Katedra Aplikovanej informatiky
Fakulta Matematiky, Fyziky a Informatiky
Univerzita Komenského, Bratislava

Algoritmy na rekonštrukciu genómov pomocou preusporiadania

Diplomová práca

2014 Bc. Veronika Ženišová

Katedra Aplikovanej informatiky
Fakulta Matematiky, Fyziky a Informatiky
Univerzita Komenského, Bratislava

Algoritmy na rekonštrukciu genómov pomocou preusporiadania

(Diplomová práca)

Evidenčné číslo: 4fbddf6b-0583-4e14-9049-2c75fa573c56
Študijný program: Aplikovaná informatika
Študijný odbor: 2511 Aplikovaná informatika
Školiace pracovisko: Katedra aplikovanej informatiky
Školiteľ: Mgr. Tomáš Vinař, PhD.

Bratislava, 2014 Bc. Veronika Ženišová

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Veronika Ženišová
Študijný program: aplikovaná informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: 9.2.9. aplikovaná informatika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: slovenský

Názov: Algoritmy na rekonštrukciu genómov pomocou preusporiadania

Cieľ: Cieľom práce je preštudovať metódy rekonštrukcie evolučných histórií pomocou operácie reverzie, naimplementovať algoritmus umožňujúci takúto analýzu a aplikovať ho na reálne dáta z kvasinkových mitorchondriálnych genómov.

Vedúci: Mgr. Tomáš Vinař, PhD.

Dátum zadania: 27.11.2008

Dátum schválenia: 29.04.2011

doc. RNDr. Roman Ďurikovič, PhD.
garant študijného programu

.....
študent

.....
vedúci

Prehlasujem, že túto diplomovú prácu som vypracovala samostatne s použitím citovaných zdrojov.

V Bratislave, 5. mája 2014.

.....

Veronika Ženišová

Za podporu a povzbudenie chcem poďakovať školiteľovi
Mgr. Tomášovi Vinařovi, PhD. a taktiež môjmu manželovi.

Abstrakt

Autor:	Bc. Veronika Ženišová
Názov:	Algoritmy na rekonštrukciu genómov pomocou preusporiadania
Univerzita:	Univerzita Komenského v Bratislave
Fakulta:	Fakulta matematiky, fyziky a informatiky
Katedra:	Katedra aplikovanej informatiky
Školiteľ:	Mgr. Tomáš Vinař, PhD.
Počet strán:	62
Rok:	2014

Diplomová práca sa zaoberá rekonštrukciou evolučnej histórie pomocou operácie reverzie. Ide o hľadanie najmenšej reverzálnej vzdialenosti, čo je najmenší počet krokov potrebných k transformácii genómu na iný za použitia operácie reverzie. Na základe hodnôt reverzálnych vzdialeností je pomocou programu PIVO: Phylogeny by IteratiVe Optimization generovaný fylogenetický strom skúmaných genómov a ich predkov. V práci sú stručne opísané princípy príbuzných modelov riešenia, základné štruktúry a metódy implementovaného algoritmu, samotná implementácia a výsledky behu algoritmu na mitochondriálnych cirkulárnych genómoch kvasiniek triedy Hemiascomycetes.

Kľúčové slová: genóm, preusporiadanie genómu, reverzia, reverzálna vzdialenosť

Abstract

Author:	Bc. Veronika Ženišová
Title:	Algorithms for genome reconstruction by reversals
University:	Comenius University, Bratislava
Faculty:	Faculty of Mathematics, Physics and Informatics
Departement:	Department of Applied Informatics
Supervisor:	Mgr. Tomáš Vinař, PhD.
Number of Pages:	62
Year:	2014

This Diploma Thesis focuses on reconstruction of evolutionary history by operation of reversal. The minimal count of changes needed to transform one genome into another is called reversal distance. This Diploma Thesis proposes a model of solution of the problem of finding the shortest reversal distance between two recent genomes, short description of relative models and description of implemented model. We have made an implementation in Java platform and common with software PIVO: Phylogeny by IteratiVe Optimization, we generate a phylogeny tree of 16 circular mitochondrial genomes of Hemiascomycetes.

Keywords: genome, genome rearrangement, reversal, reversal distance

Obsah

Zoznam obrázkov

Zoznam tabuliek

1 Úvod

Bioinformatika patrí v súčasnosti k vedeckým odborom, ktoré sa vyvíjajú veľmi rýchlo. Zaoberá sa skúmaním biologických dát najmä z oblasti molekulárnej biológie, pričom pomocou matematických a štatistických prístupov a algoritmov využíva výpočtovú silu počítača. Namiesto toho, aby vedec-biológ riešil zdĺhavé výpočty pomocou papiera a pera, používa rýchly a presný algoritmus. Problémom však zostáva zadefinovať postupy, ktoré by verne reprezentovali javy odohrávajúce sa v prírode.

1.1 Motivácia

Jedným z veľkých problémov v bioinformatike je rekonštrukcia evolučných zmien v genóme na základe štruktúry DNA. Využíva sa fakt, že genóm ako celok je tvorený segmentmi, ktoré je možné identifikovať, označiť. Tieto segmenty sa nazývajú markéry. Aj keď sa markéry postupom evolúcie rôzne modifikujú mutáciami, ich charakteristické črty, vlastnosti (napríklad produkty génov v nich obsiahnutých – RNA a bielkoviny) zostávajú v genóme prítomné. Porovnávanie genómov dvoch druhov môže priniesť nový pohľad na fylogenézu, vývoj druhov, ktorý bol v minulosti definovaný najmä podľa paleontologických nálezov.

Dva organizmy, ktoré vznikli zo spoločného predka jedinou výraznou zmenou, sú si geneticky viac podobné ako iné dva, ktorých genóm prešiel od momentu spoločného predka mnohými zmenami. Veľmi zjednodušene môžeme povedať, že miera príbuznosti dvoch organizmov je priamo úmerná počtu zmien, ktoré spôsobili transformáciu jedného genómu na iný. Ak by sme si to chceli vysvetliť na príklade: koncom 80-tych rokov minulého storočia skupina

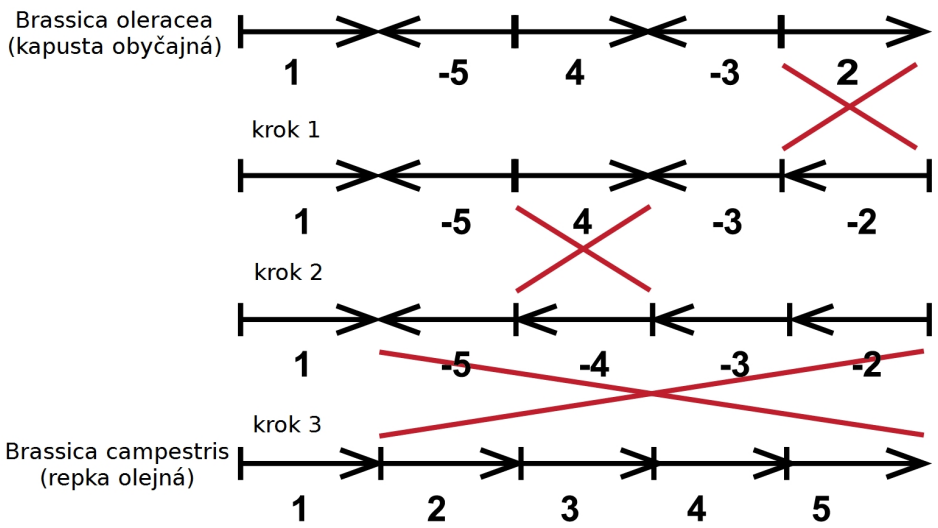
vedcov porovnala dva evolučne príbuzné druhy – *kapustu obyčajnú* a *repku olejnú*. Zistili, že ich jednotlivé gény, úseky genómu, sa zhodujú v 99%. No aj napriek veľkej podobnosti génov sa ich genómy, teda postupnosti génov, výrazne odlišujú, a to v poradí génov. (Hannenhalli – Pevzner, 1999) Je zrejmé, že ak by bol genóm chápaný ako postupnosť väčších stálych blokov a zmeny sa udiali len v preusporiadaní, otočení, či inej obdobnej zmene týchto blokov, stačí pre rekonštrukciu evolúcie nasimulovať zmeny ovplyvňujúce bloky genómu – markéry.

Označme skupiny génov markérmi – na obrázku 1.1 sú vyjadrené číslami. Podľa toho, na ktorom vlákne DNA sa daný markér vyskytuje, nadobúda kladnú, alebo zápornú orientáciu. Vieme, že vlákna DNA sú navzájom komplementárne. Opačne orientované bloky zachovávajú informáciu, rozdielom je, že gény v rámci nich majú navzájom opačné poradie. Môžeme vidieť, že genómy kapusty a repky ako sekvencie markérov vieme pretransformovať jeden na druhý len v troch krokoch. Počet potrebných krokov označujeme ako *genómová vzdialenosť*[1]. V prípade repky a kapusty je rovná 3 a toto relatívne malé číslo je dôkazom blízkej genómovej príbuznosti spomínaných druhov. Pre porovnanie podľa Hannenhalli – Pevzner (1995) je možné genóm človeka pretransformovať na genóm myši pomocou 131 zmien.

Príbuznosť organizmov sa v biológii vyjadruje pomocou štruktúry fylogenetického stromu, ktorý je v podstate zobrazením biologickej taxonómie, histórie vývoja organických štruktúr. Jednotliví zástupcovia vetiev sú natoľko príbuzní, ako vzdialení sú v strome. Na obrázku 1.2 je zobrazený fylogenetický strom so znázornenými hlavnými evolučnými taxónmi[2] biológie.

[1] angl. genomic distance

[2] taxón je vrchol vo fylogenetickom strome, ktorý zahŕňa aj všetkých predstaviteľov podstromu s koreňom v tomto vrchole; je to skupina druhov istej evolučnej vetvy, napr. trieda, rod, kmeň



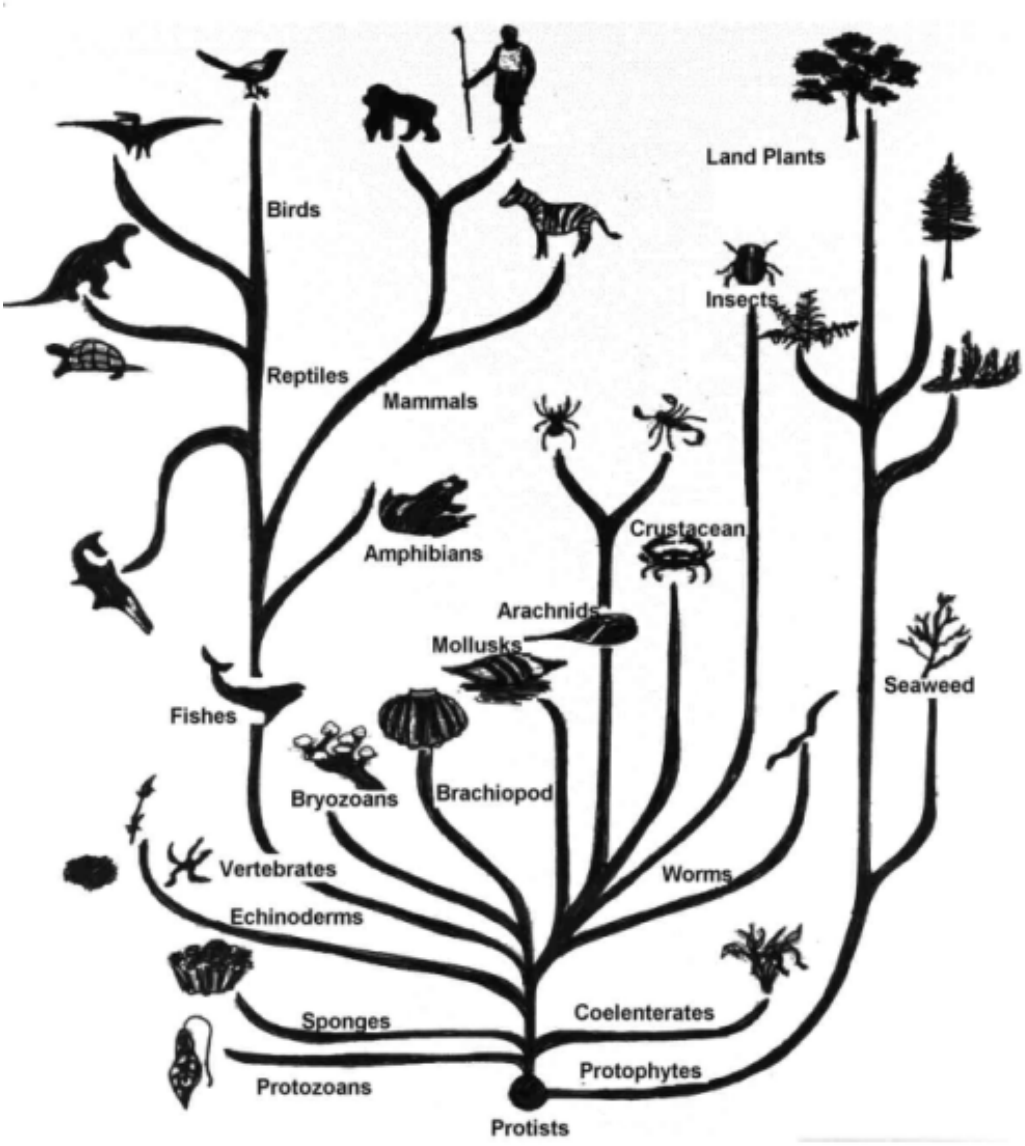
Obr. 1.1: **Transformácia genómu kapusty obyčajnej na repku olejnú** (Pevzner, 2000)

1.2 Ciele práce

Ako sme už spomenuli aj v predchádzajúcom texte, hľadanie genómovej vzdialenosti dvoch druhov je zaujímavým bioinformatickým problémom.

V tejto diplomovej práci budeme popisovať metódy hľadania genómovej vzdialenosti ako vstupného údaju na rekonštrukciu evolučnej histórie. Jednej z možných metód – rekonštrukcii evolučnej histórie pomocou operácie reverzie – sa budeme venovať podrobne a ponúkneme popis algoritmu riešenia na základe tejto metódy a jeho implementáciu.

Na záver zhrnieme a zanalyzujeme výsledky získané ako výstup z naimplementovaného algoritmu, kde ako vstupné údaje použijeme reálne dáta mitochondriálnych genómov kvasiniek.



Obr. 1.2: **Fylogenetický strom**
(Thomas, 2011)

1.3 Prehľad literatúry

Samotnú myšlienku transformácie genómu pomocou preusporiadavania blokov génov vyjadrili ako prvý Dobzhansky a Sturtevant (1938), keď pomocou 17-stich inverzií transformovali genóm muchy rodu Drosophila na iný genóm muchy rovnakého rodu. Išlo o chápanie genómu ako sekvencie konzervovaných blokov génov, nie jednoduchej sekvencie nukleotidov ako doposiaľ. Aj keď nový pohľad na genóm bol pokrokový, manuálne preusporiadavanie väčších genómov nebolo reálne.

Postupom času sa objavovali rôzne riešenia problému, ktoré sa od seba odlišovali nie len zložitosťou, ale aj spôsobom chápania entít, modelom riešenia problému. Principiálne ide vždy o hľadanie najmenšieho počtu krokov potrebných k transformácii genómu na iný, avšak modely sa líšia najmä definovanými operáciami.

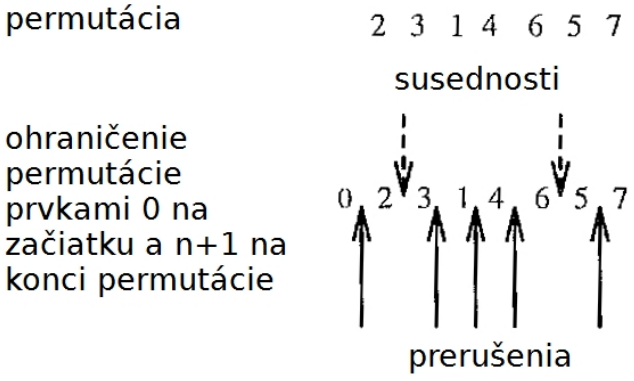
V tejto časti sa budeme venovať popisu niektorých ťažiskových modelov hľadania najkratšej reverzálnej vzdialenosti.

Dekompozícia maximálneho cyklu v grafe prerušení

Metóda je popísaná a postupne rozvíjaná predovšetkým v článkoch Transforming men into mice (Hannenhalli – Pevzner, 1995), Transforming cabbage into turnip (Hannenhalli – Pevzner, 1999) a Computational Molecular Biology: An Algorithmic Approach (Pevzner, 2000). V tejto práci sú definované mnohé dátové štruktúry, ktoré boli použité aj v ďalších prístupoch a aj v našej práci.

V modeli chápeme permutáciu ako postupnosť markérov. Na oboch koncoch markéry 0 a n+1, kde n je počet markérov v permutácii, vyjadrujú ukončenia genómu. Významovo reprezentujú biologické teloméry. Našim cieľom je preusporiadať elementy tak, aby sme dostali sekvenciu rastúcich čísel, ktorá zjednodušene reprezentuje utriedený genóm.

Medzi každými dvoma markérmi v sekvencii môže byť buď susednosť[3], alebo prerušenie[4]:
Nech permutácia $\pi = (\pi_1\pi_2 \dots \pi_n)$. Hovoríme, že medzi elementmi π_i a π_{i+1} je susednosť, ak $|\pi_i - \pi_{i+1}| = 1$, $0 \leq i < n$, teda rozdiel ich absolútnych hodnôt je 1. Inak je medzi týmito elementmi prerušenie. (Hannenhalli – Pevzner, 1999) To znamená, že ak elementy nasledujú v usporiadanom rade za sebou, nachádzame tu susednosť a ak je postupnosť pretrhnutá, je medzi elementmi prerušenie (obr.1.3). Počet prerušení a miera neusporiadanosti bola daná do súvisu už v prvom článku popisujúcom tento problém – Sturtevant – Dobzhansky (1936). Teda čím viac prerušení, tým viac operácií je potrebných na usporiadanie sekvencie.

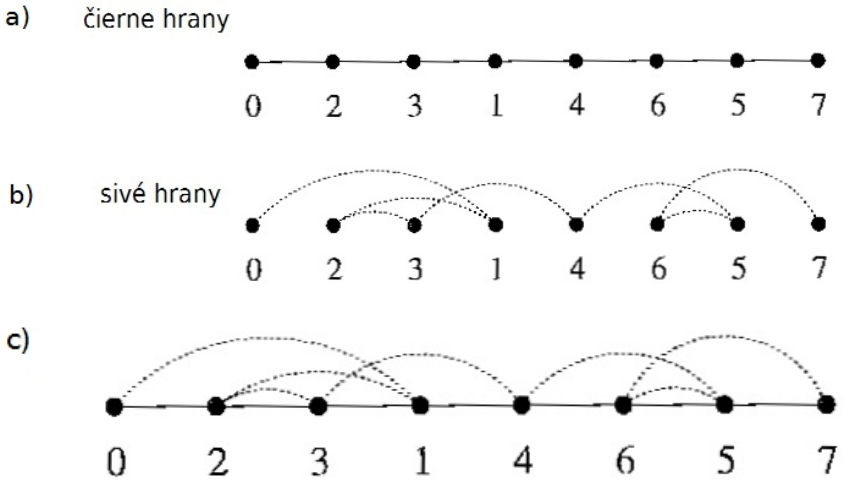


Obr. 1.3: **Sekvencia s miestami postupností a prerušení**
(Pevzner, 2000, str.181)

Keď máme permutáciu takto označenú hraničnými bodmi, môžeme vytvoriť graf prerušení. V tomto grafe sú dva typy hrán – sivé a čierne. Nech permutácia $\pi = (\pi_1\pi_2 \dots \pi_n)$. Čierne hrany spájajú elementy permutácie π_i a π_{i+1}, kde $0 \leq i \leq n$ a sivé hrany spájajú dva vrcholy π_i a π_j také, že medzi π_i a π_j je susednosť v analyzovanom genóme, $0 \leq \{i, j\} \leq n$ a bez ujmy na všeobecnosť $j \neq i + 1$". Hrany v grafe vytvárajú cykly (obr.1.4). V časti a) sú zvýraznené čierne hrany grafu, v časti b) sú zaznačené sivé hrany a c)

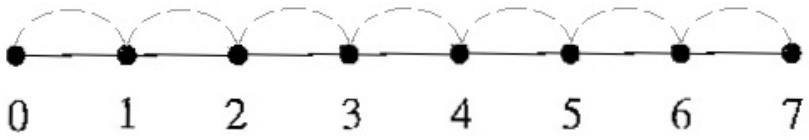
[3]angl. adjacency
[4]angl. breakpoint

znázorňuje spojenie oboch typov hrán do výsledného grafu.



Obr. 1.4: **Čierne a sivé hrany v grafe prerušení**
(Pevzner, 2000, str.181)

V grafe usporiadanej permutácie budú mať všetky cykly dĺžku 2, budú tvorené vždy jednou sivou a jednou čiernou hranou (obr.1.5). Každá hrana patrí práve do jedného cyklu. V grafe prerušení neusporiadanej permutácie sa pomocou sivých a čiernych hrán vytvoria medzi vrcholmi cykly väčšej dĺžky ako 2. Podľa Hannenhalli – Pevzner (1995, 1999); Pevzner (2000) sa k utriedenej permutácii dopracujeme práve dekompozíciou maximálneho cyklu v grafe prerušení. Počet krokov potrebných na riešenie dekompozície je totožný s počtom krokov potrebných na transformáciu genómu. Dekompozícia cyklu súvisí s vytváraním nových cyklov menšej dĺžky, cieľom je dosiahnuť $n + 1$ cyklov, kde n je dĺžka pôvodného genómu bez telomér.



Obr. 1.5: **Graf prerušení usporiadanej sekvencie**

Popísali sme model riešenia, jeho hlavné entity a žiadaný cieľový stav. Zostáva popísať krok výpočtu, povolené operácie, ktorými je možné sa do spomínaného cieľového stavu dopracovať. Pre tento model je definovanou operáciou reverzia: Nech permutácia $\pi = (\pi_1, \pi_2, \ldots \pi_i, \ldots \pi_j, \ldots \pi_n)$. Operácia reverzie $\rho(i, j)$ nad *oznamienkovaným* podintervalom ohraničenom indexmi i, j permutácie π vykoná zmenu v sekvencii nasledovne (Pevzner, 2000):

$$\begin{aligned} \pi &= (\pi_1 \ \ldots \ \underline{\pi_i \ \pi_{i+1} \ \ldots \ \pi_j} \ \ldots \ \pi_n) \\ \pi \cdot \rho(i,j) &= \pi' \\ \pi' &= (\pi_1 \ \ldots \ \underline{-\pi_j \ \ldots \ -\pi_{i+1} \ -\pi_i} \ \ldots \ \pi_n) \end{aligned}$$

Použitím reverzie na ľubovoľnom podintervale sekvencie ale nie je zaručené, že dosiahneme požadovaný stav. Je treba zvoliť taký interval, ktorý výslednú sekvenciu posunie smerom k cieľovej. Aby sme takýto krok rozpoznali, definujeme operáciu bezpečnej reverzie, ktorá po vykonaní nad permutáciou π zníži počet cyklov, ktorých dĺžka je väčšia ako 2 o jeden: $\Delta c = c(\pi\rho) - c(\pi) = -1$, kde Δc je zmena počtu cyklov definovaná ako počet cyklov v permutácii π pred a po vykonaní reverzie ρ, ďalej $c(\pi)$ je počet cyklov v permutácii π a nakoniec $\rho(i, j)$ je operácia reverzie nad podintervalom permutácie π definovanom hraničnými prvkami i a j. Ak sa eliminuje jeden z cyklov, ktorý treba ďalej spracovávať, znamená to, že sa eliminovali aj dve prerušenia na jeho koncoch, teda klesá aj celkový počet prerušení v permutácii. (Hannenhalli – Pevzner, 1999)

Popísaným spôsobom, však, nie je možné dekomponovať cykly, ktoré sú označené ako prekážky (angl. hurdles). Sú to spojité neorientované cykly, ktoré je potrebné upraviť istým spôsobom tak, aby sme ich podinterval dokázali utriediť. Keďže obsahujú iba neorientované prvky, musíme zvoliť iný prístup ako reverziu nad orientovaným párom. Prekážky odstraňujeme tzv. vystrihovaním a spájaním, teda vykonaním reverzie na podintervale prekážky, alebo intervale medzi dvoma prekážkami. Pre popísanie princípu modelu ich nie je potrebné ďalej definovať, avšak pre vysvetlenie algoritmu naimplementovaného v tejto diplomovej práci budú vysvetlené v rámci popisu algoritmu

v časti 3.2.3. V závislosti od počtu takýchto komponentov v grafe aplikujeme reverziu na určený podinterval sekvencie. Po vykonaní bezpečnej reverzie počet prekážok nevzrastie.

Ak je graf permutácie π označený ako pevnosť[5], nie je možné ho utriediť popísanými operáciami. Obsahuje totiž prekážky, ktoré po odstránení vytvoria nové prekážky. Pevnosť akoby bránila triedeniu za použitia vymenovaných operácií. Preto je potrebné najprv eliminovať pevnosť a až potom pristúpiť k samotnému odstraňovaniu prekážok a preusporiadavaniu permutácie. Táto operácia taktiež neznižuje počet cyklov, ktoré je potrebné eliminovať. Ak je, teda, graf označený ako pevnosť, vo výslednej postupnosti krokov bude figurovať jedna reverzia, ktorú nie je možné označiť ako bezpečnú, počet izolovaných cyklov dĺžky 2 nevzrastie.

Celkový počet potrebných operácií – genómová vzdialenosť – je vyjadrená: Ak π je permutácia s n elementmi, $c(\pi)$ je počet cyklov v permutácii π a h je počet prekážok v π, potom genómová vzdialenosť d je definovaná

$$d(\pi) = \begin{cases} n + 1 - c(\pi) + h(\pi) + 1 & \text{ak } \pi \text{ je pevnosť} \\ n + 1 - c(\pi) + h(\pi) & \text{inak,} \end{cases}$$

teda reverzálna vzdialenosť je rovná počtu cyklov dĺžky 2 v utriedenej permutácii mínus počet cyklov v aktuálnej v permutácii – týmto dostávame počet cyklov, ktoré je nutné dekomponovať; plus počet prekážok, ktoré je potrebné odstrániť a nakoniec plus 1 ak je permutácia označená ako pevnosť, pretože jedna reverzia nepriblíži aktuálnu permutáciu k cieľovému stavu, ale odstráni prekážky.

Metóda dvojitého preseknutia a spojenia

Metóda dvojitého preseknutia a spojenia[6] sa výrazne odlišuje od predchádzajúcej metódy. Je to diametrálne odlišný model riešenia, ktorý používa

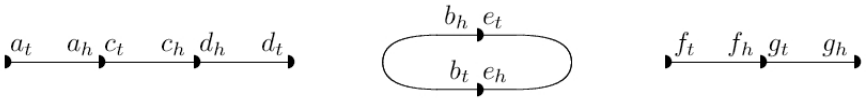
[5] anglicky fortress

[6] Double Cut and Join

odlišné operácie. Popis sa nachádza napríklad v Bergeron et al. (2006); Mixtacki (2008); Kováč et al. (2010). Výhodou metódy je, že dokáže pracovať s lineárnymi a aj cirkulárnymi genómami.

Genóm je chápaný ako sekvencia markérov prenesená do grafu. Každý markér genómu je reprezentovaný hranou medzi dvojicou vrcholov a_h a a_t[7] a postupnosť týchto vrcholov vyjadruje orientáciu daného markéru. Vrcholy grafu sú tvorené hranou medzi dvojicou po sebe nasledujúcich markérov. Genóm je ohraničený telomérami. Jednotlivé vrcholy môžu byť interné (sú stupňa 2) a externé (sú stupňa 1), spoločne vytvárajú cestu. Dvojice elementov hlavičky a chvostu dvoch susedných génov, v závislosti od ich orientácie, vytvárajú tzv. susednosti[8]. Môžu vytvoriť nasledovné kombinácie častí dvoch markérov $\{a_h, b_t\}, \{a_h, b_h\}, \{a_t, b_t\}, \{a_t, b_h\}$. Teloméry nesusedia so žiadnym génom, reprezentujú koncové (terminálne) časti cesty.

Napríklad, ak genóm A obsahuje nasledovné gény s danou orientáciou: $(a, c, -d), (b, e, b), (f, g)$, potom sekvencia elementov pre hľadanie najkratšej genómovej vzdialenosti pomocou dvojitého preseknutia a spojenia bude: $A = \{\{a_t\}, \{a_h, c_t\}, \{c_h, d_h\}, \{d_t\}, \{b_h, e_t\}, \{e_h, b_t\}, \{f_t\}, \{f_h, g_t\}, \{g_h\}\}$ a jeho grafová reprezentácia (obr. 1.6) môže byť:



Obr. 1.6: **Grafová reprezentácia genómu A**
(Bergeron et al., 2006)

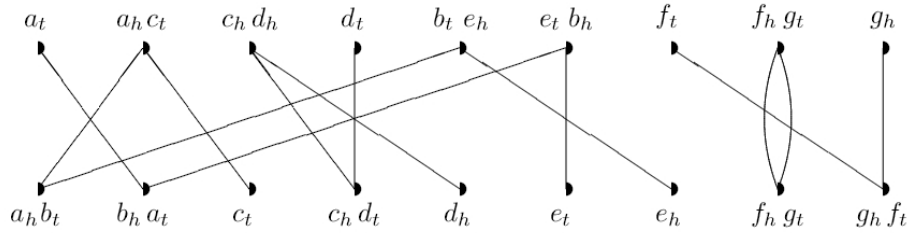
Ak máme zápis genómu ako postupnosti hlavičiek a chvostov, vytvoríme graf susedností[9] genómu, ktorý vyjadruje momentálny stav usporiadanosti genómu A voči genómu B (obr. 1.7).

[7]skratky z anglických head, tail – hlavička a chvost markéru, v kladnej orientácii nasledujú v poradí tail, head

[8]angl. adjacency

[9]angl. adjacency graph

$$A = \{\{a_t\}, \{a_h, c_t\}, \{c_h, d_h\}, \{d_t\}, \{b_h, e_t\}, \{e_h, b_t\}, \{f_t\}, \{f_h, g_t\}, \{g_h\}\}$$
$$B = \{\{a_h, b_t\}, \{b_h, a_t\}, \{c_t\}, \{c_h, d_t\}, \{d_h\}, \{e_t\}, \{e_h\}, \{f_h, g_t\}, \{g_h, f_t\}\}$$



	1	2	3	4	5	6	7	8	9
first	a_t	a_h	c_h	d_t	b_h	e_h	f_t	f_h	g_h
second	–	c_t	d_h	–	e_t	b_t	–	g_t	–

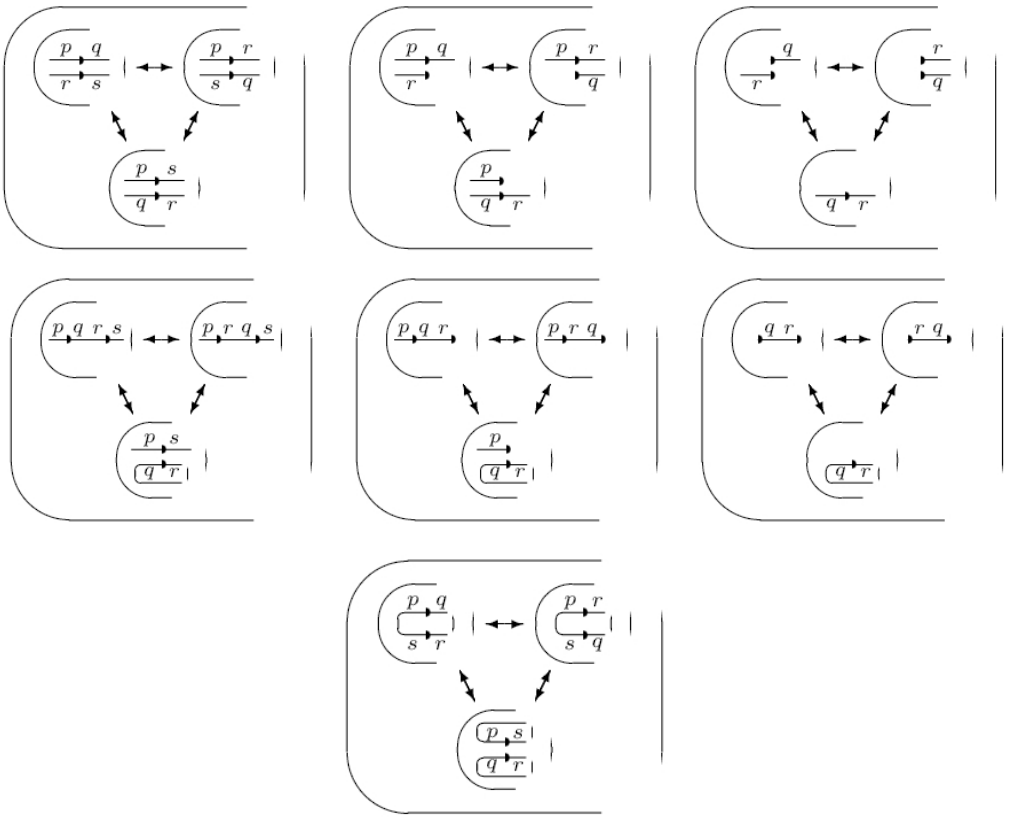
	a	b	c	d	e	f	g
head	2	5	3	3	6	8	9
tail	1	6	2	4	5	7	8

Obr. 1.7: **Graf susedností genómov A a B**
(Bergeron et al., 2006)

Vrcholmi grafu susedností sú elementy sekvencií aktuálneho a cieľového genómu, totožné vrcholy sú pospájané hranami. Takáto grafová reprezentácia prenesená do tabuľky je využívaná počas výpočtu. Tabuľka zachytáva elementy a ich indexy v rámci sekvencie – v prvom prípade je kľúčom index prvku v genóme, v druhom prípade sú dopĺňané naopak indexy elementov v sekvencii.

Ďalšou odlišnosťou tohto modelu riešenia problému najkratšej genómovej vzdialenosti sú povolené operácie. Principiálne ide o prerušenie hrán incidentných s podintervalom, na ktorom operáciu vykonávame a opätovné pripojenie k niektorej z voľných telomér genómu, alebo môže odseknutý podinterval vytvoriť samostatnú cestu. Akoby sme kus sekvencie vyrezali (teda presekli na dvoch miestach) a vložili na iné miesto (opäť spojili so sekvenciou). Tým dochádza k translokácii cesty, fúzii ciest, štiepeniu cesty v grafe, inverzii, cirkularizácii (vytváraniu nových cyklov), či linearizácii (dekompozícii cyklov). Operácie sú bližšie popísané v časti 2.3. Na obrázku 1.8 sú zobrazené možné scenáre úpravy genómu vykonaním operácie dvojitého preseknutia a spojenia.

Sledom operácií dvojitého preseknutia a spojenia postupne upravujeme



Obr. 1.8: **Operácie metódy dvojitého preseknutia a spojenia** *(Bergeron et al., 2006)*

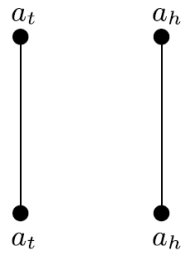
genóm tak, že sa z grafu eliminujú cesty a cykly obsahujúce viac ako 2 vrcholy. Pre triedenie genómu A podľa genómu B, pričom tieto dva genómy pozostávajú z rovnakej abecedy N markérov platí, že $A = B \Leftrightarrow N = C + I/2$, kde C je počet cyklov v grafe a I je počet ciest s nepárnym počtom vrcholov v grafe. Vysvetlenie a dôkaz (Bergeron et al., 2006):

$A = B \Rightarrow N = C + I/2$

V grafe utriedenej sekvencie označíme ako a počet susedností t počet telomér. Platí, že $N = a + t/2$. Je zjavné, že počet susedností je rovný počtu cyklov ($a = C$) a teloméry sú jediné elementy s nepárnym počtom vrcholov, teda $t = I$. Z toho vyplýva, že $N = C + I/2$.

$N = C + I/2 \Rightarrow A = B$

Môžeme povedať, že $N = a + t/2$. Každý cyklus v grafe obsahuje najmenej jednu susednosť. Z toho vyplýva, že $C \leq a$ a každá nepárna cesta obsahuje práve jednu teloméru, $I \leq t$. Ak $C + I/2 = N = a + t/2$, potom $C = a$ a $I = t$. Z toho vyplýva, že všetky cykly majú dĺžku 2 a všetky nepárne cesty majú dĺžku 1, k čomu dochádza jedine v prípade, že $A = B$ (obr.1.9). Vtedy algoritmus končí, sekvencie sú usporiadané.



Obr. 1.9: **Graf susedností usporiadanej sekvencie**
(Bergeron – Stoye, 2013)

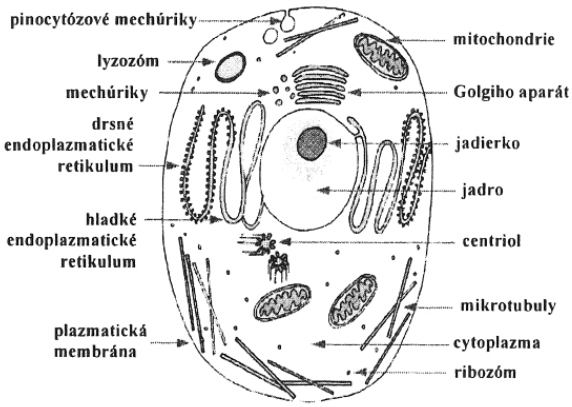
V tejto časti sme popísali dva ťažiskové modely riešenia pre problém hľadania minimálnej genómovej vzdialenosti. Existuje ešte množstvo prístupov k riešeniu problému, ktoré sú ale zväčša modifikáciou jednej z popísaných schém. Model riešenia, ktorý sme v implementácii tejto diplomovej práce použili my taktiež vychádza z modelu dekompozície maximálneho cyklu. Okrem iných odlišností, ale, neuvažuje s entitami pevností.

2 Biologické pozadie

V tejto kapitole vysvetlíme základné biologické pojmy používané pri opise problému a jeho riešenia popísaných v kapitole 3.

2.1 Genetická informácia bunky

Vo vnútri bunky sa nachádza viacero štruktúr – bunkových organel. Každá z nich má svoju špecifickú a pre bunku nepostrádateľnú funkciu. (Obr.2.1).



Obr. 2.1: **Bunka a jej organely**

(Böhmer et al., 2008)

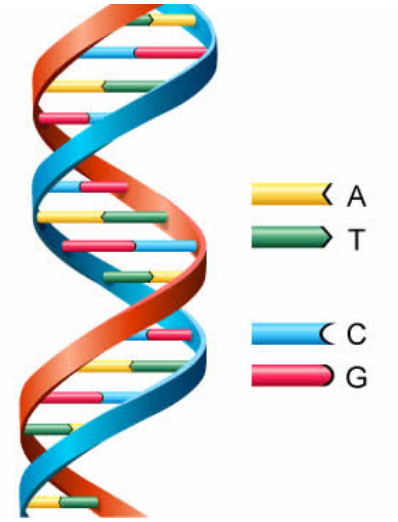
Ak sa však budeme sústreďovať len na nositeľov genetickej informácie bunky, zameriame sa len na niektoré štruktúry. Je známe, že genetickú informáciu môžeme nájsť najmä v jadre bunky. Mimo jadra je genetickú informáciu možné nájsť napr. v centrioloch, plazmidoch a mitorchondriách. (Hraška, 2005) Podľa umiestnenia môžeme hovoriť o jadrovej a mimojadrovej DNA.

2.1.1 Jadrová DNA

Pomocou DNA[1] nachádzajúcej sa v jadre je bunka schopná regulovať procesy súvisiace s jej vývojom, rastom, alebo špecifickou funkciou ako napríklad sekrécia, či prijímanie reflexných podnetov. Ak sa táto bunka rozmnoží delením, jej dcérska bunka bude niesť rovnakú genetickú informáciu o regulácii procesov.

DNA si môžno predstaviť ako reťaz nukleotidových báz – **A**denínu, **G**uanínu, **C**ytozínu a **T**ymínu. Tieto bázy sú štruktúrnymi jednotkami genetickej informácie, tvoria vnútornú pamäť bunky.

Jednotlivé nukleotidy sú viazané so svojím predchodcom, nasledovníkom a jedným nukleotidom v druhom vlákne dvojzávitnice. (Obr.2.2)



Obr. 2.2: **Dvojzávitnica DNA**
(http://education.technyou.edu.au)

Nukleotidy, ktoré vytvárajú väzbu medzi dvoma vláknami DNA, nie sú spájané ľubovoľne, ale na základe komplementarity: Ak sa v jednom vlákne

[1]DNA – angl. Deoxyribonucleic acid, kyselina deoxyribonukleová

nachádza báza A, v komplementárnom vlákne je na danom mieste T, prípadne naopak. Tento princíp platí, samozrejme, aj pre C a G. Dvojice C – G a A - T nazývame komplementárne páry. Počet molekúl adenínu (A) sa zhoduje s počtom molekúl tymínu (T), podobne to platí aj pre pomer cytozínu a guanínu. U človeka predstavujú páry A-T 62% z celkového počtu nukleotidových párov v jadrovej DNA. Doplnkovo C-G páry tvoria 38%. (Hraška, 2005)

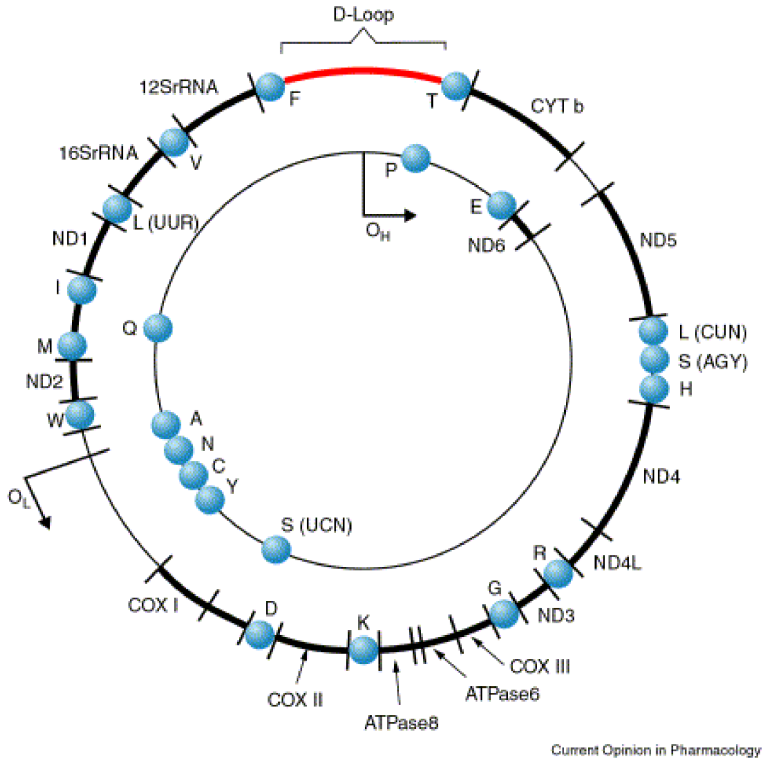
2.1.2 Gén

Gén je základnou jednotkou dedičnosti, predávania znakov a vlastností z rodičov na potomkov. Je to úsek reťazca DNA, ktorý obsahuje informáciu zakódovanú v poradí nukleotidov A,G,C,T o danom znaku, či tvorbe určitej bielkoviny. Súbor génov tvorí chromozóm, súbor chromozómov tvorí genóm. (Böhmer et al., 2008)

2.1.3 Mitochondriálna DNA

Mitochondrie sú malé organely v bunke, v ktorých priebiehajú dôležité procesy spojené so získavaním energie. Poškodenia mitochondriálnej DNA (ďalej aj ako mtDNA) zapríčiňujú vážne poruchy jedinca – v súvislosti s potrebou energie sú to najmä poruchy nervového systému.

Ako sme už spomenuli, mitochondrie disponujú vlastným genómom, čo im umožňuje autoreprodukciu nezávisle od delenia materskej bunky. (Böhmer et al., 2008) Ich genóm sa od jadrového genómu, ale, značne odlišuje. Na obrázku 2.3 je schématicky zobrazený mitochondriálny genóm človeka. Má kruhovú štruktúru, oproti jadrovej DNA obsahuje podstatne menej nukleotidov – jadrový genóm je zložený z cca 3,3 miliardy bázových párov, zatiaľ čo mtDNA je tvorená 16 568 nukleotidmi. (Sperschneider et al., 2010; Taanman, 1999)



Obr. 2.3: **Mitochondriálny genóm človeka**
(Schaefer et al., 2001)

Má štruktúru dvojvlákna – vonkajšie vlákno tzv. H-vlákno[2] kóduje väčšinu génov mitochondriálneho genómu. Krátky úsek – D-loop[3] predstavuje miesto, kde je H-vlákno mtDNA zdvojené. Nachádzajú sa tu kontrolné body pre procesy transkripcie a translácie, ktorými sa mtDNA replikuje. (Wanrooij – Falkenberg, 2010)

Mitochondriálny genóm človeka je tvorený len 37-mimi génmi (DiMauro – Schon, 2001). Tie neobsahujú intróny (proteíny nekódujúce úseky) a často sa prekrývajú. Kontrolné body v replikačnom procese zabezpečujú, že v prípade chybného, nefunkčného genómu sa proces jeho replikácie zastaví. Zhustenie génov v mtDNA teda spôsobuje, že gény sú do značnej miery konzervované,

[2]Heavy strand, angl. ťažké, plné vlákno
[3]D = displacement, angl. posunutie, loop – angl. slučka

pretože v prípade zásadnej mutácie práve v mieste génu dochádza k zmene jeho prejavu, funkcie. Ak by bola táto zmena zaznamenaná, v replikácii zmutovaného genómu by sa nepokračovalo.

Mitochondriálny genóm sa dedí výlučne po materskej línii, materálne. (DiMauro – Schon, 2001) Z dôvodu veľkosti, zakonzervovanosti počas dlhého časového obdobia, či vlastných kontrolných mechanizmov počas replikácie môžeme považovať mtDNA za objekt vhodný na skúmanie evolučných zmien v rámci fylogenézy.

2.2 Génové markéry

Genetické markéry sú znaky, u ktorých z ich fenotypového prejavu, teda prítomnosti vonkajšieho znaku, sme schopní odvodiť ich genotyp, resp. prítomnosť určitého génu. Z nášho pohľadu je dôležitá najmä podskupina genetických markérov – **génové markéry**. Sú to znaky, fragmenty genómu, ktoré je možné analýzou identifikovať na chromozóme určitého druhu. V procese evolučných zmien zotrvávajú markéry v rámci mitochondriálneho genómu ako konzervovaná skupina génov, môže sa však zmeniť ich vzájomné usporiadanie. (Gömöry, 2010)

V chápaní nášho problému nám markéry budú označovať zjednodušene rozpoznateľné skupiny génov, fragmenty DNA. A tak sa zo zložitej štruktúry genómu stáva podstatne jednoduchšia sekvencia markérov. V závislosti od orientácie fragmentu od začiatočnej teloméry 5' ku koncovej 3', prípadne naopak markér nadobúda znamienko $+$ resp. $-$. (Meidanis et al., 2000)

Výnimku v definovaní markérov ako skupiny génov tvoria **teloméry** – konce genómu. Majme lineárny genóm reprezentovaný k markérmi. Na oboch koncoch genómu sú pridané ďalšie dva elementy reprezentujúce začiatočnú

teloméru (z) a koncovú teloméru (k), teda symbolický začiatok a koniec sekvencie genómu:

$$\pi = (\pi_{\mathbf{z}}, \pi_1, \ \ldots \ \pi_i, \ \ldots \ , \pi_{\mathbf{k}})$$

V prípade cirkulárneho genómu v postupnosti markerov za $\pi_{\mathbf{k}}$ nasleduje $\pi_{\mathbf{z}}$. Z toho vyplýva, že genóm môže byť zapísaný viacerými analogickými reprezentáciami.

2.3 Evolučné zmeny v genóme

Evolučné zmeny sú udalosti vo vývoji druhu, ktoré sú spôsobené zmenami v štruktúre genómu, teda v poradí nukleotidov vlákna DNA, alebo konzervovaných blokov. Tieto čiastkové zmeny nazývame **mutácie**. Evolúcia je teda sekvencia genómových mutácií, ktorá má za následok zmenu tých znakov, ktoré sú kódované úsekmi ovplyvnenými mutáciami.

Evolúcia, či vývoj druhov na základe evolučných zmien, má niekoľko príznačných charakteristických čŕt. Je postupná, graduálna, deje sa na základe reťazca viacerých menších zmien v čase. Má charakter vetvenia. Nové druhy organizmov vznikajú postupnou diferenciáciou z pôvodného predka, pričom tieto sa neskôr môžu stať predkom ďalších druhov.

Jednotlivé zmeny v genóme druhu môžu zasahovať rôzny počet nukleotidov. Podľa rozsahu genómu postihnutého mutáciou rozlišujeme:

- *génové mutácie* – rozsah jedného nukleotidového páru
- *chromozómové mutácie* – rozsah maximálne jedného chromozómu
- *genómové mutácie* – postihujú celý genóm – zmena väčšej skupiny génov, prípadne zmena v počte chromozómov. Rozoznávame intrachromozomálne (úsek DNA v rámci jedného chromozómu) a interchromozomálne (úsek nad viacerými chromozómami) genómové mutácie.

Pre problém modelovania genómovej evolúcie sú využívané operácie o veľkosti chromozómových a genómových mutácií. Podľa Pevznera (2000), Boorea (2000) patria medzi evolučné zmeny daného rozsahu napr.:

delécia – operácia, ktorá odstráni časť úseku DNA na chromozóme.

duplikácia – operácia, ktorá zdvojí, pripojí kópiu časti úseku DNA na chromozóme.

štiepenie – operácia, ktorá rozdelí úsek chromozómu na dve časti. Zvyšuje sa počet chromozómov o 1.

translokácia – presun úsekov DNA medzi dvoma chromozómami. Patrí k interchromozomálnym mutáciám.

fúzia – typ translokácie, kedy sa úsek DNA jedného chromozómu presunie kompletne na druhý chromozóm, pričom vzniká jeden chromozóm s oboma úsekmi DNA.

reverzia mení poradie jednotlivých nukleotidov na chromozóme tak, že sekvenciu preusporiada v opačnom poradí oproti pôvodnej sekvencii.

Modelovanie všetkých týchto operácií v rámci jednej metódy výpočtu je, však, z hľadiska ich rôznorodosti nevýhodné a zložité. Preto sa metódy rekonštrukcie evolučnej histórie vo všeobecnosti zameriavajú na definovanie menšieho počtu operácií, ktoré logicky reprezentujú všetky udalosti nad sekvenciou DNA, pričom je zachovaná biologická relevancia simulovaných procesov, teda simulovaná zmena je biologicky reálne možná.

3 Algoritmus rekonštrukcie evolučnej histórie za použitia operácie reverzie

Rekonštrukcia evolučnej histórie je vlastne premietnutie vypočítanej genómovej vzdialenosti[1] medzi skúmanými druhmi do stromu. Môže poskytnúť model vývoja, to v akom poradí, či z akého spoločného predka sa organizmy vyvinuli. Podľa Darwina (1859) sa všetky recentné druhy vyvinuli zo spoločného predka istým počtom evolučných zmien. Predpokladáme, že evolučnú históriu je teda možné vyjadriť fylogenetickým stromom na základe vzájomných vzdialeností genómov zástupcov jednotlivých vetiev, koreňom stromu sa stáva práve spoločný prapredok. Čím väčšia je príbuznosť druhov, tým kratšia bude ich vzájomná genómová vzdialenosť a uzly reprezentujúce týchto zástupcov vo fylogenetickom strome budú mať viac spoločných predkov. Je zrejmé, že pre vytvorenie evolučnej histórie je genómová vzdialenosť dvoch druhov kľúčovou informáciou.

Definujme problém hľadania najkratšej genómovej vzdialenosti nasledovne: nech A a B označujú dva genómy reprezentované markérmi. Úlohou je nájsť najmenší počet za sebou nasledujúcich trnasformácií genómu A tak, aby výsledná sekvencia zodpovedala genómu B. Počet krokov potrebných na transformáciu označujeme ako *genómovú vzdialenosť* genómov A a B. (Bergeron et al., 2006) Menší počet krokov potrebných na transformáciu značí vyšší stupeň evolučnej príbuznosti.

Práve mitochondriálny genóm je pre svoje vlastnosti vhodným objektom takejto rekonštrukcie evolúcie. Vďaka svojej konzervovanosti – odolnosti voči

[1] angl. genomic distance

zmenám – dokáže vierohodnejšie odzrkadliť vývin druhov ako jadrový genóm, jeho relatívne malá veľkosť uľahčuje prácu s dátami.

Je zrejmé, že riešenie takéhoto problému závisí aj od charakteru krokov, ktoré vo výpočte používame. Práve charakteristika elementárnych operácií rozlišuje metódy rekonštrukcie genetickej evolúcie. Kroky výpočtu by mali simulovať možné evolučné zmeny, ktoré sa v čase uskutočnili v genóme.

Problém nájdenia transformácie genómu A na genóm B, kde oba genómy sú zložené z rovnakej abecedy markérov, sa považoval do roku 1995 za NP-ťažký problém. Vtedy Hannenhalli a Pevzner predstavili prvý polynomiálny variant riešenia problému (Hannenhalli – Pevzner, 1995), ktorý sa od pôvodného líšil v tom, že jednotlivé markéry boli orientované podľa vlákna DNA, na ktorom sa nachádzajú, markéry genómu predstavujú označenú (angl. signed) premutáciu. Následne sa pre tento problém definovalo niekoľko ďalších možných prístupov k riešeniu.

V nasledujúcej časti popíšeme zjednodušený algoritmus riešenia problému. Ďalej sa pokúsime definovať postupy a štruktúry, ktoré nám zjednodušia výpočet na zložitejších dátach.

3.1 Triviálny algoritmus

Ako sme už spomenuli, na vstupe do algoritmu máme dve permutácie markérov. Prvá – permutácia A – označuje sekvenciu markérov tak, ako sa nachádzajú v genóme zástupcu taxónu T_A. Druhá permutácia B vyjadruje poradie markérov u taxónu T_B. Pre získanie genómovej vzdialenosti chceme trasformovať genóm A na genóm B v čo najkratšom slede krokov:
$A = (1, 3, 5, 4, 6, 2)$ a pre jednoduchosť $B = (1, 2, 3, 4, 5, 6)$. Cieľom je transformovať A na B.

Pri riešení tohoto problému využijeme ako krok výpočtu operáciu definovanú v Bergeron (2005) a Hannenhalli – Pevzner (1999) – operáciu **reverzie**[2].

Definujme postup najprv neformálne: Ak v permutácii $A = (1, 3, 5, 4, 6, 2)$ prevedieme operáciu reverzie ρ nad podsekvenciou $(3, 5, 4, 6, 2)$, tak dostávame permutáciu $A' = (1, 2, 6, 4, 5, 3)$. Vidíme, že prvky danej podsekvencie sme preusporiadali tak, že sme ich na pôvodné miesto vložili v opačnom poradí. Takto by sme pôvodnú permutáciu vedeli preusporiadať na genóm B ďalšími dvoma krokmi, kedy v prvom kroku vykonáme reverziu nad prvkami 4 a 5 a v nasledujúcom nad $(6, 5, 4, 3)$. Vzdialenosť týchto genómov za použitia opísanej reverzie je teda 3. V ďalšom texte budeme interval reverzie definovať indexmi hraničných prvkov vybraného podintervalu sekvencie.

Operácia reverzie nad permutáciou bez znamienok: Nech je daná sekvencia markerov genómu A nasledovne:

$$\pi_A = (\pi_1 \ \ldots \ \pi_i \ \pi_{i+1} \ \ldots \ \pi_j \ \ldots \ \pi_n)$$

Operácia reverzie ρ nad podintervalom ohraničeným prvkami i a j neoznamienkovanej permutácie π_A spôsobí zmenu v genóme takto:

$$\pi_A = (\pi_1 \ \ldots \ \underline{\pi_i \ \pi_{i+1} \ \ldots \ \pi_j} \ \ldots \ \pi_n)$$
$$\pi_A \cdot \rho(i, j) = {\pi_A}'$$
$${\pi_A}' = (\pi_1 \ \ldots \ \underline{\pi_j \ \ldots \ \pi_{i+1} \ \pi_i} \ \ldots \ \pi_n)$$

Nech $\pi = (\ldots \pi_{i-1}, 1, 4, 3, 2, \pi_{i+4} \ \ldots)$, potom po reverzii $\rho(i, i+2)$, teda na podintervale $< 1, 4, 3 >$, bude sekvencia $\pi' = (\ldots \ 3, 4, 1, 2, \ \ldots)$.

Hľadanie najkratšej genómovej vzdialenosti týmto algoritmom môže byť podľa Bergeron (2005) až NP-ťažké, a preto sa budeme venovať polynomiálnemu variantu problému. Ide o problém hľadania najkratšej reverznej vzdialenosti nad permutáciou prvkov so znamienkami, ktoré vyjadrujú na ktorom

[2]angl. reversal

vlákne DNA sa daná sekvencia genómov nachádza. Za použitia operácie reverzie hovoríme namiesto genómovej vzdialenosti o **vzdialenosti reverzálnej**.

3.1.1 Algoritmus hľadania najkratšej reverzálnej vzdialenosti nad oznamienkovanou permutáciou.

Ak by sme chceli pracovať so sekvenciou, v ktorej sa vyskytujú elementy z oboch vlákien mtDNA, čo je pri aplikácii na reálne dáta veľmi pravdepodobné, genóm potom budeme reprezentovať pomocou permutácie, v ktorej je nutné zdôrazniť na ktorom vlákne DNA sa markér nachádza. Táto informácia bude vyjadrená znamienkom pred označením markéru. Ak pred číslom v permutácii znamienko nie je, budeme toto chápať ako kladné. Vieme, že vlákna DNA sú navzájom komplementárne, mení sa len smer poradia nukleotidov, a tak zmenou znamienka, orientácie markéru ako skupiny nukleotidov, neprichádzame o žiadnu informáciu.

Operáciu reverzie modifikujeme tak, že okrem zmeny poradia elementov zmeníme aj ich orientáciu. Ilustrujeme príkladom:
Nech $\pi = (\ldots\ 1, 4, -3, -2,\ \ldots)$, potom po reverzii $\rho(k+1; k+3)$, kde k je počet elementov pred 1, a teda nad intervalom $< 1, 4, -3 >$ bude výslená sekvencia $\pi' = (\ldots\ 3, -4, -1, -2,\ \ldots)$.

3.1.2 Odstránenie znamienok

Aj keď je vyjadrenie orientácie markérov počas výpočtu žiadúce, ako nevhodnú možno označiť reprezentáciu zápornými hodnotami. Znižuje prehľadnosť výpočtu pri modifikácii dát do pomocných štruktúr, a preto zavedieme inú reprezentáciu orientácie. Každý element genómu nahradíme dvomi prvkami nasledovne:

- ak je element kladný, má znamienko +, bude nahradený prvkami **2x-1** a **2x**, kde x je hodnota prvku.
- ak je element záporný $(-)$, bude nahradený dvoma prvkami **2x** a **2x-1**, kde x je hodnota prvku.

Permutáciu $\pi = (1, 4, -3, -2)$ by sme podľa horeuvedených pravidiel upravili na $\pi' = (1, 2, 7, 8, 6, 5, 4, 3)$. Takto získavame síce väčšie množstvo elementov, no zachovávame informáciu o vlákne, pri zjednodušení problému spracovania znamienok. Vidíme, že počet elementov v upravenej permutácii sa oproti pôvodnej zdvojnásobil.

3.1.3 Ohraničenie telomérami

Ďalšia úprava predstavuje ohradenie sekvencie markérov rámcovými prvkami – **telomérami**[3] **0** a **n+1**, kde n je najväčší, maximálny prvok permutácie. Rámcové prvky budú vždy kladné. Reprezentujú terminálne časti – konce lineárneho genómu. Spomínaná permutácia $\pi = (1, 4, -3, -2)$, po odstránení znamienok $\pi' = (1, 2, 7, 8, 6, 5, 4, 3)$, bude s ohradením rámcovými prvkami vyzerať nasledovne: $\pi'' = (0, 1, 2, 7, 8, 6, 5, 4, 3, 9)$.

3.1.4 Orientované páry

Ak teda máme sekvenciu markérov dvoch genómov, ktoré vieme opísaným spôsobom upraviť, ako budu vyzerať kroky výpočtu a akým spôsobom ich budeme voliť? Odpoveďou na túto otázku sú tzv. **orientovené páry**. Ich početnosť v sekvencii v danom momente výpočtu, teda po vykonaní reverzie ρ_i, bude reprezentovať skóre kroku i. (Bergeron, 2005) Pre zjednodušenie vysvetlenia budeme pracovať opäť s pôvodnou, nezdvojenou permutáciou so znamienkami.

[3] viac v kapitole 2.2

Daná je permutácia $\pi = (\pi_1 \ \ldots \ \pi_i \ \ldots \ \pi_n)$. Nech prvý prvok permutácie je **0**, prvky sekvencie π nasledujú v nezmenenom poradí za ňou a posledný prvok bude **n+1**. π bude mať tvar: $\pi = (0 \ \pi_1 \ \ldots \ \pi_n \ n+1)$

Orientovaný pár v permutácii π, ktorá má n elementov so znamienkami, je dvojica prvkov takých, že absolútna hodnota elementov (keďže hodnoty prvkov môžu naberať aj záporné hodnoty) $|\pi_i| - |\pi_j| = \pm 1$, $0 \leq i < n$, $0 < i \leq n$, pričom prvky π_i a π_j majú navzájom opačné znamienka. (Bergeron, 2005)

Orientované páry v permutácii sú elementy, ktoré naznačujú, že reverzia použitá na úsek ohraničený týmto párom, vytvorí žiaducu podsekvenciu bez nových neorientovaných neusporiadaných vrcholov. (Bergeron, 2005) Prítomnosť orientovaných vrcholov nám totiž zabezpečuje aj prítomnosť ďalšieho možného kroku, až pokým sekvencia nie je v požadovanom stave. K významu reverzie nad orientovanými pármi sa po vysvetlení ďalších pojmov ešte vrátime. Uvedieme príklad:
Genóm $\pi = (-5 \quad -6 \quad 1 \quad 2 \quad 3 \quad 4)$. K sekvencii pridáme rámcové prvky: $\pi' = (0 \quad -5 \quad -6 \quad 1 \quad 2 \quad 3 \quad 4 \quad 7)$. Nech $\pi'' = (0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7)$ je permutácia prvkov, na ktorú chceme π' transformovať.
Orientované páry v π sú $[4, -5]$ a $[-6, 7]$. Ďalší postup bude nasledovný:

Operácia	**Aktuálny stav permutácie**									
	$\pi_0 = ($	0	-5	**-6**	**1**	**2**	**3**	**4**	7	) $= \pi'$
$\pi_0 \cdot \rho(1,6)$	$\pi_1 = ($	0	-5	-4	-3	-2	-1	6	7	)
	$\pi_1 = ($	0	**-5**	**-4**	**-3**	**-2**	**-1**	6	7	)
$\pi_1 \cdot \rho(0,5)$	$\pi_2 = ($	0	1	2	3	4	5	6	7	) $= \pi''$

Ako vidno pri prvom kroku, kedy sa v permutácii nachádzali dva orientované páry $[4, -5]$ a $[-6, 7]$, sme využili operáciu nad podsekvenciou ohraničenou 4 a -5. Dôvodom pre tento krok bolo vyššie skóre orientovaného páru $[4; -5]$:

Skóre orientovaného páru definujeme ako počet orientovaných párov vo výslednej sekvencii po aplikovaní operácie reverzie nad subsekvenciou ohraničenou týmto orientovaným párom. Príspevok do skóre orientovaného páru, ktorý obsahuje rámcový prvok je 0.

Ak by sme sa v prvom kroku rozhodli aplikovať reverziu nad podsekvenciou ohraničenou $[-6, 7]$, vo výsledku $\pi_{1'} = (0\ -5\ -6\ -7\ -4\ -3\ -2\ -1)$ sa vyskytuje jeden orientovaný pár $[0, -1]$, no keďže obsahuje nulu ako rámcový prvok je jeho skóre a taktiež výsledné skóre operácie je 0.

Môžeme konštatovať, že **skóre** $(\rho(1, 6))$ > **skóre** $(\rho(2, 7))$, a preto sme nad permutáciou π_0 aplikovali $\rho(1, 6)$. Triviálny princíp algoritmu na výpočet reverzálnej vzdialenosti môže, teda, znieť: *Vyznač orientácie, ohranič telomérami, následne opakovane, pokiaľ sú v permutácii orientované páry, nájdi pár s najvyšším skóre a aplikuj reverziu nad týmto párom, pokým permutácia nie je utriedená.*

Ak popísaný algoritmus aplikuje k operácií reverzie nad permutáciou π a tým ju transformuje na permutáciu π', potom reverzálna vzdialenosť $d(\pi)$ permutácie π je rovná reverzálnej vzdialenosti doposiaľ upravenej permutácie π' a k krokom, ktorými sme sa k π' dopracovali: $d(\pi) = d(\pi') + k$.

3.2 Modifikovaný algoritmus

V predchádzajúcom texte sme popísali naivný, triviálny algoritmus, ktorý je však pri reálnom vstupe nedostačujúci a nevyhovujúci, pretože dosiahnutie riešenia nie je spoľahlivé. Triviálny algoritmus predpokladá, že postupnosťou reverzií nad vhodným orientovaným párom v sekvencii sa dopracujeme ku konečnému správnemu výsledku. Existuje však možnosť, že algoritmus sa zastaví v momente, kedy sa v permutácii už nenachádzajú žiadne orientované páry nad ktorými by sme mohli previesť ďalší krok a zároveň permutácia ešte

nie je utriedená. Takýto stav môže dokonca nastať už na samom začiatku, kedy sú všetky elementy rovnakej orientácie, no neutriedené. Napríklad pri permutácii $(0, 2, 4, 3, 1, 5)$ podľa triviálneho algoritmu nevieme zvoliť ďalší krok. V nasledujúcom texte sa budeme venovať jeho modifikácii, aby sme boli schopní dosiahnuť výsledok z ľubovoľných vstupných údajov.

Je potrebné permutáciu upraviť tak, aby sme získali orientovaný pár a pomocou neho sa dopracovali k riešeniu. Pre riešenie opísaného stavu bol definovaný pojem prekážka[4]. Predtým, však, pristúpme k vysvetleniu drobného zjednodušenia.

3.2.1 Prečíslovanie elementov

V ďalšom texte pre zjednodušenie a lepšiu názornosť zmeníme cieľovú permutáciu na sekvenciu čísel rastúcich v smere od začiatočného hraničného elementu smerom ku koncovému. Toto zjednodušenie je možné vykonať, pretože jednotlivé prvky permutácie A, ktorú chceme preusporiadať podľa poradia v permutácii B je možné prečíslovať tak, aby permutácia B bola usporiadaná rastúco v opísanom smere. Táto zmena nám umožňuje pracovať ďalej už iba s jednou permutáciou a druhú vnímať ako rastúci rad kladných čísel.

Príklad: V nasledujúcom príklade pre lepšiu názornosť vyjadríme pôvodné hodnoty prvkov písmenami abecedy, modifikované poradie zaznačíme arabskými číslicami.
Pôvodná permutácia A: b d c a e
Pôvodná permutácia B: a e d c b

Prečíslovaná permutácia B: 1(a) 2(e) 3(d) 4(c) 5(b)
Prečíslovaná permutácia A: 5 3 4 1 2

[4]anglicky hurdle

3.2.2 Zlievanie podintervalov

Niekedy nastane stav, kedy je vhodné spojité podintervaly v sekvencii upraviť tak, že elementy v rámci nich označíme jediným prvkom a ostatné markéry v sekvencii adekvátne prečíslujeme. Uľahčí to prácu so sekvenciou, pretože po tejto operácii bude pravdepodobne výrazne redukovaná jej veľkosť. Príklad: Sekvencia {0 2 3 4 5 6 1 7} bude mať po úprave tvar {0 2 1 3}. Podinterval {2 3 4 5 6} bol označený jediným elementom 2 a markér 7 ďalej vystupuje v sekvencii ako element 3.

3.2.3 Prekážky

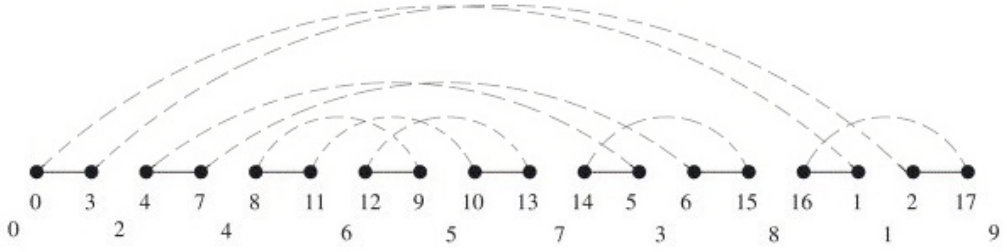
Podľa Bergeron (2005) sú prekážky definované ako ohraničené intervaly, ktoré neobsahujú žiaden ďalší, kratší, ohraničený interval. Podľa Hannenhalli – Pevzner (1999) sú prekážky spojité neorientované komponenty, ktoré sú minimálne vzľadom na ich interval. Teda sú to najmenšie spojité intervaly prvkov s rovnakou orientáciou. Čo to ale znamená?

Ohraničený interval je interval, ktorý na číselnej osi, prípadne na osi utriedených elementov tvorí úsečku. Podľa Bergeron (2005) je to minimálny interval, ktorý v cirkulárnom poradí obsahuje všetky elementy medzi maximálnym a minimálnym prvkom intervalu. O cirkularite hovoríme, ak za posledným prvkom intervalu nasleduje opäť počiatočný prvok. Pre ďalšie vysvetlenie si zobrazíme problém graficky.

3.2.4 Graf prerušení

Na obrázku 3.1 je zobrazená permutácia s odstránenými znamienkami markérov. Každý z nových elementov predstavuje vrchol grafu. Vidíme, že všetky vrcholy sú stupňa 2. Medzi vrcholmi sú hrany: každé dva za sebou nasledujúce vrcholy sú spojené **čiernou**, rovnou hranou, dva vrcholy nasledujúce za

sebou v utriedenej permutácii sú spojené **sivou**, oblúkovou hranou. (Hannenhalli – Pevzner, 1995). Popísaný graf nazývame **graf prerušení**[5].



Obr. 3.1: **Graf prerušení permutácie** {0 2 4 6 5 7 3 8 1 9}
(Bergeron, 2005, str.140)

Samotný pojem prerušenia definujeme podľa Pevznera (2000):
daná je permutácia $\pi = (\pi_1 \pi_2 \dots \pi_n)$. Hovoríme, že medzi elementmi π_i a π_{i+1} je

a) **susednosť**[6] ak $|\pi_i - \pi_{i+1}| = 1$, $0 \leq i < n$

b) **prerušenie**[7] inak

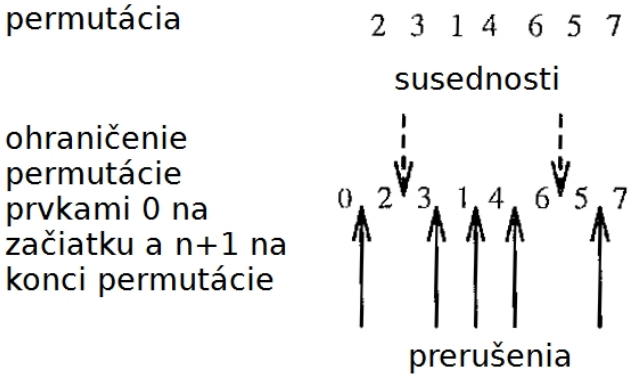
Ak teda v permutácii dva vedľa seba stojace elementy za sebou nasledujú aj v usporiadanej sekvencii a rozdiel ich absolútnych hodnôt je 1, tvoria susednosť. Ak nie – a teda absolútna hodnota rozdielu medzi ich hodnotami je väčšia ako 1, tvoria prerušenie. Voľne možno prerušením nazvať miesto, kde podsekvencia dvoch elementov nie je usporiadaná v smere od začiatku smerom ku koncovému prvku, rad stúpajúcich hodnôt prvkov je prerušený.

Príklad: Nech je daná permutácia $\pi = (2, 3, 1, 4, 6, 5, 7)$. K prvkom π pridáme podľa predchádzajúceho návodu hraničné prvky a skúmame prítomnosť

[5]angl. breakpoint graph
[6]angl. adjacency
[7]angl. breakpoint

prerušenia medzi prvými dvomi elementmi sekvencie: 0 a 2. Absolútna hodnota ich rozdielu je väčšia ako 1, a tak môžeme tvrdiť, že medzi elementmi sa vyskytuje prerušenie, poradie prvkov bude v rámci sekvencie reverzií zmenené. Ak by sme, ale, vzali nasledujúce dva prvky, a teda 2 a 3, vidíme, že 3 je nasledovníkom 2 a absolútna hodnota ich rozdielu je 1. Elementy budú na pozíciách vedľa seba aj v usporiadanej sekvencii a nie je nutné ich ďalej preusporiadavať.

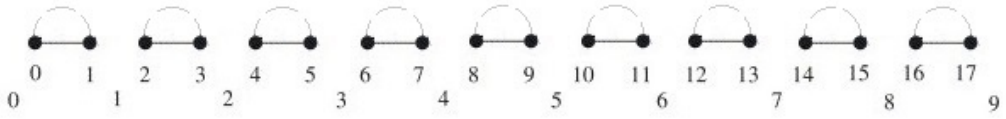
Pre ilustráciu uvádzame obrázok 3.2, v ktorom sú na miestach v sekvencii zvýraznené body susedností a prerušení.



Obr. 3.2: **Sekvencia s miestami postupností a prerušení**
(Pevzner, 2000, str.181)

V čom ale spočíva význam takéhoto porovnávania? Čierne hrany v grafe prerušení znázorňujú aktuálny stav permutácie. Sivé hrany vyjadrujú cieľový stav grafu, ktorý chceme dosiahnuť. Čím viac prerušení v permutácii nachádzame, tým viac komponentov grafu bude mať dĺžku väčšiu ako 2, a teda tým viac reverzií bude potrebných na to, aby sa aktuálny a cieľový stav stotožnili. V grafe prerušení usporiadanej permutácie sú iba izolované cykly s dvoma vrcholmi, medzi ktorými je jedna sivá a jedna čierna hrana, cesty týchto hrán sú rovnaké. Voľne možno povedať, že počet prerušení je úmerný počtu krokov, potrebných na usporiadanie. V (Pevzner, 2000) Pavel Pevzner hovorí, že myšlienku reverzálnej vzdialenosti a prerušení dal do súvisu už

v roku 1938 Strutevant a Dobzhansky. Prerušenia sú podľa Pevznera to, čo robí permutácie ťažko utriediteľné.



Obr. 3.3: **Graf prerušení utriedenej permutácie** {0 1 2 3 4 5 6 7 8 9}

Sivé hrany v grafe prerušení môžu byť **orientované** a **neorientované**. Hrana (π_k, π_l) je orientovaná ak súčet indexov $k+l$ je párny a ak jej koncové body tvorili v pôvodnej permutácii orientovaný pár. (Kaplan et al., 1999; Bergeron, 2005) Ak je totiž súčet indexov vrcholov hrany párny, znamená to, že oba spájajú buď začiatky, alebo konce čiernych hrán. V prípade cyklu s najmenším počtom vrcholov (4) spájajú takéto hrany vrcholy predstavujúce dva elementy s opačnou orientáciou, počet hrán v cykle nezodpovedá žiadanému počtu v cieľovom grafe.

Komponent v grafe je spojitý útvar, ktorý obsahuje maximálnu množinu vrcholov, pre ktoré platí, že každý vrchol je dosiahnuteľný z ľubovoľného vrchola. Postupným prechádzaním sivých a čiernych hrán cez všetky vrcholy podgrafu komponentu dosiahneme opäť počiatočný vrchol. Komponent, ktorý obsahuje aspoň jednu orientovanú hranu **je orientovaný**, inak je neorientovaný.(Kaplan et al., 1999; Bergeron, 2005)

V grafe na obrázku 3.1 môžeme identifikovať tri spojité komponenty, ktoré sú definované postupnosťou sivých a čiernych hrán medzi vrcholmi: [4,15], [8,13], [16,3]. Intervaly prislúchajúce k týmto komponentom sú:
$[4, 15] = 4\ 7\ 8\ 11\ 12\ 9\ 10\ 13\ 14\ 5\ 6\ 15$
$[8, 13] = 8\ 11\ 12\ 9\ 10\ 13$
$[16, 3] = 16\ 1\ 2\ 17\ 0\ 3.$

Podľa definície od Bergeron (2005) môžeme zo zoznamu prekážok v tejto permutácii vyradiť interval $[4, 15]$ pretože jeho interval obsahuje kratší ohraničený podinterval $[8, 13]$. Mohlo by sa zdať, že komponent $[16, 3]$ obsahuje

oba menšie podintervaly. Tu musíme ale prihliadať na fakt, že ak sekvenciu vnímame ako cirkulárnu, tak prvý pár elementov 0 a 3 môže byť bez straty informácie umiestnený aj za markér 17 a vtedy je zjavné, že komponent neobsahuje žiadne kratšie intervaly, čo potvrdzuje aj vymenovanie elementov týchto intervalov uvedené vyššie.

Ak teda permutáciu nemožno utriediť sériou reverzií nad orientovanými pármi, je potrebné identifikovať a odstrániť prekážky. Eliminovať prekážky je možné dvoma spôsobmi v závislosti od celkového počtu týchto komponentov v sekvencii:

- Ak je počet prekážok v permutácii rovný 1, použijeme operáciu **vystrihovania prekážok**[8] na intervale prekážky.
- Ak je počet prekážok aspoň 3, použijeme **spájanie prekážok**[9] na intervaloch ľubovoľných dvoch prekážok, ktorých intervaly nie sú susedné.
- Ak sú prítomné 2 prekážky, spojíme tieto dva elementy.

V nasledujúcom texte definujeme operácie eliminácie prekážok – vystrihovanie a spájanie.

3.2.5 Vystrihovanie prekážok

Túto operáciu vykonávame v jedinom prípade a to ak sa v našej permutácii vyskytuje len jedna prekážka. Na podintervale medzi i a $i+1$ označeného komponentu vykonávame operáciu reverzie nasledovne:

$$i \; \boxed{\pi_{j+1}\, \pi_{j+2}\, \ldots} \; i+1 \; \ldots \; \pi_{j+k-1}\, i+k$$

[8] angl. hurdle Cutting
[9] angl. hurdle merging

Pre názornosť uvedieme príklad: permutáciu {0 2 4 3 1 5} tvorí len jeden komponent, ktorý môžeme označiť ako prekážku. Reverziu vykonáme na podintervale medzi bodmi 0 a 1 nasledovne:

$$\{0\ \boxed{2\ 4\ 3}\ 1\ 5\} \longrightarrow \{0\ \boxed{\text{-3 -4 -2}}\ 1\ 5\}$$

Vidíme, že reverziou na vhodnom podintervale sme získali 2 orientované vrcholy $(1, -2)$ a $(-4, 5)$ a modifikovaná permutácia je následne utriediteľná štyrmi reverziami nad orientovanými pármi.

3.2.6 Spájanie prekážok

Spájanie prekážok je v princípe spojenie dvoch intervalov označených ako prekážka. Reverzia bude v tomto prípade vykonaná na intervale medzi ich hraničnými prvkami, a to koncovým prvkom jednej prekážky a začiatočným prvkom druhej prekážky nasledovným spôsobom:

$$i\ \ldots\ \boxed{(i+k)\ \ldots\ i'}\ \ldots\ (i'+k')$$

Napríklad spojením dvoch prekážok {12 1 2 13 0 3} a {4 9 10 7 8 5 6 11} v permutácii (0 3 4 9 10 7 8 5 6 11 12 1 2 13) dostávame permutáciu (0 3 4 9 10 7 8 5 6 12 11 1 2 13). Reverziu sme previedli na {11 12}. V nezdvojenej permutácii sa z (0 2 5 4 3 6 1 7) stáva (0 2 5 4 3 $-$ 6 1 7).

3.2.7 Bezpečná reverzia

Bezpečná reverzia je taká, ktorá nevytvorí nové neorientované komponenty, pričom povolené sú izolované vrcholy, teda vrcholy usporiadaných prvkov. (Bergeron, 2005) Podľa Hannenhalli – Pevzner (1999); Bergeron (2005) optimálnu postupnosť transformácie jednej permutácie na druhú tvoria práve kroky bezpečných reverzií.

Pri popise operácií spájania a vystrihovania prekážok sa ozrejmuje význam zobrazovania sekvencií a ich komponentov do grafu. Aj napriek tomu, že samotný graf nie je nutnou súčasťou riešenia, bez grafu sa identifikujú komponenty genómu podstatne ťažšie. Zostaneme teda ešte pri grafických zobrazeniach sekvencie.

3.2.8 Graf prekrytia

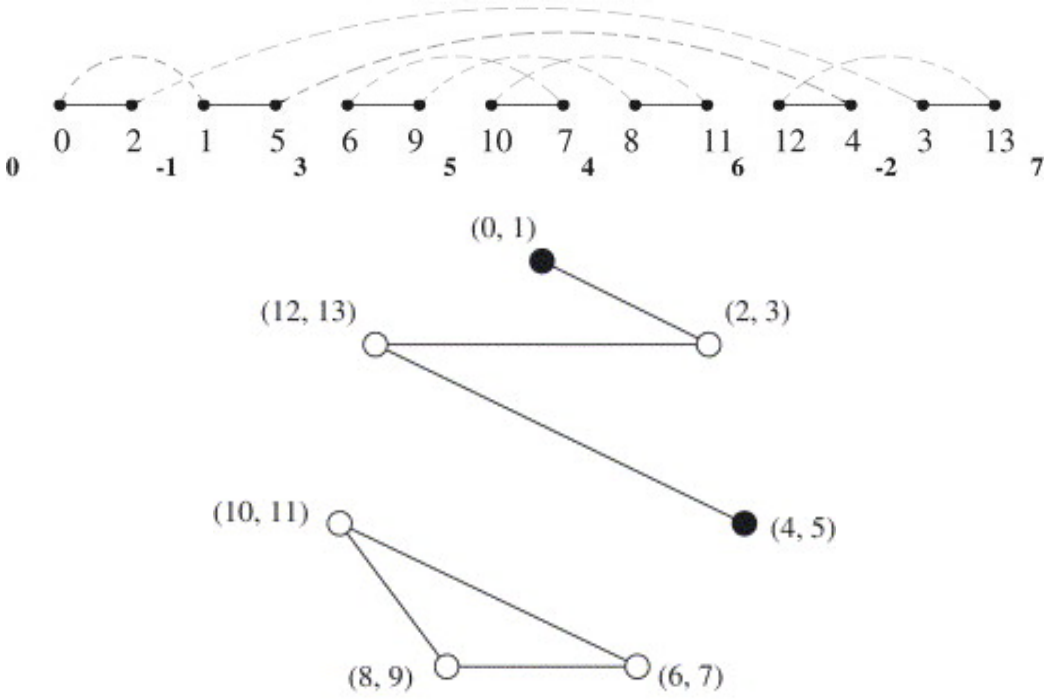
Graf prekrytia[10] vychádza zo štruktúry grafu prerušení. Spracováva informáciu o sivých hranách, resp. o intervaloch pod týmito hranami. Je ďalšou pomocnou štruktúrou v probléme usporiadavania pomocou reverzií.

Graf prekrytia je graf, ktorého **vrcholy** predstavujú **sivé hrany** v grafe prerušení. Ak sa intervaly sivých hrán vo vrcholoch prekrývajú – tzn. ak hraničné body jednej sekvencie ohraničujú interval, v ktorom sa nachádza hraničný bod druhého intervalu, no ich prienik je prázdny, vrcholy reprezentujúce tieto sivé hrany budú spojené hranou. Týmto spôsobom sa orientujeme len na detekovanie častí v postupnosti, ktoré nie sú vhodne usporiadané a v grafe prerušení vystupujú už len príslušné podintervaly definované hranami. Táto metóda sa opiera o fakt, že ak bude sekvencia usporiadaná, žiadne dve hrany v grafe prekrytia sa nebudú prekrývať a žiadne dve hrany nebudú orientované, pretože vždy budú tvoriť izolovaný cyklus medzi dvomi usporiadanými vrcholmi, teda budú reprezentovať podinteval dĺžky 2.

Ak je daná permutácia $\pi = (-1, 3, 5, 4, 6, -2, 7)$, na obrázku 3.4 vidíme jeho transformáciu z grafu prerušení na graf prekrytia.

Môžeme si všimnúť, že ako aj v grafe prerušení, tak aj grafe prekrytia je možné rozpoznať dva samostatné komponenty definované hranami $(0,1) - (2,3) - (12,13) - (4,5)$ a $(10,11) - (6,7) - (8,9)$.

[10] angl. overlap graph



Obr. 3.4: **Graf prerušení a graf prekrytia permutácie** π
(Bergeron, 2005)

V predchádzajúcom texte bolo spomenuté, že v usporiadanej sekvencii sú v rámci grafu prerušení všetky hrany neorientované a tvoria samostatné cykly. Je zrejmé, že po transformácii na graf prekrytia bude táto usporiadaná sekvencia reprezentovaná izolovanými vrcholmi, pretože žiadne dve hrany sa neprekrývajú. Zároveň chceme, aby každý izolovaný cyklus v grafe prerušení obsahoval práve dva vrcholy. Ak je sivá hrana v grafe prerušení orientovaná, jej koncové vrcholy majú oba index s rovnakou paritou, a teda obsahuje najmenej 3 prvky sekvencie. V usporiadanej sekvencii však budú všetky vrcholy neorientované a izolované, obsahujú práve 2 vrcholy. Práve preto sa na orientovanosť hrán budeme aj naďalej sústrediť a prenesieme ju aj do reprezentácie v grafe prekrytia.

Na obrázku 3.4 si môžeme všimnúť, že vrcholy majú dve rôzne označenia. Vrcholy $(0, 1)$ a $(4, 5)$ sú zvýraznené plnou guličkou. Ak sa v súvislosti s pre-

došlým odstavcom budeme sústrediť na orientovanosť, tak zistíme, že práve tieto dva intervaly sú v BG orientované: pre $(0, 1)$, začiatočný vrchol 0 je na indexe 1 a koncový vrchol 1 má index 2, ich súčet je 2, a teda párny.

Ak je súčasťou komponentu v grafe prekrytia aspoň jeden orientovaný vrchol, ktorý predstavuje orientovanú hranu v grafe prerušení, tak hovoríme, že komponent je orientovaný, inak je neorientovaný. (Bergeron, 2005; Kaplan et al., 1999).

Graf prekrytia ešte výraznejšie zjednodušuje reprezentáciu dát a sústredenie na orientovanosť ako graf prerušení. Negrafickým maticovým zápisom susednosti vrcholov vznikne matica susednosti.

3.2.9 Matica susednosti

Matica susednosti, ako aj v predchádzajúcom texte popísané štruktúry, vyjadruje stav permutácie v danom momente výpočtu. Táto dátová štruktúra použitá v implementácii zjednodušuje rozhodovanie o nasledujúcom kroku. Prostredníctvom n vektorov dĺžky n, kde n je počet vrcholov grafu prekrytia, vyjadruje binárnymi booleovskými hodnotami, ktoré sivé hrany sa v grafe prerušení prekrývajú. Počet vrcholov je teda zhodný s počtom sivých hrán, každý z nich reprezentuje hranu $(2i, 2i + 1)$ v zdvojenej permutácii bez znamienok.

I-ty riadok vyjadruje s ktorými sivými hranami sa v grafe prerušení i-ta sivá hrana prekrýva, alebo s ktorými vrcholmi je i-ty vrchol grafu prekrytia spojený. Je zrejmé, že ak je vrchol i v grafe prekrytia spojený s vrcholom j, teda v matici na mieste $[i][j]$ je 1, platí aj opačný vzťah, a teda aj na mieste $[j][i]$ bude hodnota 1. Matica je preto diagonálne symetrická, na diagonále budú nuly.

V matici sú okrem vektorov reprezentujúcich susednosti vrcholov $v[i]$ aj ďalšie dva binárne vektory:

Tabuľka 3.1: **Matica susednosti grafu**

Matica susednosti reprezentujúca vzťahy medzi vrcholmi $v[0]$ až $v[7]$ grafu na obrázku 3.4

	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_0	0	1	0	0	0	0	0	0
v_1	1	0	0	0	0	0	1	0
v_2	0	0	0	0	0	0	1	0
v_3	0	0	0	0	1	1	0	0
v_4	0	0	0	1	0	1	0	0
v_5	0	0	0	1	1	0	0	0
v_6	0	1	1	0	0	0	0	0
v_7	0	0	0	0	0	0	0	0
p:	1	0	1	0	0	0	0	0
s:	1	0	1	2	2	2	0	0

- Prvý, označený **p**, vyjadruje **paritu vrcholu**: $p[j]$ je nulové, ak vrchol v_j má v grafe prekrytia párny počet incidentných hrán, naopak, pri nepárnom počte hrán má $p[j]$ hodnotu 1.

- Vektor **s** – **skóre vrcholu**: Podľa Bergeron (2005) môžeme skóre ľubovoľnej reverzie označiť ako $s_i = T + U_i - O_i - 1$, kde T je celkový počet orientovaných vrcholov v grafe počiatočnej permutácie. U je počet neorientovaných a O počet orientovaných vrcholov incidentných k vrcholu v_i v grafe prekrytia. Keďže je v danom momente výpočtu počet orientovaných vrcholov T v pôvodnom grafe konštantný pre všetky vrcholy a neovplyvňuje vzájomný rozdiel skóre vrcholov, môžeme ho spolu s konštantou -1 z vyjadrenia vynechať. Skóre vrcholu v_i v matici susednosti definujeme teda ako $s_i = U_i - O_i$, alebo rozdiel počtu neorientovaných a orientovaných vrcholov susediacich s vrcholom v_i .

Pre permutáciu π = (-1, 3, 5, 4, 6,-2, 7), ktorej graf prekrytia je zobrazený

na obrázku 3.4 je matica susednosti zobrazená v tabuľke 3.1.

Kroky výpočtu budú spočívať vo **výbere orientovaného vrcholu** (sivej hrany) **s najvyšším skóre**. Dvojica bodov sivej hrany určuje podinterval permutácie, nad ktorým sa uskutoční reverzia. Napríklad, ak z danej matice zvolíme z dvoch možností (v_0 a v_2) vrchol v_2, operácia reverzie sa bude prevádzať na podintervale $4; 5$ v pôvodnej permutácii bez znamienok.

Ak je permutácia utriedená a všetky vrcholy sú izolované, žiadne dve hrany v grafe prerušení sa neprekrývajú. Z toho vyplýva, že príslušná matica susednosti utriedenej permutácie bude nulová.

Počet vykonaných operácií – vystrihovaní a spájaní prekážok a reverzií určuje požadovaný údaj o reverzálnej vzdialenosti dvoch organizmov.

3.3 Cirkulárny genóm

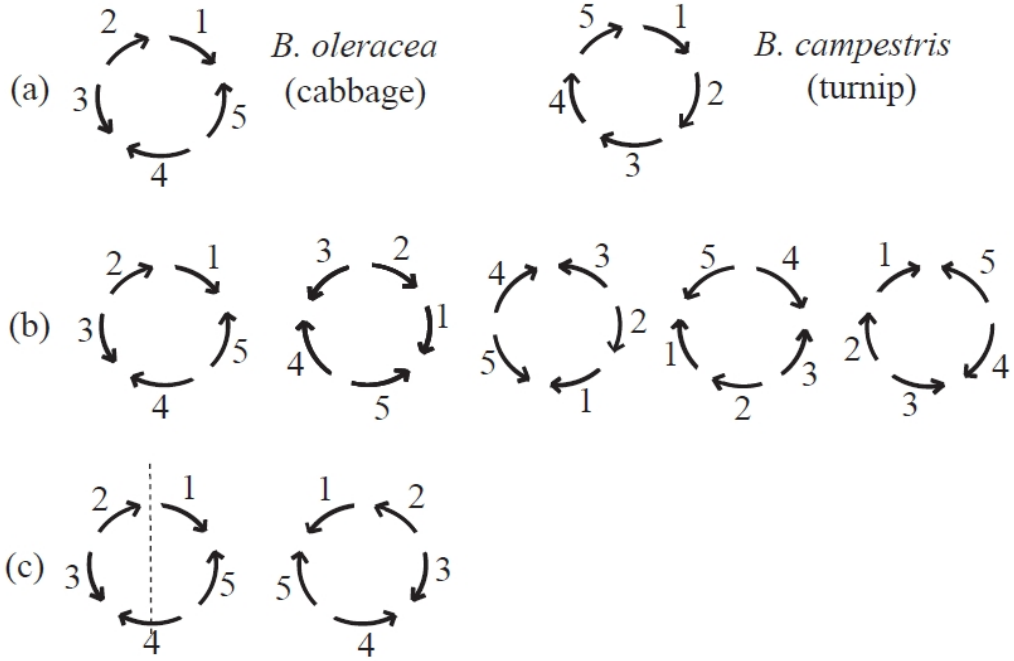
Ako bolo spomenuté v úvode práce, za cieľ sme si stanovili aplikovať algoritmus na kvasinkový mitochondriálny genóm. Hlavnou črtou tohoto genómu je cirkularita, a preto je potrebné popísaný algoritmus prispôsobiť. V tejto časti sa budeme zaoberať popisom cirkulárneho genómu a operácií súvisiacich s odlišnosťou vlastností oproti genómu lineárnemu.

3.3.1 Charakteristika

Cirkulárny genóm môžeme, rovnako ako genóm lineárny, chápať ako postupnosť genetických markérov, kde každý z nich má definovanú orientáciu. Nasledovníkom koncového elementu je ale opäť začiatočný element genómu. (Meidanis et al., 2000)

Na obrázku 3.5 v časti a) vidíme znázornený príklad cirkulárneho genómu dvoch rastlinných druhov – kapusty obyčajnej a repky olejnej. Jednotlivé

markéry sú označené číslami, orientácia je vyjadrená šípkami. Vo všeobecnosti platí, že kladný smer majú elementy s orientáciou v smere hodinových ručičiek, záporný s opačnou, teda proti smeru hodinových ručičiek. V časti b) a c) sú zobrazené ekvivalentné zápisy toho istého genómu, pričom varianty v b) časti vznikajú rotáciou genómu, v c) časti varianty vznikajú reflexiou cez znázornenú os. Rotáciu a reflexiu vysvetlíme v ďalšom texte tejto kapitoly. Varianty sa líšia rôznymi začiatočnými elementmi.



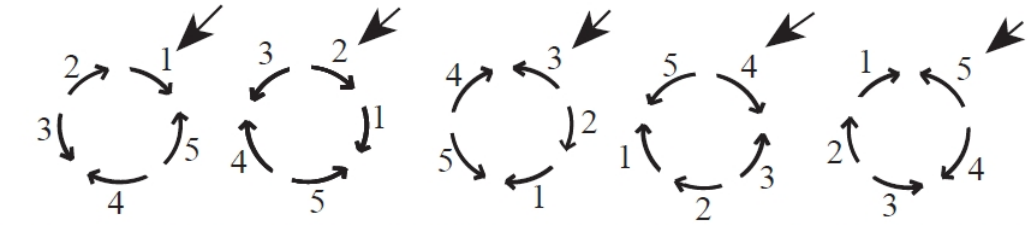
Obr. 3.5: **Príklad cirkulárneho genómu**
(Meidanis et al., 2000)

Vďaka označenému začiatočnému elementu môžeme permutáciu zapísať ako sekvenciu oznamienkovaných celých čísel. (obr.3.6) Ako bolo spomenuté, nové zobrazenia toho istého genómu vznikajú pomocou operácie rotácie alebo reflexie (Meidanis et al., 2000):

Rotácia posúva elementy permutácie doľava – proti smeru hodinových ručičiek o jednu pozíciu, pričom prvok na koncovej, poslednej pozícii sa po

rotácii stáva opäť prvkom počiatočným. Rotácia pre nás zabezpečuje vhodné vstupné zobrazenie genómu.

Reflexia prevracia postupnosť elementov permutácie a zároveň aj ich orientáciu.

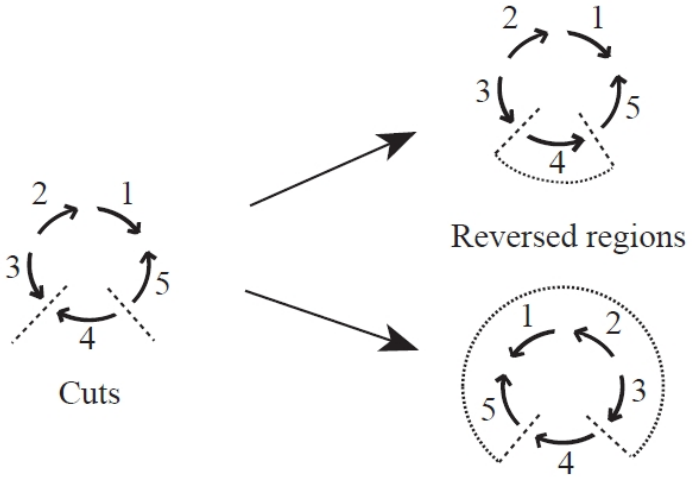


Obr. 3.6: **Ekvivalentné zápisy cirkulárneho genómu s rôznymi počiatočnými elementmi**
(Meidanis et al., 2000)

Vidíme, že cirkulárny genóm je možné zapísať ako postupnosť orientovaných elementov. V tomto tvare sa s genómom dá pracovať podobne ako s lineárnym genómom. Problém je ale v tom, že zobrazenia genómov nemusia byť navzájom porovnateľné. Preto pre zápis cirkulárneho genómu v uvedenej podobe používame variantu, kde začiatočným elementom je vždy markér 1 s kladnou orientáciou.

3.3.2 Reverzie nad cirkulárnym genómom

V predchádzajúcom texte sme uviedli, že reverzia je operácia nad podintervalom genómu. Tento podinterval je daný dvomi ohraničujúcimi bodmi. Keďže ale pracujeme s cirkulárnym genómom, tieto dva body nám definujú dva navzájom doplňujúce sa podintervaly. Situáciu môžeme vidieť na obrázku 3.7.



Obr. 3.7: **Reverzia nad cirkulárnym genómom**
(Meidanis et al., 2000)

K operácii reverzie nad cirkulárnym genómom je teda možné pristupovať dvojako. Obe tieto operácie sú, ale, vzhľadom na cirkularitu genómu významovo rovnocenné. (Meidanis et al., 2000) Výsledné permutácie je možné prečíslovať tak, aby sme získali totožnú sekvenciu. A preto môžeme nutné zmeny nášho popísaného algoritmu zúžiť na stanovenie kladného markéru 1 ako počiatočného prvku genómu a spojenie telomér do jedného vrchola, nakoľko nie je potrebné definovať dva konce genómu. Podľa Meidanis et al. (2000) je následne možné použiť operácie definované pre lineárny genóm, efekt použitia bude rovnaký.

3.4 Zhrnutie

Na vstupe do algoritmu máme dva genómy. Aby sme si zjednodušili prácu, genómy vhodne prečíslujeme tak, že cieľový genóm bude tvorený sekvenciou elementov označených radom rastúcich celých čísel, ďalej už pracujeme iba s jedným genómom. Tento ohraničíme telomérou na konci podľa úpravy pre cirkulárny genóm a zrotujeme tak, aby začínal kladnou 1. Zlejeme spojité

podintervaly do jedného prvku a permutáciu adekvátne prečíslujeme podľa vykonaných zmien. Pre odstránenie znamienok zdvojíme elementy sekvencie. Vyhľadáme prekážky a prípadné odstránime vhodnou operáciou spájania alebo vystrihovania. Vytvoríme graf prerušení, z neho graf prekrytia a štruktúry v grafe prekrytia prenesieme do matice susednosti. Ďalej iteratívne vykonávame reverziu na orientovanom elemente s najvyšším skóre, až pokým matica nie je nulová. V prípade sekvencií, ktoré tvorí vyšší počet elementov môžeme zlievanie podintervalov a následné prečíslovanie zopakovať, no je potrebné opäť vytvoriť všetky súvisiace dátové štruktúry. Vzhľadom k tomu je nutné zvážiť celkový prospech vykonania tejto operácie. Celkový počet potrebných reverzií bude výsledkom hľadania minimálnej reverzálnej vzdialenosti vstupných génov.

4 Implementácia

Na implementáciu zadania sme použili platformu Java. Samotná realizácia je rozvrhnutá do viacerých tried reprezentujúcich objekty výpočtu – graf prekrytia, komponent grafu, sivú hranu, maticu susednosti, uzol a strom riešenia a samotnú kalkuláciu výpočtu.

Vstup do algoritmu tvorí textový súbor, ktorého názov je do algoritmu zadaný ako argument programu. Na začiatku súboru sú dve celé čísla N a M. Značia, že budeme počítať reverzálnu vzdialenosť $N \times M$ génov. Za nimi nasledujú riadky s $N + M$ markérmi cirkulárnych genómov, každý genóm je ukončený symbolom @ na znak cirkularity menovaného genómu:

g_1@ (ďalších N riadkov je N genómov)
g_2@ (na každom riadku je jeden genóm)
...
g_N@ (posledný N-tý genóm)
h_1@ (ďalších M riadkov je M ďalších genómov)
h_2@
...
h_M@

Výstupom je výpis v konzole. Je to tabuľka rozmerov $nn \times mm$, kde jednotlivé hodnoty tvorí výsledok kalkulácie genómu N_{nn} a M_{mm}, pričom $1 \leq nn \leq N$ a $1 \leq mm \leq M$. Program teda spočíta reverzálnu vzdialenosť genómu N voči všetkým M genómom a následne ukončí riadok.

4.1 Vstupné dáta

Implementácia bola odskúšaná na simulovaných a reálnych dátach získaných z Valach et al. (2011). Ide o osekvenovaný cirkulárny mitochondriálny genóm 16-stich druhov triedy Hemiascomycetes. Každý z týchto genómov sa skladá z 25-tich unikátnych markérov. Vstupné sekvencie sú priložené ako súčasť CD.

4.2 Vytváranie fylogenetických stromov

Popísali sme spôsob, akým je možné získať údaj o genómovej vzdialenosti. Avšak, v úvode sme si za cieľ okrem implementácie algoritmu stanovili aj vygenerovanie fylogenetického stromu na základe porovnaní výstupných dát z jednotlivých triedení. Na vytvorenie takéhoto stromu sme použili časť programu PIVO: Phylogeny by IteratiVe Optimization (Kováč et al., 2010). Tento program, na základe vopred stanoveného stromu vypočíta, ako vyzerali genómy predkov vstupných druhov a vygeneruje fylogenetický strom. Pôvodne bol na výpočet použitý model dvojitého preseknutia a spojenia, ten sme nahradili modelom výpočtu z našej práce.

Principiálne sa PIVO snaží vytvoriť strom, v ktorom je minimalizovaný celkový počet evolučných zmien, v našom prípade reverzií. Z genómov listov vypočíta vzájomné genómové vzdialenosti. Potom označí skupinu kandidátov na predka zo susedných vrcholov a ich vhodnou kombináciou vypočíta medián. Tento v kombinácii s postavením genómov vo vrcholoch dáva isté skóre stromu. Iteratívne sa skúmajú rôzne lokálne zmeny postavenia vrcholov a ich predka v strome a zachováva sa strom s najlepším skóre, teda s najnižším počtom všetkých operácií naprieč analyzovanou evolúciou.

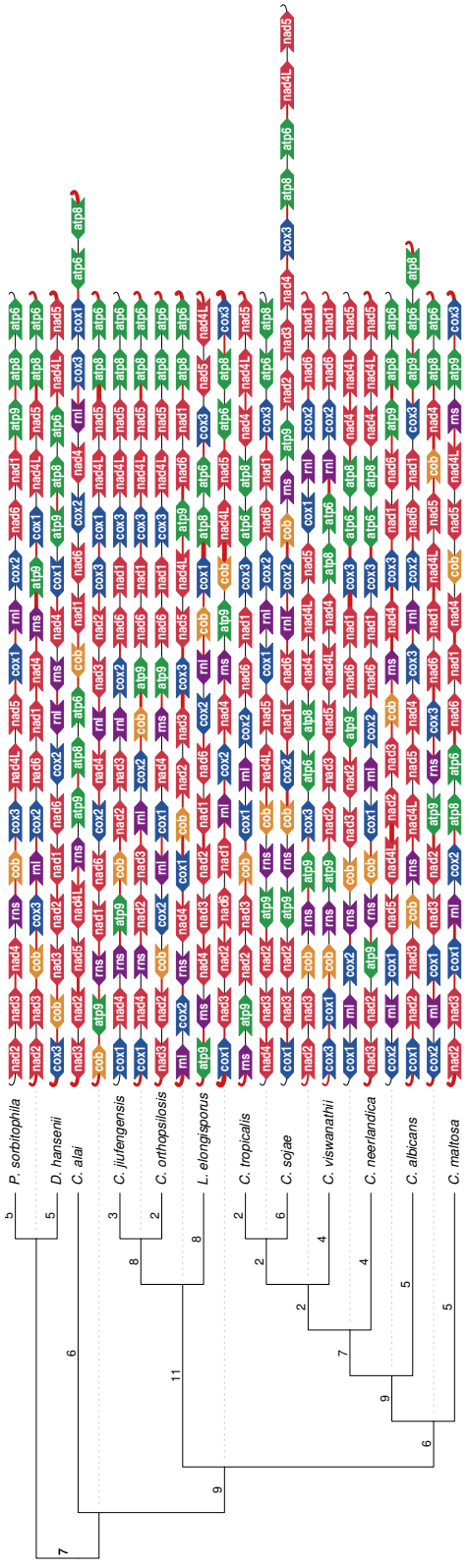
5 Výsledky a zhodnotenie

Implementáciu sme spustili najprv na simulovaných syntetických dátach, ktoré odsimulovali prítomnosť problémových štruktúr v genómoch a zvládnutie ich riešenia. Všetky sekvencie boli úspešne utriedené, simulovaný dataset je priložený na CD. Obsahoval 26 unikátnych genómov s 9-timi markérmi. Pri nastavení N a M množín na vstupe na hodnoty $N = 13$ a $M = 13$ je možné v krátkom čase odsimulovať 169 jednoduchých triedení, takže tento dataset bol pre overenie vyhovujúci.

Overenie na reálnych kvasinkových genómoch prebehlo taktiež korektne. Po spojení s programom PIVO (Kováč et al., 2010) sme získali fylogenetický strom s celkovým počtom reverzií 116. Strom, ktorý vznikol manuálnym usporiadaním druhov a je v súčasnosti akceptovaný v biológii má skóre 78. Na pohľad je rozdiel skóre značný, ale keďže PIVO iteratívne hľadá lepšiu konšteláciu vrcholov tak, aby strom mal nižšie skóre a spolu s našim programom bol spustený na daných dátach iba niekoľkokrát, predpokladáme, že viacnásobnou iteráciou chodu algoritmu by sa skóre zlepšilo, no hľadanie stromu s minimálnym skóre nebolo cieľom našej práce.

Každopádne funkčnosť implementácie algoritmu popísaného v tejto diplomovej práci bola overená na reálnych dátach mitochondriálnych genómov kvasiniek triedy Hemiascomycetes. Výsledný fylogenetický strom z cirkulárnych genómov skúmaných kvasiniek a ich predkov je na obrázku 5.1.

Na obrázku je v ľavej časti vyobrazený výsledný strom s vypočítanými reverzálnymi vzdialenosťami medzi jednotlivými vrcholmi, pričom listy tvoria vstupné genómy zobrazených druhov. V pravej časti vidíme postupnosť markérov, šípka vyjadruje orientáciu daného markéru.



Obr. 5.1: **Výsledný fylogenetický strom**

6 Záver

Podľa zadania bolo cieľom práce preštudovať metódy rekonštrukcie evolučných histórií pomocou operácie reverzie, naimplementovať algoritmus umožňujúci takúto analýzu a aplikovať ho na reálne dáta z kvasinkových mitochondriálnych genómov.

V úvode práce sme stručne popísali dva ťažiskové príbuzné modely riešenia problému hľadania najkratšej genómovej vziadlenosti, biologické pozadie problému a následne sme popísali aj algoritmus rekonštrukcie evolučnej histórie za použitia operácie reverzie. Tento algoritmus sme následne naimplementovali a overili na dátach reálnych genómov kvasiniek. Dataset genómov je priložený, je možné túto implementáciu spustiť a overiť. Všetky tri vytýčené ciele boli teda splnené.

Nad rámec zadania sme náš algoritmus spojili s programom z Kováč et al. (2010) a vygenerovali fylogentický strom a jeho vizualizáciu. Takýmto spôsobom je možné vizualizovať výsledky z ľubovoľných vstupných dát, vďaka grafickému zobrazeniu sa výsledok stáva prehľadnejší, pridaná informačná hodnota sú genómy predkov vstupných druhov.

V ďalšej práci by bolo vhodné zefektívniť štruktúry v implementácii a zakomponovať bitový paralelizmus operácií popísaný v Bergeron (2005), prípadne inak optimalizovať implementáciu.

Literatúra

Bergeron, A. *A very elementary presentation of the Hannenhalli-Pevzner theory. Discrete Applied Mathematics*. **2005**, 146, 2, s. 134 – 145.

Bergeron, A. – **Stoye**, J. *The Genesis of the DCJ Formula*. In **Chauve**, C. – **El-Mabrouk**, N. – **Tannier**, E. (Ed.) *Models and Algorithms for Genome Evolution*, 19 / *Computational Biology*. Springer London, **2013**. s. 63 – 81. doi: 10.1007/978-1-4471-5298-9˙5. Dostupné z: <http://dx.doi.org/10.1007/978-1-4471-5298-9_5>. ISBN 978-1-4471-5297-2.

Bergeron, A. – **Mixtacki**, J. – **Stoye**, J. *A Unifying View of Genome Rearrangements*. In **Bücher**, P. – **Moret**, B. (Ed.) *Algorithms in Bioinformatics*, 4175 / *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, **2006**. s. 163 – 173. Dostupné z: <http://dx.doi.org/10.1007/11851561_16>. ISBN 978-3-540-39583-6, 10.1007/11851561$_1$6.

Böhmer, D. et al. *Úvod do biológie a humánnej genetiky*. Asklepolis, Bratislava, **2008**. ISBN 978-80-7167-122-0.

DiMauro, S. – **Schon**, E. A. *Mitochondrial DNA mutations in human disease. American Journal of Medical Genetics*. **2001**, 106, 1, s. 18 – – 26. ISSN 1096-8628. doi: 10.1002/ajmg.1392. Dostupné z: <http://dx.doi.org/10.1002/ajmg.1392>.

Gömöry, D. *Genetika a šľachtenie lesných drevín, Návody na cvičenia*. Technická univerzita vo Zvolene, Lesnícka fakulta, **2010**. Dostupné

z: <http://www.tuzvo.sk/files/LF-KF/Pedago-Predmety/skripta.genetika.NCV-1.pdf>.

Hannenhalli, S. – **Pevzner**, P. A. *Transforming men into mice (polynomial algorithm for genomic distance problem). Foundations of Computer Science, Annual IEEE Symposium on.* **1995**, 0, s. 581. ISSN 0272-5428. doi: http://doi.ieeecomputersociety.org/10.1109/SFCS.1995.492588.

Hannenhalli, S. – **Pevzner**, P. A. *Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. J. ACM.* January **1999**, 46, s. 1 – 27. ISSN 0004-5411. doi: http://doi.acm.org/10.1145/300515.300516. Dostupné z: <http://doi.acm.org/10.1145/300515.300516>.

Hraška, Š. *Molekulárne základy dedičnosti.* Nepublikovaný materiál, **2005**.

http://education.technyou.edu.au. *Dvojzávitnica DNA* [online]. [cit. 1. 1. 2012]. Dostupné z: <http://education.technyou.edu.au/sites/default/files/images/bio/helix.jpg>.

Kaplan, H. – **Shamir**, R. – **Tarjan**, R. E. *A Faster and Simpler Algorithm for Sorting Signed Permutations by Reversals. SIAM J. Comput.* December **1999**, 29, s. 880 – 892. ISSN 0097-5397. doi: 10.1137/S0097539798334207. Dostupné z: <http://portal.acm.org/citation.cfm?id=337729.337790>.

Kováč, J. – **Brejová**, B. – **Vinař**, T. *A New Approach to the Small Phylogeny Problem. ArXiv e-prints.* December **2010**. Dostupné z: <http://compbio.fmph.uniba.sk/pivo/>.

Meidanis, J. – **Walter**, M. E. M. T. – **Dias**, Z. *Reversal Distance of Signed Circular Chromosomes*, **2000**.

Mixtacki, J. *The double cut and join operation and its applications to genome rearrangements.* PhD thesis, Bielefeld University, **2008**.

Pevzner, P. A. *Genome Rearrangements In: Computational Molecular Biology: An Algorithmic Approach*. Cambridge, MA, MIT Press, **2000**. ISBN 0-262-16197-4.

Schaefer, A. M. – **Taylor**, R. W. – **Turnbull**, D. M. *The mitochondrial genome and mitochondrial muscle disorders. Current Opinion in Pharmacology*. **2001**, 1, 3, s. 288 – 293. ISSN 1471-4892. doi: DOI:10.1016/S1471-4892(01)00051-0. Dostupné z: <http://www.sciencedirect.com/science/article/B6W7F-431YRPR-J/2/540f0b21281913e380574f350815637c>.

Sperschneider, V. – **Sperschneider**, J. – **Scheubert**, L. *Bioinformatics: Problem Solving Paradigms*. Springer, **2010**. ISBN 9783642097263.

Sturtevant, A. H. – **Dobzhansky**, T. *Inversions in the Third Chromosome of Wild Races of Drosophila Pseudoobscura, and Their Use in the Study of the History of the Species. Proceedings of the National Academy of Sciences of the United States of America*. July **1936**, 22, 7, s. 448 – – 450. ISSN 0027-8424. Dostupné z: <http://view.ncbi.nlm.nih.gov/pubmed/16577723>.

Taanman, J.-W. *The mitochondrial genome: structure, transcription, translation and replication. Biochemica et Biophysica Acta 1410*. **1999**.

Thomas, S. B. *Fylogenetický strom* [online]. [cit. 4. 8. 2011]. Dostupné z: <http://evillusion.files.wordpress.com/2010/09/treeolif.jpg>.

Valach, M. et al. *Evolution of linear chromosomes and multipartite genomes in yeast mitochondria. Nucleic Acids Research*. **2011**, 39, 10, s. 4202 – – 4219.

Wanrooij, S. – **Falkenberg**, M. *The human mitochondrial replication fork in health and disease. Biochimica et Biophysica Acta (BBA) - Bioenergetics*. **2010**, 1797, 8, s. 1378 – 1388. ISSN 0005-2728. doi: DOI:10.1016/j.bbabio.2010.04.015. Dostupné z: <http://www.sciencedirect.com/science/article/B6T1S-4YX7KHX-1/2/cb21fc38fbda081dce5ff75082c857fa>.

Prílohy

1. CD nosič obsahujúci zdrojové kódy textu, implementácie a dokumentácie k nej.