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

Badanie 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. W szczególności drzewo ewolucyjne genów nie musi pokrywać się z odpowiadającym im drzewem 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 zdarzeń ewolucyjnych i przyporządkować je we właściwe miejsca historii ewolucyjnej.

Należy zaznaczyć, że wybranie właściwego scenariusza ewolucyjnego nie jest zadaniem łatwym, ponieważ takich scenariuszy, wyjaśniających ewolucję danej rodziny genów, może być nieskończenie wiele. Problem jeszcze bardziej komplikuje się gdy w badaniach uwzględnimy wiele drzew genów.

W niniejszej pracy proponowane są algorytm, które oceniają 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 złożonym i wymagającym specyficznego podejścia. Jednym z możliwych sposobów przedstawienia historii ewolucyjnej gatunków jest drzewo filogenetyczne, które przedstawiają zależności ewolucyjne pomiędzy umieszczonymi na nim gatunkami lub genami.

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 genów 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 drzewo to ukorzenione, binarne drzewo T o zbiorze krawędzi skierowanych E_T i zbiorze węzłów V_T:

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

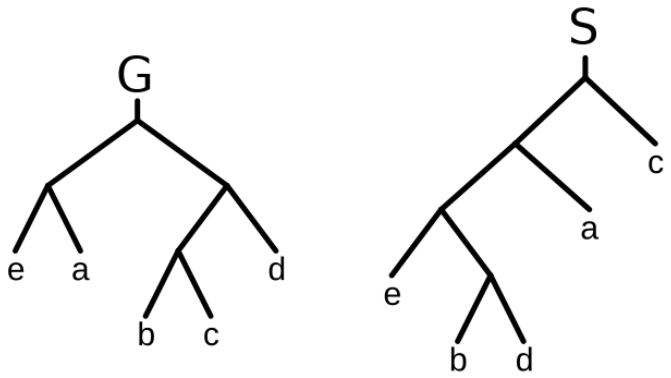
gdzie E_T zawiera pary węzłów (V,W) takie, że V,$W \in V_T$. Węzły z których nie wychodzą żadne krawędzie nazywane są liśćmi, a korzeń jest węzłem do którego nie prowadzą żadne krawędzie (nieposiadającym rodzica). Węzeł V jest przodkiem węzła W jeśli istnieje ścieżka skierowana z węzła V do węzła W.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

Związki między gatunkami przedstawia się za pomocą drzewa T, zwanego drzewem gatunków S.

Definicja 1.2.1 *Drzewo gatunków to takie drzewo S, gdzie każdy liść reprezentuje gatunek, a węzły wewnętrzne nazywamy specjacjami. Zbiór wszystkich gatunków (liści) oznaczony jest jako $L(S)$.*

Związki między genami w danej rodzinie przedstawia się za pomocą drzewa T, zwanego drzewem genów G.

Definicja 1.2.2 *Drzewo genów to takie drzewo G, gdzie każdy liść reprezentuje gen i jest etykietowony gatunkiem, z którego dany gen został zsekwencjonowany. Zbiór wszystkich gatunków (etykiet) oznaczony jest jako $L(G)$.*



Rysunek 1.1: Przykładowe drzewa genów G i gatunków S.

1.3. Uzgodnienie drzew

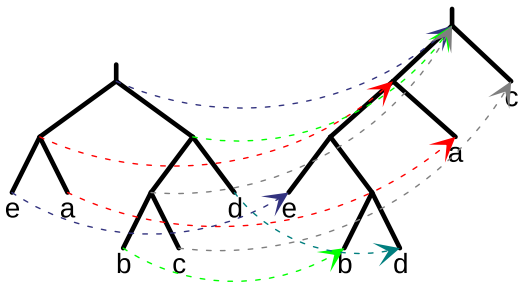
Bardzo częste różnice struktury drzewa genów w stosunku do historii ewolucyjnej opisanej drzewem gatunków wymagają mapowania węzłów drzewa G na węzły znajdujące się w drzewie S. Jest to krok niezbędny by zrozumieć w jaki sposób ewolucja gatunków wpływała na strukturę ich genomów.

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 i X to taki węzeł oznaczony jako $LCA(V, W)$, który jest przodkiem obu węzłów i którego długość ścieżki od korzenia drzewa jest największa. Najniższego wspólnego przodka dwóch węzłów można policzyć w czasie stały po liniowym, jednorazowym preprocesingu.

Definicja 1.3.1 *Dla drzewa genów G i drzewa gatunków S takich, że $L(G) \subseteq L(S)$ mapowanie LCA to funkcja MAP_{LCA}: $V_G \to V_S$ gdzie dla każdego węzła g z drzewa genów G $MAP_{LCA}(V_G)$ to węzeł taki, że:*

$$MAP_{LCA}(g) = \begin{cases} \textit{etykieta } g & \textit{jeśli } g \textit{ jest liściem,} \\ LCA(MAP_{LCA}(g_1), MAP_{LCA}(g_2)) & \textit{jeśli } g \textit{ ma synów } g_1 \textit{ i } g_2. \end{cases}$$



Rysunek 1.2: Mapowanie MAP_{LCA} dla przykładowego drzewa genów G i gatunków S.

1.3.2. Drzewa DLS

Scenariusz ewolucyjny jest przedstawieniem ewolucji danej rodziny genów, przy uwzględnieniu ewolucji gatunków, które owe geny zawierają. Jako reprezentację scenariusza ewolucyjnego można stosować drzewa nazwane drzewami DLS (*Duplication-Loss-Speciation Tree*) [3]. Drzewa te posiadają dwa rodzaje węzłów wewnętrznych i dwa rodzaje liści. Pierwszy rodzaj węzła opisuje zjawisko duplikacji, czyli powielenia tego samego genu do dwóch kopii (oznaczmy jako DUP), zaś drugi jest wyrażeniem zjawiska specjacji (oznaczmy jako SPEC). Pierwszy z rodzajów liści jest opisuje stratę genu (oznaczmy jako LOSS), a drugi jest reprezentacją sekwencji genu obecnego w gatunku o danej etykiecie. Drzewo DLS dla ustalonego drzewa genów G i ustalonego drzewa gatunków S, gdzie $L(G) \subseteq L(S)$ definiuje się w następujący sposób:

1. s jest drzewem DLS z jednym węzłem oznaczającym, że sekwencja genu jest obecna w gatunku s,
2. A- jest drzewem DLS z jednym węzłem oznaczającym stratę genu, gdzie A jest niepustym zbiorem gatunków,
3. (R_1,R_2)+ jest drzewem DLS, którego korzeń jest węzłem duplikacyjnym i dzieci korzenia R_1 i R_2 są drzewami DLS takimi, że L(R_1) = L(R_2),
4. (R_1,R_2)$\sim$ jest drzewem DLS, którego korzeń jest węzłem specjacyjnym i dzieci korzenia R_1 i R_2 są drzewami DLS takimi, że L(R_1) $\cap$ L(R_2) = 0.

Należy zaznaczyć, że z każdego drzewa DLS możliwe jest odczytanie drzewa genów za pomocą którego zostało zbudowane drzewo DLS [3].

Klasyczną miarą kosztu ewolucyjnego danego drzewa DLS jest liczba duplikacji potrzebnych do uzgodnienia drzewa genów i gatunków.

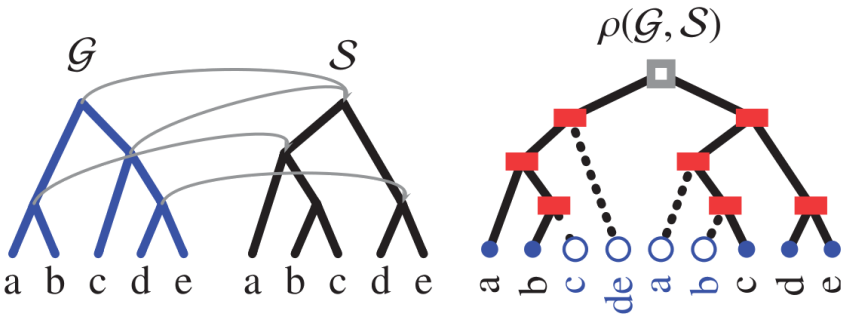
1.3.3. Scenariusz LCA

Z wykorzystaniem mapowania LCA węzłów drzewa genów G do drzewa gatunków S można stworzyć drzewo DLS, które reprezentuje scenariusz LCA, to znaczy taki, który zawiera minimalną liczbę duplikacji potrzebnych do uzgodninienia drzew [**?**].

Definicja 1.3.2 *Scenariusz LCA dla drzewa G o korzeniu g i drzewa S o korzeniu s, gdzie $L(G) \subseteq L(S)$, jest drzewem DLS $p(g, MAP_{LCA}(g))$, takim, że $p(g, s) = s$ kiedy g i s są liśćmi oraz etykietą g jest s. W innym przypadku:*

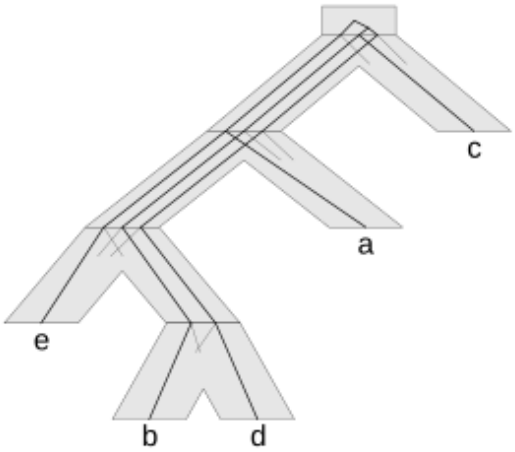
$$p(g,s) = \begin{cases} (p(g,u), L(T(v))-) \sim & \text{jeśli } MAP_{LCA}(g) \in T(u), \quad (LOSS) \\ (p(z,u), p(q,v)) \sim & \text{jeśli } MAP_{LCA}(z) \in T(u) \wedge MAP_{LCA}(q) \in T(v), \quad (SPEC) \\ (p(z,s), p(q,s))+ & \text{jeśli } MAP_{LCA}(g) = MAP_{LCA}(z) = s, \quad (DUP) \end{cases}$$

gdzie u i v są dziećmi s a z i q są dziećmi g.



Rysunek 1.3: Drzewo DLS $p(\mathcal{G},\mathcal{S})$ będące scenariuszem zbudowanym w oparciu o mapowanie LCA, które uzgadnia drzewo genów $\mathcal{G}$ i drzewo gatunków $\mathcal{S}$. Węzeł oznaczony szarym kwadratem jest węzłem duplikacyjnym, a czerwony kwadrat jest specjacją. Gałęzie narysowane linią przerywaną prowadzą do liści, które są reprezentacją straty genów w danym gatunku. Rysunek pochodzi z pracy [3].

Przedstawieniem drzewa DLS jest "wbudowanie"drzewa genów, w kontener będący drzewem gatunków.

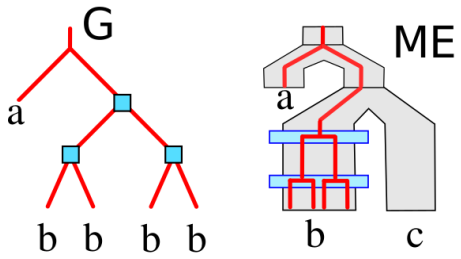


Rysunek 1.4: Przykład wbudowania dla drzew z rysunku 1.2 [1].

1.4. Klastrowanie duplikacji

Problemem podczas poszukiwania drzew DLS o najmniejszym koszcie ewolucyjnym jest odpowiednie klastrowanie duplikacji, które następują bezpośrednio po sobie. Zależnie od obranej

metody klastrowania ich liczba na danym węźle drzewa gatunków może się znacząco różnić. W pracy tej zakłada się, że dzieci nie mogą znajdować się w tym samym klastrze co ojciec (klastrowanie ME).



Rysunek 1.5: Klastrowanie ME dla przykładowego drzewa genów G. Rysunek zaadaptowany z pracy [4].

Niech S to drzewo gatunków, a $\mathcal{G} = \{G_1, G_2, ..., G_n\}$ to drzewa genów gdzie $L(G_i) \subseteq L(S)$ dla każdego i. Niech $\mathcal{T} = \{T_1, T_2, ..., T_n\}$ to drzewa DLS stworzone w oparciu o $\mathcal{G}$. Wówczas mówimy, że dwa węzły duplikacyjne d_1 i d_2 z $\mathcal{T}$ można sklastrować (oznaczymy jako $d_1 \Longleftrightarrow d_2$) jeśli poniższe warunki są spełnione:

1. $L(d_1) = L(d_2)$,
2. d_1 i d_2 nie leżą na jednej ścieżce (są nieporównywalne) albo $d_1 = d_2$.

MEScore($\mathcal{T}$, S) to minimalna wielkość podziału zbioru wszystkich węzłów duplikacyjnych obecnych w scenariuszach tak, że każde dwie duplikacje z tego samego podziału są klastrowalne.

Definicja 1.4.1 *MEScore($\mathcal{G}$, S)* $= \min_{\cup P = Dup(\mathcal{G})}\{|P| : \forall_{A \in P} \forall_{d_1, d_2 \in A} d_1 \Longleftrightarrow d_2\}$ *gdzie Dup($\mathcal{G}$) jest zbiorem wszystkich węzłów duplikacyjnych obecnych w $\mathcal{G}$.*

Należy wspomnieć, że klastrowanie ME nie jest jedynym możliwym rodzajem. Istnieją również klastrowania:

- GD, gdzie każda duplikacja musi zostać opisana jako oddzielny klaster.
- EC, gdzie wszystkie duplikacje następujące po sobie są sklastrowane razem.

Definicje na podstawie pracy [4].

1.5. Modele scenariuszy ewolucyjnych

Drzewo DLS, 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. Formalnie drzewo DLS nazywamy semi-normalnym jeśli:

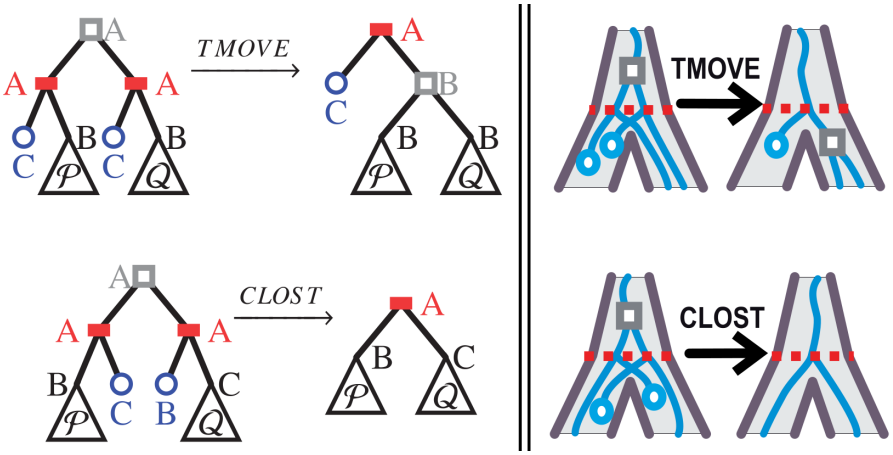
1. nie istnieje węzeł duplikacyjny, którego dzieckiem jest strata,
2. nie istnieje węzeł specjacyjny, którego dzieci są stratami.

1.5.1. Transformacje scenariuszy ewolucyjnych

Drzewo DLS podlega ściśle określonym transformacjom (zwanych również ruchami), które zmieniają jego strukturę, ale zachowując poprawność danego drzewa:

1. TMOVE: $((C-,P)\sim,(C-,Q)\sim))+ \Longrightarrow (C-,(P,Q)+)\sim$,
2. CLOST: $((P,L(Q)-)\sim,(L(P)-,Q)\sim)+ \Longrightarrow (P,Q)\sim$,

gdzie C, P, Q są drzewami DLS.



Rysunek 1.6: Ruchy TMOVE i CLOST wraz z ich biologiczną interpretacją. Szary kwadrat reprezentuje duplikację. Czerwony prostokąt i czerwona linia przerywana są reprezentacją specjacji, a niebieskie koło jest stratą. Rysunek zaadaptowany z pracy [3].

Stosowane są również odwrotności zdefiniowanych powyżej transformacji, które oznaczone są indeksem górnym $^{-1}$ np. TMOVE^{-1}.

Dla ustalonego drzewa genów G i ustalonego drzewa gatunków S, gdzie $L(G) \subseteq L(S)$, możliwe jest uzyskanie zbioru scenariuszy semi-normalnych, który reprezentowany jest jako zbiór drzew DLS, poprzez zastosowanie ruchów TMOVE^{-1} i CLOST^{-1} na drzewie DLS będącym scenariuszem LCA.

Na rysunku 1.7 przedstawione są wszystkie możliwe przekształcenia TMOVE i CLOST dla przykładowych drzew G i S.

1.5.2. Opis modeli dopuszczalnych scenariuszy

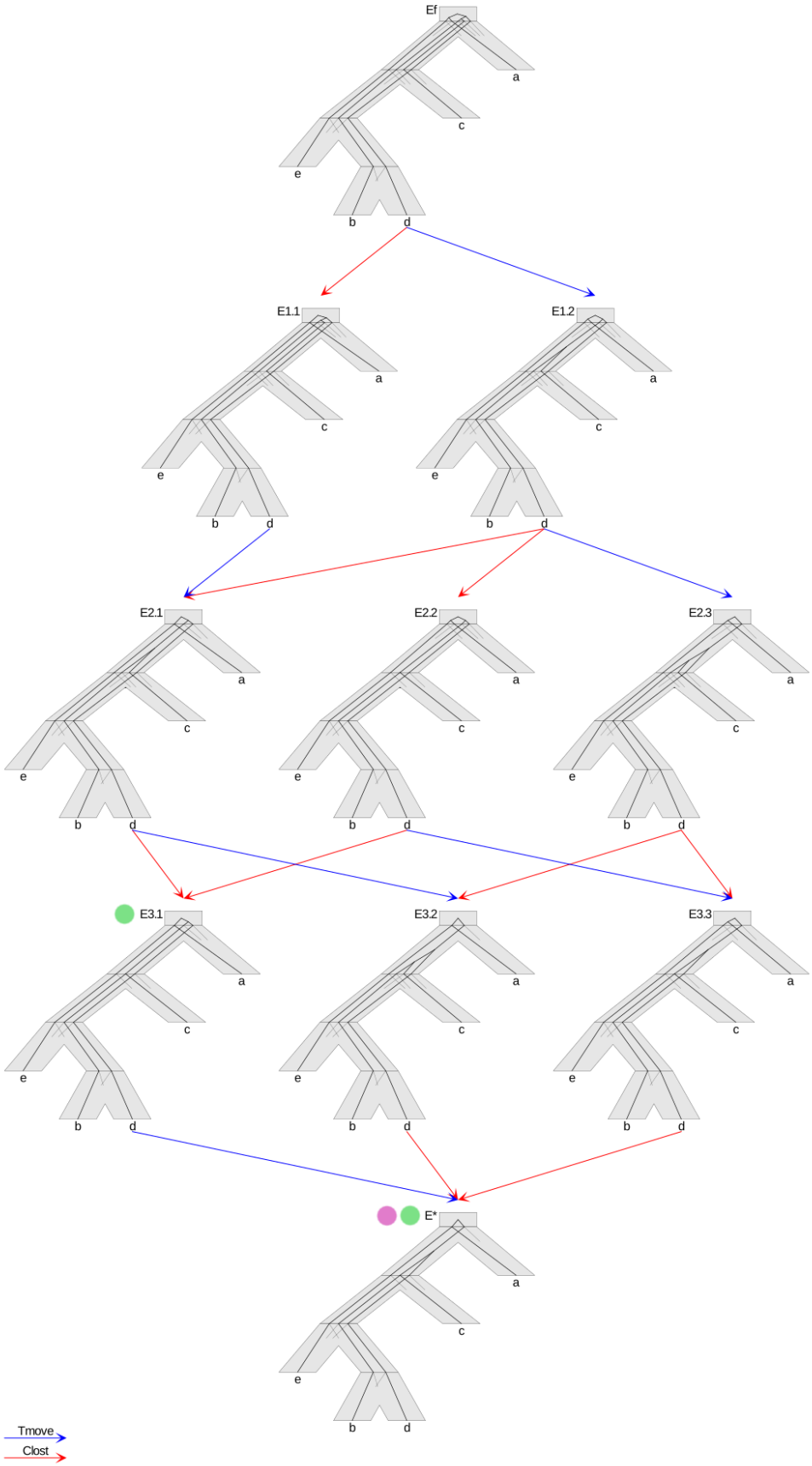
W praktyce nie zawsze rozważa się wszystkie możliwe scenariusze do klastrowania, ponieważ może prowadzić to do nieprawdopodobnych klastrowań duplikacji np. gdzie wszystkie duplikacje ulokowane zostały w korzeniu drzewa DLS.

W literaturze wyodrębniono kilka dopuszczalnych modeli, a w pracy tej zajęto się podanymi niżej modelami, które dopuszczają podane warunki:

1. Model PG [4]: Model ten dopuszcza tylko takie drzewa DLS, które zachowują minimalny koszt duplikacyjny. Oznacza to, że są to scenariusze osiągalne ze scenariusza LCA reprezentowanego drzewem DLS za pomocą transformacji TMOVE^{-1}.

2. Model FHS [4]: Model ten dopuszcza każde możliwe przekształcenie drzew DLS, nawet jeśli takie drzewo nie zachowuje minimalnego kosztu duplikacyjnego. Oznacza to, że są

to scenariusze osiągalne ze scenariusza LCA reprezentowanego drzewem DLS za pomocą transformacji TMOVE^{-1} i CLOST^{-1}.



Rysunek 1.7: Diagram przekształceń scenariuszy semi-normalnych. Na dole rysunku, oznaczone filetowym kołem, znajduje się drzewo uzyskane za pomocą mapowania LCA. Model PG zawierać będzie tylko drzewa oznaczone zielonym kołem, podczas gdy dla modelu FHS wszystkie drzewa obecne na diagramie są częścią zboru scenariuszy semi-normalnych [1].

Rozdział 2

Heurystyka

2.1. Opis problemu

Ogólnym problemem, niezależnym od obranego modelu, jest szereg komplikacji, które mogą zaistnieć przed samymi obliczeniami drzew DLS. Do utrudnień mogą należeć między innymi:

1. braki w samych sekwencjach genomowych,

2. straty genów i ich brak w materiale kopalnym,

3. błędy sekwencjonowania,

4. metody obliczania drzew genów i gatunków są heurystykami, a przez to możliwe są błędy.

Szczególnie dwa pierwsze punkty związane z brakiem sekwencji są problematyczne dla badaczy, ponieważ w przypadku straty genów duplikacje genomowe będą położone niżej na drzewie DLS, co nie zawsze musi odpowiadać rzeczywistości. Z tego powodu często okazuje się, że klastrowania są bardzo podobne do tego wyznaczonego przez mapowanie LCA. Nie zawsze jest to rozwiązaniem poszukiwanym i zbliżonym do rzeczywistego. Należy też pamiętać, że specjacja jest górnym ograniczeniem podczas przekształcania drzewa DLS. Nie zawsze możliwe jest przesunięcie duplikacji bliżej korzenia i przekroczenie bariery specjacji, ponieważ taki scenariusz nie byłby semi-normalny.
W pracy tej proponowana jest nowa heurystyka, która

2.2. Opis algorytmu

Input:

1. Drzewo gatunków S z węzłami $< s_1, s_2, ..., s_m >$,
2. Drzewa genów $G_1, G_2, ..., G_z$ gdzie $L(G_i) \subseteq L(S)$,
3. Dla każdego drzewa genów G_i zbiór scenariuszy R_i w postaci wektorów $< {e_i}^1, {e_i}^2, ..., {e_i}^m >$, gdzie dla każdego j, ${e_i}^j$ to liczba epizodów duplikacyjnych przypisanych do węzła s_j w i-tym scenariuszu.

Output:

1. Scenariusz w postaci wektora $< e^1, e^2, ..., e^m >$, gdzie dla każdego j, e^j to minimalna liczba epizodów duplikacyjnych przypisanych do węzła s_j.

V^{max} = wektor $< e^1, e^2, ..., e^m >$, gdzie dla każdego j, e^j to maksymalna liczba epizodów duplikacyjnych przypisanych do węzła s_j w każdym scenariuszu R_i dla każdego drzewa G.;
$V^* = V^{max}$;
for $s_x \in S$ *w ustalonej kolejności* **do**
    ${V_{new}}^* = V^*$;
    while *zmiana w węźle* ${V_{new}}^*[s_x]$ *i* ${V_{new}}^*[s_x] > 0$ **do**
        ${V_{new}}^*[s_x]$ - -;
        if ${V_{new}}^* \subseteq R_i$ *dla każdego drzewa* G **then**
            $V^* = {V_{new}}^*$;
        end
    end
end

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

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

2.3. Dokumentacja użytkowa i opis implementacji

Opisana heurystyka zaimplementowana została w języku Python w wersji 3.7.4 przy użyciu paradygmatu obiektowego, gdzie zbiór wszystkich drzew DLS dla wszystkich drzew, zbiór drzew semi-normalnych otrzymanych z jednego drzewa genów, a także pojedyncze drzewo DLS stanowią oddzielne klasy. Fragment kodu można zobaczyć w dodatku A. Program pythonowy przyjmuje na wejściu listę plików w których zawarte są wyliczone scenariusze dla danego drzewa genów.

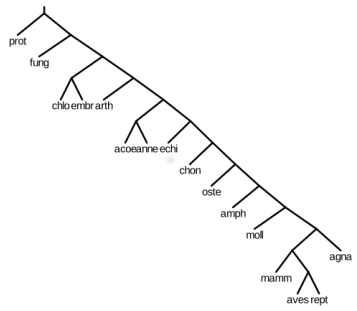
Algorytm ten, dla wygody użycia, został obudowany skryptem napisanym w języku bash, który pozwala na wyliczenie scenariuszy w modelu FHS i PG z wykorzystaniem programu DLSgen autorstwa dra hab. Pawła Góreckiego.[?] Program ten wylicza dla danego drzewa genów i drzewa gatunków scenariusze ewolucyjne,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

2.4. Testy algorytmu

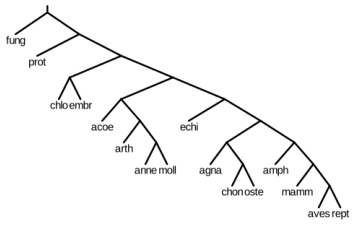
Do sprawdzenia wyników wyliczanych przez proponowany algorytm użyty został program RME napisany przez dra Jarosława Paszka. Program ten korzystając z dostępnych metod algorytmicznych wylicza dokładne i najniższe możliwe wartości kosztu ewolucyjnego dla podanych modeli z użyciem klastrowania ME. [2]
Z 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.

2.4.1. Testy algorytmu na danych rzeczywistych

Zbiorem danych dla testu na danych rzeczywistych był zbiór Guigo zawierający 53 ukorzenione drzewa genów pochodzące od 16 eukariontów. Do zbioru tego załączono dwa drzewa gatunków S_1 z pracy [5] i S_2 z pracy [6].

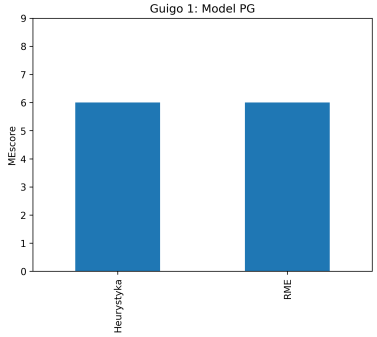


(a) Drzewo S_1

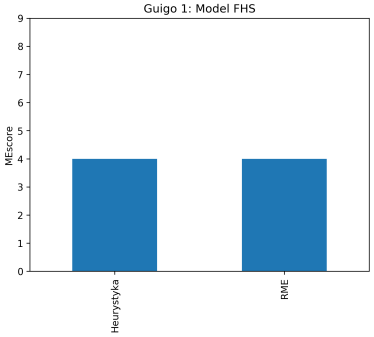


(b) Drzewo S_2

Rysunek 2.1: Drzewa gatunków ze zbioru Guigo

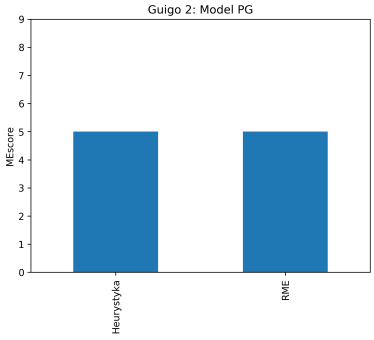


(a) Test algorytmu dla modelu PG

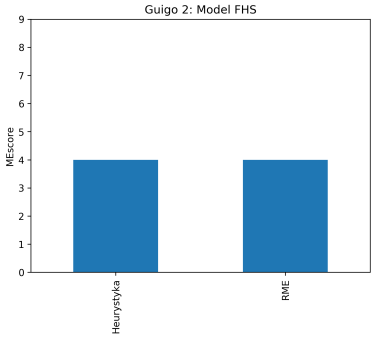


(b) Test algorytmu dla modelu FHS

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



(a) Test algorytmu dla modelu PG



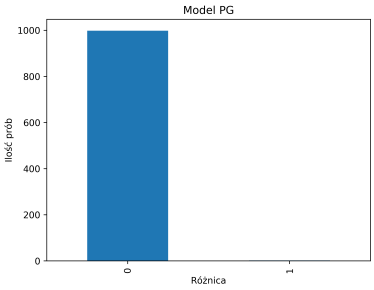
(b) Test algorytmu dla modelu FHS

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

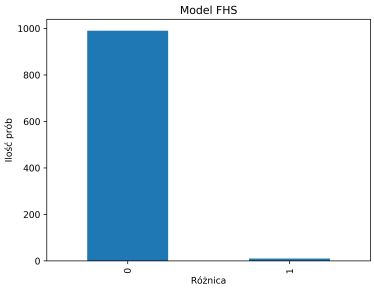
Dla wszystkich przypadków algorytm przy losowym wyborze współrzędnych jest w stanie osiągnąć wyniki identyczne jak te wyliczone przez program RME.

2.4.2. 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 liśćmi oraz wygenerowane losowo, w oparciu o algorytm YULA, 48 ukorzenione drzewa genów etykietowane gatunkami z wylosowanego wcześniej drzewa 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 reprezentowanymi drzewami DLS pod kątem ilości duplikacji. Należy jednak wspomnieć, że samą ideę da się znacząco usprawnić.
Obecnie to krok w którym konieczne jest wyliczenie zbioru drzew semi-normalnych dla danego drzewa 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. Należy jednak podkreślić, że ponieważ to właśnie program DLSgen był swego rodzaju "wąskim gardłemńiemożliwe jest przetestowanie algorytmu na danych bardziej złożonych niż zbiór Guigo lub jego pochodne.
Testy pokazują również, że opisany algorytm zwraca wyniki, które różnią się w bardzo niewielkim stopniu 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 drzew DLS zawartych w badanych zbiorach, który wynosił maksymalnie 9 (średnia arytmetyczna = 6,1) dla modelu PG i 7 (średnia arytmetyczna = 4,5) dla modelu FHS. Proponowany algorytm wymaga w związku z tym kolejnych, bardziej rozbudowanych testów.

3.1. Perspektywy rozwoju

Trudno przewidzieć wszystkie możliwości rozwoju algorytmu, ale tę bardziej oczywistą można wskazać już teraz. Jest to uniezależnienie algorytmu od kroku w którym wyliczane są scenariusze i klastrowanie duplikacji bezpośrednio na podstawie drzew genów. Jest to krok kluczowy dla dalszego rozwoju heurystyki, ponieważ pozwoli on na używanie jej przy rozwiązywaniu rzeczywistych problemów. Będzie możliwe wtedy również przeprowadzenie testów dla 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. Drzewo binarne nie zawsze jest najlepszym przedstawieniem 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. Zapewnia to możliwość wykorzystania algorytmu nie tylko dla drzew binarnych. Algorytm funkcjonuje obecnie jednak w dosyć prymitywnej formie i wymaga wzmożonej oraz dokładnej pracy.

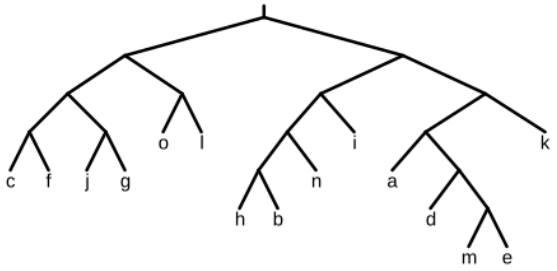
Dodatek A

Pętla programu zapisana w języku Python wykonywana dla losowego wybierania indeksów

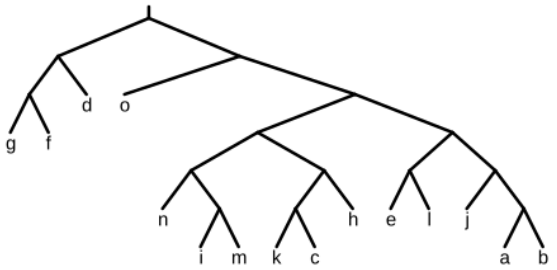
```
max_trees = []
        for scenario in self:
            all_dup_pref = [tree.duplication_prefix for tree n scenario]
            max_trees.append(self.rate_scenario(all_dup_pref))
        max_tree = self.rate_scenario(max_trees)

        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] Wygenerowane za pomocą serwisu: http://gsevol.azor.mimuw.edu.pl

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

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

[4] Jarosław Paszek, Paweł Górecki: *Efficient Algorithms for Genomic Duplication Models*

[5] Guigo, R., Muchnik, I.B., Smith, T.F.: *Reconstruction of ancient molecular phylogeny*, Molecular Phylogenetics and Evolution 6(2), 189–213 (1996)

[6] Page, R.D.M., Charleston, M.A.: *Reconciled trees and incongruent gene and species trees*, DIMACS Series in Discrete Mathematics and Theoretical Computer Sciences, vol. 37 (1997)