwe jest otrzymanie skończonego zbioru możliwych scenariuszy ewolucyjnych, które zwane są semi-normalnymi.

1.4.1. Transformacje scenariuszy ewolucyjnych

Za podstawę do uzyskania zbioru scenariuszy semi-normalnych dla danego drzewa G może służyć uzyskane dzięki mapowaniu LCA drzewo uzgadniające, które w kolejnych krokach poddawane będzie ściśle określonym transformacjom:

Definicja 1.4.1 *TMOVE.*

Definicja 1.4.2 *CLOST.*

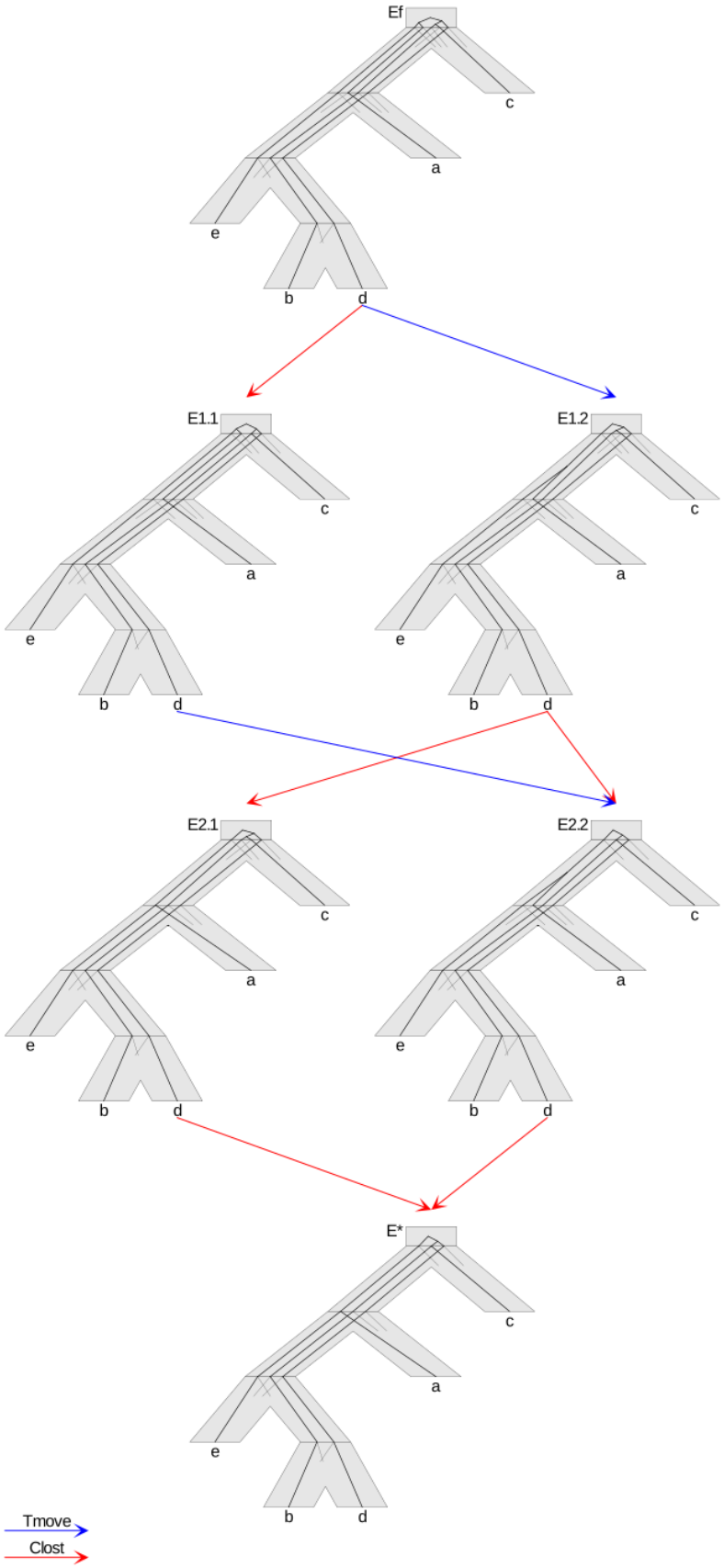
Stosując kombinacje podanych ruchów można wyselekcjonować takie scenariusze, które określą oczekiwany zbiór (Patrz rysunek **??**).

1.4.2. Opis modeli

W pracy tej zajęto się podanymi modelami, które dopuszczają podane warunki:

1. Model PG [**?**]: Model ten dopuszcza tylko takie scenariusze, które zachowują minimalny koszt ewolucyjny. Dozwolony rodzaj transformacji scenariusza to TMOVE.

2. Model FHS [**?**]: Model ten dopuszcza każde możliwe przekształcenie scenariusza ewolucyjnego. Dozwolone rodzaje transformacji to TMOVE i CLOST.



Rysunek 1.5: Diagram redukcyjny obrazujący możliwość uzyskania scenariuszy semi-normalnych. Na dole rysunku znajduje się drzewo uzyskane za pomocą mapowania LCA. Model PG zawierać będzie tylko właśnie to drzewo, podczas gdy dla modelu FHS wszystkie drzewa obecne na diagramie są częścią zboru scenariuszy semi-normalnych.

Rozdział 2

Heurystyka

2.1. Opis algorytmu

Algorytm zakłada, że na wejściu dostępne są dane:

- Drzewa genów G_1 ... G_k,
- m_i scenariuszy drzewa G_i opisanych jako wektory v_{i1}, v_{i2} ... v_{im} z wyliczonymi wartościami kosztu ewolucyjnego, mierzonego jako ilość duplikacji dla każdego węzła.

Istnieje wektor przybliżający rozwiązanie ME, który nazwijmy V* i który pasuje do każdego scenariusza.
Jednym z takich wektorów jest wektor V_{max}, który na każdej współrzędnej x zawiera maksymalną wartość $v_{im}[x]$ dla każdego scenariusza m w każdym drzewie genów G_i. Oczywiście nie jest to rozwiązanie najlepsze, ponieważ jest one kosztowne, ale pasuje do każdego scenariusza.

Bazując na wyliczonym wektorze V_{max} heurystyka wylicza wektor V* i w pętli poprawia go w następujący sposób:

- Obniż jedna z wartości w wektorze V*,
- Zaakceptuj w/w zmianę jeśli dla każdego drzewa genów istnieje scenariusz, który jest zgodny z takim wektorem epizodów,
- Zakończ działanie jeśli nie da się poprawić żadnej współrzędnej.

Wybór współrzędnej może, zależnie od potrzeby, może być dokonywany w inny sposób:

- od końca wektora (od korzenia),
- od początku (od liści),
- losowo.

2.2. Dokumentacja użytkowa i opis implementacji

Opisana heurystyka zaimplementowana została w języku Python w wersji 3.7.4 przy użyciu paradygmatu obiektowego, gdzie zbiór wszystkich scenariuszy dla wszystkich drzew, pojedynczy diagram redukcyjny otrzymany z jednego drzewa genów, a także pojedynczy scenariusz stanowią oddzielne klasy. Program pythonowy przyjmuje na wejściu listę plików w których zawarte są wyliczone scenariusze dla danego drzewa genów.

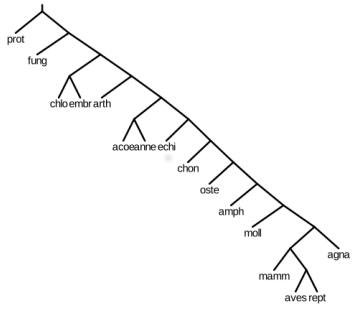
Algorytm ten, dla wygody użycia, został obudowany skryptem napisanym w języku bash, który pozwala na wyliczenie scenariuszy w modelu FHS i PG z wykorzystaniem programu DLSgen autorstwa dra hab. Pawła Góreckiego.[?] Program ten wylicza dla danego drzewa genów i drzewa gatunków scenariusze ewolucyjne, które wykorzystywane są jako dane wejściowe dla proponowanej heurystyki.

2.3. Testy algorytmu

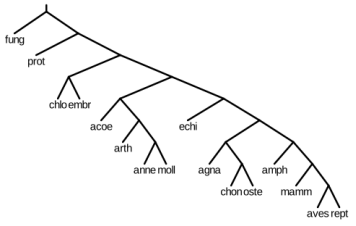
Do sprawdzenia wyników wyliczanych przez proponowany algorytm użyty został program RME napisany przez dra Jarosława Paszka. Program ten korzystając z dostępnych metod algorytmicznych wylicza dokładne i najniższe możliwe wartości kosztu ewolucyjnego dla podanych modeli. [2]
Z powodu trudności w obliczeniu danych wejściowych dla heurystyki obecnie nie ma możliwości dla przetestowania algorytmów dla dużych zbiorów danych.
Ze względu na czytelność wykresów w dalszej części pracy we wszystkich testach uwzględniony został tylko losowy wybór współrzędnej. W trakcie testów okazało się również, że taki wybór prowadzi zawsze do najlepszych wyników.

2.3.1. Testy algorytmu na danych rzeczywistych

Zbiorem danych dla testu na danych rzeczywistych był zbiór Guigo zawierający 53 ukorzenione drzewa genów pochodzące od 16 eukariontów. Zbiór ten zawiera dwa drzewa gatunków S_1 i S_2. [?]

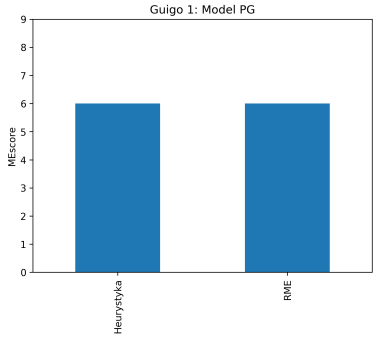


(a) Drzewo S_1

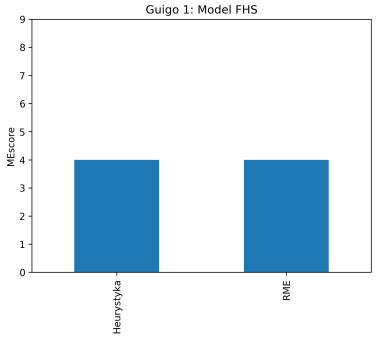


(b) Drzewo S_2

Rysunek 2.1: Drzewa gatunków ze zbioru Guigo

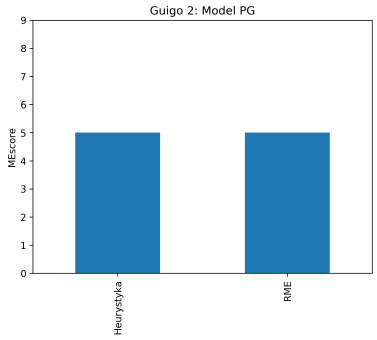


(a) Test algorytmu dla modelu PG

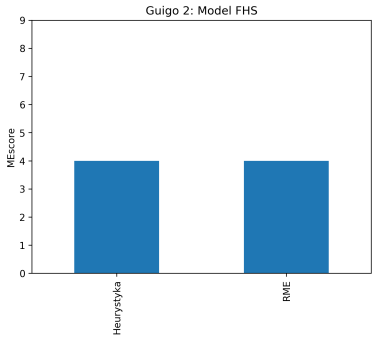


(b) Test algorytmu dla modelu FHS

Rysunek 2.2: Testy algorytmu na danych rzeczywistych dla drzewa gatunków S_1



(a) Test algorytmu dla modelu PG



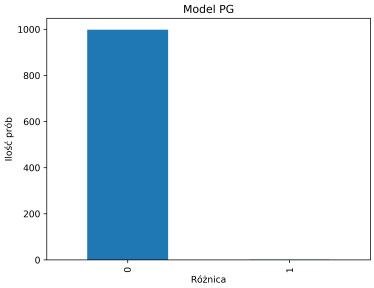
(b) Test algorytmu dla modelu FHS

Rysunek 2.3: Testy algorytmu na danych rzeczywistych dla drzewa gatunków S_2

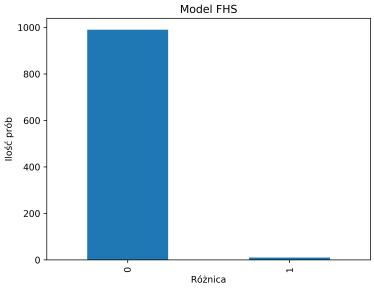
Dla wszystkich przypadków algorytm przy losowym wyborze współrzędnych jest w stanie osiągnąć wyniki identyczne jak te wyliczone przez program RME.

2.3.2. Testy algorytmu na danych symulowanych

Dane dla testów syntetycznych zostały wygenerowany w sposób losowy jednak wielkość, struktura i ilość drzew genów zostały dopasowane do zbioru Guigo. Zbiory symulowane zawierają syntetyczne drzewo gatunków z 15 etykietami oraz losowo wygenerowane 48 ukorzenione drzewa genów oparte o etykiety obecne w drzewie gatunków. Przykładowe drzewa gatunków (w formie graficznej) i genów (w formie tekstowej) można znaleźć w dodatkach do pracy. Test został przeprowadzony dla 1000 losowych zestawów.



(a) Test algorytmu dla modelu PG



(b) Test algorytmu dla modelu FHS

Rysunek 2.4: Testy algorytmu na danych symulowanych. Przez różnicę rozumiany jest wynik odjęcia od wyliczeń otrzymanych przez program RME wyliczeń otrzymanych dzięki proponowanemu algorytmowi.

W obu przypadkach wyniki obliczeń heurystyki nie odbiegają znacząco od dokładnych wartości kosztu ewolucyjnego otrzymanych przez program RME. Za pomocą wykresu nie da się nawet dokładnie określić dla ilu zbiorów heurystyka nie wyliczyła prawdziwego, minimalnego kosztu.
Dla modelu PG tylko 4 zbiory z 1000 nie dały tego samego wyniku, a dla modelu FHS było to 17 zbiorów z 1000.

Rozdział 3

Podsumowanie

W pracy przedstawiono pierwszy i dosyć intuicyjny pomysł na ocenę scenariuszy ewolucyjnych pod kątem ilości duplikacji. Należy jednak wspomnieć, że samą ideę da się znacząco usprawnić i obniżyć złożoność obliczeniową nawet do poziomu liniowego.
Obecnie to właśnie krok w którym konieczne jest wyliczenie scenariuszy dla drzew genów jest krokiem najbardziej wymagającym czasowo i obliczeniowo. W związku z tym trudno mówić o czasie potrzebnym algorytmowi na własne obliczenia. Nie udało się zaobserwować, by algorytm kiedykolwiek potrzebował więcej niż jedną sekundę na wczytanie danych i więcej niż 0.3 sekundy na obliczenie drzewa o najmniejszym koszcie ewolucyjnym. Ponadto obecne wymaganie dotyczące danych wejściowych uniemożliwia przetestowanie algorytmu na danych bardziej złożonych niż zbiór Guigo.
Testy pokazują również, że opisany algorytm zwraca wyniki, które różnią się w bardzo niewielkim stopniu (o ile w ogóle) od rzeczywistego minimalnego kosztu ewolucyjnego, gdyż maksymalnie tylko dla 2,8% danych algorytm nie uzyskał najniższego możliwego wyniku. Maksymalna różnica jaką udało się zaobserwować wynosiła 1, co również nie jest dużą wartością. Może to jednak wynikać z dosyć niskiego kosztu ewolucyjnego danych zbiorów, który wynosił maksymalnie 9 (średni koszt ewolucyjny to 6,1) dla modelu PG i 7 (średni koszt ewolucyjny to 4,5) dla modelu FHS. Widać jednak, że proponowany algorytm wymaga kolejnych, bardziej rozbudowanych testów.

3.1. Perspektywy rozwoju

Trudno przewidzieć wszystkie możliwości rozwoju algorytmu, ale te bardziej oczywiste można wskazać już teraz. Są to:

- Uniezależnienie algorytmu od kroku w którym wyliczane są scenariusze i klastrowanie duplikacji bezpośrednio na podstawie drzew genów.
- Uliniowienie algorytmu.

Szczególnie punkt pierwszy proponowanych usprawnień jest kluczowy dla dalszego rozwoju heurystyki, ponieważ pozwoli on na używanie i dotestowanie go na danych dużo bardziej skomplikowanych i bardziej przystających do obecnych problemów niż zbiór Guigo.

3.2. Perspektywy wykorzystania

Podstawową zaletą przedstawionej heurystyki jest jej elastyczność. Ocena scenariuszy nie zależy od obranego modelu, a obecnie wydaje się, że same obliczenia nie są obarczone dużym

błędem. Kolejną niewątpliwą zaletą jest fakt, że w istocie również struktura drzewa nie ma dla algorytmu dużego znaczenia. Obecnie powoli odchodzi się od drzewa binarnego jako metody przedstawienia historii ewolucji i algorytm jest na taką zmianę gotowy. Z punktu widzenia algorytmu drzewa są tablicą zawierającą ilość klastrów duplikacyjnych dla danego węzła, a które umieszczone są w niej w porządku prefiksowym, co zapewnia możliwość wykorzystania algorytmu nie tylko dla drzew binarnych. Algorytm funkcjonuje obecnie jednak w dosyć prymitywnej formie i wymaga wzmożonej i dokładnej pracy, ale sam algorytm rokuje niezwykle pozytywnie.

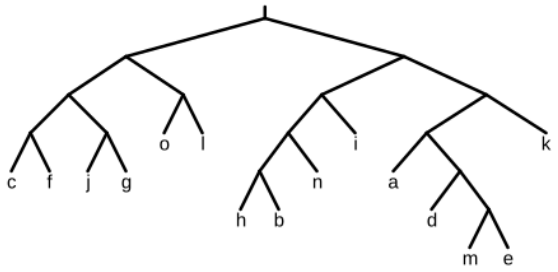
Dodatek A

Pętla programu zapisana w języku Python wykonywana dla losowego wybierania indeksów

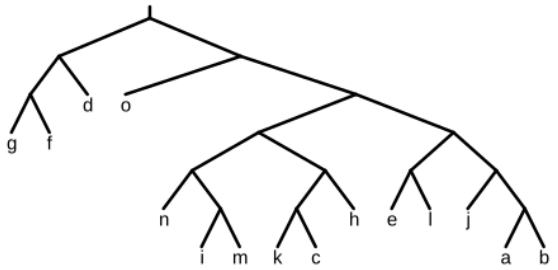
```
max_trees = []
        for scenario in self:
            all_dup_pref = [tree.duplication_prefix for tree n scenario]
            max_trees.append(self.rate_scenario(all_dup_pref))
        max_tree = self.rate_scenario(max_trees)

        if select_type == "random":

            index_list = [x for x in range(len(max_tree)) if x != 0]

            while index_list:
                index_list_position = random.randint(0, len(index_list) - 1)
                index = index_list[index_list_position]

                max_tree_temp = max_tree[:]
                max_tree_temp[index] -= 1

                for scenario in self:
                    for tree in scenario:
                        for i in range(len(tree.duplication_prefix)):
                            if max_tree_temp[i] - tree.duplication_prefix[i] < 0:
                                break
                        else:
                            break
                    else:
                        index_list.pop(index_list_position)
                        break
                else:
                    max_tree = max_tree_temp
            return max_tree, sum(max_tree)
```

Dodatek B

Przykładowe drzewa gatunków dla danych syntetycznych



Rysunek B.1: Drzewo gatunków S_1



Rysunek B.2: Drzewo gatunków S_2

Dodatek C

Przykładowe drzewa genów dla danych syntetycznych

Format tekstowy:

((b,l),h)
((a,k),b)
((i,k),a)
((l,b),o)
(h,(j,o))
(b,(i,c))
(b,(m,j))
((m,a),b)
((g,m),f)
((a,l),k)
(o,(j,h))
((k,d),i)
(e,(c,o))
(f,(a,h))
(j,(k,d))
((k,e),c)
(j,(e,b))
((i,n),l)
(f,(b,h))
(c,(k,g))
((l,o),j)
((a,f),d)
((f,h),j)
(h,(d,n))
((b,j),i)
((a,e),(g,l))
((b,c),(n,o))
(m,((j,h),d))
((e,(f,g)),a)
(((m,d),g),a)
((h,l),(n,b))
((d,(b,a)),k)

(((k,n),c),m)
(((d,c),b),(j,a))
(c,((l,(g,j)),k))
(((h,g),(a,e)),l)
((o,(f,e)),(j,h))
((g,l),((c,m),n))
((e,(f,(j,o))),k)
(f,(a,((n,h),(b,c))))
((a,(o,h)),((e,i),n))
((e,d),((a,b),(n,g)))
(((m,(d,b)),j),(f,e))
((l,h),(((a,j),((m,g),f)),e))
((n,(i,e)),(((a,k),f),(l,b)))
((j,e),(i,((a,(k,(l,d))),n)))
((((f,k),(b,(o,e))),((g,n),j)),m)
(h,((b,(m,f)),((c,j),(a,(k,l)))))

Bibliografia

[1] Wszystkie obrazy wygenerowane zostały za pomocą serwisu http://gsevol.azor.mimuw.edu.pl

[2] Jarosław Paszek, Paweł Górecki *https://www.mimuw.edu.pl/ jpaszek/rme.html*

[3] *DLS-trees: a model of evolutionary scenarios, Theoretical Computer Science 359 (1-3) 2006, s. 378–399. zobacz*