VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

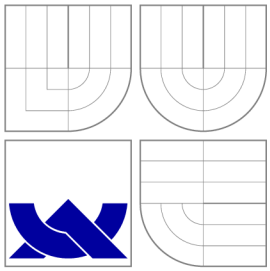
APLIKACE PRO ZPRACOVÁNÍ DAT Z OBLASTI EVOLUČNÍ BIOLOGIE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

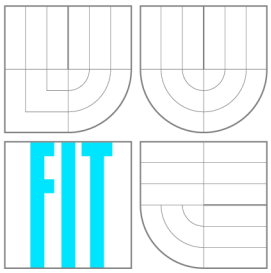
AUTOR PRÁCE
AUTHOR

IVAN VOGEL

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

APLIKACE PRO ZPRACOVÁNÍ DAT Z OBLASTI EVOLUČNÍ BIOLOGIE

APPLICATION FOR THE DATA PROCESSING IN THE AREA OF EVOLUTIONARY BIOLOGY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — IVAN VOGEL
AUTHOR

VEDOUCÍ PRÁCE — Ing. PAVEL OČENÁŠEK, Ph.D.
SUPERVISOR

BRNO 2011

Abstrakt

Tvorba fylogenetických stromů je v současnosti běžnou vizualizační metodou na vyjádření evolučních vztahů druhů. Tato práce se soustředí na vysvětlení matematické teorie popisující molekulární fylogenetiku a na návrh modifikovaného algoritmu na odvozování evolučního stromu založeného na vnitroskupinové analýze nukleotidových a aminokyselinových sekvencí. Dále popisuje objektový návrh a implementaci těchto metod v jazyce Python a integraci nástroje do velkého bioinformatického portálu. Navžená řešení dávají lepší výsledky oproti konvenčním metodám při zhlukování předdefinovaných shluků a jsou co do vstupních dat široce použitelné. Práce také závěrem diskutuje možné jiné aplikace navrhnutých metod a ich rozšíření na iné obory informačních technologií.

Abstract

Phylogenetic tree inference is a very common method for visualising evolutionary relationships among species. This work focuses on explanation of mathematical theory behind molecular phylogenetics as well as design of a modified algorithm for phylogenetic tree inference based on intra-group analysis of nucleotide and amino acid sequences. Furthermore, it describes the object design and implementation of the proposed methods in Python language, as well as its integration into powerful bioinformatic portal. The proposed modified algorithmic solutions give better results comparing to standard methods, especially on the field of clustering of predefined groups. Finally, future work as well as an application of proposed methods to other fields of information technology are discussed.

Klíčová slova

substituční model, fylogenetický strom, markovský řetězec, vnitroskupinová analýza, neighbor joining, Mobyle

Keywords

substitution model, phylogenetic tree, Markov chain, intragroup analysis, neighbor joining, Mobyle

Citace

Ivan Vogel: Aplikace pro zpracování dat z oblasti evoluční biologie, diplomová práce, Brno, FIT VUT v Brně, 2011

Aplikace pro zpracování dat z oblasti evoluční biologie

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Pavla Očenáška, Ph.D.

.....
Ivan Vogel
25.5.2011

Poděkování

Týmto chcem poďakovať môjmu konzultantovi, Mgr. Františkovi Zedkovi, za cenné odborné rady pri písaní tejto práce a priateľský prístup počas celej doby našej spolupráce.

Obsah

Kapitola 1

Úvod

Od čias Darwina a publikovania jeho kľúčového diela evolučnej biológie [4] ubehlo už vyše storočie. Za tú dobu zaznamenal tento odbor veľký pokrok. Vďačí za to okrem iného úspechom molekulárnych biológov (sekvenovanie DNA) a čoraz tesnejšej väzbe na moderné výpočtové metódy bioinformatiky, bez ktorých by množstvo otázok zostalo dodnes nezodpovedaných. Táto práca sa snaží poskytnúť plynulý prechod od evolučnej biológie cez matematiku až po informatiku, aby vzápätí mohla podať poskytnuté a osvojené znalosti a postupy do rúk molekulárnej fylogenetiky.

V úvode sa čitateľ dozvie základné poznatky z oblasti evolučnej a molekulárnej biológie. V ďalších častiach sa potom práca podrobnejšie venuje molekulárnej fylogenetike ako podmnožine spomínanej oblasti.

Kapitola 3 poskytuje chronologický pohľad na tvorbu fylogenetického stromu, je nasledovaná kapitolou mapujúcou dostupnosť komerčného aj nekomerčného programového vybavenia venujúceho sa fylogenetickej analýze biosekvencií.

Na základe teoretických poznatkov, skúseností s prácou s dostupným programovým vybavením a pokynov konzultanta tejto práce je navrhnutá nová aplikácia na rekonštrukciu fylogenetických stromov.

Ústredným bodom práce je, čo do významu, reprezentácia skupiny sekvencií pomocou sekvenčného profilu získaného frekvenčnou analýzou a následné matematické odvodenie originálnych metrík na určenie podobnosti takýchto skupín. Tieto metriky sa opierajú o teoretické základy uvedené v úvodných kapitolách práce, konkrétne o substitučné modely.

Spolu so spomínanými metrikami bude predstavený originálny algoritmus TDCG založený na divizívnom prístupe. Navrhnuté riešenia budú implementované v jazyku Python a naviac bude predstavený postup, ako vytvorené nástroje integrovať do bioinformatického portálu Mobyle.

Ako bude zrejmé zo záverečných kapitol, práca poskytuje bohaté možnosti na rozšírenia zasahujúce do viacerých odborov bioinformatiky, umelej inteligencie ale aj spracovania v špecializovanom hardware.

Úplnym záverom bude zhodnotený prínos vytvorených riešení v kontexte existujúcich metód na poli bioinformatickej analýzy.

Kapitola 2

Biologická evolúcia

Biologická evolúcia je dlhodobý, samovoľne prebiehajúci proces, v ktorého priebehu vznikajú, príp. jednorázovo vznikli z neživých systémov systémy živé, a tieto živé systémy sa potom ďalej vyvíjajú a vzájomne diverzifikujú. Evolučná biológia ako vedná disciplína sa zaoberá štúdiom vlastností biologickej evolúcie a jej konkrétnymi mechanizmami [8].

2.1 Vysvetlenie základných mechanizmov

V rámci biologickej evolúcie funguje niekoľko procesov, vďaka ktorým sa organizmy vyvíjajú, vzájomne divergujú (vznikajú nové druhy) a kumulujú zmeny vo svojej štruktúre a správaní. Medzi tieto mechanizmy patria predovšetkým:

- Prirodzený výber (selekcia) – mechanizmus popisujúci systematický výber z množiny náhodne vzniknutých dedičných zmien (mutácií) také, ktoré sú účelné z hľadiska života ich nositeľov. Prežijú teda predovšetkým najzdatnejší jedinci, pretože ich účelné vlastnosti im zabezpečia nezanedbateľný priestor v ďalších generáciách.

- Mutagenéza – ide o hybný mechanizmus celej evolúcie, nakoľko ako jediný zo spomínaných javov generuje genetickú variabilitu. Hľadiac na túto problematiku očami informatika, ide o určitý náhodný prepis programového kódu (daného sekvenciou nukleotidov, prípadne aminokyselín). Týmto princípom sa v posledných desaťročiach hojne inšpirovala aj samotná informatika [29]. Nakoľko je výskyt mutácií kľúčový pre metódy molekulárnej fylogenetiky, bude im venovaná samostatná sekcia 2.3.

- Genetický drift – alebo genetický posun. Ide o náhodné posuny vo frekvencii výskytu jednotlivých aliel (konkrétnych variant génov) v genofonde určitej populácie. Je to spôsobené tým, že v rámci populácie ani zdaľeka nedôjde ku vzajomnej kombinácii všetkých aliel zastúpených v genofonde. Reálne vznikne oveľa menej genotypov, ako je teoreticky možné [8].

- Molekulárny ťah – zahŕňa procesy súvisiace so šírením repetitívnych úsekov DNA (napr. génov v niekoľkých kópiách) v rámci genómu i celého genofondu. Svojou funkciou pripomína genetický drift, na rozdiel od neho však nepracuje na báze náhody. V rámci molekulárneho ťahu nahrádza jedna alela inú alelu nie preto, že by bola výhodnejšia, ale preto, že je účelnejšie rozložená na úrovni DNA (väčší počet kópií).

2.2 Biosekvencie

Sekcia poskytuje stručný popis nukleotidových a polypeptidových reťazcov.

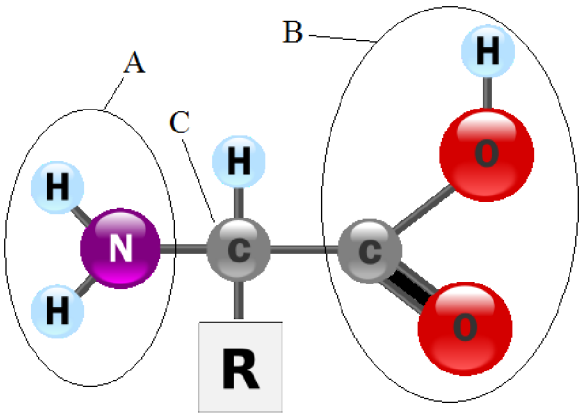
2.2.1 Štruktúra DNA a RNA

Základnou jednotkou DNA je nukleozid, ktorý je zložený z cukru deoxyribózy, na ktorú ja naviazaná niektorá zo štyroch dusíkatých báz, t.j. dvoch pyrimidínových báz - tymínu (T) a cytozínu (C), resp. purínových báz - adenínu (A) a guanínu (G). Susedné nukleozidy sú spojené medzi 5'- a 3'- atómom uhlíka. Nukleozid vytvára spolu s fosfátovou skupinou nukleotid. Molekulu DNA tvoria dva lineárne reťazce, v ktorých sa nepravidelne striedajú jednotlivé nukleotidy. Lineárne reťazce sú spojené tzv. vodíkovými mostíkmi medzi komplementárnymi dusíkatými bázami, t.j. AT(dva mostíky) alebo GC(tri mostíky). Reťazce sú antiparalelné, t.j. 5'-koncom jedného vlákna sa páruje s 3'-koncom druhého vlákna. V priestore vytvárajú dvojzávitnicu. Molekula RNA je podobná DNA s tým rozdielom, že miesto cukru deoxyribózy sa tu nachádza ribóza a tymín je nahradený uracilom (U) [8].

Postupnosť nukleotidov tvorí primárnu štruktúru nukleovej kyseliny. Kóduje gény, regulačné oblasti a všetky informácie potrebné pre štruktúru a funkciu každej bunky organizmu. Z hľadiska bioinformatiky sa na molekuly DNA a RNA pozerá ako na reťazec nad abecedou nukleotidov.

2.2.2 Štruktúra proteínov

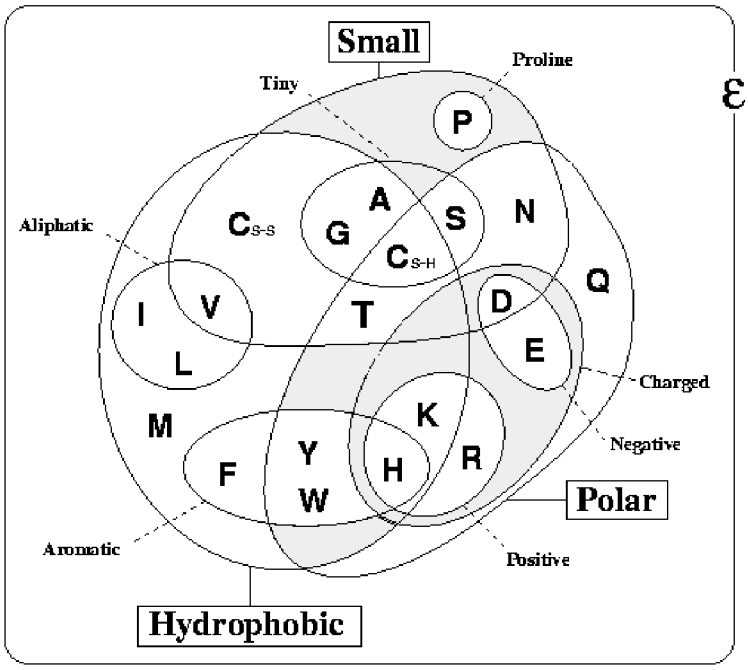
Proteíny sú tvorené sekvenciou aminokyselín. Každá aminokyselina je zložená z tzv. aminoskupiny (obr.2.1, množina A), alfa uhlíka (obr.2.1, množina C) a karboxylovej skupiny (obr.2.1, množina B). Na alfa uhlík sa naväzuje tzv. postranný reťazec(obr.2.1, množina R), ktorého štruktúra odlišuje od seba jednotlivé aminokyseliny.



Obrázok 2.1: Schéma aminokyseliny [36]

Proteíny tvorí dovedna 20 aminokyselín. Na základe chemicko–fyzikálnych vlastností ich možno rozdeliť do skupín, ako znázorňuje Vennov diagram na obrázku 2.2.

Proteíny vykonávajú v organizmoch rozmanité funkcie – stavebné, katalyzačné (enzýmy), transportné (hemoglobín), pohybové, zásobné, signálne i regulačné. Z hľadiska bioinformatiky sa na primárnu štruktúru proteínu hľadí ako na reťazec nad abecedou aminokyselín [23].



Obrázok 2.2: Vennov diagram znázorňujúci vzťahy jednotlivých aminokyselín [3]

2.2.3 Centrálna dogma molekulárnej biológie

Popisuje prenos informácie z DNA do proteínov [38]. Štandardná cesta spočíva v nasledovných krokoch:

- replikácia alias kopírovanie DNA do DNA – za účasti viacerých enzýmov (okrem iných DNA–polymerázy) sa dvojzávitnica DNA rozpletie a ku každému z pôvodných reťazcov, ktoré slúžia ako templát, sa dosyntetizuje komplementárny reťazec. Vzniknú tak z jednej dsDNA[1] dve identické dsDNA.
- transkripcia alias prepis informácie z DNA do mRNA (mRNA je messenger RNA čiže akýsi posol, ktorý donesie prepísanú informáciu z DNA na preklad do ribozómov)
- translácia alias preklad informácie z mRNA do proteínu. Každá aminokyselina je kódovaná tzv. kodónom alebo tripletom (trojica nukleotidov) na mRNA. Kodóny sa prekladajú v bunkových organelách, ribozómoch, na sekvenciu aminokyselín podľa tabuľky na obrázku 2.3. Transport aminokyselín do ribozómu zabezpečujú molekuly tRNA.

Obrázok 2.4 schmematicky zhŕňa uvedené poznatky. Tzv. otvoreným čítacím rámcom nazývame oblasť medzi START a STOP kodónom (viď obrázok 2.3). Táto oblasť potenciálne kóduje proteín. (Pozn.: Sekcia 2.3 diskutuje okrem iného drastické následky posunu čítacieho rámca v dôsledku bodových mutácií.)

[1] double stranded DNA, t.j. dvojreťazcová DNA

	T			C			A			G		
T	TTT	Phe	F	TCT	Ser	S	TAT	Tyr	Y	TGT	Cys	C
	TTC			TCC			TAC			TGC		
	TTA	Leu	L	TCA			TAA	STOP		TGA	STOP	
	TTG			TCG			TAG			TGG	Trp	W
C	CTT	Leu	L	CCT	Pro	P	CAT	His	H	CGT	Arg	R
	CTC			CCC			CAC			CGC		
	CTA			CCA			CAA	Gln	Q	CGA		
	CTG			CCG			CAG			CGG		
A	ATT	Ile	I	ACT	Thr	T	AAT	Asn	N	AGT	Ser	S
	ATC			ACC			AAC			AGC		
	ATA			ACA			AAA	Lys	K	AGA	Arg	R
	ATG	Met	M	ACG			AAG			AGG		
G	GTT	Val	V	GCT	Ala	A	GAT	Asp	D	GGT	Gly	G
	GTC			GCC			GAC			GGC		
	GTA			GCA			GAA	Glu	E	GGA		
	GTG			GCG			GAG			GGG		

Obrázok 2.3: Štandardný genetický kód

2.3 Mutácie

Pod pojmom mutácie rozumieme také zmeny v štruktúre genetického materiálu, ktoré potenciálne menia zmysel genetickej informácie, dodržiavajú však syntaktické pravidlá zápisu. Pokiaľ sú tieto pravidlá porušené, hovoríme o poškodení DNA.

Mutácie možno rozdeliť podľa ich fyzickej povahy nasledovne:

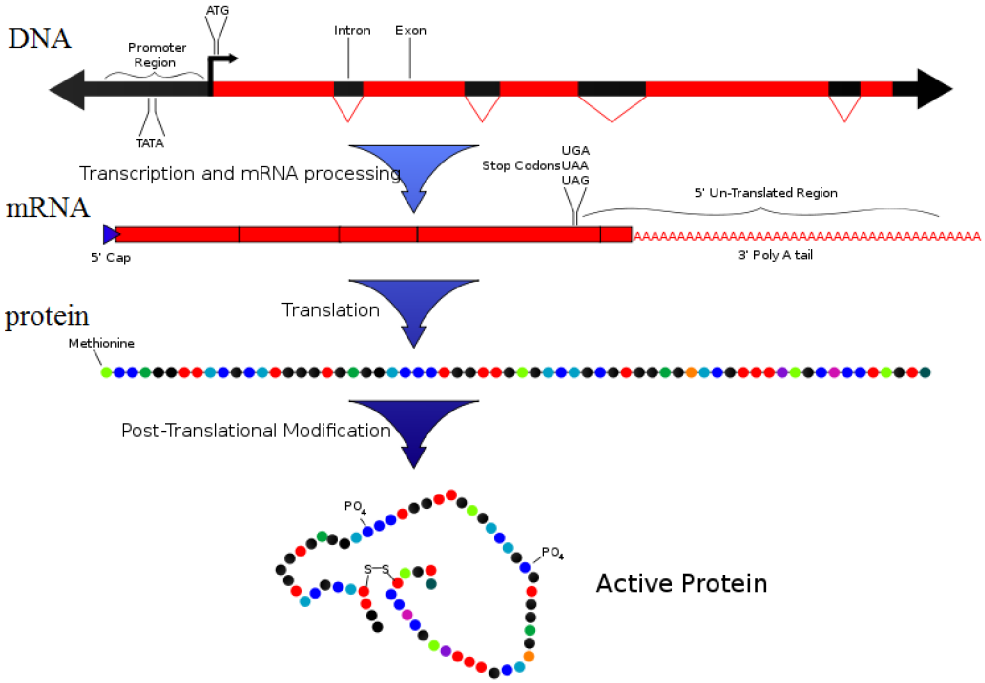
- bodové
- reťazcové
- na úrovni chromozómov
- na úrovni celého genómu

Z hľadiska tejto práce sú kľúčové bodové mutácie. Delia sa na:

- zámenové mutácie – substitúcie (transverzie a tranzície)
- delécie
- insercie

V úsekoch kódujúcich proteíny treba rozdeliť bodové mutácie podľa toho, aký majú vplyv na výsledný proteín.

- synonymné mutácie (samesense) – sú dôsledkom degenerovanosti genetického kódu (jednu aminokyselinu kóduje viacero rôznych kodónov); mutácie tohto typu sú z hľadiska prirodzeného výberu väčšinou neviditeľné, teda neutrálne
- nesynonymné (missense) mutácie – spôsobujú, že je jedna aminokyselina nahradená inou; pokiaľ má nahradená aminokyselina podobné fyzikálno-chemické vlastnosti, ide o konzervatívnu zámenu [8], ktorá nemusí výrazne meniť biologickú funkciu.
- nezmyselné (nonsense) mutácie – triplet kódujúci aminokyselinu mutuje na jeden z troch terminačných kodónov, čo má veľmi vážne následky pri syntéze proteínu (predčasné zastavenie translácie)



Obrázok 2.4: Centrálna dogma molekulárnej biológie [36]

Najdrastickejšie zmeny v preklade spôsobujú insercie a delécie, v dôsledku ktorých sa posunie celý čítací rámec a teda nukleotidová sekvencia sa preloží do úplne odlišného sledu aminokyselín.

2.4 Zarovnanie biologických sekvencií

Zarovnanie (*alignment*) biologických sekvencií patrí k jednej z najdôležitejších úloh bioinformatiky. Dáva odpoveď na otázky o podobnosti génov, odlišnosti organizmov a často pomáha určiť aj funkciu daného génu. V neposlednom rade zarovnanie poskytuje informáciu o konzervovaných častiach sekvencií a o evolučnej príbuznosti jednotlivých druhov, čo je v prípade tejto práce kľúčové. Základný typ zarovnania je párové zarovnanie dvoch sekvencií. Môže byť lokálne alebo globálne. V prípade lokálneho zarovnania sa hľadá určitý spoločný sekvenčný motív, napríklad výskyt nejakej krátkej sekvencie v rámci celého genómu. Základným algoritmom na riešenie problému lokálneho zarovnania je algoritmus od Watermana a Smitha [30]. Využíva prístupy dynamického programovania.

V prípade globálneho zarovnania sa predpokladá určitá miera podobnosti sekvencií po celej ich dĺžke, ide zväčša o tzv. homologné sekvencie s rovnakou alebo veľmi podobnou dĺžkou a podobnou evolučnou históriou. Základným algoritmom globálneho zarovnania je algoritmus Needleman–Wunsch [25], ktorý podobne ako Waterman-Smith využíva dynamické programovanie.

Spomenuté metódy sú výpočtovo veľmi náročné, preto sa v praxi využívajú heuristické algoritmy, ako napríklad BLAST [1] a FASTA [18].

Rozšírením párového zarovnania je viacnásobné zarovnanie, kde sa požaduje väčšinou zarovnanie na globálnej úrovni. Na vstupe sa teda predpokladá množina sekvencií, ktorej

jednotlivé prvky sa upravujú vkladaním medzier (tzv. indelov – insercií a/alebo delécií). Hľadá sa teda taká konfigurácia jednotlivých reťazcov, kde sa pod sebou nachádzajú vždy znaky identické, v prípade aminokyselín znaky maximálne podobné podľa substitučnej matice(viď sekciu 3.1).

V prípade globálneho zarovnania sú to napríklad algoritmy CLUSTAL [11] a MUSCLE [6]. Nakoľko tento typ zarovnania bude využitý pri fylogenetickej analýze, budú algoritmy diskutované podrobnejšie.

2.4.1 CLUSTAL

Algoritmus CLUSTAL [11] je úplne automatizovaný algoritmus viacnásobného zarovnania. Pozostáva z troch základných krokov:

- Zostrojenie matice párových vzdialeností jednotlivých sekvencií. Na určenie párových vzdialeností je na výber buď rýchla, ale menej presná metóda, ktorá vyčíta hodnoty skóre z najdlhších diagonál bodového diagramu, alebo presnejšia, no výpočtovo náročnejšia metóda, kde sa optimalizuje zarovnanie každej dvojice sekvencií. Tento krok je časovo najzložitejšou časťou algoritmu, pre N sekvencií je nutné vykonať $N(N-1)/2$ porovnaní, s počtom sekvencií rastie teda čas kvadraticky.

- Na základe matice podobností sa vytvorí zhlukovou analýzou tzv. guide tree alebo vodiaci strom.

- Podľa vytvorenej matice podobností sa potom prevedie párové globálne zarovnanie dvoch najpodobnejších sekvencií a postupne sa k nim zarovnávajú ďalšie, vzdialenejšie sekvencie.

Algoritmus funguje dobre pre sekvencie, ktoré majú veľký počet konzervovaných oblastí, a teda nie je nutné pridávať veľký počet medzier. V opačnom prípade hrozí vznik medzerových artefaktov – CLUSTAL totiž pri zarovnávaní medzery pridáva, nevie ich však odstraňovať. Ak sa teda raz "nesprávne"rozhodne, túto skutočnosť kompenzuje vkladaním ďalších medzier, čo má za následok vytvorenie medzerových zhlukov a signalizuje, že zarovnanie nie je optimálne.

Implementáciou algoritmu CLUSTAL je program ClustalW[2] s textovým užívateľským rozhraním a s podporou vo viacerých webových službách a jeho rozšírenie ClustalX s grafickým užívateľským rozhraním.

2.4.2 MUSCLE

Pre popis algoritmu je nutné najprv čitateľa oboznámiť s tzv. počítaním k-tic.

Počítanie k-tic pre 2 sekvencie

Na vstupe predpokladáme sekvencie X a Y. Pomocou posuvného okienka *sliding window* sa vytvorí množina všetkých k-tic W_k. Následne sa vytvorí množina výskytov týchto podreťazcov pre obidve sekvencie zvlášť, teda množiny c_X a c_Y. Medzi týmito výskytmi sa vypočíta Euklidovská vzdialenosť (viď rovnica 2.1), ktorá určí mieru podobnosti týchto sekvencií. Je totiž štatisticky dokázané, že existuje korelácia medzi zhodou v početnosti výskytu podreťazcov a podobnosťou celých sekvencií [22].

[2]dostupný na http://www.clustal.org/

$$d(X,Y) = (c_X - c_Y).(c_X - c_Y) = \sum_{i=1}^{n} (c_{Xi} - c_{Yi})^2 \tag{2.1}$$

Skóre Sum of Pairs

Ide o metriku na posudzovanie kvality viacnásobného zarovnania, ktorá vychádza z myšlienky, že skóre viacnásobného zarovnania je vysoké, pokiaľ je vysoké i skóre dvojíc zarovnaných sekvencií.

Výpočet prebieha nasledovne: Pre každý stĺpec viacnásobného zarovnania je vypočítané skóre všetkých dvojíc znakov:

$$S_{Col}(a_1, a_2, \ldots, a_n) = \sum_{i \neq j} S(a_i, a_j) \tag{2.2}$$

$$S(a_i, a_j) = \begin{cases} M(a_i, a_j) & \text{if}(a_i \text{ and } a_j \neq -) \\ 0 & \text{if}(a_i \text{ and } a_j = -) \\ -g & \text{otherwise} \end{cases} \tag{2.3}$$

$M(a_i, a_j)$ je položka skórovacej matice pre nukleotidy, resp. aminokyseliny a_i, a_j a g je penalizácia za vloženie medzery.

Popis

Algoritmus MUSCLE pozostáva z 3 fáz:

1. *Draft progressive* – cieľom je vytvoriť prvotné viacnásobné zarovnanie sekvencií, dôraz sa kladie na rýchlosť, nie na presnosť. Fáza pozostáva z týchto krokov:
   - Výpočet vzdialeností k-tic ($k$mer distance) pre každý pár vstupných sekvencií, vytvorenie matice vzdialeností $D1$.
   - Zhlukovanie podľa matice $D1$ pomocou algoritmu UPGMA (popis viď podsekcia 3.3.2), ktorého výsledkom je binárny strom $TREE1$.
   - Na základe $TREE1$ sa vytvorí prvotné zarovnanie. Pre každý vnútorný uzol stromu sa vytvorí profil podľa synovských uzlov. Uzly stromu sa prechádzajú algoritmom prefix order (teda synovia pred rodičmi). Na konci tohto kroku sa nachádza v koreňovom uzle prvotné viacnásobné zarovnanie sekvencií označované ako $MSA1$.
2. *Improved progressive* – najpravdepodobnejším zdrojom chýb v prvej fáze je odhad vzdialeností k-tic, táto fáza preto spresňuje pôvodný odhad a pozostáva z nasledovných krokov:
   - Z $MSA1$ vypočítať pomocou Kimurovej vzdialenosti (popis viď 3.1.2) a skonštruovať maticu vzdialeností $D2$.
   - Z matice $D2$ sa vytvorí pomocou UPGMA binárny strom $TREE2$
   - Podobne ako v poslednom kroku prvej fázy sa podľa stromu $TREE2$ vytvorí viacnásobné zarovnanie $MSA2$. To sa však počíta len pre tie podstromy, ku ktorým došlo k zmene v porovnaní so stromom $TREE1$

3. *Refinement* – Iteratívne spresňovanie výsledku.

    - Zo stromu $TREE3$ sa vyberie hrana (zoznam hrán je zoradený podľa klesajúcej vzdialenosti od koreňového uzlu). Strom sa na základe tejto hrany rozdelí na dva podstromy. Pre každý z nich sa vytvorí nový profil úpravou profilu z pôvodného stromu.
    - V prípade, že sa zvýši skóre SP (Sum of Pairs) nového viacnásobného zarovnania, tak sa toto uchová, inak sa zahodí.
    - Predchádzajúce kroky sa opakujú tak dlho, kým konvergujú alebo keď sa dosiahne limit daný užívateľom.

Výstup v podobe viacnásobného zarovnania je vždy v treťom kroku 1., 2. aj 3. fázy. To sú tým pádom miesta, kde možno *a*lignment ukončiť. Prvé dve fázy sa niekedy nazývajú MUSCLE-p. Pokiaľ značí L dĺžku sekvencií a N ich počet, potom je časová zložitosť prvých dvoch fáz spolu $O(N^2 + NL + L^2)$. Tretia fáza pridáva k časovej zložitosti ešte výraz $O(N^3L)$.

Kapitola 3

Molekulárna fylogenetika

Fylogenéza ako vznik a vývoj jednotlivých vývojových línií organizmov je predmetom štúdia fylogenetiky. Fylogenetika sa snaží zrekonštruovať priebeh tzv. kladogenézy. Kladogenéza je poradie a spôsob vetvenia všetkých vývojových línií v priebehu evolúcie [8].

Molekulárna fylogenetika zapája do skúmania organizmov ich molekulárne znaky, predovšetkým poradie nukleotidov a aminokyselín.

Molekulárne znaky majú prakticky zásadne kvalitatívny charakter. Genetická informácia je zapísaná v DNA v digitálnej forme, takže je možné ju prekódovať do alfabetických znakov a odovzdať v tejto podobe bez akejkoľvek straty či skreslenia informácie. Kapitola v niekoľkých sekciách oboznámi čitateľa s najdôležitejšími metódami spracovania biosekvencií z pohľadu molekulárnej fylogenetiky a tvorby fylogenetických stromov.

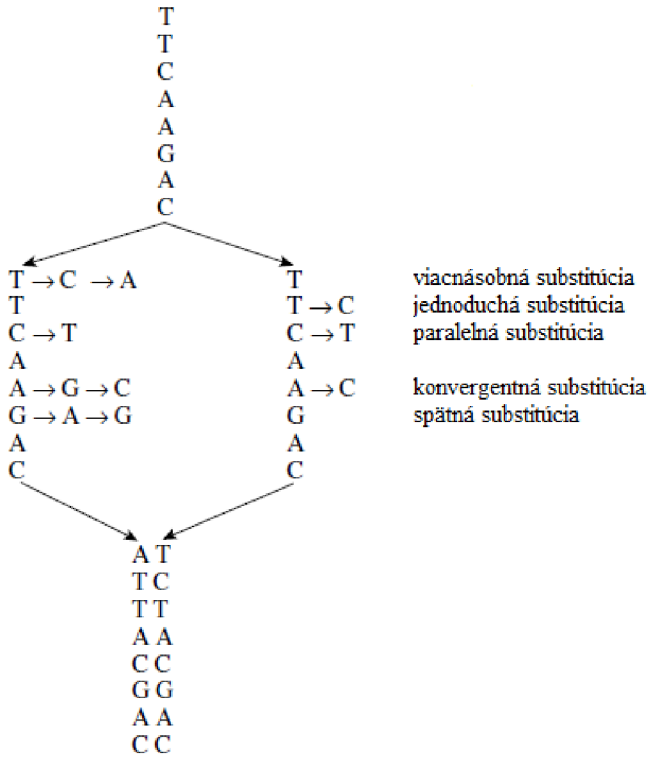
3.1 Substitučné modely

Substitučné modely reprezentujú techniku zisťovania vzdialeností medzi zarovnanými sekvenciami, t.j odhad ich evolučnej (substitučnej) vzdialenosti d. Pri výpočte berú modely do úvahy typy substitúcií, ako sú zobrazené na obrázku 3.1. V tejto sekcii bude znázornený všeobecný model spojitých markovských reťazcov, ako aj jeho implementácia v konkrétnych substitučných modeloch.

3.1.1 Markovské reťazce

Na modelovanie substitučných dejov sa používajú tzv. markovské reťazce v spojitom čase [37]. Každá pozícia v sekvencii je modelom reprezentovaným markovským reťazcom. Platia preň nasledovné podmienky:

- model je stochastický
- jednotlivé pozície v rámci biosekvencie na sebe navzájom v čase nezávisia, každá evolvuje samostatne
- čas je spojitou veličinou, $T =< 0, \infty)$
- stavy sú diskrétnou veličinou, spravidla ide o konečnú množinu
- markovská vlastnosť – pravdepodobnosť hodnoty nasledujúceho stavu závisí len na hodnote aktuálneho stavu a už nie na stavoch minulých, systém teda nedisponuje žiadnou pamäťou



Obrázok 3.1: Možné substitúcie na fragmente DNA [39]

- stacionarita, resp. homogénnosť – spôsob závislosti bezprostredne budúceho stavu na prítomnom stave sa nemení v čase [19]

Všeobecný model Nech je stav reťazca v čase t $X(t)$. Popisuje ho tzv. matica intenzít prechodu homogénneho markovského reťazca [19] $Q = q_ij$, kde q_ij je intenzita prechodu zo stavu i do stavu j, ktorú formálne popisuje rovnica3.1

$$Pr\{X(t + \triangle t) = j|X(t) = i\} = q_ij\triangle t, \text{pre každé j} \neq i \tag{3.1}$$

Interval $(t, t+\triangle t)$ je infinitezimálny. Pre diagonálne prvky q_ii musí platiť $q_ij = -\sum_{j\neq i} q_ij$, aby bol súčet prvkov v každom stĺpci rovný nule. Prvok $-q_ii$ teda udáva intenzitu prechodu zo stavu i do ľubovoľného iného stavu systému.

Z matice Q sa odvodzuje ďalej tzv. matica prechodov $P(t)$ pre ľubovoľné $t > 0$: $P(t) = p_{ij}(t)$, kde prvok $p_ij(t) = PrX(t) = j|X(0) = i$. Je riešením diferenciálnej rovnice

$$\mathrm{d}P(t)/\mathrm{d}t = P(t)Q, \tag{3.2}$$

ktorá má riešenie:

$$P(t) = e^{Q}t \tag{3.3}$$

(Pozn.: Odkazy na podrobné odvodenie uvedeného vzťahu 3.3 diskutuje Yang [39])

Markovský reťazec je ďalej definovaný vektorom počiatočných pravdepodobností $\pi^{(0)}$. Za čas t sa pravdepodobnostné rozloženie zmení z $\pi^{(0)}$ na $\pi^{(t)}$, čo vyjadruje vzťah:

$$\pi^{(t)} = \pi^{(0)} P(t) \tag{3.4}$$

Príklad: Sú dané iniciálne pravdepodobnosti výskytu jednotlivých nukleotidov v dlhej sekvencii, teda vektor $\pi^{(0)} = (\pi_t^{(0)}, \pi_c^{(0)}, \pi_a^{(0)}, \pi_g^{(0)})$. Po uplynutí času t zodpovedajú proporcie nukleotidov vektoru $\pi^{(t)}$. Pre $\pi_T^{(t)}$, teda pre frekvenciu nukleotidu T, platí: $\pi_T^{(t)} = \pi_T^{(0)} p_T T(t) + \pi_C^{(0)} p_C T(t) + \pi_A^{(0)} p_A T(t) + \pi_G^{(0)} p_G T(t)$, čo zodpovedá rovnici 3.4.

Pre $t \longrightarrow \infty$ sa systém ustáli tak, že pravdepodobnosť jednotlivých stavov zodpovedá vektoru stacionárnych pravdepodobností π. Platí:

$$\pi P(t) = \pi \tag{3.5}$$

Systém dosiahne ekvilibrium a ďalej svoje pravdepodobnostné rozloženie stavov nemení.

3.1.2 Modely nukleotidových substitúcií

Pri popise nukleotidových substitúcií pomocou markovských reťazcov platí všeobecný popis uvedený na začiatku kapitoly. Množine stavov zodpovedajú jednotlivé nukleotidy, teda $S = \{T, C, A, G\}$

p-vzdialenosť

Ide o základnú metriku na určovanie odhadu vzdialenosti dvoch nukleotidových sekvencií. Je daná vzťahom:

$$\hat{p} = n_d / n \tag{3.6}$$

kde n_d značí počet rozdielnych nukleotidov a n celkový počet porovnávaných dvojíc nukleotidov. Táto metóda je relatívne presná v prípade, že sa porovnávajú veľmi podobné sekvencie. So zvyšujúcim sa p, teda so vzrastajúcou vzdialenosťou sekvencií, klesá presnosť odhadu p, pretože sa v modeli vôbec nezohľadňujú paralelné a spätné substitúcie.

Jukes and Cantor

Metóda autorov Jukes a Cantor [13] predpokladá rovnakú pravdepodobnosť substitúcie medzi jednotlivými nukleotidmi. Jej matica intenzít prechodu jednotlivých nukleotidov má teda nasledovnú formu:

$$Q = \{q_{ij}\} = \begin{bmatrix} -3\lambda & \lambda & \lambda & \lambda \\ \lambda & -3\lambda & \lambda & \lambda \\ \lambda & \lambda & -3\lambda & \lambda \\ \lambda & \lambda & \lambda & -3\lambda \end{bmatrix} \tag{3.7}$$

Stĺpce, resp. riadky matice zodpovedajú nukleotidom T,C,A a G, presne v tomto poradí. Matica prechodov je po výpočte diferenciálnej rovnice (viď úvod tejto kapitoly):

$$P(t) = \begin{bmatrix} p_0(t) & p_1(t) & p_1(t) & p_1(t) \\ p_1(t) & p_0(t) & p_1(t) & p_1(t) \\ p_1(t) & p_1(t) & p_0(t) & p_1(t) \\ p_1(t) & p_1(t) & p_1(t) & p_0(t) \end{bmatrix}, \text{s hodnotami} \begin{cases} p_0(t) & = \frac{1}{4} + \frac{3}{4} e^{-4\lambda t}, \\ p_1(t) & = \frac{1}{4} - \frac{1}{4} e^{-4\lambda t}, \end{cases} \tag{3.8}$$

Vlastnosti matice prechodov:

- súčet prvkov v stĺpci je vždy 1, nakoľko sa markovský reťazec musí nachádzať v niektorom zo štyroch možných stavov
- $P(0) = I$, I je jednotková matica, ktorá značí, že nedochádza k žiadnej evolúcii, pravdepodobnosť nukleotidovej substitúcie je teda nulová

Vektor stacionárnych pravdepodobností je $\pi = (\pi_T, \pi_C, \pi_A, \pi_G) = (1/4, 1/4, 1/4, 1/4)$, čo logicky vyplýva z rovnakej pravdepodobnosti zmeny z nukleotidu na iný nukleotid.

Model sa musí vysporiadať s prítomnosťou viacnásobných substitúcií, a to tým, že sa zohľadnia všetky možné evolučné cesty, ako sa dostať z nukleotidu i do nukleotidu j. Tento jav popisuje Chapmanov-Kolmogorovov teorém [39]:

$$p_{ij}(t_1 + t_2) = \sum_k p_i k(t_1) p_k j(t_2), \tag{3.9}$$

ktorý hovorí, že pravdepodobnosť, že sa nukleotid i zmení v čase $t_1 + t_2$ na nukleotid j, je daný súčtom pravdepodobností všetkých medzistavov k.

Nakoľko je model Jukes a Cantor najjednoduchší, je vhodné na ňom ukázať odvodenie evolučnej vzdialenosti dvoch nukleotidových sekvencií: Z rovnice 3.7 vyplýva, že intenzita prechodu z jedného ľubovoľného nukleotidu na iný je vždy 3λ. Za predpokladu, že sa sekvencie vyvíjali oddelene počas obdobia t časových jednotiek, divergovali zo spoločného predka pred $t/2$ časovými jednotkami.Evolučná vzdialenosť týchto sekvencií je $d = 3\lambda t$. Podľa rovnice 3.8 je pravdepodobnosť toho, že sekvencie majú na príslušnej pozícii odlišnú hodnotu nukleotidu, daná vzťahom:

$$p = 3p_1(t) = \frac{3}{4} - \frac{3}{4}e^{-4\lambda t} = \frac{3}{4}e^{-4d/3} \tag{3.10}$$

Po nahradení p za p-vzdialenosť ($\hat{p}$) a logaritmovaní predchádzajúcej rovnice vzniká vzťah pre výpočet odhadu evolučnej vzdialenosti dvoch nukleotidových sekvencií pomocou modelu Jukes a Cantor:

$$\hat{d} = -\frac{3}{4} ln(1 - \frac{4}{3}\hat{p}) \tag{3.11}$$

Kimurov dvojparametrový model

Kimurov dvojparametrový model [14] berie do úvahy rozdielny výskyt tranzícií a transverzií v porovnávaných sekvenciách. Pod pojmom tranzícia rozumieme substitúciu pyrimidínu za pyrimidín alebo purínu za purín, zatiaľ čo transverzia je substitúcia z jednej do druhej skupiny. Modelu zodpovedá nasledovná matica intenzít prechodov:

$$Q = \{q_{ij}\} = \begin{bmatrix} -(\alpha + 2\beta) & \alpha & \beta & \beta \\ \alpha & (\alpha + 2\beta) & \lambda & \lambda \\ \beta & \beta & -(\alpha + 2\beta) & \alpha \\ \beta & \beta & \alpha & (\alpha + 2\beta) \end{bmatrix} \tag{3.12}$$

Rýchlosť substitúcií je daná súčtom transverzií a tranzícií, a teda $\alpha + 2\beta$. Kimura odvodil podiel tranzícií (P) a transverzií (Q) v čase t po divergovaní dvoch sekvencií s nasledovným výsledkom:

$$P = (1/4)(1 - 2e^{-4(\alpha+\beta)t} + e^{-8\beta t}) \tag{3.13}$$

$$Q = (1/2)(1 - e^{-8\beta t}) \tag{3.14}$$

Celkový počet substitúcií medzi dvoma sekvenciami je teda daný ako:

$$d \equiv 2rt = 2\alpha t + 4\beta t = -(1/2)\ln(1 - 2P - Q) - (1/4)\ln(1 - 2Q) \tag{3.15}$$

Odhad $\hat{d}$ sa získa tak, že za P a Q sa dosadia empirické hodnoty získané z porovnávaných sekvencií.

Tamurov trojparametrový model

Spomenutý Kimurov model počíta s približne rovnakým výskytom všetkých štyroch nukleotidov. V reálnych dátach tomu tak nemusí vždy byť, obzvlášť výskyt nukleotidov G+C sa môže odkláňať od očakávaných 0,5. Tamurov model [31] rozširuje Kimurov model o túto informáciu (odhad).

$$Q = \begin{bmatrix} -(1-\theta)(\alpha+2\beta) & \beta(1-\theta) & \beta\theta & \alpha\theta \\ \beta(1-\theta) & -(1-\theta)(\alpha+2\beta) & \alpha\theta & \beta\theta \\ \beta(1-\theta) & \alpha(1-\theta) & -(1-\theta)(\alpha+2\beta) & \beta\theta \\ \alpha(1-\theta) & \beta(1-\theta) & \beta\theta & -\theta(\alpha+2\beta) \end{bmatrix} \tag{3.16}$$

Odhad počtu substitúcií pomocou Tamurovho modelu je potom:

$$d = -h\ln(1 - P/h - Q) - (1/2)(1-h)\ln(1-2Q) \tag{3.17}$$

$$h = 2\theta(1-\theta) \tag{3.18}$$

Premenná h (rovnica 3.18) je odchýlka od predpokladanej hodnoty 0,5 (tzv. *G+C-content bias*) a θ je podiel nukleotidov G a C (tzv.*G+C content*) v porovnávaných sekvenciách. Hodnota θ vo vzorci 3.17 je spoločná pre obidve sekvencie. V skutočnosti sa však tieto hodnoty môžu líšiť. V tomto prípade možno vzťah rozšíriť o hodnoty θ_1 a θ_2, ktoré prislúchajú porovnávaným sekvenciám. Upravujú rovnicu 3.18 nasledovne:

$$h = \theta_1 + \theta_2 - 2\theta_1\theta_2 \tag{3.19}$$

Popis ďalších modelov možno nájsť v [16] a [39].

3.1.3 Modely aminokyselinových substitúcií

Podsekcia diskutuje metriky, ktoré sa využívajú pri odhade evolučnej vzdialenosti aminokyselinových sekvencií.

p-vzdialenosť

Podobne ako pri nukleotidových sekvenciách, ide aj v prípade aminokyselín o základnú metriku na určovanie podobnosti. Vzorec je identický s rovnicou 3.6.

PC-vzdialenosť

Vzťah p a t nie je lineárny. Rozdiel medzi n_d a skutočným počtom substitúcií začne narastať, pokiaľ dochádza k viacnásobným aminokyselinovým substitúciám. Z toho dôvodu sa pri odhade počtu substitúcií zavádza koncept Poissonovho pravdepodobnostného rozloženia.

Odvodenie poissonovskej vzdialenosti: Pre jednoduchosť sa predpokladá rovnaká substitučná rýchlosť na všetkých pozíciách aminokyseliny a označí sa r (počet substitúcií za rok). Priemerný počet substitúcií na jednu aminokyselinovú pozíciu za t rokov je potom rt. Pravdepodobnosť výskytu k aminokyselinových substitúcií ($k = 0, 1, 2, 3, \ldots$) na každej pozícií je daná Poissonovým rozložením:

$$P(k;t) = e^{-rt}(rt)^k/k! \tag{3.20}$$

Pravdepobnosť toho, že na určitej pozícii nedošlo k žiadnej substitúcii, je $P(0;t) = e^{-rt}$. Pravdepodobnosť, že v sekvencii dlhej n aminokyselín nedôjde ani k jednej zmene, je teda ne^{-rt}. V praxi však neexistuje žiadna informácia o tom, ako vyzerala sekvencia predka, z ktorej vychádza rovnica 3.20. Počet substitúcií sa teda odhaduje porovnaním sekvencií, o ktorých sa predpokladá, že divergovali pred t rokmi zo spoločného predka. Ako už bolo spomenuté, e^{-rt} je pravdepodobnosť, že na pozícii v polypeptide nedôjde k substitúcii. Pravdepodobnosť q, že k tomu nedôjde ani v druhej sekvencii na rovnakej pozícii, je:

$$q = (e^{-rt})^2 = e^{-2rt}, \tag{3.21}$$

Hodnota q sa vyjadrí pomocou opačného javu $\hat{p}$ získaného z porovnávaných sekvencií, teda $q = 1 - \hat{p}$. Stále platí vzťah pre počet substitúcií $d = 2rt$, po dosadení do rovnice 3.21 a logaritmovaní vzniká vzťah pre výpočet PC (Poisson correction)-vzdialenosti:

$$\hat{d} = -ln(1 - \hat{p}) \tag{3.22}$$

Tento model je aminokyselinovou variantou modelu Jukes a Cantor, viď 3.1.2.

Gamma rozloženie substitočných rýchlostí

Doposiaľ sa v práci predpokladalo, že na všetkých pozíciách polypeptidov je substitučná rýchlosť rovnaká. Skutočnosť je však taká, že menej dôležité oblasti podliehajú mutáciám častejšie ako funkčne dôležitejšie časti genómu [16]. Vzťah pre výpočet tzv. gamma vzdialenosti znázorňuje rovnica 3.23

$$\hat{d_G} = a[(1 - \hat{p})^{-1/a} - 1] \tag{3.23}$$

Gamma parameter a je kľúčový, ovplyvňuje pravdepodobnostné rozloženie substitučnej rýchlosti, viď tabuľka 3.1

Parameter a	**Substitučná rýchlosť** r
$a = \infty$	rovnaká pre všetky pozície
$a = 1$	exponenciálne rozloženie (veľké zmeny r medzi jednotlivými pozíciami)
$a < 1$	rozloženie je ešte ostrejšie ako exponenciálne, vyskytuje sa však často $r \approx 0$

Tabuľka 3.1: Závislosť pravdepodobnostného rozloženia substitučnej rýchlosti na gama parametri a

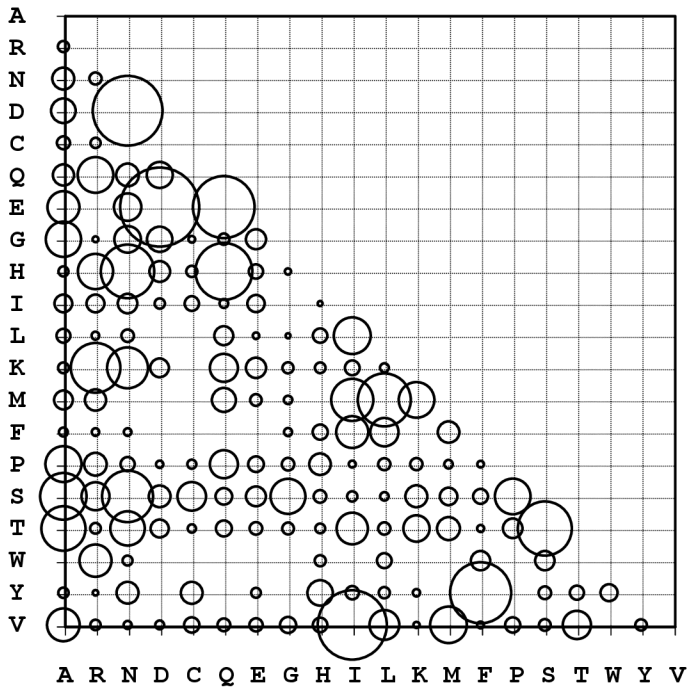
Empirické modely

Empirické modely majú priamu náväznosť na teóriu markovských reťazcov diskutovaných v úvode tejto kapitoly. Množinu možných stavov v tomto prípade tvorí dvadsať aminokyselín, teda $S = \{A, R, C, D, C, E, Q, G, H, I, L, K, M, F, P, S, T, W, Y, V\}$.

Štúdie dokazujú, že k substitúcii dochádza častejšie medzi aminokyselinami s podobnými biochemickými vlastnosťami (viď obrázok 2.1). Z tohto faktu ďalej vyplýva, že substitučné rýchlosti na jednotlivých pozíciách polypeptidu sa výrazne líšia v závislosti od prítomnosti tej ktorej aminokyseliny.

V článku Dayhoffovej [5] sa prvýkrát spomína vytvorenie tzv. aminokyselinovej substitučnej matice na základe empirických dát pozostávajúcich z proteínov rozmanitých typov (hemoglobín, cytochróm c, fibrinopeptidy). Postupovalo sa tak, že sa najprv vytvoril evolučný strom najpodobnejších aminokyselinových sekvencií a z neho sa odvodili relatívne frekvencie substitúcií jednotlivých aminokyselín. Zo získaných údajov sa potom vytvorila empirická aminokyselinová substitučná matica PAM (accepted point mutations) pre všetkých dvadsať aminokyselín. Počet substitúcií v skúmaných dátach bol v priemere jedna na sto aminokyselinových pozícií, matica sa teda označuje aj 1 PAM. V teórii markovských reťazcov to znamená maticu prechodov $P(0.01)$ (viď rovnica 3.3). Pomocou $P(0.01)$ môže byť ďalej skonštruovaná matica intenzít prechodov Q (podrobnejšie viď zdroj [39]).

Obrázok 3.2 ilustruje pravdepodobností zámen jednotlivých aminokyselín empirického modelu, podľa ktorého bola zostavená matica PAM.



Obrázok 3.2: Zámeny aminokyselín podľa Dayhoffovej, čím väčšia bublina, tým väčšia pravdepodobnosť zámeny, prebraté z [39]

3.1.4 Modely kodónových substitúcií

Vychádzajúc z teórie markovských reťazcov je množina stavov v tomto prípade definovaná ako $S = \{61 \text{ kodónov univerzálneho genetického kódu}\}$. Stop kodóny vnútri funkčných proteínov nie sú možné a preto nie sú brané do úvahy. Prvok matice intenzít prechodov q_{ij} zodpovedá intenzite prechodu z kodónu i do kodónu j. Prvky matice Q sa vytvárajú podľa

nasledovných pravidiel:

$$q_{ij} = \begin{cases} 0, & \text{keď sa } i \text{ a } j \text{ odlišujú v dvoch, resp. troch kodónových pozíciach} \\ \pi_j, & \text{keď sa } i \text{ a } j \text{ odlišujú synonymnou transverziou} \\ \kappa\pi_j & \text{keď sa } i \text{ a } j \text{ odlišujú synonymnou tranzíciou} \\ \omega\pi_j & \text{keď sa } i \text{ a } j \text{ odlišujú nesynonymnou transverziou} \\ \omega\kappa\pi_j & \text{keď sa } i \text{ a } j \text{ odlišujú nesynonymnou tranzíciou} \end{cases} \tag{3.24}$$

Vo vzťahu 3.24 je κ pomer tranzícií a transverzií, ω je pomer nesynonymných a synonymných prechodov, π_j je stacionárna pravdepodobnosť výskytu kodónu j. Na výpočet matice prechodových pravdepodobností $P(t) = p_{ij}(t)$ sa používajú numerické algoritmy diskutované v literatúre [39].

Odhad počtu synonymných a nesynonymných substitúcií Obvykle sa odhadujú dve veličiny, a to d_S ako počet synonymných substitúcií na synonymnú pozíciu a d_N ako počet nesynonymných substitúcií na nesynonymnú pozíciu. Budú predstavené heuristické počítacie metódy, pričom popis *maximum likelihood* metódy možno nájsť v literatúre [39].

Počítacie metódy - Neiova a Gojoboriho [39] Ako obvykle bude algoritmus aplikovaný na dve homológne sekvencie. Metóda pozostáva z troch fáz:

1. Zrátanie synonymných a nesynonymných pozícií: Každý kodón má tri nukleotidové pozície, ktoré možno rozdeliť do kategorií synonymných a nesynonymných. Napríklad pre kodón TTT (Phe) platí, že má deväť bezprostredných susedov (zámenou jediného nukleotidu), a to TTC (Phe), TTA (Leu), TTG (Leu), TCT (Ser), TAT (Tyr), TGT (Cys), CTT (Leu), ATT (Ile) a GTT (Val). Zo všetkých kodónov kóduje rovnakú aminokyselinu ako originálny kodón (TTT) len jeden kodón, a to TTC. Počet synonymných pozícií pre pôvodný kodón je teda $3 \times 1/9 = 1/3$, počet nesynonymných pozícií je $3 \times 8/9 = 8/3$. Táto procedúra sa vykoná pre celú sekvenciu 1 a sekvenciu 2 a sčíta sa celkový počet synonymných a nesynonymných miest. Následne sa vypočíta priemerný počet synonymných miest S a N, pričom $S + N = 3 \times L_c$, kde L_c je počet kodónov sekvencie.

2. Zrátanie synonymných a nesynonymných rozdielov sekvencií: Sekvenčne sa porovnávajú jednotlivé kodóny sekvencie 1 a sekvencie 2 a počíta sa počet synonymných a nesynonymných rozdielov. V prípade identických kodónov je problém triviálny, počet synonymných aj nesynonymných substitúcií je nulový. Podobne, keď sa líšia kodóny v jednej pozícií, ide o jednu synonymnú alebo nesynonymnú substitúciu. Keď sa však kodóny líšia v dvoch alebo dokonca všetkých troch pozíciách, je nutné preskúmať všetky evolučné cesty. Napríklad pre prechod z kodónu CCT do CAG sú možné dve rôzne cesty, a to:

   - cesta CCT(Pro)↔ CAT(His) ↔ CAG(Gln), ktorá obsahuje dve nesynonymné zmeny
   - cesta CCT(Pro)↔ CCG(Pro) ↔ CAG(Gln), ktorá obsahuje po jednej synonymnej a nesynonymnej zmene

   Naznačeným spôsobom sa porovnávajú sekvencie po celej dĺžke a vyhodnotí sa počet synonymných zmien S_d a počet nesynonymných zmien N_d.

3. Výpočet podielu zmien a zarátanie viacnásobných substitúcií do výsledku: Podiel synonymných substitúcií na synonymnú pozíciu sa vypočíta ako $p_S = S_d/S$, zatiaľ čo podiel nesynonymných substitúcií na nesynonymnú pozíciu je $p_N = N_d/N$. Finálne je nutné zohľadniť viacnásobné mutácie a upraviť vzorce pomocou modelu Jukesa a Cantora (viď sekcia 3.1.2):

$$d_S = -\frac{3}{4}\ln(1 - \frac{4}{3}p_S), \tag{3.25}$$

$$d_N = -\frac{3}{4}\ln(1 - \frac{4}{3}p_N). \tag{3.26}$$

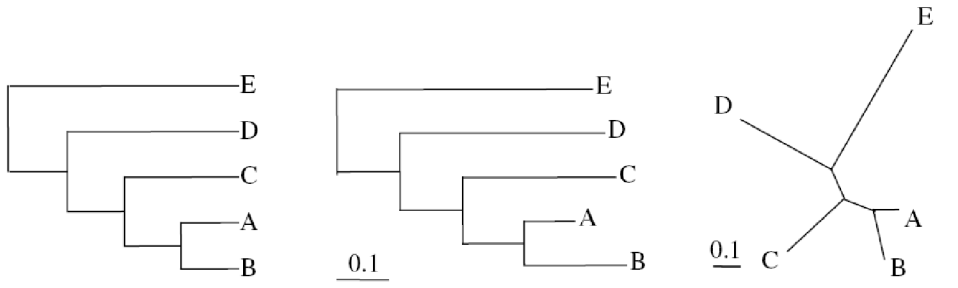
3.1.5 Posudzovanie medzier viacnásobného zarovnania pri odhade substitučnej vzdialenosti

Ako bolo naznačené v sekcii 2.4, v matici viacnásobného zarovnania sa vyskytujú medzery (*gaps*). Z hľadiska výpočtu evolučnej vzdialenosti je jedno, či na tej ktorej pozícií chýba informácia o nukleotide alebo je v nej medzera. V zásade existujú dva prístupy, ako postupovať v tejto situácii:

- *Complete-deletion* – v prípade, že obsahuje niektorý stĺpec matice viacnásobného zarovnania medzeru alebo v ňom chýba informácia v ľubovoľnom riadku, vynechá sa celý takýto stĺpec z analýzy.
- *Pairwise-deletion* – v tomto prípade sa nevyradí celý stĺpec, len pozície v pároch sekvencií, ktorých sa neprítomnosť informácie alebo medzera bezprostredne dotýka.

3.2 Fylogenetické stromy

Fylogenetické stromy patria v súčasnosti k štandardným metódam vizualizácie evolučných vzťahov organizmov. Vyjadrujú hierarchiu vývoja určitej skupiny na základe podobnosti DNA, resp. aminokyselinových sekvencií.



Obrázok 3.3: Druhy fylogenetických stromov

Matematicky je fylogenetický strom graf definovaný ako množina uzlov a množina hrán, ktoré ich spájajú. Strom je súvislý graf bez slučiek.

Z hľadiska fylogenetiky majú určité časti stromu významné vlastnosti:

- **Listové uzly**– reprezentujú existujúce organizmy
- **Vnútorné uzly**– reprezentujú spoločného predchodcu, väčšinou ide o virtuálneho predka, o ktorom neexistujú žiadne genetické informácie
- **Koreň**– pokiaľ je prítomný, ide o koreňový strom (obrázok 3.3a,b), v opačnom prípade je to nekoreňový strom (obrázok 3.3c); koreň reprezentuje evolučne najstaršieho predka, často tzv. outgroup
- **Dĺžky hrán**– vyjadrujú evolučnú vzdialenosť medzi organizmami, prípadne čas, za ktorý sa organizmus divergoval od svojho predka

Strom, ktorého topológia ignoruje evolučné dĺžky jednotlivých vetví, sa nazýva kladogram (viď obrázok 3.3a). Strom, ktorého vetvy proporcionálne zodpovedajú evolučným vzdialenostiam, sa nazýva fylogram (viď obrázok 3.3a).

3.3 Algoritmy na generovanie fylogenetických stromov

V snahe vytvoriť čo najrelevantnejší fylogenetický strom sa ponúkajú v zásade dve možnosti – vytvoriť strom postupne na základe nejakého výpočtového algoritmu, alebo vygenerovať celú množinu stromov a o jeho relevantnosti rozhodovať na základe určitých vopred stanovených kritérií optimality.

Do prvej kategórie patria tzv. vzdialenostné metódy. Ich vstupom je spravidla matica vzdialeností, ktorá vyjadruje párové vzdialenosti všetkých analyzovaných sekvencií. Prvky matica vzdialeností sú výsledkom aplikácie substitučných modelov popisovaných v sekcii 3.1. Nespornou výhodou skupiny týchto algoritmov je ich rýchlosť. Výstup poskytuje jeden konkrétny strom, topológia tohto stromu sa však môže občas líšiť v závislosti na poradí, v akom jednotlivé taxonomické jednotky vstupovali do analýzy. Konkrétnym algoritmom budú venované nasledujúce podsekcie.

Druhá kategória zahŕňa metódu maximálnej parsimónie, metódu maximálnej vieryhodnosti a metódu minimálnej evolúcie, ktoré diskutuje ďalej literatúra [39].

3.3.1 Metóda Neighbor-joining

Algoritmus bol navrhnutý Saitouom a Neiom [28] v roku 1987. Cieľom metódy je nájsť uzly a a b, ktoré sú najbližšie k sebe a zároveň najvzdialenejšie k ostatným. Vzdialenosť uzlov od seba $D(a, b)$ sa získa z matice vzdialeností, ktorej prvky sú vypočítané pomocou zvoleného substitučného modelu. Vzdialenosť uzlu a od ostatných je daná vzťahom:

$$u(a) = \frac{1}{N-2} \sum_{i \epsilon \text{MnozinaUzlov}} D(a, i) \tag{3.27}$$

Metóda hľadá takú dvojicu, pre ktorú platí vzťah:

$$\min[D(a, b) - u(a) - u(b)] \tag{3.28}$$

Beh algoritmu je daný nasledovne:

1. Vypočítaj maticu vzdialeností $D(i, j)$ a vytvor nekoreňový strom so spoločným stredom a uzlami zodpovedajúcimi vstupným sekvenciám.

2. Nájdi také dva uzly A a B pre ktoré platí vzťah 3.28 a spoj ich do spoločného uzlu U

3. Novovzniknutým hranám priraď dĺžky podľa vzťahu:

$$L(a,u) = \frac{D(a,b) + u(a) - u(b)}{2} \tag{3.29}$$

$$L(b,u) = \frac{D(a,b) + u(b) - u(a)}{2} \tag{3.30}$$

4. Vypočítaj vzdialenosť uzlu U od ostatných uzlov podľa vzťahu:

$$D(u,i) = \frac{D(a,i) + D(b,i) - D(a,b)}{2} \tag{3.31}$$

5. Vráť sa k bodu 2, až kým zostane len jediný uzol [21].

3.3.2 Metóda UPGMA

Algoritmus UPGMA[1] je jednoduchý aglomeratívny hierarchický algoritmus na rekonštrukciu fylogenetického stromu. Počíta s rovnakými rýchlosťami evolúcie vo všetkých vetvách vzniknutého stromu (konštantné molekulárne hodiny). Jeho priebeh sa dá zhrnúť v nasledovných bodoch: Na vstupe sa predpokladá matica vzdialeností D.

1. Vyber z matice D sekvencie s_A a s_B s najnižšou vzdialenosťou.

2. Dvojicu spoj do jednej sekvencie s_{AB} a prepočítaj vzdialenostnú maticu podľa vzťahu $D_{AB,C} = (D_{A,C} + D_{B,C})/2$.

3. Spojené uzly zakresli do stromu.

4. Vráť sa ku kroku 1 až kým nie sú spojené všetky dvojice.

Hlavnou nevýhodou algoritmu UPGMA je fakt, že generuje ultrametrické stromy (vzdialenosti synovských uzlov od rodičovského uzlu sú vždy po dvojiciach rovnaké). Berie totiž do úvahy konštantné molekulárne hodiny[2] pre všetky vetvy stromu. Táto hypotéza bola však v minulosti vyvrátená [20].

3.4 Štatistická relevantnosť získaných riešení

Samotné vytvorenie fylogenetického stromu neposkytuje žiadny údaj o tom, do akej miery riešenie spĺňa kritériá na presnosť a relevantnosť topológie. Z toho dôvodu sa používajú na analýzu dát, z ktorých bol strom vygenerovaný, tzv. resamplingové metódy. Medzi najpoužívanejšie metódy patrí dozaista boostraping, ktorý bude predstavený. Predchádza mu vysvetlenie tvorby konsenzuálneho stromu v nasledujúcej podsekcii. Sekcia čerpá zo znalostí získaných z [39] a [8].

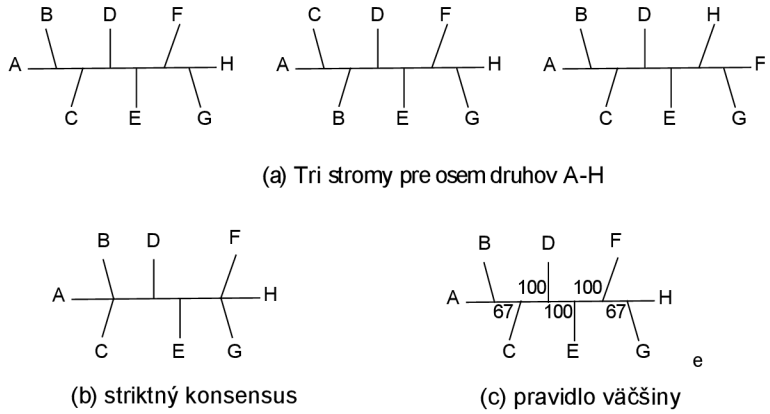
[1] z anglického *Unweighted Pair Group Method with Arithmetic Mean*

[2] táto hypotéza predpokladá, že miera mutácií v DNA alebo proteíne je za určitý čas približne konštantná

3.4.1 Konsensuálny strom

Konsensuálny strom je strom vytvorený z množiny viacerých stromov. Sumarizuje spoločné znaky (spoločnú topológiu) tejto množiny. Rozlišujú sa dva prístupy:

- striktne konsensuálny strom (*strict consensus tree*) - tento strom zobrazuje vetvy a uzly, ktoré sa nachádzajú u každého vstupného stromu, vo forme tzv. polytómií zobrazuje tie uzly, ktoré nie sú prítomné u všetkých stromov. Pre stromy z obrázka 3.4(a) je zobrazený ich *strict consensus tree* zobrazený na obrázku 3.4(b). Vetva (A,B) sa nachádza v prvom a treťom strome, ale nenachádza sa v druhom strome, avšak vetva (A,B,C) je vo všetkých troch stromoch. Z toho vyplýva, že *strict consensus tree* spracuje vetvu (A, B, C) ako trichotómiu, podobne ako aj vetvu (F, G, H). Ide o pomerne konzervatívnu sumarizáciu a v praxi sa tento typ konsensuálneho stromu príliš nepoužíva, nakoľko často vedie k hviezdicovej topológii (ako priamy dôsledok polytómie) [39].
- konsensuálny strom podľa pravidla väčšiny (*majority-rule consensus tree*) - zobrazuje uzly a vetvy, ktoré majú podporu aspoň v polovici množiny vstupných stromov. Často sa v tomto prípade píše ku každému uzlu výsledného stromu jeho výskyt v percentách (obrázok 3.4), čo sa často využíva pri metóde bootstrapping (viď ďalej).



Obrázok 3.4: Tri rôzne stromy pre osem druhov (a) a z nich vychádzajúce konsenzuálne stromy - *strict consensus* (b) a *majority-rule* (c)

3.4.2 Bootstrapping

Táto metóda pozostáva z nasledujúcich krokov. Predpokladá sa, že už bol vytvorený originálny fylogenetický strom z pôvodnej vzorky dát.

1. **Vytvorenie pseudovzoriek** – zo zarovnaných vstupných sekvencií sa náhodne vyberie stanovený počet stĺpcov (spravidla rovný dĺžke vstupných sekvencií). Výsledkom je dátový súbor, ktorý má rovnakú veľkosť ako pôvodný, no niektoré pozície sa do neho náhodne nedostali a naopak niektoré pozície sú zastúpené viackrát. Takýchto pseudovzoriek sa vytvorí väčšie množstvo (nazývajú sa aj replikácie).
2. **Vygenerovanie množiny stromov** – z každej pseudovzorky sa vytvorí strom rovnakou metódou, akou bol vytvorený originálny strom

3. **Ohodnotenie uzlov hodnotami bootstrapovej podpory** – v tejto fáze sa berú uzly pôvodného stromu jeden po druhom a zisťuje sa, v koľkých percentách stromov získaných z pseudovzoriek sa tento uzol vyskytuje. Pod pojmom rovnaký uzol sa rozumie taký uzol, z ktorého vychádza vetva obsahujúca identickú množinu druhov. Poradie vetvenia v rámci danej vetvy ani umiestnenie vetvy v rámci stromu nemá žiadny význam. Ku každému uzlu pôvodného stromu sa potom uvedie príslušné percento výskytu, tzv. bootstrappingová hodnota alebo podpora. Keď je hodnota pre nejaký uzol blízka 100, má existencia tohto uzlu vysokú podporu vo vychodzích dátach, naopak keď je nízka, hovoríme o nízkej podpore tohto uzlu.

Alternatívne možno krok 3. nahradiť iným postupom, ktorý bude aplikovaný v praktickej implementácii:

- z vygenerovanej množiny stromov vytvoriť konsensuálny strom pomocou majority-rule
- sekvenčne porovnávať topológiu originálneho stromu a konsensuálneho stromu uzol po uzle, pokiaľ dôjde k situácii, že sa nejaký uzol originálneho stromu v konsensuálnom strome vôbec nevyskytuje, ohodnotiť tento uzol nulou, inak mu priradiť hodnotu podpory uzlu konsensuálneho stromu

Kapitola 4

Počítačové programy venujúce sa problematike

Na fylogenetickú analýzu existuje niekoľko veľmi komplexných programových riešení. Prednostne budú popisované tie, ktoré implementujú dištančné algoritmy. Pri každom bude uvedený krátky popis a analýza z hľadiska dostupnosti, funkcionality (implementované algoritmy a substitučné modely), komplexnosti, rozšíriteľnosti, dostupnosti a platformy, na ktorých pracujú.

4.1 Program MEGA

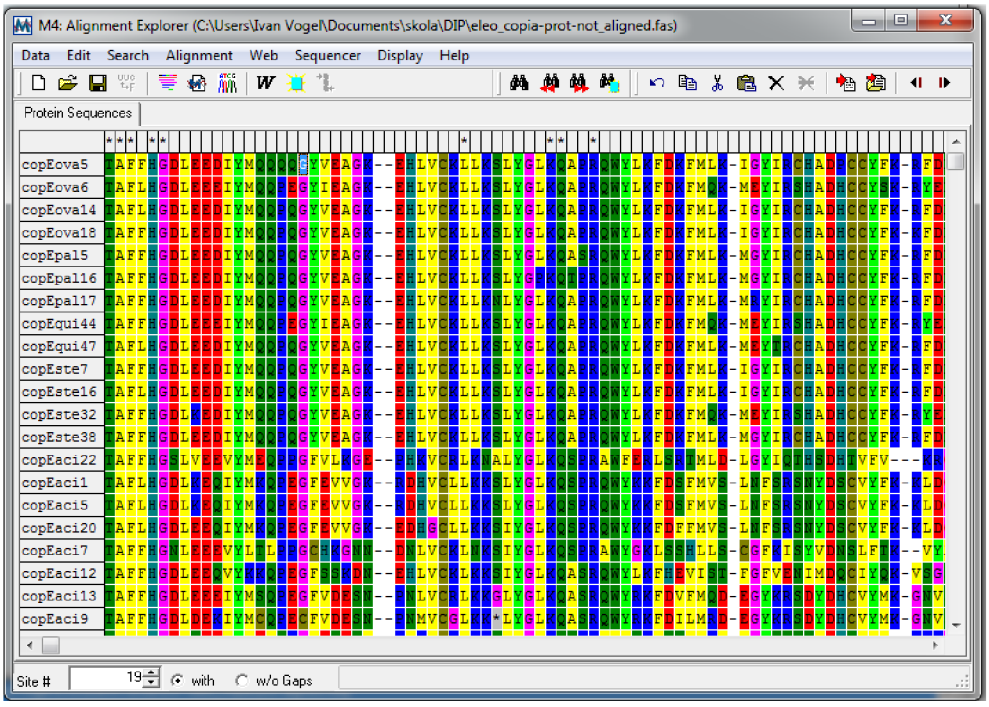
Program MEGA[32] patrí k jedným z najkomplexnejších voľne dostupných riešení. Samotná aplikácia sa dá rozdeliť na tri časti, a to:

- *Sequence Explorer* – slúži ako editor a prehliadač vstupných sekvencií. Podporuje farebné označenie niektorých význačných pozícií, ako napríklad konzervovaných úsekov. Nutné podotknúť, že sekvencie treba pred samotným prehliadaním prekonvertovať z konvenčného FASTA[1] formátu do formátu MEGA (zabezpečuje vstavaný konvertor)
- *Alignment Explorer* – umožňuje zarovnanie sekvencií pomocou vstavanej implementácie programu CLUSTALW, viď obrázok 4.1
- *Tree Explorer* – zobrazuje výsledok fylogenetickej analýzy.

Z dištančných algoritmov je k dispozícii metóda Neighbor-Joining alebo UPGMA, zo znakových metód je to Maximum Parsimony, metódy maximálnej pravdepodobnosti zastupuje algoritmus Minimum evolution. V programe sú okrem iných implementované všetky modely nukleotidových, aminokyselinových a kodónových substitúcií, ktoré sú popisované v tejto práci a s ktorými sa bude ďalej pracovať. Funkčnosť dopĺňa množstvo štatistických výpočtov nad sekvenciami a stromami vrátane bootstrappingu.

Funkčnou špecialitou programu MEGA je vkladanie sekvencií do disjunktných skupín, postup znázorňuje obrázok 4.2. Vygenerovaný fylogenetický strom je ukázaný na obrázku 4.2. Program MEGA sa teda v konečnej fáze riadi výlučne evolučnými vzdialenosťami a akékoľvek počiatočné rozdelenie do zhlukov ignoruje, resp. ho naznačuje len popiskami.

[1] popis viď http://www.ncbi.nlm.nih.gov/BLAST/fasta.shtml



Obrázok 4.1: Prehliadač zarovnaní programu MEGA

Pozn.: Táto "nedokonalosť"je jedným z hlavných podnetov na vytvorenie vlastnej modifikovanej verzie algoritmu Neighbor-joining, ktorej sa venuje nasledujúca kapitola.

Program je primárne určený pre platformu Windows, existujú však verzie pre platformy Mac a Linux. K dispozícii je popri staršej verzii 4.0 aj novšia verzia 5.0.

4.2 Phylip

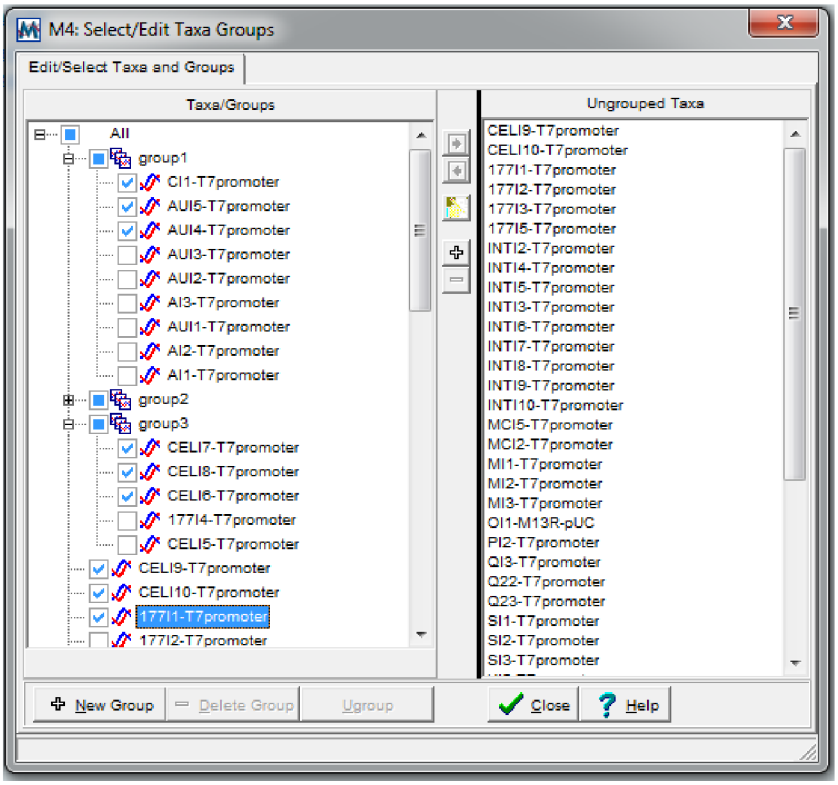
Phylip[7][2] je pravdepodobne najznámejšou knižnicou na fylogenetickú analýzu. Ide o balíček multiplatformných konzolových aplikácii napísaných v jazyku C. Pre každý krok fylogenetickej analýzy je k dispozícii niekoľko metód. Phylip neobsahuje program na zarovnanie, implicitne počíta s už zarovnanými sekvenciami, resp. prijíma na vstupe výstupný súbor programu Clustal vo formáte PHYLIP[3].Phylip okrem zarovnania implementuje podobne ako MEGA všetky metódy spomínané v tejto práci. Naviac sa tu nachádza množstvo znakových algoritmov, algoritmov na výpočet konsensuálneho stromu a dalších štatistických ukazateľov.

Programy z balíčka sa ovládajú prostredníctvom jednoduchého menu v textovom režime, ktoré umožňuje nastavenie parametrov konkrétneho algoritmu a samotné spustenie výpočtu. Vstupné dáta sa načítajú spravidla z obyčajného textového súboru (*"flat ASCII"* alebo *"Text Only"* formát). Prehľad programov z balíčka, ktoré súvisia s fylogenetickou analýzou na dištančnej báze:

- **dnadist** - program na výpočet vzdialeností sekvencií DNA pomocou substitučných modelov, výstupom je dištančná matica

[2] the PHYLogeny Inference Package

[3] špecifikácia dostupná na http://evolution.genetics.washington.edu/phylip/doc/sequence.html



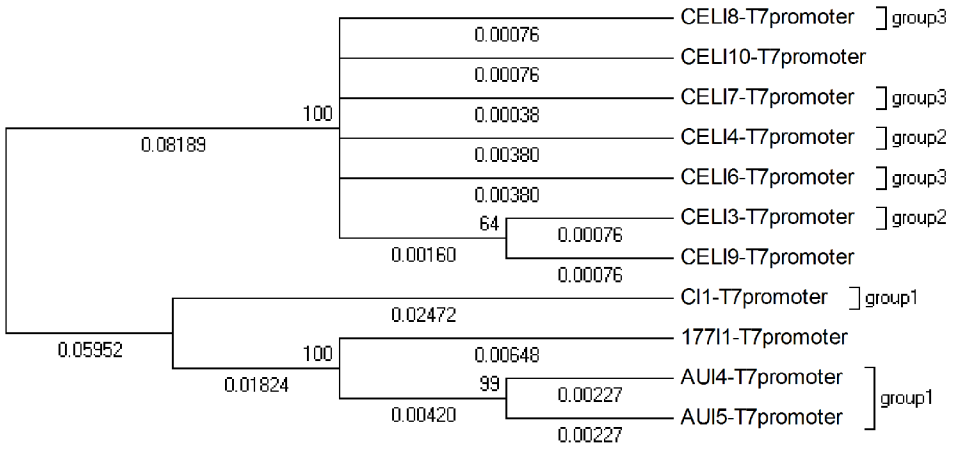
Obrázok 4.2: Princíp vkladania sekvencií do skupín

- **protdist** - obdoba pre proteínové sekvencie, využíva na odhad vzdialenosti metódu maximálnej vierohodnosti (*maximum likelihood*) založenej na rôznych substitučných maticiach (okrem iných aj Dayhoffovej PAM matici, viď 3.1.3), program je schopný korigovať vzdialenosti pre gamma rozloženie substitučných rýchlostí 3.1.3.
- **seqboot** - program na bootstrapping, načíta vstupnú množinu sekvencií a prevedie resampling, počet replikácií sa definuje na vstupe
- **neighbor** - implementácia algoritmov Neighbor-Joining (viď 3.3.1) a UPGMA (viď 3.3.2)
- **drawgram** - program vizualizuje koreňové stromy - fylogramy, kladogramy aj fenogramy
- **drawtree** - podobne ako predchádzajúci program, ale vizualizuje nekoreňové fylogenetické stromy
- **consense** - vypočíta konsensuálny strom potrebný pri bootstrappingu (viď sekcia 3.4)

4.3 Mobyle@Pasteur

Portál Mobyle@Pasteur[4] bioinformatický webový framework vyvinutý špeciálne pre účely bioinformatickej analýzy vyvinutý na Pasteurovom inštitúte v Paríži. Portál integruje

[4]dostupný na http://mobyle.pasteur.fr/cgi-bin/portal.py



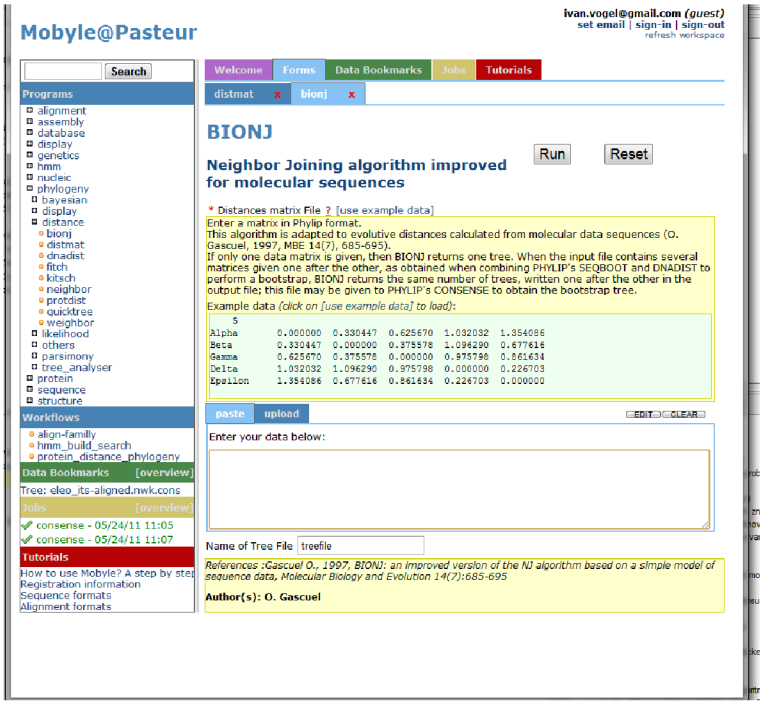
Obrázok 4.3: Fylogenetický strom s definovanými skupinami, vygenerovaný pomocou metódy Neighbor-Joining, matica vzdialeností vytvorená pomocou modelu Jukes-Cantor

množstvo fylogenetických nástrojov, pričom výpočtové jadro tvorí práve v predchádzajúcej sekcii spomínaný balíček Phylip. Jeho silnou stránkou je možnosť kombinácie vstupov a výstupov jednotlivých analýz a tým pádom možnosť vytvorenia *workflow*[5] bioinformatických nástrojov. Nakoľko sa v práci na tento portál ešte naviaže, bude mu venovaná samostatná kapitola. Ukážka užívateľského rohrania je na obrázku:

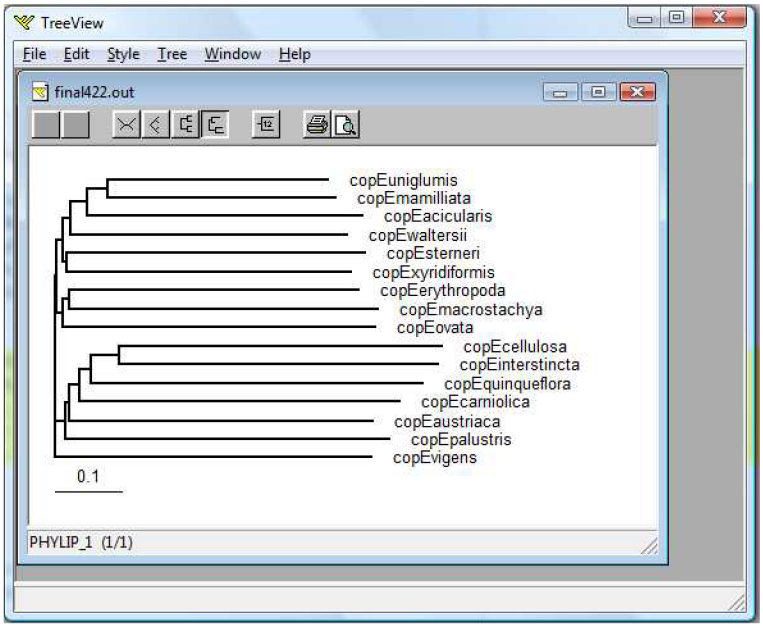
4.4 TreeView

TreeView[6] je jednoduchý, ale veľmi užitočný program na vizualizáciu fylogenetických stromov. Existuje pre platformy Windows, Linux/Unix a Mac OS. Je schopný načítať stromy vo formáte NEWICK, NEXUS,PHYLIP, CLUSTALW a iných. So stromom je možné vykonávať viacero užitočných operácií - zakorenenie (definovanie outgroup), usporiadanie uzlov, zmena na kladogram, zobrazenie radiálneho stromu atď. Obrázok 4.5 zobrazuje prostredie programu.

[5]postupnosť krokov, ktoré na seba nadväzujú, vedúca k určitému cieľu
[6]program dostupný na http://taxonomy.zoology.gla.ac.uk/rod/treeview.html



Obrázok 4.4: Užívateľské rozhranie vybranej utility v portále Mobyle



Obrázok 4.5: Ukážka programu TreeView

Kapitola 5

Špecifikácia požiadaviek

5.1 Neformálna špecifikácia

Na základe požiadaviek Mgr. Zedka je potrebné vytvoriť nové algoritmy a aplikáciu na generovanie fylogenetických stromov pre potreby Ústavu botaniky na Prírodovedeckej fakulte Masarykovej univerzity. Požadovaná je nasledovná funkcionalita:

1. načítanie sekvencií nukleotidov, resp. aminokyselín vo formáte FASTA
2. preklad nukleotidov do sekvencie aminokyselín a vice versa
3. implementácia substitučných modelov nukleotidov Jukesa a Cantora, Kimurovej 2-parametrovej metódy a Tamurovej 3-parametrovej metódy
4. implementácia substitučných modelov aminokyselín, a to PC-vzdialenosti
5. implementácia algoritmu vnútroskupinovej analýzy v kombinácii s metódou Neighbor-joining pre účely tvorby genetického stromu na báze expertných znalostí
6. implementácia bootstrappingu

S výnimkou predposledného bodu bol ku všetkým ostatným podaný teoretický základ. Návrh nového algoritmu k bodu 5 popisuje nasledujúca sekcia. Diagram prípadov užitia na obrázku 5.1 sumarizuje základnú funkcionalitu.

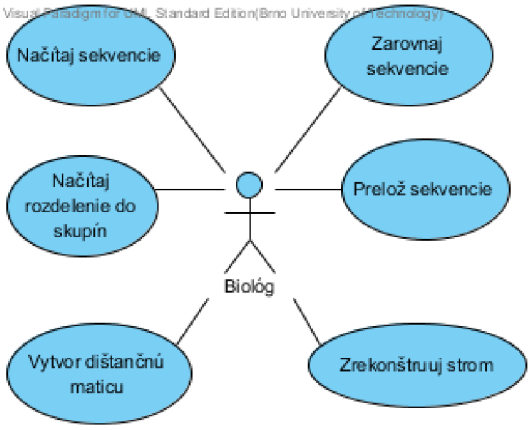
5.2 Zhlukovanie na báze expertných znalostí

Ľubovoľná dištančná metóda na tvorbu fylogenetických stromov vychádza z predpokladu, že do analýzy vstupujú sekvencie ako samostatné jednotky. Požadované je rozšírenie tejto funkcionality. Do analýzy budú vstupovať expertom preddefinované zhluky. Ide teda o nepovinné apriori vloženie sekvencií do disjunktných skupín, pričom ďalej bude táto skupina vo fylogenetickom strome reprezentovaná ako jednoduchý uzol.

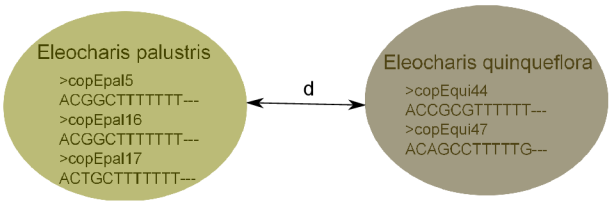
Tento postup naznačuje už program MEGA spomínaný v sekcii 4.1. Ten však pri výstupe nezobrazuje preddefinované skupiny ako samostatné zhluky, ale len znázorňuje vo výslednom strome, ktoré sekvencie boli zadelené do ktorej skupiny, viď obrázok 4.3. Tento prístup sa javí pre praktické účely Mgr. Zedka ako nedostatočný.

Na obrázku 5.3 sú pre ilustráciu zobrazené sekvencie druhov z rodu Eleocharis a ich zadelenie do skupín na základe znalostí experta.

Z praktického hľadiska má zhlukovanie na báze expertných znalostí nasledovný význam:

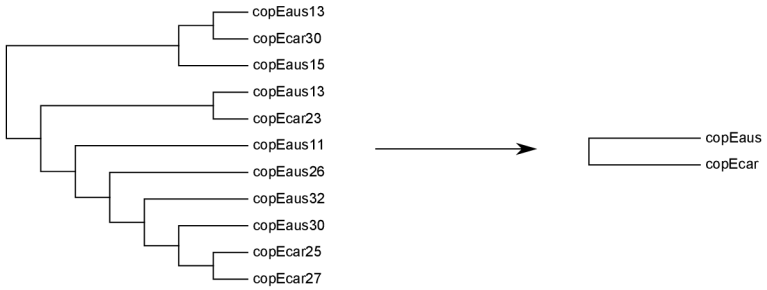


Obrázok 5.1: Diagram prípadov užitia



Obrázok 5.2: Definícia hlavného problému - vhodná dátová štruktúra pre reprezentáciu a relevantné určenie vzdialenosti preddefinovaných zhlukov

- Pokiaľ je priemerná vzdialenosť medzi členmi preddefinovaného zhluku (priemerná vnútroskupinová vzdialenosť) malá v porovnaní so vzdialenosťou od sekvencií mimo tejto skupiny (priemerná mimoskupinová vzdialenosť), je v rámci zjednodušenia vhodné reprezentovať túto skupinu vo výslednom strome ako jeden uzol (so zachovaním informácie o vnútroskupinovej variabilite). Zjednoduší sa tak pohľad na fylogenetické vzťahy medzi rodmi organizmov.
- Pokiaľ prvky skúmanej preddefinovanej skupiny (rodu) prirodzene netvoria jeden podstrom, zaujíma nás, v snahe dozvedieť sa o rode nové evolučné poznatky, pozícia takéhoto jeho uzlu vo výslednom strome
- Pokiaľ je z fylogenetického stromu zrejmé, že prvky z dvoch rôznych rodov majú k sebe blízko, t.j., že sa často vyskytujú spolu v jednej vetve ale na rôznych miestach v rámci stromu, je znova možné tento vzťah zjednodušiť (situáciu ilustruje obrázok 5.3).



Obrázok 5.3: Ukážka očakávaného zjednodušenia pomocou požadovaného algoritmu, vľavo podstrom fylogenetického stromu druhov z rodu Eleocharis, vpravo agregovaný strom

Kapitola 6

Analýza požiadaviek a návrh algoritmov

Jadrom problému, ktorý je potrebné vyriešiť, je spôsob, akým sa vysporiadať s prítomnosťou viacerých sekvencií pre jeden uzol fylogenetického stromu. Možné sú nasledovné alternatívy:

1. Náhodne vybrať jednu reprezentatívnu sekvenciu z každej skupiny.
2. Vytvoriť konsensuálnu sekvenciu a tou potom reprezentovať skupinu.
3. Vytvoriť metódu odhadu vzdialenosti, pri ktorej by sa zohľadnila variabilita sekvencií v rámci skupiny.

Pri variantoch 1 a 2 dochádza k strate informácie. Pri variante 1 sa vôbec nezohľadňuje variabilita v rámci skupiny. Pri variante 2 sa síce čiastočne variabilita zohľadňuje, ku strate informácie však dochádza v dôsledku väčšinového princípu výberu nukleotidov / aminokyselín na jednotlivých pozíciách. Príklady z praxe sú uvedené v kapitole 8.

6.1 Dátová reprezentácia preddefinovaného zhluku

Pre účely analýzy je nevyhnutné vhodne zvoliť, ako reprezentovať preddefinovaný zhluk na počítači. Bolo navrhnuté vytvorenie reprezentatívnej sekvencie metódou tzv. priemerovania sekvencií (frekvenčnej analýzy). Predpokladá sa zhluk o troch krátkych úsekoch DNA, ako zobrazuje obrázok 6.1 vľavo. Na základe analýzy jednotlivých znakov sa vypočíta pravdepodobnosť výskytu znaku na danej pozícií a tieto hodnoty sa vložia do tabuľky. Tabuľka potom reprezentuje daný zhluk a pri odhade pomocou substitučných modelov sa zohľadňujú relatívne výskyty znakov na tej ktorej pozícii.

6.2 Základný algoritmus

Vychádza sa z nasledovnej všeobecnej schémy dištančného algoritmu:
Vstup: reťazce DNA, resp. aminokyselín
Výstup: fylogenetický strom

1. Zarovnaj sekvencie pomocou algoritmu MUSCLE
2. Na základe vybraného substitučného modelu odhadni vzájomnú vzdialenosť a vytvor maticu vzdialeností

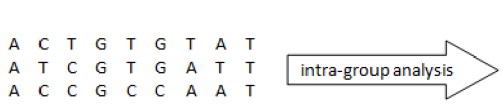
3. Pomocou algoritmu NJ s maticou vzdialeností na vstupe vytvor fylogenetický strom.

4. Vypočítaj podpory jednotlivých uzlov pomocou metódy bootstrapping.

Upravuje sa vstup a 2. bod všeobecnej schémy nasledovne:
Vstup: ako vo všeobecnom algoritme + údaje o príslušnosti sekvencií k jednotlivým zhlukom

1. Aplikuj priemerovanie/Vytvor sekvenčný profil preddefinovaného zhluku a vytvor tabuľku reprezentujúcu zhluk.

2. Vytvor maticu podobností pre všetky zhluky a pokračuj bodom 3 všeobecnej schémy dištančného algoritmu.



N/P	1	2	3	4	5	6	7	8	9
A	1	0	0	0	0	0	0,67	0,67	0
T	0	0,33	0,33	0	0,67	0	0,33	0,33	1
C	0	0,67	0,67	0	0,33	0,33	0	0	0
G	0	0	0	1	0	0,67	0	0	0

Obrázok 6.1: Vnútroskupinová analýza preddefinovaného zhluku

6.3 Stavový diagram

Obrázok 6.2 znázorňuje možné stavy dát v aplikácii. Ako vidieť, po načítaní sekvencií sú k dispozícii dva možné prechody do ďalšieho stavu v závislosti od toho, či sa pracuje so zarovnanými sekvenciami alebo nie.

6.4 Vlastné rozšírenie substitučných modelov

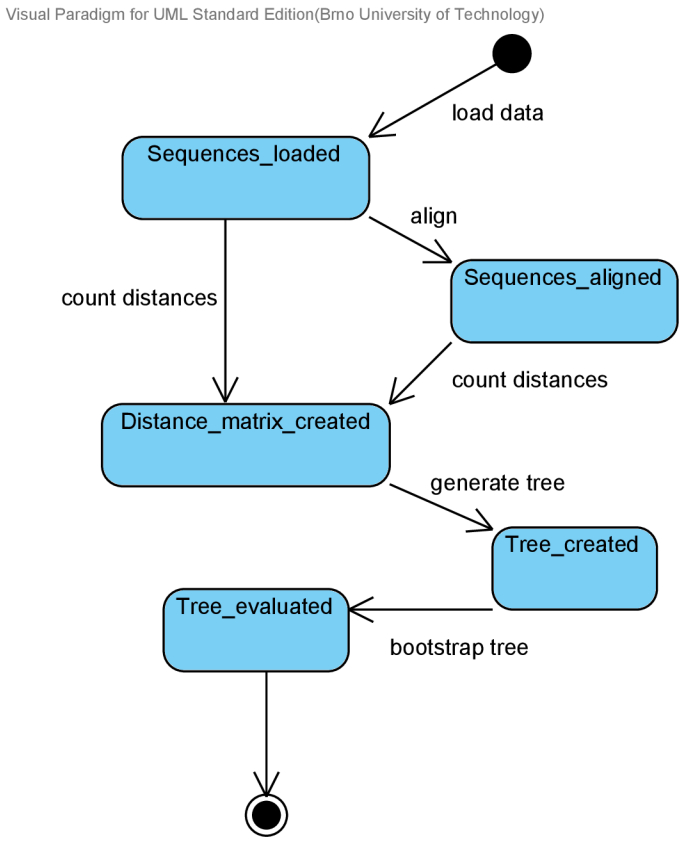
Je nutné navrhnúť vhodnú metriku, ako definovať podobnosť medzi dvoma zhlukmi, prípadne medzi zhlukom a samostatnou sekvenciou (čo je de facto tiež zhluk obsahujúci len jeden prvok). Naväzujúc na predchádzajúcu sekciu bola vytvorená originálna metodika určovania vzdialenosti preddefinovaných zhlukov.Opiera sa o teoretický výklad z kapitoly 3.

6.4.1 Rozšírenie modelu Jukes-Cantor

Predpokladajme dva preddefinované zhluky, označme ich $\mathbb{A}$ a $\mathbb{B}$. Poznáme ich pozične špecifické vektory na pozícii i (rovnica 6.1)

$$v_{\mathbb{A}}[i] = \begin{pmatrix} p_{\mathbb{A}}^{T} \\ p_{\mathbb{A}}^{C} \\ p_{\mathbb{A}}^{A} \\ p_{\mathbb{A}}^{G} \end{pmatrix}, v_{\mathbb{B}}[i] = \begin{pmatrix} p_{\mathbb{B}}^{T} \\ p_{\mathbb{B}}^{C} \\ p_{\mathbb{B}}^{A} \\ p_{\mathbb{B}}^{G} \end{pmatrix} \tag{6.1}$$

Pravdepodobnosť substitúcie nukleotidu T na T (čo znamená, že obidva preddefinované zhluky majú na tejto pozícii rovnaký nukleotid a teda k substitúcii vskutku nedochádza)



Obrázok 6.2: Stavový diagram v jazyku UML

dostaneme vynásobením $p_{\mathbb{A}}^T$ a $p_{\mathbb{B}}^T$, rovnako pre tri zostávajúce nukleotidy. Aby sme dostali pravdepodobnosť N, že na pozícií i medzi dvoma zhlukmi nedochádza k substitúcii, aplikujeme na vektory $p_{\mathbb{A}}^T$ a $p_{\mathbb{B}}^T$ skalárny súčin. Pravdepodobnosť substitúcie P_s na tejto pozíci je teda $P_s = 1 - N$. Súčet týchto hodnôt na každej pozícii nám dáva p vzdialenosť ($\hat{p}$)(viď 3.1.2) dvoch zhlukov. Dosadením tohto odvodenia do vzorca 3.1.2 dostávame **odhad počtu substitúcií medzi dvoma preddefinovanými zhlukmi pomocou modifikovaného modelu Jukes-Cantor**:

$$\hat{d} = -\frac{3}{4} ln \left(1 - \frac{4}{3} \frac{\sum_{i=1}^{n} (1 - v_{\mathbb{A}}[i] \cdot v_{\mathbb{B}}[i])}{n}\right) \tag{6.2}$$

6.4.2 Rošírenie Kimurovho dvojparametrového modelu

V prípade Kimurovho modelu je nutné rozlíšiť pri počítaní substitúcií tranzície a transverzie medzi dvoma preddefinovanými zhlukmi. Predpokladáme pozične špecifické vektory ako v rovnici 6.1. Potom:

- pre pravdepodobnosť tranzícií na pozícii i platí:

$$P(i) = p_{\mathbb{A}}^A . p_{\mathbb{B}}^G + p_{\mathbb{A}}^G . p_{\mathbb{B}}^A + p_{\mathbb{A}}^T . p_{\mathbb{B}}^C + p_{\mathbb{A}}^C . p_{\mathbb{B}}^T \tag{6.3}$$

- pre pravdepodobnosť non-substitúcií platí $N(i) = v_{\mathbb{A}} \cdot v_{\mathbb{B}}$

- súčet pravdepodobností všetkých možných prechodov nukleotidu z vektoru $v_{\mathbb{A}}[i]$ na niektorý nukleotid z vektoru $v_{\mathbb{B}}[i]$ musí byť rovný 1. Platí teda, že súčet transverzií, transzícií a non-substitúcií musí byť 1. Z toho vyplýva odvodenie pre pravdepodobnosť transverzií

$$Q = 1 - P - N \tag{6.4}$$

Dosadením vzťahu 6.4 do rovnice 3.15 a postupnou úpravou dostaneme vzťah uvedený v rovnici 6.5.

$$\hat{d} = -\frac{\ln(N-P)}{2} - \frac{\ln(2P+2N-1)}{4} \tag{6.5}$$

$$N = \frac{\sum_{i=1}^{n} v_{\mathbb{A}}[i] \cdot v_{\mathbb{B}}[i])}{n} \tag{6.6}$$

$$P = \frac{\sum_{i=1}^{n} p_{\mathbb{A}}^{A}[i].p_{\mathbb{B}}^{G}[i] + p_{\mathbb{A}}^{G}[i].p_{\mathbb{B}}^{A}[i] + p_{\mathbb{A}}^{T}[i].p_{\mathbb{B}}^{C}[i] + p_{\mathbb{A}}^{C}[i].p_{\mathbb{B}}^{T}[i]}{n} \tag{6.7}$$

6.4.3 Rozšírenie Tamurovho trojparametrového modelu

Tamurov trojparametrový model je rozšírením Kimurovho modelu s predpokladom nerovnakého podielu jednotlivých nukleotidov v sekvencii (viď podsekcia 3.1.2). Upravuje sa výpočet hodnôt premenných $\theta 1, \theta 2$ zo vzorca 3.19 v súlade s predchádzajúcimi definíciami nasledovne:

$$\theta 1 = p_{\mathbb{A}}^{G} + p_{\mathbb{A}}^{C} \tag{6.8}$$

$$\theta 2 = p_{\mathbb{B}}^{G} + p_{\mathbb{B}}^{C} \tag{6.9}$$

Modifikovaný vzorec pre výpočet Tamurovej vzdialenosti je potom:

$$\hat{d} = -h\ln(N-P) - \frac{1-h}{2}\ln(2P+2N-1) \tag{6.10}$$

$$h = \frac{n\sum_{i=1}^{n} p_{\mathbb{A}}^{G}[i] + p_{\mathbb{A}}^{C}[i] + p_{\mathbb{B}}^{G}[i] + p_{\mathbb{B}}^{C}[i] - 2\sum_{i=1}^{n} p_{\mathbb{A}}^{G}[i] + p_{\mathbb{A}}^{C}[i].\sum_{i=1}^{n} p_{\mathbb{B}}^{G}[i] + p_{\mathbb{B}}^{C}[i]}{n^2} \tag{6.11}$$

6.4.4 Rozšírenie PC-vzdialenosti

Ako už bolo uvedené, PC vzdialenosť je aminokyselinovou variantou modelu Jukes Cantor. Analogicky sa teda odvodí aj modifikovaná metrika na určovanie vzdialenosti preddefinovaných zhlukov aminokyselinových sekvencií. Podobne ako v prípade modelu Jukes Cantor spočíva modifikácia v rozšírení p vzdialenosti na preddefinované zhluky. Výsledný vzťah udáva rovnica

$$\hat{d} = -ln(1 - \frac{\sum_{i=1}^{n}(1 - v_{\mathbb{A}}[i] \cdot v_{\mathbb{B}}[i])}{n}) \tag{6.12}$$

6.5 Divizívne algoritmy

Divizívne algoritmy sa vo všeobecnosti vo fylogenetickej analýze nepoužívajú priliš často. Nakoľko je však vytváranie stromov preddefinovaných zhlukov pomerne špecifický (a novátorský) prípad fylogenetickej analýzy, môžu tieto algoritmy poskytnúť výhody čo do

výkonnosti aj presnosti. V súvislosti s touto tematikou bolo naštudovaných niekoľko článkov. Článok popisujúci metódu PTDC (Phylogeny by Top Down Clustering) [2] poslúžil ako inšpirácia pre návrh vlastného riešenia (viď nasledujúca podsekcia).

PTDC je rekurzívny algoritmus využívajúci prístup zhora nadol. Metóda v každom kroku rozdelí vstupnú množinu sekvencií na také dva zhluky, ktoré sú od seba najvzdialenejšie. Vzdialenostná funkcia bola prebratá z metódy UPGMA 3.3.2 a popisuje ju rovnica 6.13.

$$d_{i,j} = \frac{1}{|C_1||C_2|} \sum_{p \in C_1, q \in C_2} d_{p,q}, \tag{6.13}$$

kde $|C_1|$ a $|C_2|$ sú počty sekvencií v zhlukoch C_1 a C_2 v tomto poradí a $d_{p,q}$ značí vzdialenosť medzi sekvenciami p a q. Algoritmus pracuje zhora nadol tak, že hľadá vždy dva najvzdialenejšie zhluky v rámci celej množiny zhlukov. Potom rekurzívne volá seba samého na obsahy práve rozdelených zhlukov. Pokiaľ zhluk obsahuje len jednu sekvenciu, vytvorí sa len jeden uzol, pokiaľ obsahuje práve dve sekvencie, vytvoria sa dva synovské uzly, inak sa rekurzívne pokračuje ďalej. Autori tohto algoritmu použili heuristiku, aby sa pri rozhodovaní o optimálnom rozdelení zhluku nemusela prehliadať celá množina možných rozdelení. Spočíva v tom, že sa vyberajú také rozdelenia na C_1 a C_2, kde všetky sekvencie v C_1 obsahujú nejaký reťazec s v určitom regióne a práve všetky sekvencie v C_2 reťazec s v tom istom regióne neobsahujú.

6.5.1 Algoritmus TDCG

TDCG (Top Down Clustering of Groups) je originálny zhlukovací algoritmus pracujúci na princípe zhora nadol. Bol čiastočne inšpirovaný algoritmom PTDC, ktorého princíp je popísaný v úvode tejto sekcie. Je zamýšľaný na prácu s preddefinovanými zhlukmi, ale teoreticky môže byť použitý aj na jednoduché sekvencie.

Jadro algoritmu tvorí rekurzívne volanie procedúry Split (viď pseudokód nižšie). Cieľom je to, aby v každom kroku bola minimalizovaná *intra-group* vzdialenosť medzi skupinami, ktoré boli zaradené do rovnakého zhluku a zároveň maximalizovať *inter-group* vzdialenosť medzi skupinami z dvoch rôznych zhlukov.

V uvedenom pseudokóde figurujú dve pomocné funkcie:

- Funkcia $CountGroupDistances()$ spočíta vzdialenosti medzi jednotlivými preddefinovanými skupinami pomocou niektorej z modifikovaných metrík uvedených v kapitole 6.4 a vytvorí trojuholníkovú dištančnú maticu.
- Funkcia $FindGreatestDistance()$ potom prechádza vytvorenú vzdialenostnú maticu a hľadá dva maximálne vzdialené preddefinované zhluky. Tie budú následne centrá nových zhlukov.

6.6 Algoritmus na porovnanie konsensuálneho stromu s originálnym stromom

6.6.1 Reprezentácia fylogenetického stromu pomocou dvojrozmerného poľa

Postup pri uložení evolučného stromu do poľa je nasledovný:

Procedure 1 Split(M,tree)

Input: matica viacnásobného zarovnania M o rozmeroch $k \times n$ a ich rozdelenie do jednotlivých skupín (aj jedna sekvencia tvorí elementárnu skupinu)
Output: fylogenetický strom skupín definovaných na vstupe

```
if M obsahuje len jednu skupinu s názvom name then
    vytvor koreň r,označ ho name
    return
end if
if M obsahuje dve skupiny then
    vytvor strom s dvoma synmi a spoj ho s nadradeným uzlom (aktuálny koreň generovaného podstromu)
    return
end if
distances = CountGroupDistances(M)
(center1,center2) = FindGreatestDistance(distances)
cluster1.add(center1),cluster2.add(center2)
for all group in M do
    if distance[group,center1] < distance[group,center2] then
        cluster1.add(group)
    else
        cluster2.add(group)
    end if
end for
tree.addNode(node_for_cluster1)
tree.addNode(node_for_cluster2)
Split(cluster1, node_for_cluster1)
Split(cluster2, node_for_cluster2)
return
```

1. Zoradiť lexikograficky jednotlivé názvy terminálnych uzlov stromu, teda názvy taxonomických jednotiek, stĺpce vytvorenej matice budú potom označené takto zoradenými názvami.

2. Prechádzať jednotlivé uzly podľa poradia, v akom boli zaradené do zhluku (pri) a označiť v riadku:

   - v prípade dvoch terminálnych uzlov označiť obidva stĺpce prislúchajúce týmto terminálnym uzlom hodnotou 1
   - v prípade, že sa terminálny uzol pripája k väčšiemu zhluku, označiť v matici len tento uzol hodnotou 1

Hodnoty uzlov sa delegujú zhora nadol, v poslednom riadku by sa teda podľa správnosti mali vyskytovať len jednotky. Podľa matice sa dá následne spätne zrekonštruovať topológia fylogenetického stromu, predpokladá sa, že dĺžky vetví a prípadné bootstrapové hodnoty sa uložia v asociatívnom poli. Obrázok 6.3 zobrazuje fylogenetický strom z obrázka 3.3b v podobe dvojdimenzionálneho poľa.

	A	B	C	D	E
Internal_node1	1	1	0	0	0
Internal_node2	1	1	1	0	0
Internal_node3	1	1	1	1	0
Internal_node4	1	1	1	1	1

Obrázok 6.3: Fylogenetický strom z obrázka 3.3b v podobe matice

6.6.2 Porovnanie topológií

Predpokladá sa, že na vstupe je originálny a konsensuálny strom s bootstrapovými hodnotami (uloženými napríklad v asociatívnom poli), obidva vo forme dvojdimenzionálneho poľa. Cieľom je porovnať tieto dve topológie a priradiť hodnoty podpory neterminálnym uzlom originálneho stromu. Postup je priamočiary:

1. Načítaj riadok matice originálneho stromu. Hľadaj tento riadok v matici konsensuálneho stromu. Môžu nastať tieto prípady:

   - Riadok sa v konsensuálnom strome nenachádza ⇒ označ tento riadok hodnotou 0
   - Riadok sa v konsensuálnom strome nachádza a prislúcha mu určitá hodnota podpory ⇒ skopírovať túto hodnotu k originálnemu stromu

Na výstupe je originálny strom s príslušnými hodnotami podpory pre neterminálne uzly.

Kapitola 7

Návrh a implementácia

7.1 Analýza vstupných dát

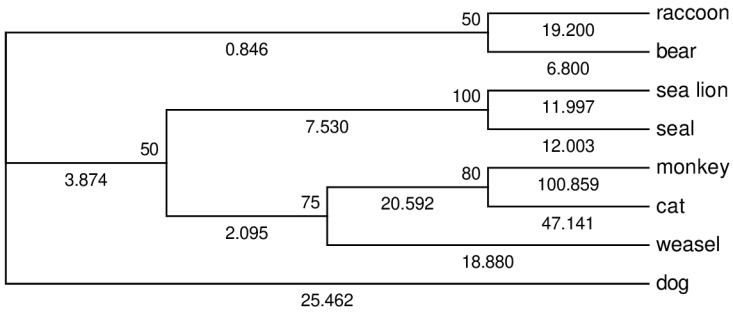
7.1.1 Vstupné dáta

Vstupný dátový súbor je textový súbor obsahujúci dve a viac nukleotidových alebo aminokyselinových sekvencií vo formáte FASTA. NCBI [1] definuje tento formát nasledovne:

- Prvý riadok každého záznamu sekvencie tvorí hlavička, začína znakom >, a ďalej obsahuje jednoznačný identifikátor sekvencie a ďalšie voliteľné informácie
- Ďalšie riadky obsahujú samotnú sekvenciu zostavenú z jednopísmenných kódov podľa štandardu UIPAC[2]-IUB[3].

7.1.2 Výstupné dáta

Za základný výstup sa považuje fylogenetický strom vo formáte NEWICK. Je to textový formát popisujúci topológiu stromu, dĺžky vetví aj bootstrappové hodnoty pomocou alfanumerických znakov, zátvoriek a čiarok. Nasledovný strom:



Obrázok 7.1: Ilustratívny fylogenetický strom

sa vo formáte NEWICK zapíše nasledovne:

[1] z angl. The National Center for Biotechnology Information
[2] z angl. International Union of Pure an Applied Chemistry
[3] z angl. International Union of Biochemistry and Molecular Biology

```
((raccoon:19.19959,bear:6.80041):0.84600[50],((sea_lion:11.99700,
seal:12.00300):7.52973[100],((monkey:100.85930,cat:47.14069):
20.59201[80], weasel:18.87953):2.09460[75]):3.87382[50],dog:25.46154);
```

7.2 Logický návrh

Prvotný logický návrh je zobrazený na obrázku 7.2. Rozdeľuje funkcionalitu fylogenetickej knižnice do jednotlivých modulov. Už v tejto fáze sa kladie dôraz na perspektívnu rozšíriteľnosť jednotlivých častí, snaha bola teda, aby boli relácie medzi jednotlivými dvojicami modulov asymetrické (ak modul A referencuje modul B, tak by to opačne už nemalo platiť) a teda aby sa minimalizoval počet závislostí. V návrhu sa ďalej predpokladá, že z hľadiska implementácie bude dôraz kladený na funkcionalitu, ktorá doposiaľ na trhu neexistuje (preddefinované zhluky) a naopak algoritmy, pre ktoré už existujú knižnice, budú využité v maximálnej možnej miere ako podpora nových riešení.

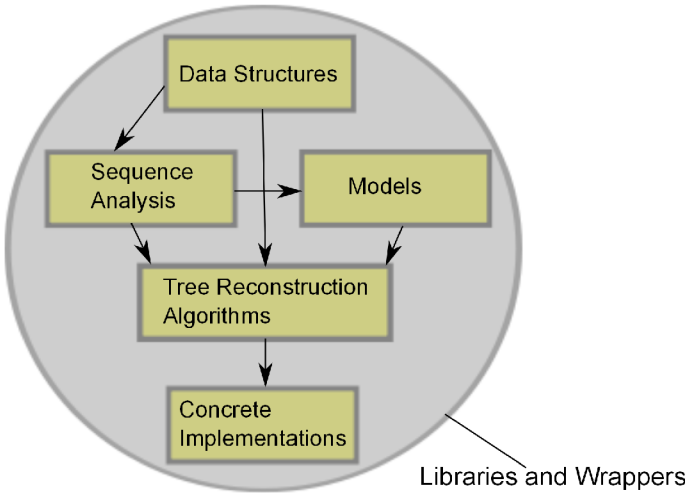
Popis jednotlivých častí logického návrhu:

- Data Structures - modul obsahuje podporné dátové štruktúry pre fylogenetickú analýzu, akými sú dištančná matica, fylogenetický strom alebo kolekcie vstupných sekvencií.
- Sequence Analysis - obsahuje frekvenčnú analýzu preddefinovaných skupín.
- Models - modul implementuje všetky substitučné modely
- Tree Reconstruction Algorithms - zabezpečuje komunikáciu medzi jednotlivými modulmi, predstavuje šablóny pre kroky fylogenetickej analýzy.
- Concrete Implementations - modul združuje konkrétne operácie nad vstupnými dátami a zastrešuje celú fylogenetickú analýzu.
- Libraries and Wrappers - schematicky znázorňuje fakt, že jednotlivé moduly využívajú podľa potreby už hotové dátové štruktúry a algoritmy, ktoré sú k dispozícii prostredníctvom voľne dostupných knižníc.

7.3 Objektový návrh jadra fylogenetickej knižnice v jazyku UML

Na základe špecifikácie bol vytvorený objektový návrh fylogenetickej knižnice (viď obrázok 7.3). Dôraz sa kládol na potenciálnu rozšíriteľnosť aplikácie s minimálnou nutnosťou zasahovať do už vytvoreného kódu. Z tohto dôvodu bola použitá metodika návrhových vzorov, ktorá je rozobratá v nasledujúcich podsekciách.

Na vrchole celej hierarchie tried je rozhranie Object, ktoré musia implementovať všetky triedy, od ktorých sa očakáva textový výstup ich obsahu. Táto vlastnosť bude užitočná, nakoľko je v modeli mnoho dátových štruktúr, ktorých obsah je priamo vyžadovaný (Sequence, DistanceMatrix, TreeStruct atď).



Obrázok 7.2: Navrhovaná logická schéma fylogenetickej knižnice. Smer šípky znázorňuje využitie inštancií tried toho ktorého modulu iným modulom.

7.3.1 Substitučné modely

Pri návrhu modulu substitučných modelov bol využitý *strategy pattern*. Trieda DistanceMatrix referencuje abstraktnú triedu Model, ktorá predpisuje implementovať metódu count(), ktorej parametrom sú dve generické biologické sekvencie. Vďaka tejto vlastnosti bude možné meniť modely za behu programu, kód sa sprehľadní a bude možné voľne experimentovať pri hľadaní najvhodnejšieho substitučného modelu pre danú vzorku dát.

7.3.2 Sekvencie

De facto všetky operácie sa uskutočňujú nad biologickými sekvenciami. Aby bola zabezpečená ľahšia rozšíriteľnosť a voľná kombinácia jednotlivých metód štatistickej analýzy sekvencií, bol pre návrh tejto časti zvolený *decorator pattern*. Pri rozšírení o novú štatistickú analýzu, prípadne o dátovú štruktúru, ktorá priamo vychádza z biologickej sekvencie, stačí rozšíriť knižnicu o novú triedu, ktorá dedí z triedy DecoratedSequence.

7.3.3 Dištančné algoritmy

Dištančné algoritmy majú vždy rovnakú schému – takú, ako znázorňujú metódy abstraktnej triedy DistanceAlgorithm. V tomto prípade sa dá s výhodou použiť *template pattern*. Triedy zdedené od DistanceAlgorithm buď dedia generickú implementáciu svojho predka, alebo dopĺňajú algoritmus o detaily.

7.3.4 Ďalšie dátové štruktúry a iné významné triedy

Trieda TreeStruct implementuje topológiu fylogenetického stromu vo forme 2D dátovej štruktúry. Bude sa s ňou pracovať počas celej fylogenetickej analýzy od chvíle, keď je strom vytvorený v pamäti počítača.

Trieda Parameters je implementovaná ako *singleton* a združuje všetky parametre a nastavenia programu. Singleton sa pre tento účel často používa, parametre sú totiž fixne dané na začiatku analýzy a vyžaduje sa teda ich persistencia v rámci celého behu programu.

Trieda ProgramWrapper implementuje jednoduchý wrapper externých binárnych súborov. Pre každý binárny súbor sa potom využije nová inštancia tejto triedy, ktorej sa nadefinujú parametre programu a cesta k nemu.

7.4 Použité technológie

Na implementáciu bol použitý vysokoúrovňový objektový jazyk Python. Treba podotknúť, že jazyk okrem objektovej paradigmy podporuje aj štruktúrovanú a naviac funkcionálnú paradigmu. Ide o dynamický typový jazyk. Bol vybraný pre vysokú efektivitu pri písaní kódu, veľmi dobrú podporu práce s reťazcami (vďaka jeho "skriptovacím"vlastnostiam) a bohatému výberu vstavaných knižníc.

Z externých knižníc bola použitá knižnica Pycogent[4] [15] a to predovšetkým na načítanie biologických sekvencií a v niektorých krokoch fylogenetickej analýzy (napr. algoritmus Neighbor - Joining).

Na vývoj bolo použité vývojové prostredie Eclipse spolu so zásuvným modulom implementujúcim prostredie pre vývoj v jazyku Python s názvom PyDev.

7.5 Implementácia

Implementácia bola v prostredí PyDev rozdelená na 8 balíčkov. Toto rozdelenie je určitým konsensom medzi logickým návrhom (obrázok 7.2) a diagramom tried (obrázok 7.3). Oproti logickému návrhu pridáva niekoľko podporných balíčkov.

Pozn.: V texte sa ďalej spomína pojem modul, má však odlišný význam ako pri logickom návrhu, v tomto prípade predstavuje jeden zdrojový súbor napísaný v jazyku Python. V stručnosti bude na tomto mieste popísaný každý z balíčkov spolu s modulmi, ktoré obsahuje:

- Balíček **Algorithms** – obsahuje 4 moduly:
  - Modul **DistanceAlgorithm.py** implementuje bázovú triedu, ktorá obsahuje metódy spoločné pre všetky dištančné algoritmy. Metóda *LoadSeq*() načíta sekvencie vo formáte FASTA pomocou modulu MinimalFastaParser z knižnice PyCogent. Metóda Align() zarovná vstupné sekvencie a to tak, že inštancuje triedu MuscleCommandline z knižnice PyCogent, ktorá zapúzdruje komunikáciu s programom Muscle[5]. Metóda CountMatrix() naplní hodnoty dištančnej matice získané pomocou nastaveného substitučného modelu.
  - Modul NeighborJoining.py zapúzdruje vytvorenie fylogenetického stromu pomocou metódy Neighbor-Joining, využitý je modul nj z knižnice PyCogent
  - Štruktúra modulu UPGMA.py je riešená podobne ako v predošlom prípade, použitý je modul upgma z knižnice PyCogent
  - Modul TDCG je implementáciou vlastného algoritmu Top Down Clustering of Groups s ústrednou metódou *Split*() implementujúcou rekurzívny algoritmus uvedený v podsekcii 6.5.1.

[4] z angl. Python the COmparative GENomic Toolkit

[5] voľne dostupný na stiahnutie na adrese http://www.drive5.com/muscle/downloads.htm

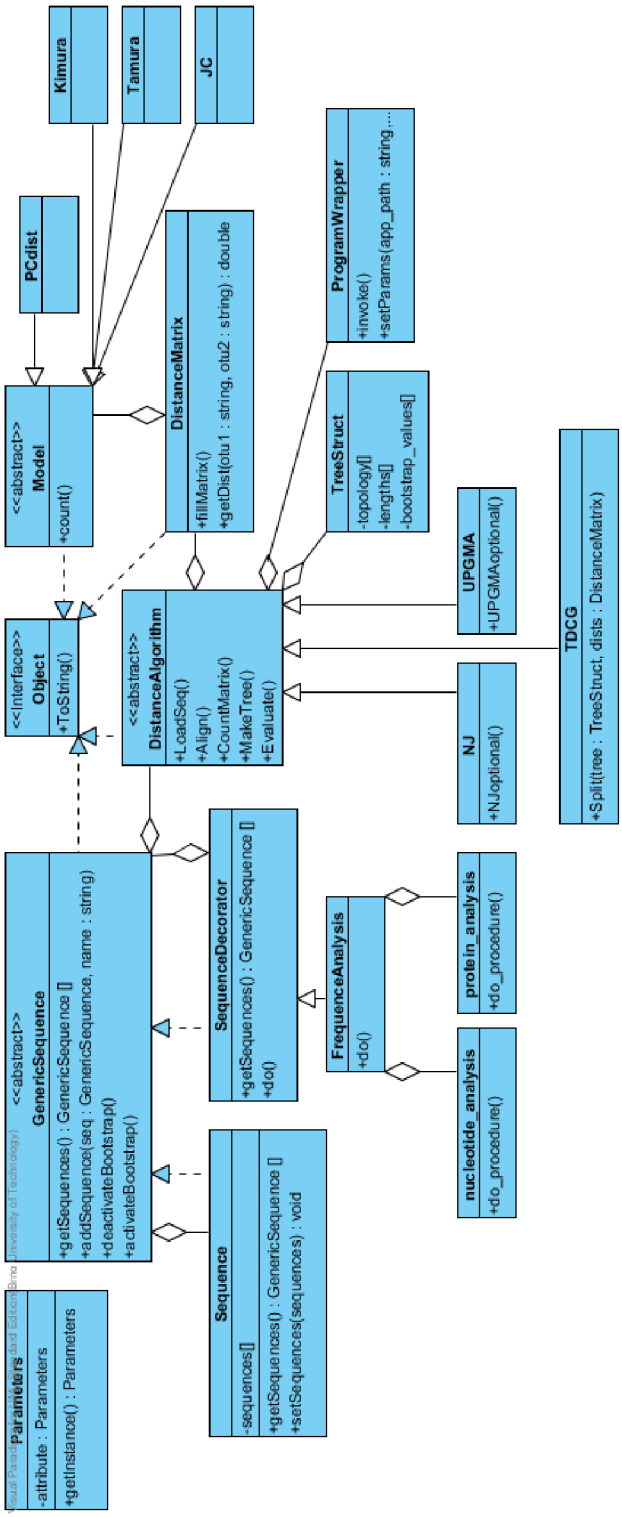
- Balíček **Exceptions** – obsahuje 2 moduly implementujúce užívateľské výnimky, ktoré sú vyvolané pri chybnom zadaní parametrov programu resp. chybnou prácou s dištančnou maticou
- Balíček **Execution** – obsahuje 2 pomocné moduly, ktoré slúžia na korektné spracovanie vstupných parametrov programu:
  - Modul **Parameters.py** obsahuje singleton triedu, ktorá po inštancovaní udržuje aktuálne parametre programu.
  - Modul **Formatters.py** obsahuje pomocné funkcie, ktoré parsujú z FASTA hlavičky sekvencie jej príslušnosť k skupine. Väčšinou ide o spoločný prefix v názve.
- Balíček **Generic** – obsahuje 1 modul, a to generickú triedu Object.py, z ktorej dedia všetky triedy, u ktorých je požadovaný textový výstup
- Balíček **Matrices** – obsahuje modul DistaneMatrix.py implementujúci metódy nad dištančnou maticou. Túto dátovú štruktúru využívajú všetky implementované algoritmy na fylogenetickú rekonštrukciu. Jednotlivé položky sa ukladajú do asociatívneho poľa (vstavaný typ *dictionary*), kde kľúčom k nim je vždy dvojica (vstavaný typ *tuple*) reťazcov.
- Balíček **Models** – obsahuje 5 modulov, ktoré implementujú jednotlivé modifikované substitučné modely zo sekcie 6.4.
- Balíček **Sequences** – spolu 3 moduly; implementácia preddefinovaných skupín sekvencií v module **GenericSequence.py** a jeho dekorátor zabezpečujúci frekvenčnú analýzu v module **FrequenceAnalysis.py**. Pri implementácii bola využitá vlastnosť jazyka Python, tzv. *first class functions*, ktorá umožňuje priraďovať za behu programu inštančné metódy objektu, tak, ako keby šlo o inštančné premenné. Bola implementovaná funkcia na frekvenčnú analýzu nukleotidov - *nucleotide_analysis*() a funkcia na frekvenčnú analýzu proteinov - *protein_analysis*(). Podľa vstupných parametrov sa až za behu programu rozhodne, ktorá z funkcií sa priradí a spustí ako inštančná metóda objektu **FrequenceAnalysis**. Podobný princíp bol využitý aj pri implementácii bootstrapingu. Pokiaľ je aktivovaný bootstraping, k inštančnej metóde triedy **GenericSequence()** sa priradí špeciálny iterátor, ktorý vyberá jednotlivé prvky (pozične špecifické vektory) náhodne. Pokiaľ táto špeciálna iterátorová metóda nie je aktivovaná, iterátor vracia prvky v štandardnom poradí.
- Balíček **Wrappers** – v aktuálnom stave implementácie obsahuje tento balíček modul **Consense.py**, ktorý zapúzdruje komunikáciu s programom consense z balíčka Phylip. Využíva pri tom vstavanú knižnicu na spúšťanie podprocesov **subprocess**.

Pozn.: Balíček **Algorithms** obsahuje ešte naviac modul **NeighborJoiningModif.py**, ktorý je vlastnou implementáciou algoritmu Neighbor-joining s využitím reprezentácie stromu vo forme dvojdimenzionálneho poľa (v implementácii je to pole asociatívnych polí), teória viď časť 6.6.1. V súčasnej implementácii ide o experimentálny modul, ktorý nebol využitý na získavanie výsledkov z reálnych dát (kapitola 8).

Výsledkom implementácie je prenositeľná konzolová aplikácia. Vstupným bodom programu je skript **analysis.py**, ktorý inštanciuje algoritmy fylogenetickej analýzy z balíčka **Algorithms**. Program v závislosti od vstupných parametrov vygeneruje dva stromy vo formáte NEWICK - originálny strom a konsensuálny strom s bootstrapovými hodnotami.

Na porovnanie týchto stromov sa v súčasnosti používa program *rbvotree*[6]. Príklady použitia, popis jednotlivých parametrov a niektoré testovacie dáta sú k dispozícií na priloženom CD (viď adresárová štruktúra v prílohe).

[6]dostupný na http://mobyle.pasteur.fr/cgi-bin/portal.py



Obrázok 7.3: Návrh fylogenetickej knižnice v jazyku UML

Kapitola 8

Praktická aplikácia vytvorených algoritmov

Kapitola popisuje nasadenie navrhnutých metód pri riešení niektorých praktických problémov.

8.1 Určovanie fylogenetického stromu ľudskej populácie

Na overenie funkčnosti a robustnosti algoritmu bola vytvorená nasledovná prípadová štúdia: Získať väčšie množstvo sekvencí mitochondriálnej DNA ľudskej populácie z rôznych oblastí zemegule a vytvoriť na základe tejto vzorky dát agregovaný fylogenetický strom. Výsledok porovnať s niektorou už prezentovanou štúdiou.

8.1.1 Mitochondriálna DNA

Mitochondriálna DNA sa nachádza v mitochondriách každej bunky ľudského tela, je teda súčasťou mimojadrovej genetickej informácie. Ľudská mitochondriálna DNA má veľkosť 16 569 párov báz, obsahuje celkom 37 génov, z toho 24 prestavujú gény pre nekódujúcu RNA. V prípade mtDNA dochádza u živočíchov k tzv. materiálnej dedičnosti – genetická informácia je dedená po matke. Tým pádom nedochádza k rekombinácii DNA obidvoch rodičov a teda mtDNA sa presúva z rodiča (matky) na potomka v nezmenenom stave. Práve vďaka tomuto špecifiku je často využívaná na rôzne genetické analýzy a kvôli pomerne vysokému podielu mutácií oproti jadrovej DNA aj na popis migrácie ľudstva a vo forenznom inžinierstve.

8.1.2 Metodika výberu a spracovania dát

Prehľad použitých sekvencií udáva tabuľka 8.1. Väčšina bola získaná zo zdroja [12] (ak je tomu inak, je referencia na sekvencie uvedená priamo v tabuľke). Dodatočne bola pridaná skupina sekvencií šimpanzov (Chimpanzee), ktorá má slúžiť ako *outgroup*. Pracuje sa s predpokladom, že priemerná vnútroskupinová vzdialenosť sekvencií jedincov, ktorí pochádzajú z rovnakého svetadiela / kontinentu, je nižšia ako vzdialenosť reprezentantov dvoch odlišných preddefinovaných skupín.

Nakoľko je veľkosť mitochondriálneho genómu príliš veľká na analýzu v bežných podmienkach (obzvlášť viacnásobné zarovnanie je mimoriadne časovo náročné), bola vybraná

Dáta	Preddefinovaná skupina	Počet sekvencií
European (Kivisild) Sardinian (Fraumene) Italian (Achilli)	Europe	215
Papua New Guinean (Ingman) Melanesian (Kivisild) Australian (Ingman)	Australia/Oceania	41
Japanese (Tanaka) Chinese (Kong)	East Asia	720
American [9]	America	5
African [10]	Africa	4

Tabuľka 8.1: Zoznam spracovávaných sekvencií ľudskej mitochondriálnej DNA (eventuálne s menom osoby, ktorá kolekciu zozbierala, pokiaľ je známe) spolu so skupinami, do ktorých boli vložené a počtom sekvencií, ktoré obsahujú

z každej sekvencie vysoko polymorfná oblasť o veľkosti približne 200 bp (pozícia 7900 až 8100).

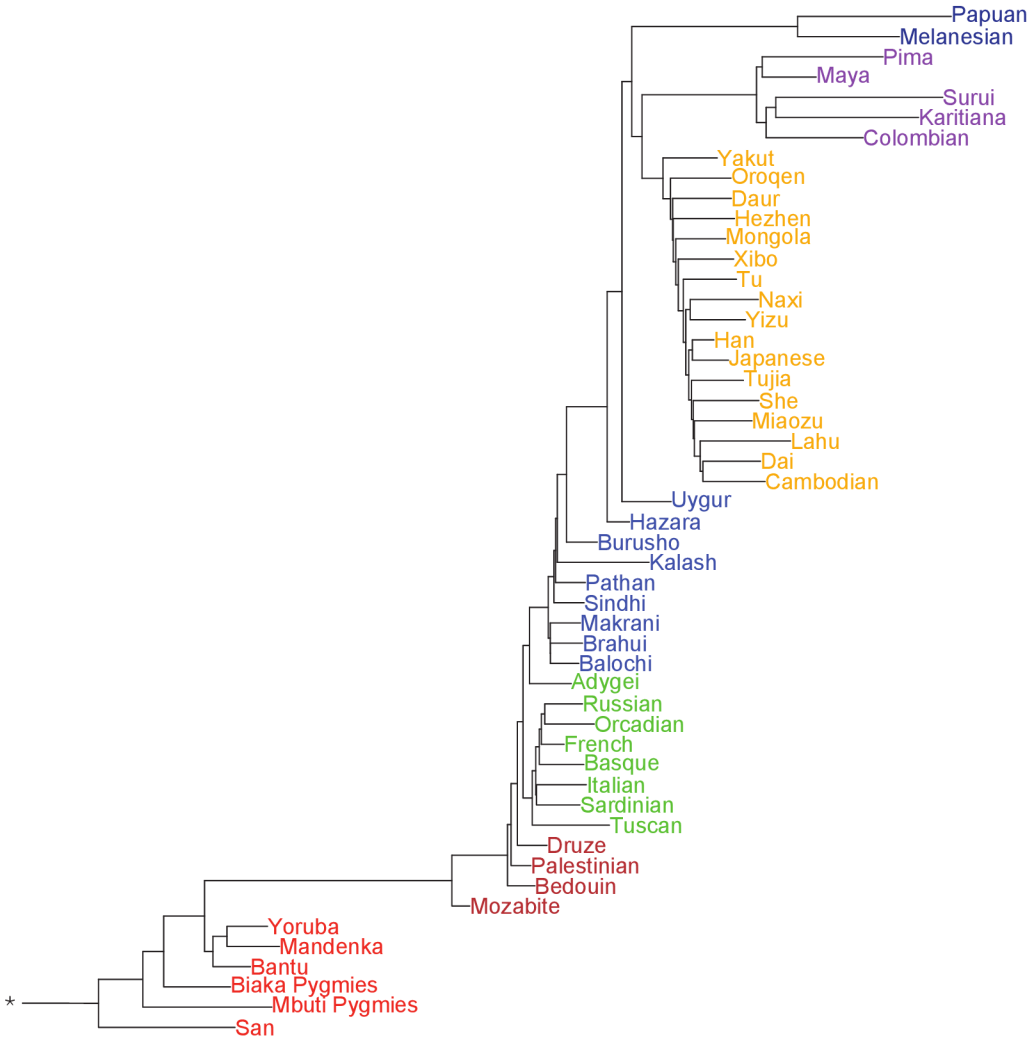
Dáta prešli všetkými štyrmi krokmi fylogenetickej analýzy pomocou dištančného algoritmu (popis viď sekcia 6.2). Zarovnanie bolo prevedené pomocou algoritmu Clustal (popis viď podsekcia 2.4.1). Na preddefinované skupiny sekvencií (tabuľka 8.1) bola aplikovaná frekvenčná analýza a následne modifikovaná metrika p vzdialenosti (obsah logaritmu v rovnici 6.2. Výslednú maticu vzdialeností potom spracoval algoritmus neighbor-joining, ohodnotenie bootstrappovými hodnotami bolo prevedené tak, že sa vytvoril konsenzuálny strom z 500 rôznych replikácií pôvodnej vzorky dát a jeho bootstrappové hodnoty sa následne priradili k pôvodnému stromu.

8.1.3 Dosiahnuté výsledky

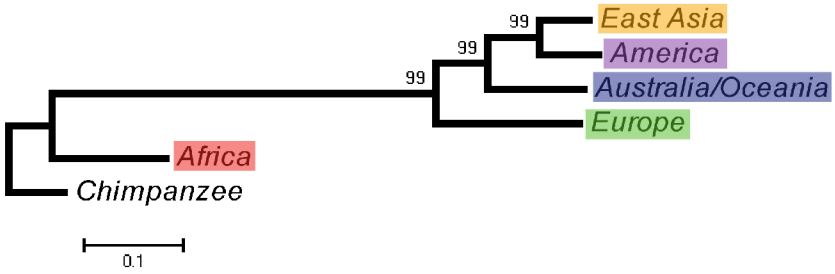
Výsledok fylogenetickej analýzy pomocou modifikovaného algoritmu zobrazuje obrázok 8.2. Skupina *Chimpanzee* bola umiestnená ako tzv. outgroup na zakorenenie stromu. Získaná topológia sa zhoduje s topológiou zverejnenou v štúdii časopisu Science [17], ktorá je zobrazená na obrázku 8.1. Spomínaná štúdia vychádzala z úplne rozdielnych dát, išlo o vzorky 51 populácii, ktoré dovedna obsahovali 650 000 lokusov s výskytom jednonukleotidových polymorfizmov. Získanú topológiu na obrázku 8.2 naviac podporujú vysoké bootstrappové hodnoty, z čoho sa dá usúdiť, že použitá modifikácia je pre tento účel zmysluplná a môže byť ešte v budúcnosti pri podobnej analýze nápomocná. Výsledky uvedené v tejto kapitole budú publikované na konferencii HCII 2011 [35].

Kvôli porovnaniu bol ďalej vytvorený fylogenetický strom pomocou konsenzuálnych sekvencií. Parametre algoritmu boli nastavené rovnako ako v predošlom prípade (p-vzdialenosť, Neighbor-joining, 500 bootstrapových replikácii) Topológia fylogenetického stromu zrekonštruovaného na základe konsenzuálnych sekvencií (obr. 8.3) nezodpovedá topologii stromu v spomínanej štúdii (obrázok 8.1; [17]), a teda ani stromu získaného pomocou vlastného modifikovaného algoritmu (obr. 8.2, popis algoritmu viď sekcia 6.4).

Do pozornosti treba dať zámenu uzlov *Europe* a *American*. Nerelevantnoť zhluku *East_Asia* s *Europe* podporuje nízka bootstrapová hodnota. Z porovnania stromu z konsenzuálnych sekvencií (obr. 8.3) a stromu vytvoreného modifikovaným algoritmom (sekcia 6.4) **je zrejmý prínos implementovaného riešenia a potvrdzuje sa tým zároveň**



Obrázok 8.1: Fylogenetický strom zo štúdie zverejnenej v časopise Science [17]

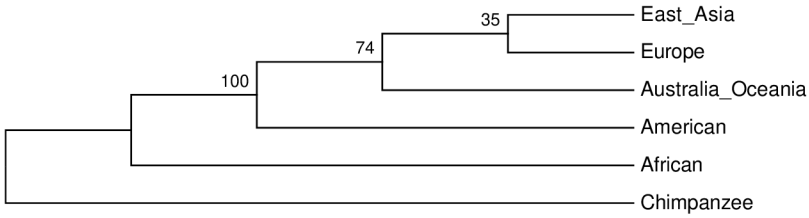


Obrázok 8.2: Získaný fylogenetický strom, farby jednotlivých uzlov korešpondujú so stromom z obrázku 8.1 tak, aby bolo vidieť zhodu v topológii.

hypotéza, že je dôležité udržiavať informáciu o všetkých prvkoch preddefinovaného zhluku.

Pozn.: Dáta boli analyzované aj algoritmom TDCG, vizualizované výstupy s krátkym

komentárom sú k dispozícií na priloženom CD.



Obrázok 8.3: Strom ľudskej populácie získaný z konsensuálnych sekvencií

8.2 Určovanie fylogenetického stromu rastlín z rodu Eleocharis

8.2.1 Popis dát a zvolené metódy

Ide o sekvencie tzv. vnútorných prepisovaných medzerníkov (ITS[1]) rastlín z rodu Eleocharis. Tieto sekvencie sa veľmi často používajú na odvodzovanie evolučných vzťahov. Na analýzu bol zvolený algoritmus Neighbor-joining, Tamurov trojparametrový substitučný model, počet replikácii pre bootstraping bol 500, metóda pre tvorbu konsensuálneho stromu bola *Majority-rule consensus* (viď sekciu 3.4). Modifikovanej metóde bolo predložené rozdelenie do 17 skupín podľa znalostí experta.

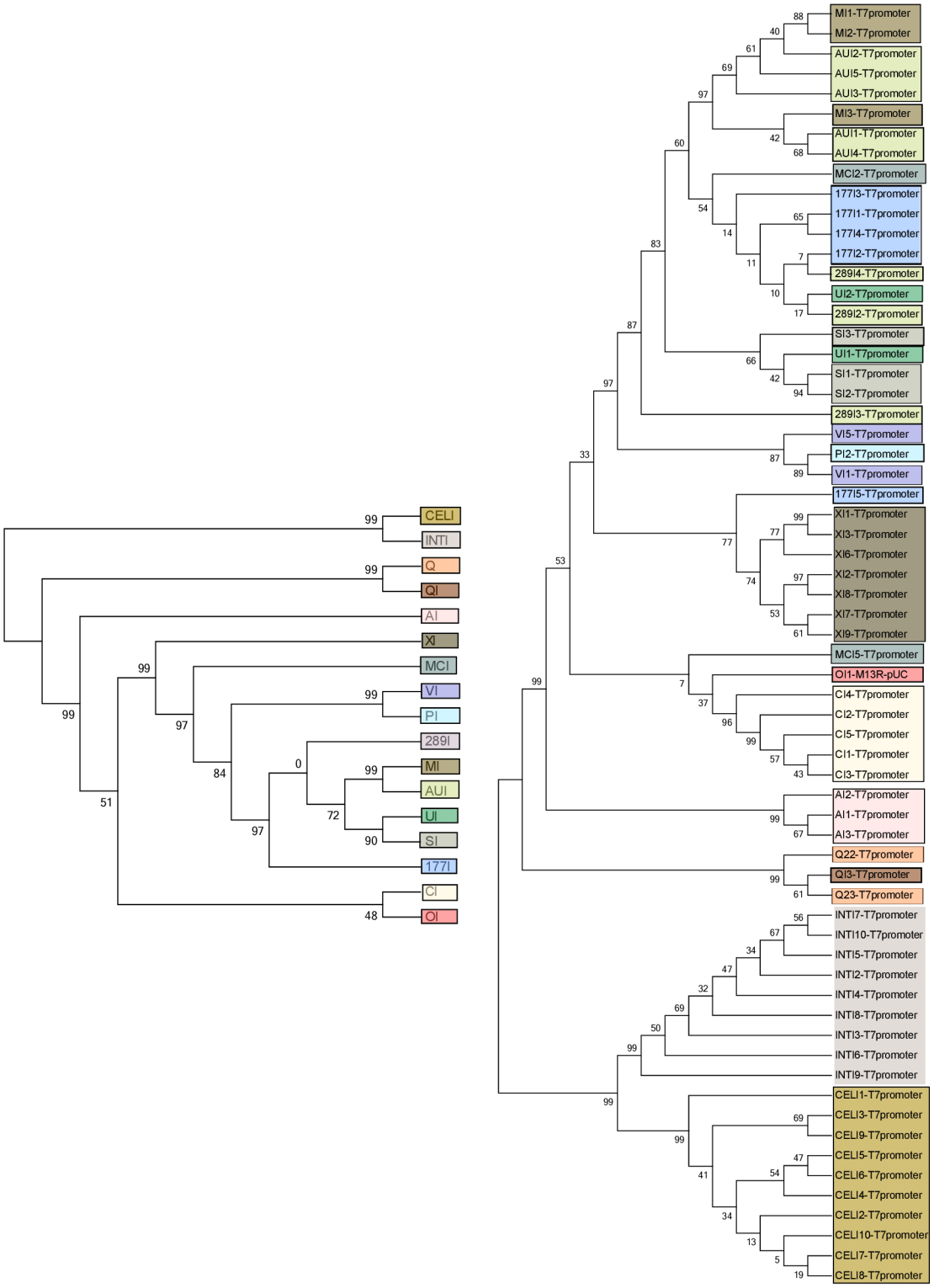
8.2.2 Interpretácia dát

Výsledok fylogenetickej analýzy modifikovanou metódou znázorňuje obrázok 8.4 vľavo. Farby jednotlivých uzlov pravého stromu (kde boli evolučné vzťahy odvodené štandardnou Tamurovou metrikou a neboli definované preddefinované zhluky) zodpovedajú farbám príslušných agregovaných uzlov stromu ľavého. Na zakorenenie bol použitý podstrom skupín sekvencií $CELI$ a $INTI$, pretože tieto sekvencie (zhluky) reprezentujú bazálne druhy rodu Eleochari.

Fylogenetický strom zástupcov rodu Eleocharis odvodený pomocou modifikovanej metriky zodpovedá doposiaľ publikovaným prácam o fylogenézii Eleocharis ([27],[24]), kde sú jasne vymedzené 4 podrody: Limnochloa (CELI, INTI), Zinserlingia (QI, Q), Scirpidium (AI) a Eleocharis (CI, OI, MCI, PI, VI, 289I, MI AUI, UI, SI, 177I). V rámci podrodu Eleocharis sa naviac vymedzuje sekcia Eleocharis, ktorá je so silnou bootstrapovou podporou (99 %) oddelená aj v prípade stromu založenom na modifikovanom algoritme (shluky XI, MCI, VI, PI, 289I, MI, AUI, UI, SI, 177I). Sekcia Eleocharis naviac v zhode s predchádzajúcimi prácami obsahuje aj druh MCI (Eleocharis macrostachya; obr. 8.4 vlevo), ktorý je reprezentovaný dvoma sekvenciami, z nich len jedna spadá do sekcie Eleocharis (obr. 8.4 vpravo). Podobnú situáciu sa dá vysledovať v rámci sekcie Eleocharis v prípade poddruhov UI a SI (Eleocharis uniglumis poddruh uniglumis a Eleocharis uniglumis poddruh sterneri), ktoré tvoria silne podporovanú skupinu (bootstrap 90%; obr. 8.4 vľavo), napriek tomu že kazdá z dvoch sekvencí poddruhu UI spadá do rozdielnych skupín (obr. 8.4 vpravo).

[1]skratka pre internal transcribed spacer

Možno teda celkovo skonštatovať, že modifikovaná Tamurova metrika nám dala relevantné výsledky a na analýzu podobného typu dát je vhodná.



Obrázok 8.4: Porovnanie stromu vytvoreného pomocou modifikovaného modelu Tamuru na zhlukovanie preddefinovaných zhlukov (vľavo) a stromu vytvoreného štandardným spôsobom

Kapitola 9

Integrácia vytvorených nástrojov

Jedným z dodatočne vytýčených cieľov práce je poskytnúť užívateľovi prehľadné grafické užívateľské rozhranie dostupné z webu. Po dôslednom zvážení a konzultácii so zadávateľom projektu bol zvolený systém Mobyle [26] ako prostriedok na zverejnenie vytvorenej aplikácie a na generovanie potrebného rozhrania k užívateľovi.

Ako už bolo spomenuté v sekcii 4.3, ide o bioinformatický framework a webový portál. V tejto kapitole bude systém popísaný z pohľadu administrátora a programátora, zameria sa teda na jeho štruktúru a ďalej popis inštalácie a postupu pri integrovaní novej aplikácie do systému.

9.1 Systém Mobyle

Systém Mobyle bol vytvorený s cieľom splniť nasledovné základné koncepty:

1. **Vytvorenie rovnakého užívateľského rozhrania pre rozličné programy** a ich spojenie do spoločného systému. Mobyle je navrhnutý tak, aby umožnil združenie prakticky neobmedzeného množstva bioinformatických aplikácii pod "jednou strechou".

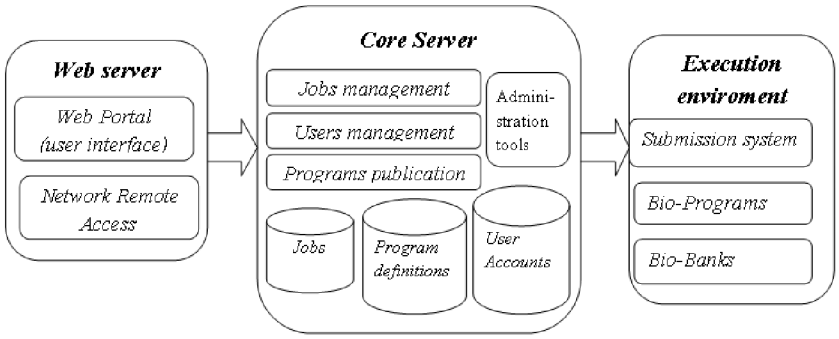
2. **Vytvorenie perzistentného pracovného priestoru (*workspace*)**. Systém umožňuje jednak prístup do systému ako hosť, ako aj vytvorenie užívateľského účtu. Užívateľský účet umožňuje trvalé uloženie analyzovaných dát a výsledkov, ku ktorým je možné sa jednoducho vrátiť.

3. **Popis rozhraní pomocou XML**. Všetky dáta vrátane popisu programov, definícií služieb a užívateľského pracovného priestoru sú uložené vo formáte XML.

4. **Distribúcia programovej služby po sieti**. Program integrovaný do systému nemusí mať len lokálnu funkciu. Vďaka funkcionalite Mobyle Net je možné programy distribuovať do vzdialených inštalácií systému Mobyle. Každá inštalácia systému teda môže pozostávať zo série lokálnych služieb a takých, ktoré sa spúšťajú vzdialene z úplne iného uložišťa [26].

9.1.1 Kompomenty systému

Obrázok 9.1 zobrazuje základné komponenty systému a ich vzájomné prepojenie.



Obrázok 9.1: Štruktúra systému Mobyle, prevzaté z [26]

9.2 Inštalácia a konfigurácia

Prerekvizitou k úspešnej inštalácii systému Mobyle je operačný systém na báze Unix a server Apache[1]. Ďalej je to základná inštalácia jazyka Python vo verzii 2.5 spolu s balíčkami uvedenými v tabuľke 9.1.

Balíček	**Hlavná funkcia**
simpletal, verzia >= 4.1	šablónovací jazyk
4suite, verzia >= 1.0.2	spracovanie XML
simplejson	JSON enkodér/dekodér pre Python
python imaging library	knižnica na spracovanie obrázkov pre Python
PyCAPTCHA	framework pre Python na tzv. CAPTCHA testy
libxml2	parser pre XML
siginterrupt	rozhranie v jazyku python pre systémovú funkciu *siginterrupt*()

Tabuľka 9.1: Zoznam pythonovských balíčkov potrebných pre inštaláciu systému Mobyle

Pozn.: Bližší popis technológií z tabuľky 9.1 a odkazy na stiahnutie možno nájsť na stránkach systému Mobyle[2].

Naviac je potrebná utilitka squizz[3], ktorá slúži na prevod medzi formátmi sekvencií a zarovnaní.

Samotná inštalácia je potom jednoduchý automatizovaný proces, ktorý sa spustí nasledovným príkazom:

python setup.py install
–install-core=/path/where/to/install/core/files
–install-cgis=/path/where/to/install/cgis/files
–install-htdocs=/path/where/to/install/html/files

- položka '–install-core' určuje umiestnenie jadra aplikácie a k nemu prislúchajúcej dokumentácie a príkladov
- položka '–install-cgis' určuje umiestnenie cgi skriptov, ktoré spúšťa webový server

[1] ide o *open-source* http server, bližšie informácie na http://httpd.apache.org/
[2] https://projets.pasteur.fr/wiki/mobyle/
[3] dostupná na ftp://ftp.pasteur.fr/pub/gensoft/projects/mobyle/

- položka '–install-htdocs' určuje koreňový adresár pre umiestnenie html súborov portálu, užívateľských sedení a programových definícií (viď ďalej)

9.2.1 Konfigurácia portálu

Na konfiguráciu portálu existuje vzorový súbor, ktorý je uložený v adresári Example/Local/Config/Config. a je nutné ho prekopírovať do Local/Config/Config.py a upraviť podľa vlastných potrieb. Upravená verzia konfiguračného súboru spolu s komentárom k jednotlivým položkám je na priloženom CD nošiči.

Portál disponuje troma druhmi služieb:

- programami,
- workflow procesmi
- a pohľadmi (*viewers*).

Všetky tri sa definujú pomocou jazyka XML. Ďalej bude diskutovaná programová služba portálu.

9.3 Integrácia nových metód

Do portálu Mobyle možno integrovať ľubovoľnú konzolovú aplikáciu spustiteľnú v bežiacom operačnom systéme. XML sa v tomto prípade využíva na:

- popis, ako invokovať program (programový *wrapper*)
- popis, ako zobraziť jednotlivé nastavenia programu vo formulári, prípadne ako zobraziť výstupy programov
- sémantickú kontrolu hodnôt a parametrov programu.

Koreňovým elementom programovej definície je tag *program*. Jeho obsah je rozdelený do dvoch častí:

- hlavička (tag *head*) obsahuje základné informácie o programe (názov, verzia, autor, umiestnenie v rámci hierarchie aplikácií atď.)
- tag *parameters*, ktorý obsahuje popis vstupných dát, popis možných nastavení programu, predpis, ako vytvoriť príkaz na spustenie služby atď.

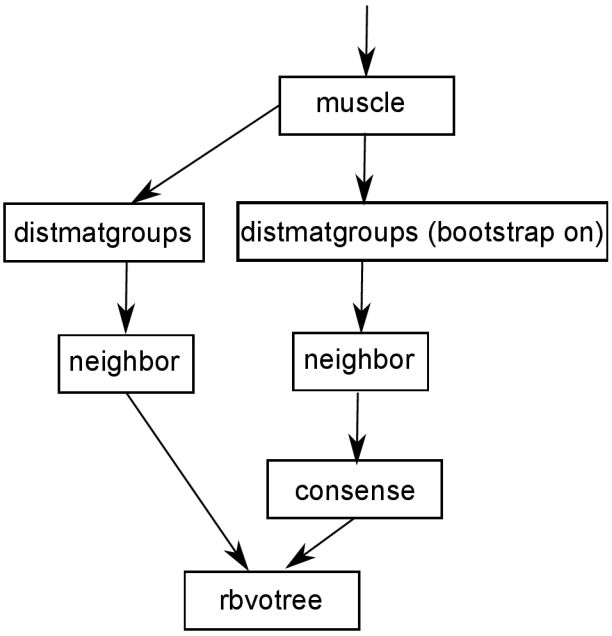
9.4 Integrácia vlastnej implementácie

Pre účely integrácie bol s využitím vybraných modulov fylogenetickej knižnice vytvorený spustiteľný skript s názvom *distmatgroups*, ktorý nahrádza pôvodný program *distmat*. Ten na vstupe prijíma zarovnané sekvencie a ich príslušnosť k skupine (spravidla je príslušnosť danej sekvencie do skupiny dopredu určeným spôsobom zakódovaná do hlavičky FASTA). Obrázok 9.2 znázorňuje pozíciu vytvoreného skriptu z hľadiska celkového behu analýzy. Po zarovnaní sekvencií sa beh rozdelí na dve časti:

- ľavá vetva generuje originálny fylogenetický strom - vytvorí dištančnú maticu skupín pomocou *distmatgroups* a zanalyzuje pomocou programu neighbor, ktorý zahŕňa algoritmy NJ alebo UPGMA

- pravá vetva vygeneruje pomocou *distmatgroups* n pseudovzoriek a z nich n dištančných matíc, tie spracuje program *neighbor* a vytvorí n stromov, z ktorých následne program *consense* vytvorí konsenzuály strom s bootstrapovými hodnotami
- ľavá a pravá vetva sa spoja v programe *rbvotree*, ktorý porovná originálny strom a konsenzuálny strom a priradí originálnemu stromu bootstrapové hodnoty
- strom sa následne vizualizuje pomocou programu *newicktops* a uloží sa vo formáte PostScript[4]

Pre programy muscle, neighbor, rbvotree, consense a newicktops existuje popis ich rozhrania vo formáte XML pre Mobyle ([5], resp. na priloženom CD), pre program *distmatgroups* musel byť tento popis vytvorený (viď priložené CD).



Obrázok 9.2: Workflow nástrojov na fylogenetickú analýzu podľa dištančnej matice s inkorporovaným vlastným bioinformatickým nástrojom

Systém Mobyle bol podľa popisu úspešne nainštalovaný a konfigurovaný na lokálnej inštalácií systému Ubuntu. V ďalšom kroku bude umiestnený do siete WWW pre potreby Mgr. Zedka.

[4] formát na grafický popis tlačiteľných dokumentov
[5] dostupné na ftp://ftp.pasteur.fr/pub/gensoft/projects/mobyle/

Kapitola 10

Rozšírenia do budúcnosti

Táto kapitola popisuje niekoľko možností, akým spôsobom rozšíriť a vylepšiť prezentované riešenie. Ďalej zahrňuje námety na návrh nových algoritmov.

10.1 Portál Mobyle

Dlhodobým zámerom autora predloženej práce je uviesť portál Mobyle do chodu v rámci niektorého zo serverov FIT na VUT v Brne. Vďaka tomu by bolo možné postupne pridávať ďalšie bioinformatické nástroje (napríklad tie, vytvorené v rámci študentských prác). Tento krok by uľahčil distribuovanie aplikácií medzi bioinformatickú komunitu, a teda by priniesol aj rýchlejšiu spätnú väzbu zo strany užívateľov a na omnoho väčšej vzorke.

10.2 Kódovanie stromu

Navrhnuté kódovanie fylogenetického stromu pomocou dvojdimenzionálneho poľa (matice, viď 6.6.1) sa javí ako vhodná dátová štruktúra pre ďalšie použitie. Myšlienky, ktoré by mohli byť ďalej rozvíjané:

- **Tvorba genetického algoritmu**. Jednotlivé riadky matice sa dajú interpretovať ako binárne čísla. Tie sú vhodné na kódovanie chromozómu pri evolučných algoritmoch a jeho následnú manipuláciu (kríženie, mutácie). Naviac obsahuje toto kódovanie presné sémantické pravidlá, takže by nebolo zložité vyhodnotiť zmysluplnosť získaného riešenia. Znalosti z evolučnej teórie by sa dali využiť pri zostrojení fitness funkcie – kvalita riešenia by sa posudzovala napríklad na základe súčtu dĺžok všetkých hrán stromu.

- **Kompresia fylogenetických stromov**. Podobne ako v predchádzajúcom bode, aj v tomto prípade sa na maticu fylogenetického stromu hľadí ako na binárny kód, resp. sled jednotiek a núl. V prípade rozsiahlejšieho stromu majú tieto sledy nezanedbateľnú veľkosť a je dobrá šanca znížiť ich priestorovú zložitosť, napríklad pomocou algoritmu RLE[1].

[1] z anglického Run-length encoding, jednoduchý algoritmus na dátovú kompresiu využívajúci dlhé sledy rovnakých hodnôt

10.3 Identifikácia pomocou preddefinovaného zhluku

V celej práci bol zvažovaný prístup, kde sa zo skupiny sekvencií vytvorí sekvenčný profil a s ním sa ďalej pracuje (kapitola 6). Dá sa však formulovať aj iný problém:
Máme k dispozícii niekoľko sekvenčných profilov a naviac sekvenciu, ktorej pôvod je nám neznámy. Chceme zistiť, ktorej zo skupín je najbližšie.
Vychádza sa z predpokladu, že sekvenčný profil v sebe obsahuje informáciu o väčšom množstve sekvencií a tým pádom nepriamo aj informáciu o konzervovaných úsekoch, ktoré sa môžu prekrývať so skúmanou sekvenciou.

10.4 Rozšírenie algoritmov na zhlukovanie preddefinovaných skupín

Jadrom práce bol návrh riešenia na zhlukovanie preddefinovaných skupín. K tejto problematike možno vyvinúť ešte iné riešenia. Ako už bolo spomenuté v sekcii 10.2, genetický algoritmus je jedno z nich. Naviac možno aplikovať iné metódy fylogenetickej analýzy - napríklad znakové. Základným problémom však aj v tomto prípade zostáva, ako reprezentovať preddefinovaný zhluk v analýze a ako definovať vzťahy medzi nimi.

10.5 Paralelizácia výpočtu

Ako naznačuje obrázok 9.2, výpočet prebieha v istej fáze v dvoch, na sebe nezávislých vetvách. Tento proces by sa dal paralelizovať na dva procesory, ak sa naviac zváži, že samotné generovanie a spracovanie pseudovzoriek v pravej vetve (obrázok 9.2) je na sebe nezávislé, možno využiť viac ako dva procesory a podľa toho upraviť algoritmus. Tento krok môže zvýšiť efektivitu výpočtu.

Kapitola 11

Záver

Jadro predloženej práce prezentuje čitateľovi nový prístup k fylogenetickej analýze biologických sekvencií. Ide o špecifické metódy, keď je predmetom analýzy odvodenie evolučných vzťahov medzi celými skupinami sekvencií. Tento prístup, ako bolo ukázané, poskytuje biológovi nový pohľad na evolučné vzťahy medzi organizmami.

Diplomová práca sa zaoberá veľmi rozsiahlou oblasťou bioinformatiky. Ani zďaleka neboli predstavené všetky dostupné metódy a mechanizmy molekulárnej fylogenetiky. Prednosť pri popise a tým pádom aj pri samotnej implementácii dostali tie, ktoré sú overené praxou (z hľadiska výkonnosti aj relevantnosti výstupov), a teda využívané Mgr. Františkom Zedkom, ktorý dohliadal na celý priebeh a smerovanie vytvoreného textu ako aj implementovaných metód a poskytol cenné interpretácie vytvorených výstupov.

Z pohľadu teórie bol výklad zameraný predovšetkým na dve oblasti: viacnásobné zarovnanie a substitučné modely. Toto zameranie bolo účelné. Viacnásobné zarovnanie je mimoriadne náročnou operáciou z hľadiska časovej zložitosti, a preto v značnej miere ovplyvňuje celkovú dobu fylogenetickej analýzy (pokiaľ nie sú k dispozícii už zarovnané sekvencie). Práve substitučné modely tvoria teoretický základ vlastného rozšírenia odhadu evolučnej vzdialenosti preddefinovaných zhlukov (kapitola 6).

V závere práce bol predstavený systém Mobyle a integrácia vytvoreného programu do tohto bioinformatického portálu. Tento portál poskytuje pre biológa veľmi silný nástroj, nakoľko umožňuje veľmi účelne kombinovať jednotlivé analýzy.

Nespochybniteľným prínosom tejto práce je návrh modifikovanej metriky na porovnávanie preddefinovaných zhlukov pomocou matematiky a štatistiky. Prezentované modifikácie modelov Jukes-Cantor, Tamura, Kimura a PC-vzdialenosti v kombinácii s konvenčnou metódou Neighbor-joining dávajú v praxi veľmi dobré výsledky, lepšie, ako konvenčné metódy (tvorba konsensuálnej sekvencie, náhodný výber). Naviac sú širšie použiteľné, ako sa pôvodne predpokladalo, fungujú totiž rovnako dobre pre ľudskú mitochondriálnu DNA ako aj pre sekvencie z rastlinného genómu 8. Výsledky práce boli prezentované na viacerých konferenciách (EDS [34] a EEICT [33]) a na medzinárodnej konferencii HCII [35].

Literatúra

[1] Altschul, S. F., Gish, W., Miller, W. et al. Basic local alignment search tool. *Journal of molecular biology*. October 1990, roč. 215, č. 3. S. 403–410. Dostupné z WWW: <http://dx.doi.org/10.1006/jmbi.1990.9999>. ISSN 0022-2836.

[2] Arslan, A. N. a Bizargity, P. Phylogeny By Top Down Clustering Using a Given Multiple Alignment. In *BIBE*. 2007. S. 809–814.

[3] Betts, M. J. a Russel, R. B. *Amino acid properties and consequences of subsitutions* [http://www.russelllab.org/aas/]. 2003 [cit. 2010-12-22].

[4] Darwin, C. *On the origin of species*. [b.m.]: Murray, London, 1859.

[5] Dayhoff, M. O. a Schwartz, R. M. Chapter 22: A model of evolutionary change in proteins. In *In Atlas of Protein Sequence and Structure*. 1978. Dostupné z WWW: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.145.4315>.

[6] Edgar, R. C. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*. 2004, roč. 32, č. 5. S. 1792–1797. ISSN 1362-4962.

[7] Felsenstein, J. *PHYLIP (Phylogeny Inference Package) version 3.6. Distributed by the author*. 2005.

[8] Flegr, J. *Evoluční biologie*. [b.m.]: Academia, 2009. ISBN 978-80-200-1767-3.

[9] Forster, P., Harding, R., Torroni, A. et al. Origin and evolution of Native American mtDNA variation: a reappraisal. *Am J Hum Genet*. 1996, roč. 59, č. 4. S. 935–945. ISSN 0002-9297.

[10] Herrnstadt, C., Elson, J. L., Fahy, E. et al. Reduced-median-network analysis of complete mitochondrial DNA coding-region sequences for the major African, Asian, and European haplogroups. *American journal of human genetics*. Květen 2002, roč. 70, č. 5. S. 1152–1171. ISSN 0002-9297.

[11] Higgins, D. G. a Sharp, P. M. CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. *Gene*. December 1988, roč. 73, č. 1. S. 237–244. Dostupné z WWW: <http://view.ncbi.nlm.nih.gov/pubmed/3243435>. ISSN 0378-1119.

[12] Ingman, M. a Gyllensten, U. mtDB: Human Mitochondrial Genome Database, a resource for population genetics and medical sciences. *Nucleic acids research.* 2006, roč. 34. ISSN 1362-4962.

[13] Jukes, T. a Cantor, C. *Evolution of protein molecules.* 1969.

[14] Kimura, M. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *Journal of Molecular Evolution.* 1980, roč. 16, č. 2. S. 111–120.
Dostupné z WWW:
<http://www.springerlink.com/content/k434l54572jk557h/>.

[15] Knight, R., Maxwell, P., Birmingham, A. et al. *Genome Biology.* č. 8. ISSN 1465-6906.

[16] Kumar, M. N. . S. *Molecular Evolution and Phylogenetics.* [b.m.]: Oxford University Press, 2000. ISBN 0-19-513584-9.

[17] Li, J. Z., Absher, D. M., Tang, H. et al. Worldwide Human Relationships Inferred from Genome-Wide Patterns of Variation. *Science.* únor 2008, roč. 319, č. 5866. S. 1100–1104. ISSN 1095-9203.

[18] Lipman, D. a Pearson, W. Rapid and sensitive protein similarity searches. *Science.* 1985, roč. 227, č. 4693. S. 1435–1441.
Dostupné z WWW:
<http://www.sciencemag.org/content/227/4693/1435.abstract>.

[19] Markl, J. *Appendix B: Markovské procesy* [prednáška z predmetu Petriho sítě I]. 2006.

[20] Martínek, T. a Rudolfová, I. *Fylogenetické stromy* [prednáška z predmetu Bioinformatika]. 2010.

[21] Martínek, T. a Rudolfová, I. *Konstrukce fylogenetických stromů* [prednáška z predmetu Bioinformatika]. 2010.

[22] Martínek, T. a Rudolfová, I. *Vícenásobné zarovnání* [prednáška z predmetu Bioinformatika]. 2010.

[23] Martínek, T. a Rudolfová, I. *Základy molekulární biologie* [prednáška z predmetu Bioinformatika]. 2010.

[24] M.S., G.-E. a P.M., P. A classification of and key to the supraspecific taxa in Eleocharis (Cyperaceae). *Taxon.* 1997, roč. 46, č. 3. S. 433–449.

[25] Needleman, S. B. a Wunsch, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology.* 1970, roč. 48, č. 3. S. 443 – 453.
Dostupné z WWW:
<http://www.sciencedirect.com/science/article/B6WK7-4DN8W3K-7X/2/0d99b8007b44cca2d08a031a445276e1>. ISSN 0022-2836.

[26] Néron, B., Ménager, H., Maufrais, C. et al. Mobyle: a new full web bioinformatics framework. *Bioinformatics*. 2009, roč. 25, č. 22. S. 3005–3011. Dostupné z WWW: <http://bioinformatics.oxfordjournals.org/content/25/22/3005.abstract>.

[27] Roalson, E. H., Hinchliff, C. E., Trevisan, R. et al. Phylogenetic Relationships in Eleocharis (Cyperaceae): C-4 Photosynthesis Origins and Patterns of Diversification in the Spikerushes. *Systematic Botany*. 2010, roč. 35, č. 2. S. 257–271.

[28] Saitou, N. a Nei, M. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*. 1987, roč. 4, č. 4. S. 406–425. Dostupné z WWW: <http://mbe.oxfordjournals.org/content/4/4/406.abstract>. ISSN 0737-4038.

[29] Sekanina, L. *Evoluční design* [prednáška z predmetu Biologií inspirované počítače]. 2010.

[30] Smith, T. F. a Waterman, M. S. Identification of common molecular subsequences. *Journal of Molecular Biology*. 1981, roč. 147, č. 1. S. 195 – 197. Dostupné z WWW: <http://www.sciencedirect.com/science/article/B6WK7-4DN3Y5S-24/2/b00036bf942b543981e4b5b7943b3f9a>. ISSN 0022-2836.

[31] Tamura, K. Estimation of the number of nucleotide substitutions when there are strong transition-transversion and G+C-content biases. *Molecular Biology and Evolution*. 1992, roč. 9, č. 4. S. 678–687. Dostupné z WWW: <http://mbe.oxfordjournals.org/content/9/4/678.abstract>.

[32] Tamura, K., Dudley, J., Nei, M. et al. MEGA4: Molecular Evolutionary Genetics Analysis (MEGA) Software Version 4.0. *Molecular Biology and Evolution*. 2007, roč. 24. S. 1596–1599. Dostupné z WWW: <http://dx.doi.org/10.1093/molbev/msm092>.

[33] Vogel, I. Modified methods for constructing phylogenetic trees of predefined groups. In *Proceedings of the 17th Conference STUDENT EEICT 2011, Brno, CZ, FEEC VUT v Brně*. 2011.

[34] Vogel, I., Ocenasek, P. a Zedek, F. Computational Molecular Evolution - From Mathematical Models to Novel Distance Metric Based on Intragroup Analysis. In *EDS '11 IMAPS CS International Conference Proceedings, Brno, VUT v Brně*. 2011.

[35] Vogel, I., Zedek, F. a Ocenasek, P. Constructing Phylogenetic Trees Based on Intra-Group Analysis of Human Mitochondrial DNA. *HCII 2011 Conference, Lecture Notes in Computer Science, Orlando, FL, USA*. 2011.

[36] Wikipedia. *Central dogma of molecular biology — Wikipedia, The Free Encyclopedia*. 2010. [Online; cit. 2010-12-23]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Central_dogma_of_molecular_biology>.

[37] Wikipedia. *Continuous-time Markov process — Wikipedia, The Free Encyclopedia.* 2010. [Online; cit. 2010-12-29].
Dostupné z WWW:
<http://en.wikipedia.org/wiki/Continuous-time_Markov_process>.

[38] Wikipedia. *Point accepted mutation — Wikipedia, The Free Encyclopedia.* 2010. [Online; cit. 2010-12-23].
Dostupné z WWW:
<http://en.wikipedia.org/wiki/Point_accepted_mutation>.

[39] Yang, Z. *Computational Molecular Evolution.* [b.m.]: Oxford University Press, 2006. ISBN 0-19-856699-9.

Dodatok A

Obsah CD

Priložené CD obsahuje nasledovné adresáre:

- *src* - obsahuje zdrojové súbory programu a knižnicu PyCogent
- *text* - obsahuje text práce a zdrojové súbor textu práce systému LaTeX
- *doc* - obsahuje programovú dokumentáciu a manuál k aplikácii
- *config* - obsahuje konfiguračný súbor *Config.py* pre portál Mobyle a XML popis rozhrania
- *mobyle* - obsahuje zdrojové súbory Mobyle-1.0.RC1
- *outputs* - obsahuje výstupy vytvorených algoritmov zverejnené v texte a ďalšie
- *inputs* - obsahuje vybrané vstupné dáta s rozdelením do skupín