UNIVERZITA KOMENSKÉHO V BRATISLAVE

Fakulta matematiky, fyziky a informatiky

SYSTÉM NA HĽADANIE ORTOLÓGOV V PRÍBUZNÝCH GENÓMOCH

2013 Martin Višňovec

UNIVERZITA KOMENSKÉHO V BRATISLAVE

Fakulta matematiky, fyziky a informatiky

e139ca49-4958-4025-9c01-2ccf1448c1ae

Systém na hľadanie ortológov v príbuzných genómoch

Diplomová práca

Študijný program: Informatika

Študijný odbor: 9.2.1 Informatika

Školiace pracovisko: Katedra Informatiky

Školiteľ: Mgr. Tomáš Vinař, PhD.

2013 Martin Višňovec

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Martin Višňovec
Študijný program: informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: 9.2.1. informatika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: slovenský

Názov: Systém na hľadanie ortológov v príbuzných genómoch

Cieľ: V súčasnosti prebieha sekvenovanie genómov mnohých organizmov. Pre jednotlivé gény v novo-osekvenovanom genóme často chceme nájsť ich ekvivalenty v iných, už známych, genómoch. Cieľom práce bude vytvoriť praktický softvér na túto úlohu, ktorý bude spájať viacero už existujúcich softvérových nástrojov.

Vedúci: Mgr. Tomáš Vinař, PhD.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: doc. PhDr. Ján Rybár, PhD.

Dátum zadania: 15.11.2011

Dátum schválenia: 08.12.2011 prof. RNDr. Branislav Rovan, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Čestne prehlasujem, že som túto diplomovú prácu vypracoval samostatne s použitím uvedených zdrojov.

V Bratislave, 23. 4. 2013 .....

Ďakujem vedúcemu diplomovej práce PhD. Tomášovi Vinařovi za odborné vedenie práce, cenné rady, ochotu a čas venovaný konzultáciám, bez ktorých by práca nedospela do finálnej podoby.

Abstrakt

Martin Višňovec: Systém na hľadanie ortológov v príbuzných genómoch
Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, Katedra informatiky, Diplomová práca, 46, 2013

V súčasnosti prebieha sekvenovanie genómov mnohých organizmov. Pre jednotlivé gény v novo-osekvenovanom genóme často chceme nájsť ich ekvivalenty v iných, už známych, genómoch. Na začiatku vysvetlíme štruktúru génu a iné biologické aspekty dôležité pre pochopenie fungovania systému. V ďalších častiach práca popisuje samotný program a porovnáva získané výsledky s iným prístupom.

Kľúčové slová: gény, homológy, ortológy, štruktúra génu, viacnásobné zarovnanie

Abstract

Martin Višňovec: Scheme for finding orthologs in related genomes
Comenius University in Bratislava, Faculty of mathematics, physics and informatics, Department of informatics, Masters work, 46, 2013

There is ongoing sequencing of the genomes of many organisms. For individual genes in the newly-sequenced genome we often want to find their equivalents in other, already known genomes. At the beginning we explain the structure of the gene and other biological aspects relevant to the understanding of the scheme. In next part we describe the scheme itself. Finally, we compare our results with different approach.

Key words: genes, homologs, orthologs, gene structure, multiple species alignment

Obsah

Zoznam obrázkov

Úvod

Genetika je pomerne mladý odbor niekde na rozmedzí biológie a informatiky. Napreduje však veľmi rýchlym tempom. Z informatického pohľadu by sme mohli zjednodušene povedať, že sa zaoberá zdrojovým kódom žijúcich (a potenciálne aj už vyhynutých) organizmov. Všetky tieto organizmy majú spoločný pôvod, pričom sa postupom času procesom evolúcie oddeľovali a špecializovali. Podarilo sa rozlúštiť abecedu tvoriacu tento zdrojový kód pozostávajúcu zo štyroch báz. Takisto už poznáme abecedu, v ktorej sa zapisujú funkčné objekty postavené podľa tohto kódu - dvadsaťjeden aminokyselín.

Poznáme pravidlá, podľa ktorých sa prepisuje zdrojový kód do týchto aminokyselín. Prečítali sme celý ľudský zdrojový kód - genóm, ako aj genómy niektorých ďalších cicavcov a iných organizmov. Pomerne dobre dokážeme už aj určiť, ktoré časti genómu tvoria súvislé 'programy' - gény, podľa ktorých sa vytvárajú funkčné objekty - proteíny. Ďalším logickým krokom je snaha zistiť funkciu jednotlivých programov. A keď sa naučíme fungovaniu kódu rozumieť, časom dokážeme tento kód cielene modifikovať alebo dokonca vytvárať nový.

Zistiť účel, na ktorý určitý gén slúži, zatiaľ nie sme schopný iba z jeho sekvencie. Funkciu proteínu určuje jeho štruktúra, ktorá závisí vo väčšej miere od vzájomnej pozície častí génu ako samotnej sekvencie báz. Túto zatiaľ nevieme dostatočne efektívne predpovedať len na základe postupnosti aminokyselín. Preto sa snažíme nájsť v iných organizmoch podobné gény, ktoré s ním majú spoločný vznik a už známu funkciu. Môžeme potom predpokladať, že bude plniť podobnú úlohu. Takéto gény so spoločnou históriou vzniku nazývame ortológy, a ich hľadaním v genómoch organizmov sa v tejto práci zaoberáme.

Práca je členená na tri hlavné časti. Na úvod práce sa venujeme biologickým základom potrebným pre pochopenie problematiky. Popisujeme tiež princípy a vlast-

nosti genetických štruktúr ako sú bázy a gény, ktoré neskôr využijeme pri samotnom hľadaní ortológov. Potom popíšeme myšlienku fungovania implementovaného systému vyhľadávajúceho potenciálne ortológy a kľúčové časti jeho fungovania. Nakoniec sa pozrieme na iný systém, ktorý je určený na túto úlohu, a porovnáme ho s našim z hľadiska výsledkov.

Prílohou k tejto práci je zdrojový kód vytvoreného programy - Corthy, ktorý vyhľadáva ortológy so zachovanou štruktúrou medzi organizmami. Spolu s programom sú priložené aj testovacie dáta a výsledky na nich. Z testovacích dát je obsiahnutá iba časť súborov kvôli ich veľkosti.

Kapitola 1

Problém hľadania ortológov

1.1 Základné pojmy

Každý živý organizmus je určený svojim genómom. Genóm jedinca predstavuje celú jeho genetickú informáciu, ktorá je uložená v podobe DNA. Táto molekula kóduje informáciu pomocou sekvencie nukleotidov. Týmito nukleotidmi sú adenín, cytozín, guanín a tymín. Z hľadiska informácie obsiahnutej v tejto štruktúra nás bude zaujímať iba postupnosť nukleotidov v molekule, môžeme si ju teda predstaviť ako reťazec nad abecedou štyroch písmen {A,C,G,T}.

Genóm organizmu sa môže skladať z jedného alebo niekoľkých takýchto súvislých reťazcov, ktoré nazývame chromozómy. Samotná DNA a teda aj chromozómy sú tvorené dvojicou vlákien nukleotidov, ktoré sú navzájom komplementárne. Cytozín sa dopĺňa s guanínom a adenín s tymínom. Nie všetky časti DNA reťazca sú rovnako dôležité z hľadiska informácie, ktorú obsahujú. Na chromozómoch sa nachádzajú súvislé úseky, z ktorých sa dekódujú proteíny. Tieto môžu byť kódované na obidvoch vláknach nukleovej kyseliny. Ostatné časti chromozómu môžu mať iný význam, alebo vôbec nemusia mať vplyv na fungovanie organizmu.

Proteíny sú molekuly s rôznymi biologickými funkciami, pričom funkcia vo veľkej miere závisí od ich tvaru. Skladajú sa z aminokyselín, teda podobne ako DNA a nukleotidy, proteín je reťazec nad abecedou aminokyselín. Každá aminokyselina je v DNA kódovaná trojicou báz - kodónom. Počet všetkých možných trojíc nukleotidov je 64, počet aminokyselín je 21. Keďže počet kodónov je vyšší, pre niektoré aminokyseliny existuje viacero rôznych zakódovaní. Niektoré kodóny majú špeciálny význam, ktorým

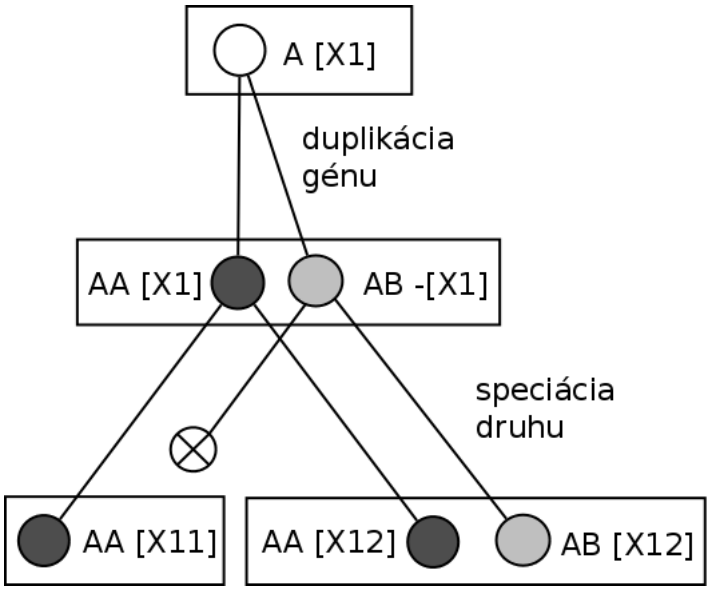
môže byť začiatok alebo koniec proteínu. Kodóny kódujúce koniec proteínu sa neprekladajú do žiadnej aminokyseliny. V ľudskom genóme začiatočný kodón predstavuje sekvencia ATG a koncovými kodónmi sú TAA, TAG alebo TGA.

Úsek DNA kódujúci proteín nazývame gén. Postupnosť nukleotidov v géne kóduje jednotlivé aminokyseliny proteínu. Časti génu, ktoré sa takto prekladajú označujeme exóny, ostatné časti neprekladajúce sa do výsledného proteínu sú intróny. Intróny sa pred prekladom génu vystrihnú, ale ich hranice nie sú tak presne určené ako pri samotných génoch. Intrón teda niekedy vystrihnutý byť vôbec nemusí, alebo sa vystrihne naopak dlhšia časť až po koniec iného intrónu spolu s exónom medzi nimi. Jeden gén môže vďaka tomu kódovať viacero rôznych proteínov, ktoré nazývame transkripty.

1.2 Typy homológov

Gény v rôznych organizmoch môžu mať podobnú sekvenciu. Rovnako niektoré gény v tom istom organizme sa môžu navzájom podobať. Je veľmi málo pravdepodobné, že by vznikli nezávisle od seba, ak je zhoda medzi nimi dostatočne veľká. Najmä pri relatívne blízkych organizmoch je to vysoko nepravdepodobné z dôvodu času, za ktorý by k tomu muselo dôjsť. Môžeme preto predpokladať, že takéto skupiny génov majú spoločný pôvod, ktorý môžeme spätne vystopovať až ku jedinému predkovi. Gény, ktorých sekvencie sa výrazne odlišujú, čo vylučuje ich spoločnú históriu a rovnaký pôvod, nazývame heterológy. Takéto gény napriek tomu môžu v organizmoch plniť podobnú úlohu.

Homológy sú gény, ktoré pochádzajú z jedného predka. Zdieľajú spolu časť svojej histórie vývinu. Majú podobnú sekvenciu báz, pričom veľkosť rozdielov závisí na dĺžke doby, od kedy došlo k rozvetveniu spoločnej histórie ako aj parametrov ovplyvňujúcich rýchlosť zmien v génoch. Môžu ale nemusia plniť v organizme podobnú úlohu. Homológy môžu vznikať rôznymi procesmi, z ktorých najvýznamnejšími sú duplikácia a speciácia. Pri duplikácií dochádza k vzniku kópie génu. Gén sa potom nachádza v organizme na dvoch, prípadne viacerých, miestach. Takto sa môže duplikovať jeden či viacero génov, alebo dokonca celý génom so všetkými génmi v ňom. Viacerými opakovaniami tohto procesu vzniká v organizme skupina génov so spoločným predkom. Takéto gény nazývame paralógy.



Obrázok 1.1: Schéma znázorňujúca procesy vzniku ortológov a paralógov.

Často sa stáva, že po duplikácií jedna z kópií prestane plniť svoju funkciu alebo sa špecializuje na nejakú inú, väčšinou podobnú. Ďalšou možnosťou je, že jeden z paralógov sa stratí v delécií, čím dôjde k jeho vymazaniu, alebo sa naruší jeho štruktúra tak, že už netvorí gén.

Druhým častým spôsobom vzniku homologických génov je speciácia. Vetvením druhu vzniknú dve kópie génu, jedna v každom novom druhu, a ďalej sa v evolúcií vyvíjajú samostatne. Takéto gény nazývame ortológmi. Je veľmi pravdepodobné, že nimi kódované proteíny v organizmoch ďalej plnia rovnakú alebo veľmi podobnú úlohu. Vďaka tomu nám umožňujú predpovedať funkciu génov v iných organizmoch. V súčastnosti stále rýchlejšie pribúdajú nové osekvenované druhy, nájdením ortológov k ľudským génom so známou funkciou môžeme potom predpokladať funkciu jednotlivých génov v danom druhu.

Na obrázku 1.1 možno vidieť proces duplikácie génu A, ktorým vzniká nová kopia génu AB, pričom pôvodný kopírovaný gén je označený AA. Následne procesom speciácie dochádza k rozdeleniu pôvodného druhu X1 na dva nové. V druhu X11 došlo po speciácií k delécií génu AB a obsahuje iba gén AA, zatiaľ čo genóm druhu X12 obsahuje oba gény AA a AB. Gén AA v druhu X11 je ortologický s génom AA v druhu X12, ale nie je ortologický s jeho génom AB. Gény AA v X11 a AB v X12 sú navzájom paralogické.

Štúdium vývoja ortológov a priebehu mutácií v nich nám umožňuje zistiť rýchlosť zmien v génoch s rôznymi funkciami. Na základe toho môžeme určiť, ktoré funkcie v organizme sa rýchlejšie vyvíjajú a menia, a ktoré sa naopak zachovávajú. Rýchlo meniacimi sú napríklad gény ovplyvňujúce imunitu, zmyslové vnemy a rozmnožovanie [8]. Vo všeobecnosti je možné predpokladať, že gény umožňujúce selekčnú výhodu pre organizmus sa menia rýchlo a gény zabezpečujúce základné životné funkcie sa takmer nemenia, keďže mutácie v týchto častiach majú väčšinou výrazne negatívne vplyv na prežitie jedinca prenášajúceho takýto gén.

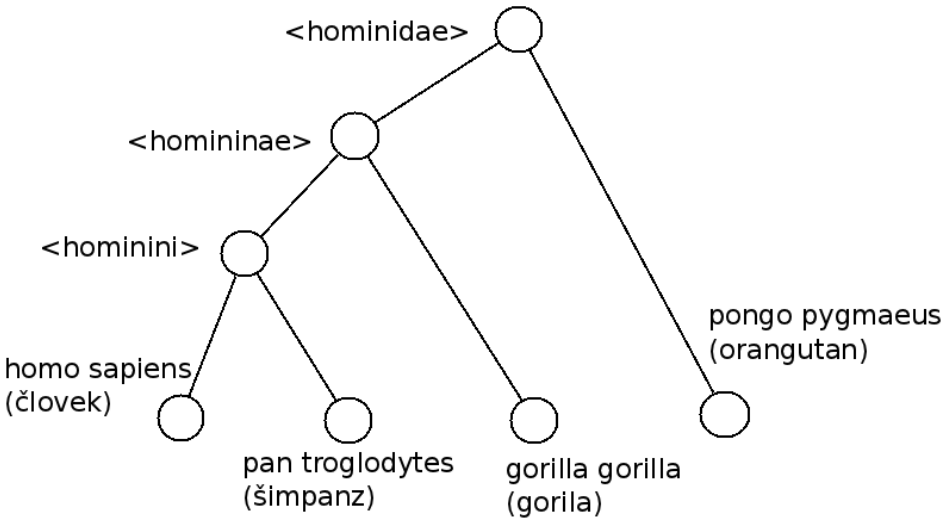
1.3 Hľadanie ortológov

Problém hľadania ortológov sa dá rozdeliť na dva podproblémy. Jedným problémom je nájsť homológy medzi genómami, a druhým určiť, ktoré z nich predstavujú ortológy. Hľadanie homológov sa vykonáva pomocou porovnávania nukleotidových sekvencií a nájdením úsekov s veľkou mierou podobnosti. Týmto spôsobom nájdeme tak ortológy, ako aj paralógy, a chceme ich navzájom odlíšiť. Ku každému génu môže byť nájdených viacero homologických génov. Tieto majú priradenú hodnotu, predstavujúcu vzájomnú podobnosť.

Častým prístupom je považovať za ortológy dvojicu génov, ktoré sú navzájom najviac zhodné. To znamená, že keď sa z daného génu vyberieme do jemu najpodobnejšiemu homológu, následne v tomto géne vyberieme tiež najpodobnejší, tak sa znovu vrátime do génu, v ktorom sme začali. Táto metóda predpokladá, že medzi ortológmi došlo k najmenšiemu počtu zmien. To však nemusí byť vždy nevyhnutne pravda. Mierne vzdialenejší gén môže byť skutočným ortológom.

Problémy spôsobuje viacero javov, ku ktorým počas evolúcie dochádza, okrem speciácie a duplikácie, ktorými sme sa už zaoberali. Pri delécií génu sa stráca jeden z možných homológov, čo komplikuje situáciu. Predstavme si, že sa jednalo o jediný ortológ k inému génu, označme ho B. Keď teraz porovnáme podobnosť génu B s jeho homológmi, dostaneme ako výsledok v skutočnosti paralóg. To je spôsobené absenciou pôvodného ortológu, v tomto prípade ku génu existujú už iba paralógy.

Horizontálny prenos génu predstavuje proces, pri ktorom dochádza k prenosu génu medzi dvomi odlišnými druhmi. To narúša klasický pohľad na evolúciu v podobe fy-



Obrázok 1.2: Fylogenetický strom pre triedu Hominidae.

logenetického stromu, kde po speciácii sa organizmy vyvíjajú ďalej nezávisle. Príklad jednoduchého stromu je na obrázku 1.2. Listy stromu zodpovedajú skúmaným organizmom a ostatné vrcholy predstavujú spoločných predkov určitej skupiny, pričom koreň je spoločným predkom všetkých. Ku medzidruhovému prenosu genetickej informácie dochádza najmä pri baktériách, pričom je známy aj prípad prenosu génu baktérie na vyšší organizmus [1]. Takto prenesené gény sa nazývajú xenológy a považujú sa za druh paralógov.

Iný problém predstavuje nahradenie časti génu sekvenciou iného génu, čím vzniká hybridný gén. Tento jav nastáva pomerne často a dáva vznik génom, ktoré majú viacero predkov. Môžu byť teda ortológmi ku viacerým génom v inom organizme, čo komplikuje odlišovanie ortológov a paralógov.

Okrem už spomínanej vzájomnej podobnosti sekvencií homológov sa pri hľadaní ortológov využívajú aj ďalšie metódy. Dodatočnú informáciu obsahuje aj relatívna pozícia génu k ostatným génom v jeho okolí [19]. Majme teda nejakú množinu génov, z ktorých niektoré môžu, ale nemusia, byť navzájom podobné. Ďalej majme druhý organizmus, v ktorom sa nachádza postupnosť génov navzájom homologických s jednotlivými génmi v prvom genóme. Jednotlivé gény majú rovnaké poradia v skupine a sú na rovnakom vlákne jedného chromozómu. Takáto zachovaná skupina homológov je veľmi často tvorená ortológmi a svedčí o spoločnej evolučnej histórií danej časti genómu.

Vo všeobecnosti možno prístupy ku hľadaniu ortológov rozdeliť na dve hlavné sku-

piny. Jednou je práca s grafom, ktorého vrcholy predstavujú gény a hrany spájajú homológy. Po vytvorení grafu sa podľa ceny hrán a prípadne ďalších parametrov oddelia ortológy od paralógov. Druhú veľkú skupinu tvoria stromové algoritmy, ktoré na vstupe potrebujú fylogenetický strom skúmanej množiny organizmov. Pomocou neho namapujú homologické gény medzi jednotlivými organizmami a snažia sa určiť ich vzájomnú históriu. Tento prístup má lepšie predpoklady na prácu s deletovanými génmi vďaka znalosti predkov a potomkov vo fylogenetickom strome. Vytvorením histórie pre zhluk homologických génov ich následne vieme rozdeliť na homológy a hľadané ortológy.

Príkladmi implementácie grafového prístupu sú napríklad Inparanoid, OrthoMCL [12] a COG/KOG, druhú skupinu stromových algoritmov prestavujú RIO alebo Orthostrapper [17]. Niektoré programy pre hľadanie ortológov obidva prístupy kombinujú, patrí medzi ne napríklad OrthoParaMap.

1.4 Existujúce riešenia

V tejto časti bližšie popíšeme prácu niekoľkých existujúcich spôsobov na určenie ortológov. Líšia sa tým, s akými biologickými procesmi pracujú a na základe toho aj tým, čo považujú za ortológy. Výber vhodnej metódy je vo veľkej miere závislý od účelu, na ktorý dáta požadujeme. Nedá sa jednoznačne povedať, že niektorá z metód je vo všeobecnosti najlepšia alebo jediná vhodná. Vybrali sme metódy, ktoré sa od seba dostatočne odlišujú, ako reprezentantov niekoľkých rôznych prístupov ku problému.

1.4.1 OrthoParaMap

Implementovaný ako sada skladajúca sa z trojice programov - DiagHunter, OrthoMap a ParaMap [2]. Prvým krokom je identifikácia homologických regiónov vzájomným porovnávaním dvojice genómov. Týmto spôsobom získame mapu, kde diagonálne čiary reprezentujú podobné úseky medzi dvomi organizmami. Jeden úsek môže byť zarovnaný k viacerým v druhom genóme, v tom prípade ho bude prekrývať niekoľko rôznych čiar. Ďalej sa pracuje s jednotlivými skupinami génov, vypočíta sa fylogenický strom pre množinu génov a namapuje sa na porovnanie genómov. Spätným mapovaním génov na fylogenický strom sa hľadajú uzly, v ktorých pravdepodobne došlo ku duplikácií jedného génu alebo väčšej skupiny. Všetky tri časti sú implementované v jazyku

Perl.

1.4.2 RIO (Resampled Inference of Orthologs)

Vstupom je sekvencia, ku ktorej chceme nájsť ortológy v iných organizmoch. Tá sa zarovná k už existujúcemu zarovnaniu rodiny génov, ktorá je sekvencii najpodobnejšia. Zarovnanie je tiež súčasťou požadovaného vstupu. Na základe sekvencií v zarovnaní sa skonštruuje strom pomocou postupného spájania najbližších susedov. Konštrukcia stromu sa robí viacnásobne vždy pre inú podmnožinu stĺpcov zarovnania, čím môže vzniknúť viacero rôznych stromov. Pre vyhodnotenie najbližších uzlov sa počíta vzdialenostná matica s využitím modelu pre substitúcie aminokyselín. Ako model slúži BLOSUM alebo PAM matica.

PAM(Point Accepted Mutation) matice sa používajú pri hľadaní zarovnania. Berú do úvahy pravdepodobnosti vzniku mutácie kódovania jednej aminokyseliny na druhú a to, ako často dochádza k zmene pri danej dvojici aminokyselín. Keďže niektoré aminokyseliny majú relatívne podobné vlastnosti, k niektorým mutáciám dochádza častejšie ako k iným. Číslo pri PAM matici udáva substitučné pravdepodobnosti pri danom počte mutácií na každých sto aminokyselín. Pri jednej aminokyseline môže pritom dôjsť k viacerým mutáciám.

Výsledný strom sa zakorení tak, aby sa minimalizoval počet duplikácií. Uzly sa vyhodnotia ako duplikačné alebo speciačné na základe fylogenetického stromu, ktorý je potrebné mať k dispozícii. Porovnávané sekvencie a do akej miery ich budeme považovať za ortológy je určená z pomeru stromov, v ktorých boli ortológmi. Systém je implementovaný ako skript v Perle, využíva niekoľko programov vytvorených v jazykoch C a Jave [20].

1.4.3 Inparanoid

Databáza ortológov pre niekoľko genómov. Rozlišuje paralógy na dve skupiny podľa toho, či duplikácia nastala pred alebo po speciácií daných organizmov. Podľa toho ich delí na vonkajšie (duplikácia, ktorou vznikli prebehla pred speciáciou) a vnútorné paralógy (duplikácia, ktorou vznikli prebehla po speciácii) [15]. Vytvára skupiny ortológov, jadro skupiny nájde pomocou najlepšej vzájomnej podobnosti medzi dvojicou homológov. Na určenie podobností využíva nástroj Blast. Tieto určí ako pár ortológov

a následne rozširuje skupinu o vnútorné paralógy. Každý gén z dvojice priradí ako ortológ vnútorným paralógom druhého génu. Umožňuje teda, aby jeden gén mal niekoľko ortológov v rovnakom organizme.

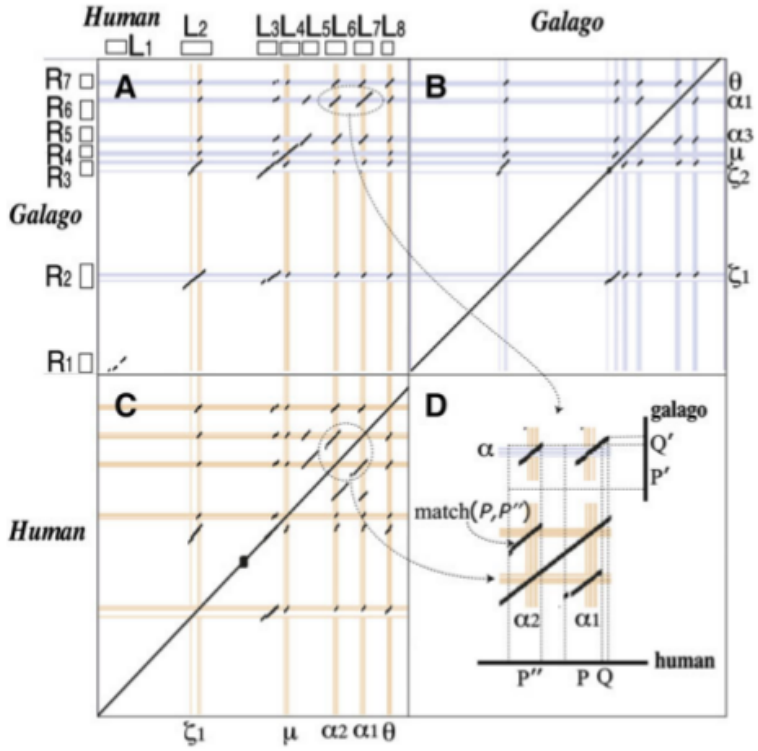
1.4.4 COG/KOG

Databáza ortológov na prokariotoch (COG) a eukariotoch (KOG). Pracuje s proteínovými doménami nachádzajúcimi sa v génoch [18]. Doména je časť sekvencie proteínu, ktorá sa nachádza vo viacerých rôznych proteínoch so samostatnou trojdimenzionálnou štruktúrou málo závislou od zvyšnej časti proteínu. Na začiatku sa rozšírené domény v proteínoch zamaskujú. Sekvencie génov sa porovnajú a vytvoria sa trojuholníky navzájom podobných sekvencií z trojice rôznych genómov. Následne sa trojuholníky so spoločnou hranou spoja do skupín. Sekvencie sa zoskupujú tak aby v nich bola zachovaná doménová štruktúra. Samostatne sa pracuje s génmi obsahujúcimi časté domény.

1.4.5 CHAP2 (The Cluster History Analysis Package)

Rozlišuje dva typy ortológie podľa pôvodu. N-ortológy, ktoré vznikajú speciáciou, a X-ortológy, ktoré vznikajú konverziou [16]. Pri konverzii je nahradená časť génu iným, čím sa gén stane hybridom dvoch génov. Stane sa teda ortológom k tým génom, ku ktorým bol ortológom skopírovaný gén. Algoritmus začína s grafom homologických úsekov. Hrany spájajú časti genómu, ktoré sú podobné. Podobné úseky určí porovnaním celých genómov navzájom a vytvorením mapy, jej ukážka je na obrázku 1.4.5.

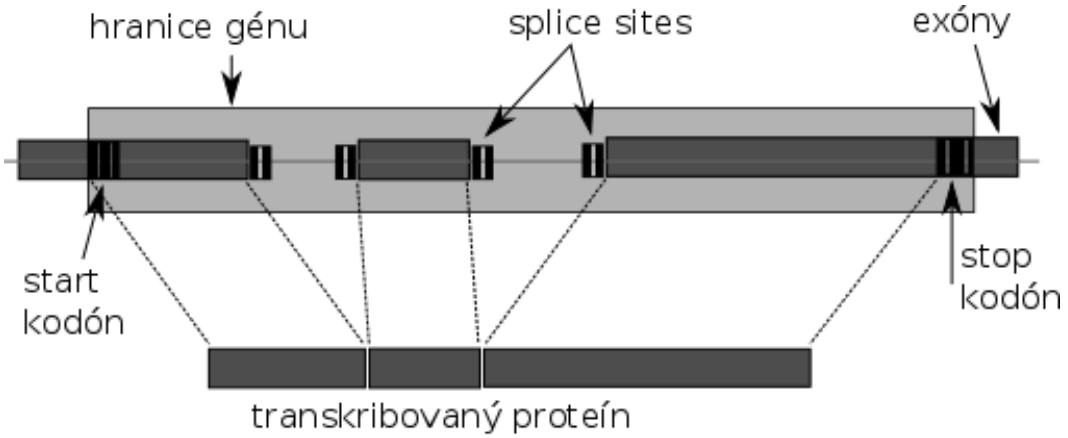
Z grafu sú potom odstránené hrany vnútri organizmu, ktoré predstavujú paralógy. Na zvyšných hranách sa nájde maximálne párenie predstavované X-ortológmi. Potom sa zarátajú konverzie, určené iným nástrojom, a na ich základe sa určia N-ortológy. Vo výsledku sú častiam génu priradené ortologické gény. Každá časť môže byť ortológom ku viacerým génom, keďže paralógy po speciácií sa oba považujú za ortológ. Program ponúka aj vizualizáciu, ktorá týmto spôsobom znázorňuje vývoj génov na danej skupine organizmov.



Obrázok 1.3: Mapa podobných úsekov pre dvojicu genómov, časť A zobrazuje medzidruhové zarovnanie, časť B a C zarovnanie samého k sebe, diagonálne čiary predstavujú podobné úseky. Zdroj: [16]

1.5 Štruktúra génu

Za gén považujeme úsek reťazca DNA kódujúci nejaký produkt, najčastejšie proteín. Je základnou funkčnou jednotkou pri evolúcii organizmov. Jednotlivé gény sa môžu prekrývať. V cicavcoch sú tvorené sekvenciami dĺžok rádovo tisícov báz, celkový počet génov napríklad v človeku je vyše 20 000 (dolný odhad), hoci jednotlivé odhady sa v rôznych štúdiach značne líšia a uvadzajú až násobky tohto čísla. Počet génov výrazne zvyšuje alternatívny zostrih. Samotný gén je tvorený z intrónov a exónov. Exóny sú kódujúce časti génu, spolu pokrývajú len niekoľko percent z celkovej dĺžky genómu. V priemere je v géne niekoľko až niekoľko desiatok exónov, ktorých dĺžka je rádovo sto báz. Intrón spravidla začína dvojicou báz GT (donor) a končí bázami AG (akceptor). Pri transkripcii génu sú intróny so sekvencie odstránené a do translácie vstupujú iba spojené exónové časti tak ako nasledujú v géne. Pred samotným génom sa v DNA sekvencii vyskytujú takzvané promoter regióny ovplyvňujúce expresiu daného génu. Obsahujú špecifické sekvencie (napr. TATA oblasť, CG oblasť, CAAT oblasť) ktoré



Obrázok 1.4: Základná štruktúra génu.

umožňujú začiatok transkripcia na danom mieste. Základná štruktúra génu je zobrazená na obrázku 1.5

Bázy tvoriace DNA sekvenciu sa v molekule nachádzajú v komplementárnych pároch, máme teda dva vlákna pričom gény môžu byť kódované na ľubovoľnom z nich. Čítanie reťazca báz na druhom vlákne prebieha v opačnom poradí. Teda ak máme číselne určenú pozíciu génu na chromozóme, v závislosti na vlákne môže na jednom okraji gén začínať a pokračovať smerom k druhému okraju, alebo naopak na danom mieste končiť. Funkcia génu do značnej miery vyplýva z tvaru výsledného proteínu, teda trojdimenzionálnej štruktúry vytváranej aminokyselinami. Aj pri väščích odlišnostiach v kódujúcej sekvencii oproti príbuzným génom, ak je zachovaná jeho štruktúra, gén bude pravdepodobne stále plniť svoju funkciu. Inak povedané, nie všetky zmeny v sekvencií majú rovnako veľký dopad na funkčnosť kódovaného génu.

1.6 Mutácie génov

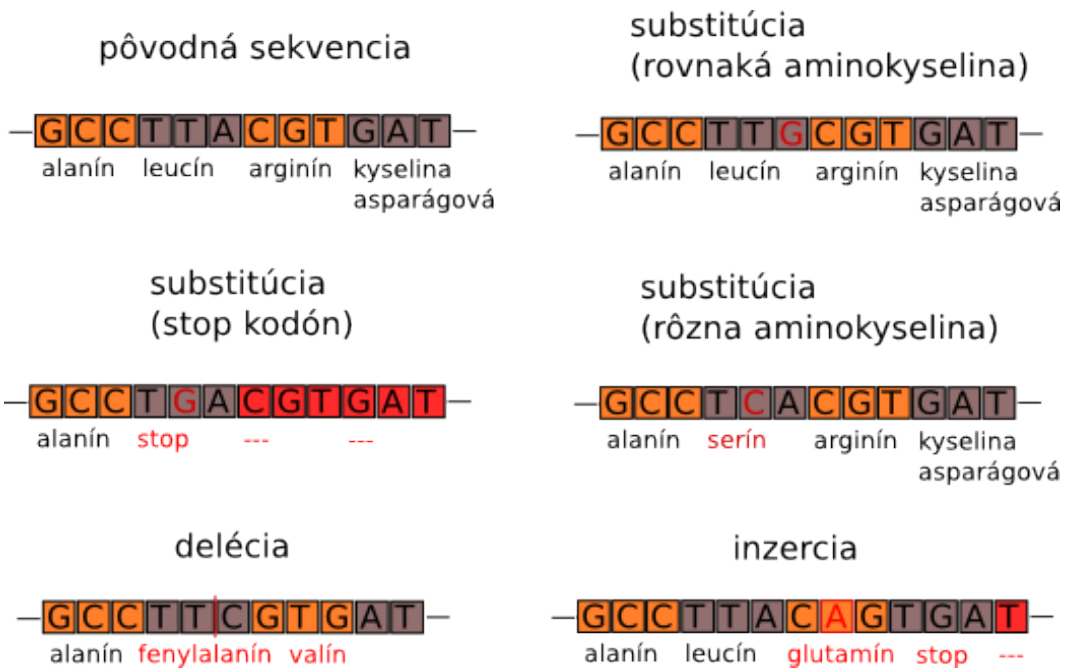
Gény nie sú nemenné a postupom času podliehajú zmenám - mutáciám, ktorými vznikajú nové zmenené verzie génu. Mutácia je proces, ktorým dochádza k zmene sekvencie DNA kódujúcej genetickú informáciu. Modifikácie DNA sa delia na viacero rôznych typov. Najjednoduchšie a zároveň najčastejšie zmeny sú jednobázové substitúcie. Tie predstavujú prepísanie jednej bázy nejakou inou bázou. Takáto zmeny môže byť v princípe troch rôznych typov v závislosti od následkov zmeny na transkribovaný proteín. Prvým typom je zmena kódujúcej aminokyseliny na nejakú inú. Druhým typom

je zachovanie aminokyseliny kódovanej danou bázou, čo je možné vďaka tomu, že počet aminokyselín je menší ako počet možností všetkých kódovaní pre ne. Tretím typom je zmena, po ktorej bude kódovaný namiesto pôvodnej aminokyseliny stop kodón. Pri transkripcii sa preto na danom mieste prepis do proteínu ukončí, čo s veľkou pravdepodobnosťou znefunkční kódovaný proteín. Vidíme teda, že zmena jedinej bázy môže mať rôzne následky počnúc malou až žiadnou zmenou a končiac úplným porušením génu, ktorý kódovala.

Pravdepodobnosť substitúcie jednej bázy na inú je odlišná pre jednotlivé dvojice. Tento jav je spôsobený chemickou štruktúrou báz. Cytozín a Tymín patria medzi pyrimidíny. Oba obsahujú šesť-členný kruh uhlíka, v ktorom sú atómy 1 a 3 nahradené dusíkom. Podobne Guanín a Adenín sú obidva puríny. Sú tvorené pyrimidínovým a imidazolovým kruhom. Imidazolový kruh je päť-členný a dva jeho atómu sú nahradené dusíkom. Pri mutáciách báz častejšie dochádza k náhrade dvoch purínových/pyrimidínových medzi sebou navzájom - tranzícia, ako ku zmene z purínu na pyrimidín a naopak - transverzia.

Ďalšími typmi zmien okrem substitúcií sú inzercie a delécie. Dochádza pri nich k pridávaniu respektíve odstraňovaniu časti sekvencie. Obidva typy zmien spolu jednotne označujeme indely. Pokiaľ počet báz, ktoré modifikujú, nie je násobkom troch, nastáva výrazne riziko poškodenia génu. To je dôsledok spôsobu akým sú v DNA kódované aminokyseliny vo forme trojíc báz. Bázy nasledujúce po takomto indele sa začnú čítať s posunom a budú tvoriť úplne odlišnú postupnosť kodónov, teda aj výsledný proteín bude mať odlišnú štruktúru a stratí svoju pôvodnú funkciu. Hrozí zároveň predčasné ukončenie transkripcie pri narazení na stop kodón, ktorý bol v pôvodnom géne súčasťou dvojice kodónov, ale posun spôsobil odlišné čítanie sekvencie. Príklady substitúcií aj inzercie a delécie sú na obrázku 1.5.

Zdvojenie časti sekvencie sa nazýva duplikácia. Je to veľmi významný typ mutácie. Predstavuje možnosť vývinu nových génov. Duplikovaná sekvencia génu môže nadobudnúť novú funkciu. Mutácie odohrávajúce sa na kópií nespôsobia znefunkčnenie kódovaného génu a teda môže dochádzať k rýchlejšiemu hromadeniu modifikácií. Vďaka existencii druhej kópie génu nie je negatívne ovplyvnená schopnosť jedinca na prežitie a šírenie zmien do ďalšej generácie. Týmto spôsobom vznikajú v genóme skupiny paralógov. Iným príkladom mutácie väčšieho rozsahu je translokácia. Translokácia



Obrázok 1.5: Typy mutácií jednej bázy a ich dopady na výsledný proteín.

je presun časti sekvencie jedného chromozómu na nejaký iný chromozóm. Ak dôjde presunom k rozdeleniu génu, funkcionalita daného génu je narušená.

Vymenované typy mutácie sa odohrávajú s rôznou pravdepodobnosťou, avšak celkovo akákoľvek mutácia je veľmi zriedkavý jav. Pri kopírovaní DNA v bunke dochádza k jednej mutácii na každých približne 50 miliónov báz, čo pri dĺžke ľudského genómu okolo troch miliárd báz znamená v priemere 120 mutácií pre každú novú bunku. To je 60 mutácií na dĺžku DNA sekvencie násobené dvomi pre diploidnú bunku obsahujúcu pár z každého chromozómu. Väčšina týchto mutácii sa odohrá v nekódujúcich častiach DNA sekvencie a nemá vplyv na kódované gény. Takéto zmeny sa prenášajú ďalej, lebo nepôsobia na organizmus negatívne.

Zmeny nekódujúcej oblasti majú väčšiu šancu udržania sa do ďalšej generácie. Časti kódujúce gény sa menia pomalšie ako nekódujúce úseky. Je to spôsobené tým, že takéto zmeny sa prejavia navonok a vplýva na ne selekcia. Väčšina zmien nie je pozitívna, a teda znižuje šance jedinca, ktorý zmenu prenáša, na prežitie. Takisto sú častejšie zmeny v kódujúcich oblastiach génov, ktoré nemenia jeho štruktúru a teda zachovávajú pôvodnú funkciu génu. Časti sekvencie kódujúce dôležité úseky génu pre jeho správnu funkcionalitu sa takmer vôbec nemenia počas vývinu druhu. Častejšie sú zmeny medzi aminokyselinami, ktoré majú podobné vlastnosti a nemenia trojdimenzionálne uspo-

riadanie výsledného proteínu.

1.7 Viacnásobné zarovnania

Jedným z používaných spôsobov na reprezentáciu homologických oblastí medzi genómami sú zarovnania. Viacnásobné zarovnanie je zarovnanie sekvencií dvoch alebo viacerých DNA sekvencií k referenčnej sekvencii. Ku každej časti referenčnej sekvencie je zarovnaný najviac jeden úsek inej sekvencie. Naopak to ale neplatí, rovnaký úsek zarovnanej sekvencie môže byť priradený viacerým miestam v referenčnej sekvencii.

Celogenómové zarovnanie sa skladá z blokov, pričom každý predstavuje viacnásobné zarovnanie nejakej časti genómu. Blok je tvorený referenčnou sekvenciou a k nej priradených sekvencií z genómov ostatných organizmov. Z každého organizmu obsahuje blok nanajvýš jeden úsek, ale z niektorých nemusí existovať žiaden dostatočne podobný úsek. To znamená, že každý blok obsahuje minimálne dve alebo viac sekvencií, každú z iného genómu. Každá sekvencia je určená chromozómom na ktorom sa nachádza a pozíciou na chromozóme, pričom táto informácia je súčasťou bloku. Referenčné úseky v jednotlivých genómoch sú navzájom disjunktné, niektoré časti genómu nemusia byť priradené v žiadnom bloku.

```
##maf version=1 scoring=autoMZ.v1

a score=3753.0
s ref.chr1      554846      44  +   13198656
ACGATGCATGCT-ATCGTAGTCGATGCTGATCGTATGCGTAGTCG
s spec1.chr1    32141       40  +   593735616
AGGATG---ACTCATGGTAGT--ATGCTGATCATATGCGAAGTCG

a score=3753.0
s ref.chr1      554899      34  +   13198656
ACGACAT------ACAGTAG--ACAAGAGACACTAGACGAGG
s spec1.chr1    312156      40  +   593735616
TCGACGACGATGTACAGTAG--ACAGGAGACAATAGTCGGGG
s spec3.chr1    465468      40  +   46549887
ACGTGATCAGG--CCAGTAGGGACCAGAGACGCTAGACGATG
```

Obrázok 1.6: Ukážka viacnásobného zarovnania.

Zarovnané sekvencie sú na väčšine pozícií zhodné, ale zároveň obsahujú rozdiely malého rozmeru. Substitúcia bázy sa prejaví odlišnosťou na dotknutej pozícii. Takto sa nedajú riešiť krátke inzercie a delécie. Sekvencie sa môžu na začiatku a konci zhodovať, ale v určitej časti jedna z nich obsahuje niekoľko báz navyše. Tie sa nemôže zarovnať k žiadnej báze v reťazci s nižším počtom báz. Z tohoto dôvodu je štvor-písmenková abeceda báz rozšírená o ďalší znak - medzeru. Z pohľadu referenčného genómu medzera v referenčnom reťazci znamená deléciu a v iných reťazcoch inzerciu. Všetky reťazce bloku sú takto rovnako dlhé, ale môžu sa líšiť v dĺžke chromozómu, ktorú pokrývajú, pretože medzery ovplyvňujú počet báz.

Príklad zarovnania troch organizmov je na obrázku 1.6. V prvom bloku je k referenčnému organizmu zarovnaný iba jeden druh. Na druhom mieste v dvojici reťazcov vidíme substitúciu báz. Na siedmej až deviatej báze je inzercia z pohľadu druhého reťazca prípadne delécia z pohľadu referenčného reťazca. V nasledujúcom bloku sú obsiahnuté všetky tri organizmy. Prvý riadok je hlavička súboru.

Kapitola 2

Systém na hľadanie ortológov

V predchádzajúcej kapitole sme uviedli doterajšie prístupy k problému nájdenia ortológov. V tejto kapitole opíšeme nami navrhnutý systém. Náš systém nemá za cieľ nájsť všetky ortologické gény. Mal by nájsť také gény, pri ktorých je s veľmi vysokou pravdepodobnosťou zachovaná ich funkcia. Okrem podobnosti samotnej sekvencie nás preto zaujímajú aj typy mutácií, ku ktorým v nej došlo. Ako sme ukázali v predchádzajúcej kapitole, rôzne mutácie majú rôzne veľký dopad na kódovaný proteín. To znamená, že sa líši aj miera, akou sa môže narušiť pôvodná funkciu proteínu.

2.1 Základná stavba systému

Podobne ako iné systémy musíme najprv získať sadu homológov a z nich následne vybrať hľadané ortológy. Náš systém získa homologické oblasti v iných genómoch z viacnásobného zarovnania. To poskytne pre každý gén nanajvýš jednu oblasť v genóme iného organizmu, ktorá je sekvenciou najpodobnejšia a môže predstavovať ortológ. Vo vstupnom zarovnaní nemusia byť obsiahnuté celé genómy organizmov. Pre referenčný genóm zarovnania bude potrebné mať navyše aj anotácie génov. Tie sú druhým vstupom systému.

Poznáme teda pozície génov v referenčnom genóme, a pre jednotlivé časti genómu poznáme im prislúchajúce časti v iných genómoch zo zarovnania. Na základe toho môžeme premapovať pozície génov na úseky v ďalších organizmoch pomocou zarovnania. Tieto úseky by mohli predstavovať príslušné gény v jednotlivých organizmoch. Pri premapovaní génov pracujeme okrem referenčného vždy postupne s

jedným porovnávaným organizmom. Odkedy došlo k speciácií zo spoločného predka v oboch genómoch, referenčnom aj porovnávanom, dochádzalo k mutáciám. Oblasť prislúchajúca premapovanému génu v porovnávanom organizme preto nemusí byť už funkčným génom. Aj v prípade, že génom je, môže byť pozmenený na výraznej časti svojej dĺžky.

Chceme vytriediť tie gény, ktoré majú svoju štruktúru zachovanú. Požadujeme, aby bola v zarovnanom géne rovnaká postupnosť a pozície exónov, z ktorých sa transkribuje výsledný proteín. Mutácie, ktoré by mohli takúto štruktúru narušiť, chceme v sekvencii vyhľadať a zaznamenať. Gény triedime pomocou sady niekoľkých filtrov, z ktorých každý overuje nejaké podmienky správnej štruktúry. Jednotlivé filtre overujú samotné zarovnanie alebo prítomnosť mutácií, ktoré by pozmenili štruktúru exónov v géne.

Listing 2.1: Stavba systému - pseudokód

```
1  corthy(parametre[]) {
2     projekt = parametre[projekt]
3     sady_genov = parametre[sady_genov]
4     klaster = parametre[klaster]
5     maf_priecinok = parametre[maf_priecinok]
6     export_priecinok = parametre[export_priecinok]
7     organizmy = parametre[organizmy]
8
9     if (nahraj_geny) {nahravanie_genov(sady_genov)}
10    if (klastruj_geny) {klustrovanie(klaster, sady_genov)}
11    if (filtruj_geny) {filtrovanie(klaster, organizmy,
         maf_priecinok)}
12    if (exportuj_geny) {exportovanie(klaster, organizmy,
         export_priecinok)}
13 }
```

Viacero génov z anotácie môže predstavovať niekoľko transkriptov nachádzajúcich sa v genóme na rovnakom mieste. S takýmito skupinami génov niekedy chceme pracovať ako s jediným génom. Proces, ktorých takéto gény spájame do skupín nazývame klastrovanie.

Program vykonáva niekoľko v princípe odlišných funkcií, z ktorých sa skladá celkový

proces. Jednotlivé fázy spracovania sú: nahratie anotácií génov do databázy, vytvorenie klastrov nad týmito génmi, filtrovanie a zaznamenanie chýb vo zvolenom zarovnaní a export čistých génov bez chýb podľa určitých kritérií.

Táto základná štruktúra je zobrazená v listingu 2.1. Každá z fáz je podrobne opísaná v samostatnej sekcii.

2.2 Inkrementačný model systému

Systém by mal byť schopný pracovať s priebežne sa meniacimi vstupnými dátami. Môže sa meniť ako zarovnanie, tak aj anotácie génov. Preto je potrebné umožňiť pridávanie viacerých množín anotácii génov. Po pridaní novej množiny je následne potrebné prepočítať gény, ktoré obsahuje. Tie sa musia overiť vo všetkých organizmoch, pre ktoré chceme získať výsledné dáta.

Podobne ako anotácie chceme vedieť pridávať aj nové zarovnania. Pri pridaní zarovnania pre nový organizmus sa spracujú existujúce sady génov. V obidvoch prípadoch nie je potrebné meniť už existujúce predpočítané dáta, iba sa k nim pridajú nové. Výsledky pre rôzne vstupné dáta sú vypočítavané navzájom nezávisle.

Môžeme teda postupne pridávať nové dáta a vždy dopočítať iba príslušnú časť. Výsledky pre predošlé dáta zostanú v systéme zachované. Jednotlivé množiny anotácií génov môžeme zhlukovať do klastrov. Týchto klastrovaní môže byť niekoľko, každé nad rôznou skupinou spomedzi dostupných anotácií. Pri procese klastrovania nie je potrebná žiadna práca so zarovnaniami.

Samotné premapovanie génov a overovanie ich štruktúry môže trvať dlhšiu dobu, ale je ho potrebné vykonať len raz. Získanie výsledkov zo systému by však malo byť pomerne rýchle, bez toho aby sa každý gén overoval. Preto budú výsledky predpočítané v databáze a pri výstupe sa použije iba ich podmnožina na základe konkrétnych parametrov požiadavky. Tomu je prispôsobený návrh databázy, ktorá je podrobne popísaná v nasledujúcej sekcii.

2.3 Štruktúra databázy

Program intenzívne využíva pri výpočte databázu. Jej stavba je zobrazená na obrázku 2.1. Do databázy sa ukladajú všetky potrebné informácie o spracovaných génoch a

po spracovaní dát sa v nej nachádzajú dáta nevyhnutné na získanie výsledkov pre požadované hodnoty.

Štruktúru databázy môžeme rozdeliť na tri časti, ktorých obsah sa plní dátami v jednotlivých fázach. Každá z týchto troch fáz môže bežať samostatne bez potreby spustenia ostatných. Množiny tabuliek, do ktorých sa zapisujú nové dáta, sú medzi fázami disjunktné.

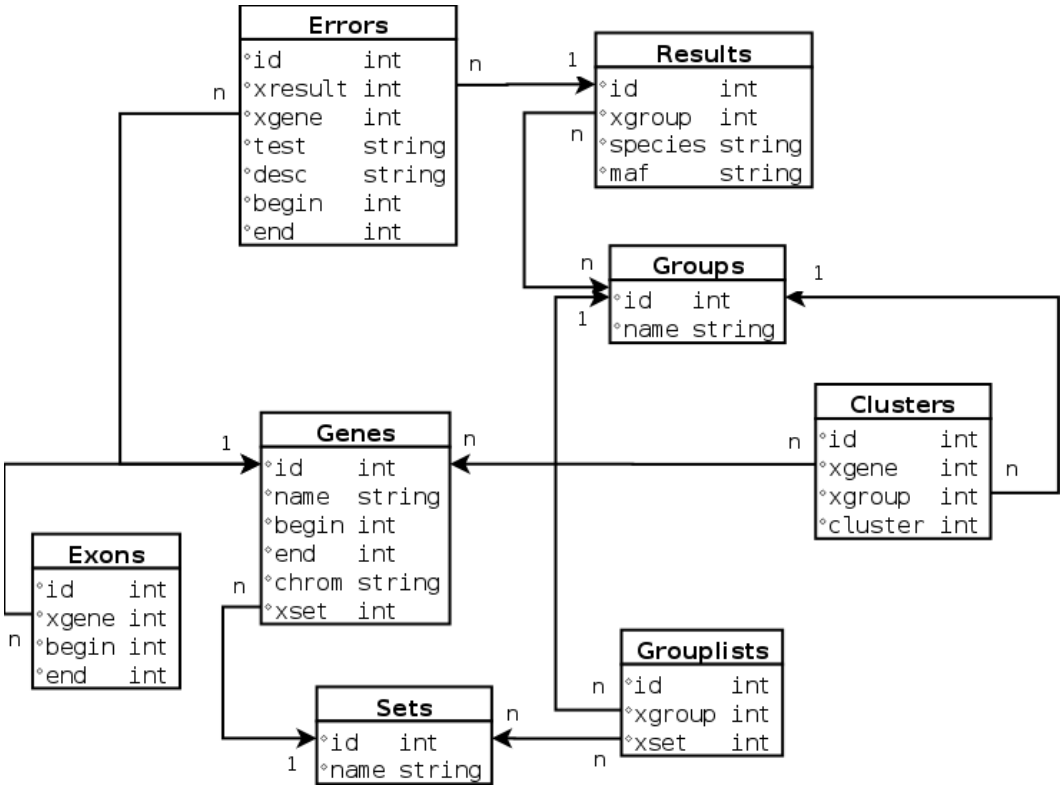
Všetky stĺpce tabuliek obsahujúce indexy sú pomenované jednotným systémom. Ich názov sa skladá z názvu odkazovanej tabuľky, ktorému predchádza písmeno 'X' pre vizuálne odlíšenie takýchto parametrov.

V prvej fáze sa načítavajú zo súboru pozície génov a ich štruktúra. Súborov môže byť viacero a každý predstavuje jednu množinu génov. Aby sme vedeli oddeliť gény do jednotlivých množín, v tabuľke Sets ukladáme záznam o každom súbore. Hranice a pozície génov sa nachádzajú v tabuľke Genes spolu s indexom ich zdroja v tabuľke Sets. Každý gén sa skladá z niekoľkých exónov, ku ktorým si spolu s ich začiatkom a koncom pamätáme v tabuľke Exons aj index na príslušný gén. Toto delenie nám umožňuje jednoduchý prístup ako ku génom, tak aj ku exónom, z ktorých sa skladajú.

Druhá fáza zhlukuje gény do klastrov, počas nej sa napĺňajú tabuľky Groups, Grouplists a Clusters. Prvá z tabuliek uchováva záznamy o jednotlivých klastroch, druhá mapuje klastre na množiny génov, ktoré sú v nich obsiahnuté. V poslednej z trojice tabuliek sa nachádzajú indexy génov s klastrovým číslom, ktoré je rovnaké pri génoch nachádzajúcich sa v tom istom klastri. Každý záznam okrem indexu na gén obsahuje aj index do tabuľky Groups, vďaka čomu je možné mať súčasne viacero rôznych nezávislých klastrovaní nad génmi.

Posledná fáza zapisujúca do databázy filtruje gény a ukladá záznamy o nájdených rozdieloch oproti očakávanej štruktúre génov. Zapisuje do tabuľky Results záznam o výsledkoch pre daný druh, zarovnanie a klastrovanie. Následne pre všetky gény v danom klastrovaní uloží do tabuľky Errors chyby jednotlivých druhov s ich pozíciou a indexom do tabuľky Results. Takto vieme neskôr vybrať výsledky pre rôzne zarovnania, druhy, aj filtrovať iba určitý druh chyby. Vďaka pozíciám chyby v kombinácií s pozíciou génu na chromozóme v tabuľke Genes vieme dotazovať aj chyby podľa ich relatívnej pozície v géne.

Program umožňuje prácu s viacerými nezávislými sadami dát pomocou oddelených



Obrázok 2.1: Návrh databázy, použité tabuľky a vzťahy medzi nimi.

projektov. Slúži na to hlavná tabuľka Projects, v ktorej sú definované ich názvy. Všetky vyššie zmienené tabuľky vždy patria ku konkrétnemu projektu, ich názvu predchádza názov projektu nasledovaný podčiarkovníkom. Teda napríklad tabuľka Genes v projekte Turtle bude mať názov *Turtle_Genes*. Pri odstránení projektu sa okrem záznamu v tabuľke Projects zároveň zmažú všetky tabuľky, ktoré sú jeho súčasťou. Umožňuje to prácu viacerých osôb súčasne bez toho, aby sa akokoľvek ovplyvňovali navzájom vo svojej činnosti.

Všetky tabuľky vytvára program automaticky pri spracovaní dát. Potrebný je len prístup ku databáze, ktorý sa načítava z konfiguračného súboru. Pomocou prepínača pri spúšťaní je možné nastaviť iný ako predvolený konfiguračný súbor a použiť prihlasovacie údaje z neho.

2.4 Nahrávanie génov

Prvou fázou celého systému je spracovanie anotácií génov. Ako vstup sa rozoberá súbor génov vo formáte GenePred a overuje sa korektná štruktúra génov. Súbor po-

zostáva z riadkov, kde každý predstavuje jeden gén a obsahuje informácie o jeho pozícii, počte exónov, pozíciach exónov a ďalšie informácie. Požadujeme, aby sa exóny v géne neprekrývali a zároveň aby koniec exónu nasledoval za jeho začiatkom. Ďalej prepočítavame dĺžku exónov, ktoré sa nachádzajú medzi začiatkom a koncom génu, túto informáciu si aj zaznamenávame. Vyraďujeme tie gény, pri ktorých je hodnota nižšia ako dĺžka dvoch kodónov. Cieľom je vytvoriť množinu génov, s ktorou sa dá pracovať v ďalších fázach.

Pre každý gén počítame aj jeho poradie v danej množine pre daný chromozóm. Vždy budeme pracovať s celým chromozómom určitej množiny génov a teda načítame celý rozsah indexov bez medzier. Pri výbere viacerých množín naraz pre ne vieme vypočítať offsety vyjadrujúce počet génov v predošlých množinách. Každému génu takto vieme jednoznačne priradiť číslo ako súčet offsetu jeho množiny a poradia v množine bez ohľadu na poradie, v ktorom gény vyberieme. Umožní nám to neskôr prístup v konštantnom čase z exónov ku génu, ktorého je exón súčasťou. Stačí si pamätať poradie génu pri exónoch a spoločné dáta priraďovať do poľa na príslušný index.

2.5 Klastrovanie génov

V genóme sa nachádzajú kódujúce úseky, ktoré sú súčasťou viacerých génov. Jedná sa buď o prekrývajúce sa gény alebo o jeden gén s viacerými alternatívnymi zostrihmi. V obidvoch prípadoch dochádza k tomu, že určitá časť ich exónov sa navzájom prekrýva. Klastrovanie predstavuje rozdelenie množiny génov na podmnožiny, ktorých všetky exóny sú navzájom disjunktné. Z nájdených ortológov potom môžeme vybrať pre každý klaster práve jeden gén. Vyberáme na základe počtu organizmov, v ktorých sa daný ortológ nachádza a sekundárne potom podľa kódovej dĺžky génov v klastri.

Použitý algoritmus pre klastrovanie sa skladá z niekoľkých krokov. Najprv sa z exónov vytvorí sieť, kde každý exón môže mať až štyroch susedov, ktorých označujeme ako horný, dolný, ľavý a pravý. Získame ju vzájomným spájaním exónov vertikálne a horizontálne. Dostaneme graf, ktorého jednotlivé komponenty predstavujú hľadané klastre génov. Prechádzame cez exóny a označujeme ich, pričom rovnakým číslom označíme všetky exóny v danom klastri. Následne tieto čísla namapujeme na gény a

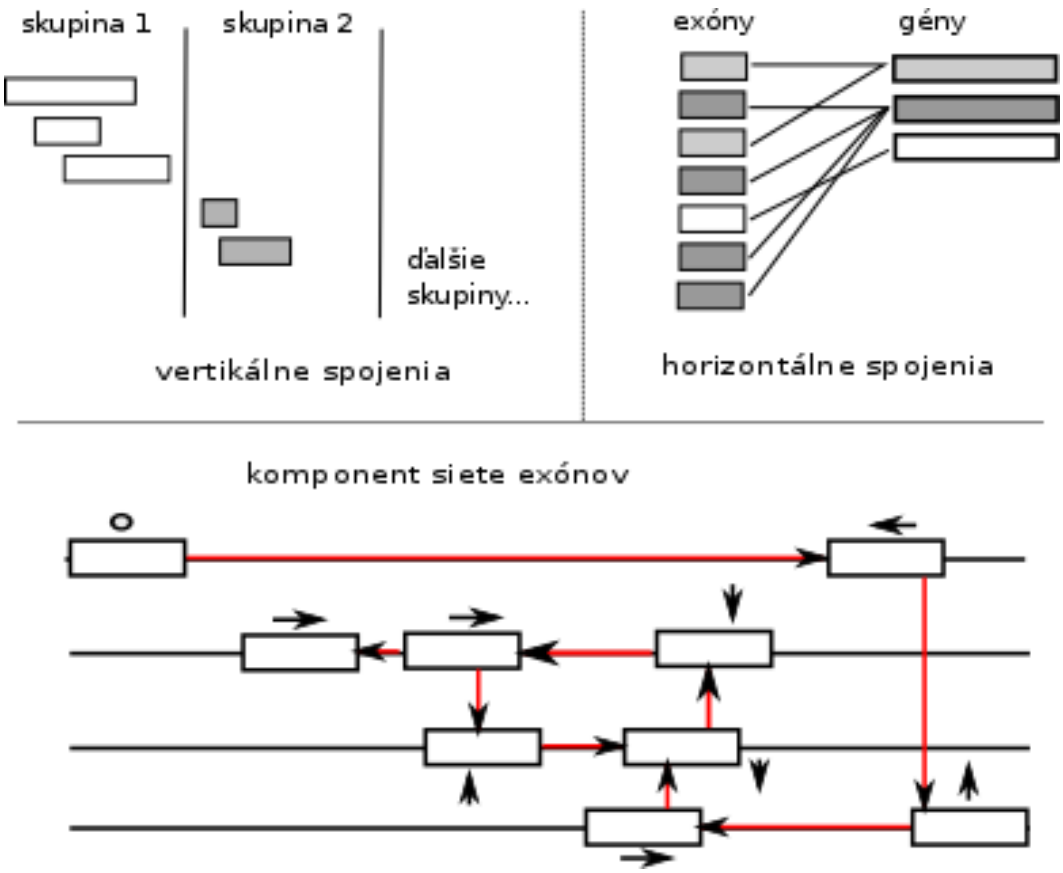
získame finálne dáta.

Podrobnejšie teraz popíšeme konštrukciu siete exónov. Dotazom z databázy dostaneme zoznam exónov usporiadaný vzostupne podľa ich začiatku. Prechádzame exóny a zhlukujeme ich do skupín. Začneme prvým exónom, ktorý bude základom pre prvú skupinu. Jeho koniec bude vytvárať hranicu skupiny. Ak ďalší exón v zozname začína za hranicou aktuálnej skupiny, vytvorí sa nová skupina a daný exón sa stane jej základom. V opačnom prípade, ak dochádza k prekryvu so skupinou, zaradí sa exón do tejto skupiny. Prepojí sa smerom nahor s predchádzajúcim exónom a ten naopak smerom nadol s ním. Ak je jeho koniec väčší ako hranica skupiny stane sa novou hranicou. Takýmto spôsobom prejdeme celý zoznam a vytvoríme vertikálne prepojenia v sieti. Popri tom si zároveň pre každý exón uložíme index génu, ktorého je súčasťou.

Následne vytvárame horizontálne prepojenia. Máme zoznam génov, ktorých parameter inicializujeme nulovými hodnotami. Prechádzame zoznam exónov a aktuálny index vždy uložíme ako parameter ku génu, na ktorý ukazuje index vytvorený v predošlom kroku. Tým máme v poli génov vždy uložený index na posledný prejdený exón, ktorý je jeho súčasťou. Ak priradením nahradíme nenulovú hodnotu, prepojíme starý index doprava s novým a naopak nový doľava so starým. Na konci máme pre každý exón vytvorené spojenia k susedom, ktoré vytvárajú graf s týmito spojeniami ako hranami.

Teraz si vytvoríme dve polia pre exóny, v jednom budeme ukladať klastrové číslo a v druhom jeden zo smerov v sieti. Znovu prechádzame zoznam exónov a zoskupujeme ich do klastrov. Keď už má nejaké číslo priradené, preskočíme daný exón, inak inkrementujeme aktuálne poradové číslo klasteru a priradíme ho. Rovnaké číslo chceme priradiť aj všetkým exónom nachádzajúcim sa v rovnakom komponente. Prechádzame graf podobným spôsobom ako bludisko, pričom chceme každou hranou prejsť práve dva krát a zároveň navštíviť všetky vrcholy. Pre každý vrchol si pri prvom prechode pamätáme vstupnú hranu do neho a daný smer si uložíme ako návratový. Ten použijeme, ak z vrcholu už nevedie žiadna nepoužitá hrana. Inak pokračujeme ľubovoľnou ešte nepoužitou hranou.

Ak vojdeme novou hranou do vrcholu s nastavenou návratovou cestou, vrátime sa rovnakou hranou hneď spätne, týmto spôsobom riešime cykly nachádzajúce sa v grafe. Na konci sa vrátime do počiatočného vrcholu a cestu uzatvoríme. Z každého exónu



Obrázok 2.2: Vizualizácia algoritmu. Zobrazená je idealizovaná situácia, graf exónov na reálnych dátach nemusí byť planárny

týmto spôsobom vyrazíme novou hranou v priemere maximálne dva krát vzhľadom na počet hrán v grafe a algoritmus jeho prechádzania. Postup vytvárania siete a nájdenia cesty v nej je na obrázku 2.2

Už nám stačí len jeden prechod poľom exónov a napamovanie klastrových čísle na príslušné gény. Celkovo tak prejdeme všetky exóny dva krát pri vytváraní siete, raz pri klastrovaní exónov a raz pri priradení výsledkov génom. Počas klastrovania exónov prechádzame každou hranou siete, tých je však nanajvýš dvojnásobne oproti počtu exónov. Celý algoritmus teda pracuje v lineárnom čase od počtu exónov. Dosiahli sme to vďaka vhodne zvolenej štruktúre grafu.

2.6 Filtrovanie génov

Najdôležitejšou súčasťou systému je vyhľadávanie chýb v štruktúre génov. V tejto fáze pracujeme s viacnásobným zarovnaním a prechádzame postupne súbory, ktoré ho tvoria. Zarovnanie očakávame v široko používanom formáte MAF. Spracúvame každý blok zarovnania, pričom si udržiavame množinu aktívnych génov a exónov, ktoré sa prekrývajú s našou pozíciou v genóme. Tak gény ako aj exóny máme v poli usporiadané podľa ich začiatku a pamätáme si indexy na pozíciu prvého a posledného vyhovujúceho zo zoznamu. Pre tieto gény overíme jednotlivé filtre a zaznamenáme prípadné chyby.

Predpokladáme, že jednotlivé bloky idú podľa poradia v referenčnom genóme a na základe toho vieme určiť chýbajúcu časť zarovnania pre aktívne gény. Následne prechádzame bázu po báze celý blok zarovnania a vykonávame filtre, pre ktoré potrebujeme poznať celú sekvenciu. Priebeh spracovania je zobrazený v listingu 2.2.

V tabuľke 2.1 je prehľad filtrov spolu s ich časovou zložitosťou. Zložitosť je ovplyvnená tým, čo v zarovnaní genómov sledujeme. Záleží ďalej od toho, či potrebujeme prejsť každý blok, alebo každú bázu každého bloku. Tiež závisí od toho, či sledujeme len hranice génov/exónov, alebo celú ich dĺžku. Na výpočet sú najnáročnejšími filtre pre čítací rámec a nezmyselné mutácie, kvôli potrebe výpočtu aktuálneho čítacieho rámcu pre každú bázu každého exónu.

Listing 2.2: Filtrovanie génov - pseudokód

```
filtrovanie(klaster, organizmy, maf_priecinok) {
   geny_z_databazy(klaster)
   foreach maf_subor in maf_priecinok
      foreach blok_zarovnania
         nacitaj_data

         foreach gen in aktivne_geny
            test_alignment
            test_synteny

         foreach baza_bloku
         foreach gen in aktivne_geny
            prepocitaj_frame
            test_start
            test_stop
            test_splice
            test_frame
            test_nonsense
}
```

Po aplikácii všetkých filtrov získame výslednú množinu ortológov medzi dvojicou organizmov. Vyradené gény, ktoré nie sú ortológy, sú označené filtrami, ktorých podmienky nespĺňali. Získané dáta je možné ďalej využiť pri štúdiu evolúcie daných organizmov, tvorbe fylogenetických stromov a iných činnostiach komparatívnej genomiky.

2.6.1 Zarovnanie a synténia

Základným filtrom je test existencie zarovnania. Bez toho nemajú ostatné filtre význam, pretože nemajú dáta, s ktorými by vedeli pracovať. Zarovnanie génu môže byť pokryté jedným alebo viacerými blokmi. Vyžadujeme pokrytie celého génu, pričom niekoľkobázové medzery medzi po sebe nasledujúcimi blokmi sú prípustné. Väčšie úseky bez zarovnania zaznamenávame do databázy a považujeme ich za chybu.

Tabuľka 2.1: Zložitosť filtrov

Filter	MAF súbor	štruktúra	časová náročnosť
Zarovnanie	bloky	gény	malá
Synténia	bloky	gény	malá
Hranice génu	bázy	gény	stredná
Hranice exónov	bázy	exóny	stredná
Čítací rámec	bázy	bázy exónov	veľká
Nezmyselné mutácie	bázy	bázy exónov	veľká

Pokiaľ je daný gén rozdelený do niekoľkých zarovnaných blokov, ďalším filtrom je potrebné overiť ich vzájomnú pozíciu v genóme. V prípade, že sa niektoré dva bloky nachádzajú na rôznych chromozómoch alebo opačných vláknach DNA, považujeme to za chybu. Takéto úseky nemôžu tvoriť v danom organizme funkčný gén, pretože gén musí byť tvorený jedinou súvislou oblasťou v rámci jedného chromozómu. Pokiaľ bloky nenasledujú za sebou, prekrývajú sa, alebo sú medzi nimi medzery väčšej veľkosti, tiež s tým ďalej pracujeme ako s chybou.

2.6.2 Hranice génu a exónov

Ďalšou vecou, ktorú overujeme v prípade existencie zarovnania, je štruktúra zarovnanej sekvencie. Požadujeme rovnaké pozície začiatku a konca génu aj jeho exónov. To znamená, že otestujeme triplety na daných pozíciach, a pre exóny dvojicu báz pred ich začiatkom a za ich koncom. Množina povolených sekvencií pre jednotlivé porovnania je v tabuľke 2.2. Pokiaľ sa hodnota líši od množiny prípustných hodnôt, považujeme to za chybu. Zaznamenáme jej pozíciu a zaznamenáme zároveň chybnú sekvenciu báz. Hodnoty pre triplety a dvojice bázy záležia od vlákna, na ktorom sa gén nachádza. Ak je na opačnom oproti referenčnému génu tak musíme použiť pre každú bázu komplementárnu bázu k nej. Jednotlivé bázy idú navyše v opačnom poradí, na konci génu je začiatok a jeho bázy postupujú smerom ku začiatku génu, kde je v tomto prípade koniec.

Tabuľka 2.2: Povolené sekvencie báz

pozícia	hodnoty - rovnaké vlákno	hodnoty - opačné vlákno
začiatok génu	ATG	CAT
koniec génu	TAA,TGA,TAG	TTA,TCA,TAC
začiatok intrónu	GT,GC	AC,GC
koniec intrónu	AG	CT

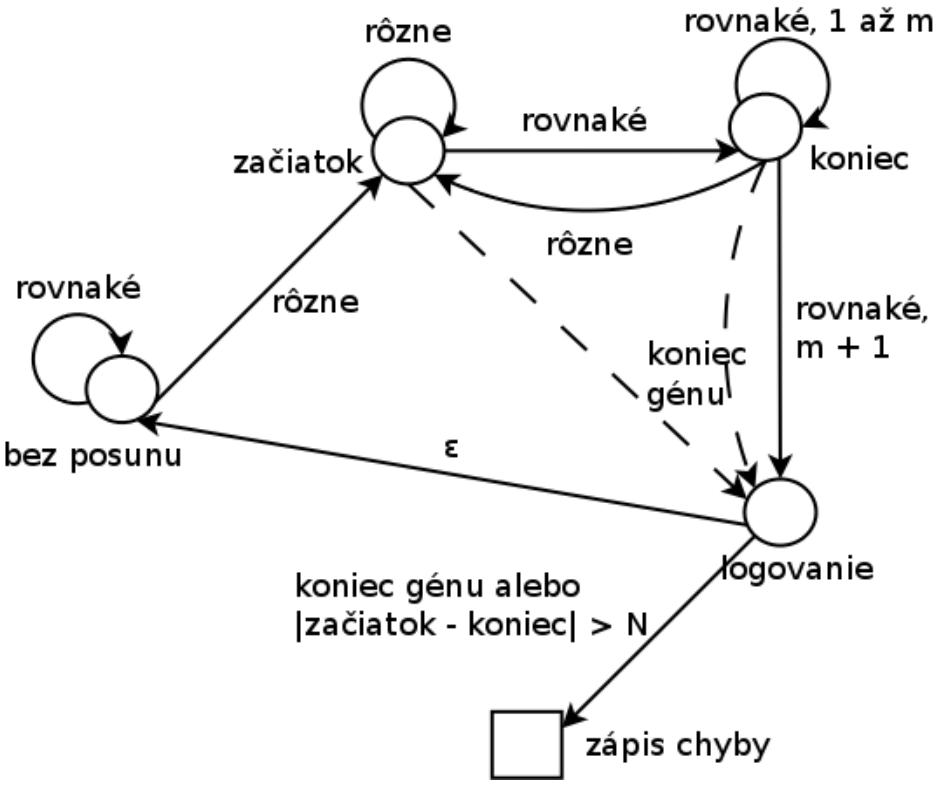
2.6.3 Čítací rámec

Filter pre čítací rámec zisťuje, či nedošlo na dlhšom úseku v géne k posunu čítacieho rámcu o číslo, ktoré nie je násobkom troch. Ak sa sekvencia génu v porovnávanom organizme dostatočne zhoduje, ale obsahuje v sebe medzeru o veľkosti napríklad dve bázy, veľmi pravdepodobne bude narušená funkcia génu. Pretože aminokyseliny sú kódované v trojbázových skupinách, takýto posun má za následok úplne novú sekvenciu kodónov a tým pádom aj výsledných aminokyselín tvoriacich proteín. V princípe existujú pre sekvenciu tri možnosti čítania kodónov v závislosti na mieste, v ktorom začneme.

Za chybu považujeme zmenu v čítacom rámci oproti referenčnej sekvencii. Preto sa v exónoch génov zameriavame na oblasti, v ktorých došlo ku indelom v sekvencii. Tento typ mutácie spôsobuje zmieňovaný posun čítacieho rámca a v zarovnaní je predstavovaný medzerami. Nájdené chyby v závislosti na ich dĺžke zaznamenávame dodatabázy.

Pre takúto kontrolu potrebujeme prejsť všetky bázy exónov a sledovať rámec v referenčnom aj porovnávanom genóme. Počítať však skutočný rámec by bolo časovo náročné. Preto pracujeme iba s oblasťami, kde dochádza k vzájomnému posunu medzi dvojicou reťazcov. Sledujeme iba pozície, kde sa nachádza medzera v práve jednom z genómov. Pre každý gén si pamätáme iba tento rozdiel v čítacích rámcoch. Viaceré takéto úseky idúce krátko po sebe považujeme za jedinú chybnú oblasť s posunom rámca. Zamedzíme tým potenciálnemu rozdeleniu oblasti na veľa krátkych, niekoľkobázových úsekov.

Keď nastane prvý rozdiel, začneme počítať frameshift chybu so začiatkom na danej pozícii. Ukončíme ju pri konci samotného génu alebo ak sa rozdiel rámcov vyrovná



Obrázok 2.3: Stavový diagram pre frameshift filter. Epsilon krok sa vykoná okamžite, prechody rovnaké/rôzne rozlišujú, či v práve jednej sekvencii sa nachádza medzera (rôzne) alebo nie (rovnaké).

a bude nulový. Po vynulovaní rozdielu niekoľko báz čakáme a ak znovu dôjde ku rozdielu, predlžujeme chybu a jej koniec posúvame ďalej. V opačnom prípade chybu na základe jej dĺžky zalogujeme. Potom ďalej prechádzame exóny a čakáme na začiatok nasledujúcej chyby alebo koniec génu. Schéma tohto postupu je znázornená na obrázku 2.3.

2.6.4 Nezmyselné mutácie

Filter pre nezmyselné mutácie preveruje existenciu stop kodónu v exóne génu pred jeho očakávaným koncom. V géne k tomu mohlo dôjsť substitúciou jednej či viacerých báz alebo posunom čítacieho rámcu indelom v sekvencii. Prepis takéhoto génu na proteín sa predčasne skončí a zvyšné aminokyseliny nie sú transkribované. Výsledkom by bol teda značne odlišný proteín. Čím skôr v géne sa predčasný stop kodón nachádza, tým je pravdepodobnejšie, že prestal plniť svoju pôvodnú úlohu. Každý výskyt považujeme za chybu a zaznamenáme ju v databáze.

Podobne ako v predchádzajúcom filtri, potrebujeme pre každú bázu v exónoch génu počítať čítací rámec. Vďaka nemu potom môžeme overovať jednotlivé triplety a porovnávať ich so stop kodónom. Triplety, ktoré porovnávame, sú rovnaké ako pri filtri na overenie konca génu a tiež sú závislé od vlákna, na ktorom sa gén nachádza. Nestačí nám tu však, narozdiel od predošlého filtra, vedieť iba rozdiel v čítacích rámcoch sekvencii. Musíme poznať ich skutočnú hodnotu počítanú od začiatku génu.

Robiť to pre každý exón nachádzajúci sa na danej báze v práve spracovávanom bloku by bolo časovo náročné. Namiesto toho používame spôsob, pri ktorom pracujeme iba s jedným číslom, vyjadrujúcim čítací rámec v referenčnej sekvencií od začiatku aktuálneho bloku zarovnania. Pre každý gén si pamätáme hodnotu čítacieho rámcu, ale nie úplne aktuálnu. Ďalej každý gén má ešte jedno pomocné číslo vyrovnávajúce starý čítací rámec s referenčnou sekvenciou. Na začiatku exónu nastavíme vyrovnávajúce číslo tak, aby v súčte s referenčným čítacím rámcom vynulovali. Aktuálny rámec počítame ako súčet starého, vyrovnávajúceho a referenčného. Na konci exónu si vždy zapamätáme aktuálny čítací rámec. Ten sa na začiatku ďalšieho opätovne stane starým číslom.

Takto poznáme pre každý gén jeho čítací rámec na danej pozícií v referenčnom genóme. Pre konkrétny organizmus k nemu ešte musíme pripočítať rozdiel rámcu na

danej pozícií oproti referenčnému, túto hodnotu nám poskytne frameshift filter, ktorý ju počíta. Celkovo teda vyzerá výpočet pre konkrétny gén nasledovne:

$$Aktualny[gen] = Stary[gen] + Vyrovnavajuci[gen] + Blokovy + Rozdiel[gen]$$

a pri začiatku exónu sa vyrovnávajúce číslo inicializuje vzorcom:

$$Vyrovnavajuci[gen] = (Stary[gen] - Blokovy)\ \%\ 3$$

2.7 Dotazovanie na výsledky

Poslednou fázou a cieľom samotného systému je získať množinu génov, ktoré neobsahovali žiadnu chybu. Vďaka ukladaniu pozícii chýb môžeme filtrovať gény s chybami na určitej časti. Pomocou parametru sa dá nastaviť, akú časť od stredu génu chceme bez chýb, pričom chyby nachádzajúce sa celé na okrajoch zanedbáme. Týmto spôsobom môžeme dostať väčšiu množinu výsledných génov.

Pre získanie výstupu v podobe množiny génov je potrebné zadať niekoľko údajov. Tými sú sady génov, z ktorých triedime, a organizmy, ktoré nás zaujímajú. Iba pre túto podmnožinu z infomácií v databáze overujeme prítomnosť chýb na zvolenej časti génov.

Pre každý gén vrátime zoznam organizmov, v ktorých prešiel testami. Nastavením parametra je možné zvoliť exportovanie týchto génov v podobe viacnásobného zarovnania. Podobne ako pri filtrovaní prechádzame pôvodné MAF súbory a pre aktívne exóny daného bloku kopírujeme ich sekvenciu do reťazca príslušného génu. Pre každý organizmus vytvárame vlastný reťazec génu. Na záver tieto reťazce prevedieme do súborov.

Výber z filtrovaných génov je možný buď samostatne alebo v podobe klastrov. Pri druhej možnosti sa z každého klastra vyberie práve jeden gén. Výber vhodného reprezentanta klastru sa riadi dvomi pravidlami. Prvým je vybrať gén, ktorý spĺňa podmienky v čo najväčšom množstve dotazovaných organizmov. Pri zhode následne určuje poradie dĺžka kódovaného proteínu. Ak aj tu nastane zhoda vyberie sa ktorýkoľvek z génov s najväčšou dĺžkou. Výsledná množina génov je závisla od dotazovanej množiny organizmov.

2.8 Implementácia systému

Popísaný systém sme sa snažili vytvoriť tak, aby bol relatívne rýchly. Spracovanie celých genómov je časovo značne náročné a aj pri rovnakej asympotickej časovej zložitosti je niekoľkonásobný rozdiel vo výkonnosti citeľný. Program sme preto implementovali v programovacom jazyku C++, a nie v skriptovacom jazyku. Ako databázový systém používame MySQL. Pristupujeme k nemu pomocou LibDBI knižnice, ktorá vytvára abstraktnú vrstvu nezávislú na konkrétnom databázovom systéme. Výmena používanej databázy napríklad za PostgreSQL by vďaka tomu bola veľmi jednoduchou záležitosťou.

Pri spustení je možné zadať niekoľko prepínačov, ktorými sa zadávajú vstupné dáta, určuje fáza spracovania a rôzne parametre. Prehľad všetkých parametrov, ich popis a použitie, je možné získať prepínačom "-h".

Pre program je k dispozícii niekoľko základných nastavení, ktoré sa dajú meniť v jeho konfiguračnom súbore. Nastavenia slúžia pre prepojenie s databázou a definíciu správania filtrov. Štandardný súbor, z ktorého sa nastavenia načítavajú, je nazvaný "config.txt". Každý riadok predstavuje dvojicu kľúč-hodnota oddelené znakom "=", riadok začínajúci znakom "#" je komentár a nespracuje sa. Zoznam nastavení je nasledovný:

db_host adresa databázového servera

db_user meno užívateľa databázy

db_passw heslo užívateľa databázy

db_name názov databázy, ktorá sa má použiť

filt_alignment maximálna povolená dĺžka bez zarovania v géne

filt_synteny maximálny rozdiel medzi dvomi po sebe idúcimi úsekmi zarovnania v porovnávanom druhu oproti referenčnému

filt_frame maximálna dĺžka sekvencie v géne s posunutým frame-om bez následného vykompenzovania

Kapitola 3

Výsledky

3.1 Testovacie dáta

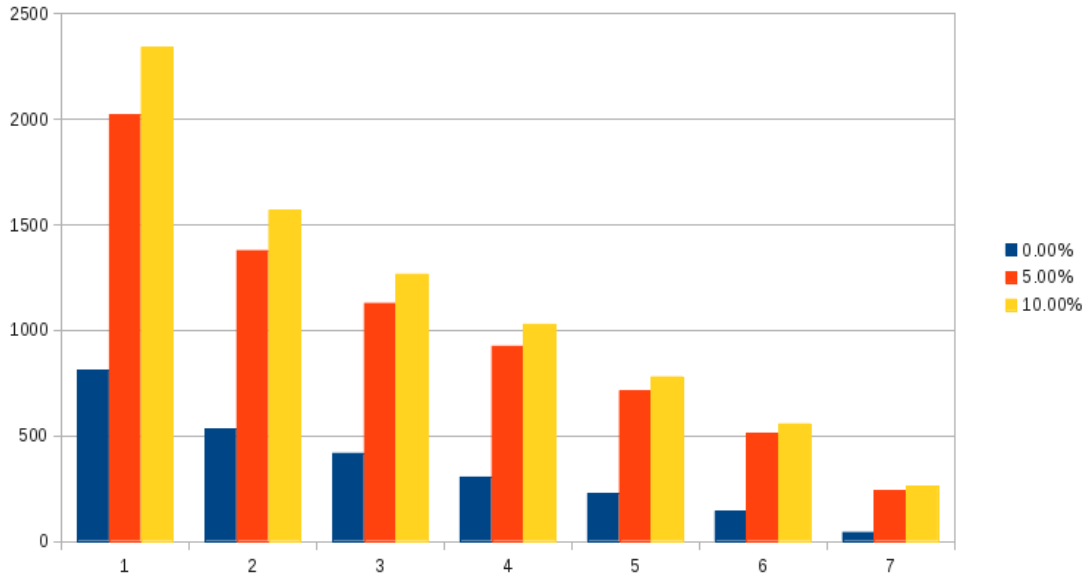
Popísaný systém sme spustili na reálnych dátach pre overenie jeho funkčnosti a získanie výstupov. Našimi vstupmi boli anotácie ľudských génov zo štyroch rôznych zdrojov. Zdrojom anotácií boli ENSEMBL databáza (143 123 génov), UCSC known genes databáza (77 614 génov), VEGA databáza (96 345 génov) a refSeq databáza (35 606 génov). Po prvotnom prefiltrovaní dát nám zostalo dohromady 218 556 génov tvoriacich 20 457 rôznych klastrov.

Ďalej sme mali k dispozícií viacnásobné zarovnanie niekoľkých organizmov ku človeku. Týmito organizmami sú:

anolis carolinensis (anoCar2), plaz
alligator mississippiensis (allMis0), plaz
python molurus (pytMol0), plaz
chrysemys picta (chrPic0), plaz
gallus gallus (galGal3), vták
taeniopygia guttata (taeGut1), vták
ornithorhynchus anatinus (ornAna1), cicavec

3.2 Dosiahnuté výsledky

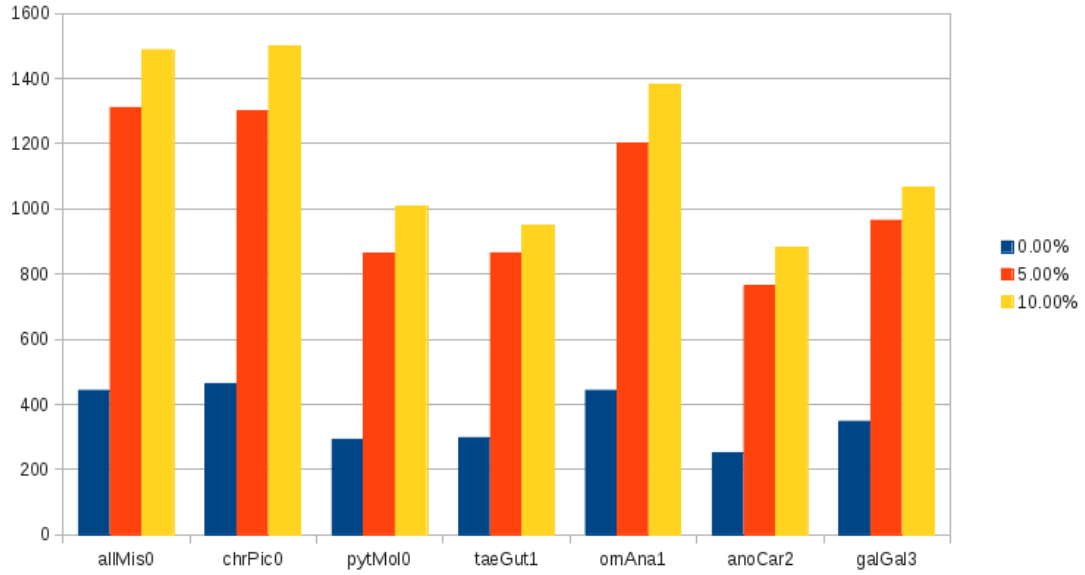
Na popísaných dátach sme spustili náš program. Následne sme exportovali výsledky tak pre jednotlivé organizmy samostatne ako aj pre všetky súčasne. Výber organizmov môže ovplyvniť výber reprezentantov z klastrov génov. Všetky exporty sme robili pre celé gény ako aj pre gény s okrajmi orezanými o 5 a 10 percent. Na obrázku 3.1 možno vidieť počet klastrov, ktoré obsahujú určitý počet organizmov. Pri orezaní okrajov o 5 percent vidno značný nárast vo všetkých skupinách, pri väčšom počte organizmov je nárast vyšší. To vypovedá o množstve génov so zachovanou štruktúrou okrem start a stop kodónu, ktoré mohli byť posunuté. Ďalšie orezanie na 10 percent prinesie už menšiu zmenu v počte klastrov, pričom rozdiel s počtom organizmov naopak klesá. V širšom okraji génov teda nedochádza k takému množstvu zmien, respektíve zmeny štruktúry sú aj na ďalších miestach génu.



Obrázok 3.1: Počet filtrovaných génov podľa počtu organizmov, v ktorých sa nachádzajú, po zhlukovaní do klastrov. Výsledky sú robené pre celé gény (0%), ako aj gény s orezanými okrajmi (5% respektíve 10%)

Počty klastrov exportované pre samostatné organizmy možno vidieť na obrázku 3.2, podobne aj pri nich je najväčšia zmena pri užšom okraji génu. Orezanie o 5 percent pri všetkých prinesie viac ako zdvojnásobenie filtrovaných génov, následné rozšírenie na 10 percent už len zlomok rastu. Počet génov so zachovalou štruktúrou v

jednotlivých organizmoch závisí tak od ich fylogenetickej príbuznosti ako aj od kvality ich osekvenovania a zarovnania. Vyšší počet možno badať u vtákopyska, aligátora a korytnačky, mierne nižší u dvojice vtákov a zvyšných plazov.



Obrázok 3.2: Počet filtrovaných génov v jednotlivých organizmoch, po zhlukovaní do klastrov. Výsledky sú robené pre celé gény (0%), ako aj gény s orezanými okrajmi (5% respektíve 10%)

3.3 Filtrované gény

Spomedzi génov, ktorých štruktúra bola v našom systéme zachovaná sme vybrali niekoľkých predstaviteľov. Jedná sa o gény, ktorých produkt má známu funkciu, pričom sme vyberali len tie, ktoré boli zachované vo väčšine z organizmov. Jedná sa prevažne o proteíny, ktorých funkcia je kľúčová pre správne fungovanie procesov v bunke. Tieto sú potrebné vo všetkých skúmaných stavovcoch a zmeny v nich často vedú k rakovine alebo iným poruchám. Gény kódujúce takéto proteíny preto patria k nadpriemerne zachovaným aj medzi relatívne vzdialenými druhmi.

ENST00000426215

Beta-katenín

Reguluje koordináciu medzibunkovej adhézie a génovej transkripcie, ktoré patria k ne-

vyhnutným procesom počas embryotického vývoja. Prenáša signály vnútri bunky, tvorí dôležitú súčasť Wnt-signálovej cesty. [7] Tá je tvorená proteínmi prenášajúcimi signáli z receptorov bunky do bunkového jadra, čo vedie ku expresií cieľových génov. Kontroluje rast bunky a bunkovú diferenciáciu. Prípadné poruchy sa ukázali byť príčinou vzniku rakoviny.

ENST00000402364

Sacsín

Je dôležitý pre funkciu centrálneho nervového systému. Pôsobí ako receptor, ktorý v kombinácii s inými proteínmi ovplyvňuje expresiu špecifických génov. Porucha tohto génu zapríčiňuje Charlevoix-Saguenay ataxiu. Je to dedičná neurologická porucha degenerujúca miechu a periférne nervy. Prejavuje sa mimo iné slabou koordináciou pri chôdzi, svalovou atrofiou.

uc001vin.3

Člen skupiny protocadherínov

Nádorový supresor, má významnú úlohu pri riadení rastu buniek.

uc001vrg.2 / ENST00000375775

Patrí medzi skupinu inhibítorov rastu. [6] Pôsobí proti vzniku rakovinových nádorov. Môže vyvolať apoptózu - proces vedúci k smrti bunky.

uc010usd.2

Enzým

Syntetizuje chondroitín sulfát nachádzajúci sa na povrchu väčšiny buniek. Tento má úlohu pri raste buniek a priestorovej distribúcií buniek počas embryotického vývoja - morfogenéze.

uc001mwu.3

Rekombinačný aktivátor

Enzým s dôležitou úlohou v procese VDJ rekombinácie imunoglobulínu a T bunkových receptorov. Tieto sú nevyhnutnou súčasťou imunitného systému stavovcov.

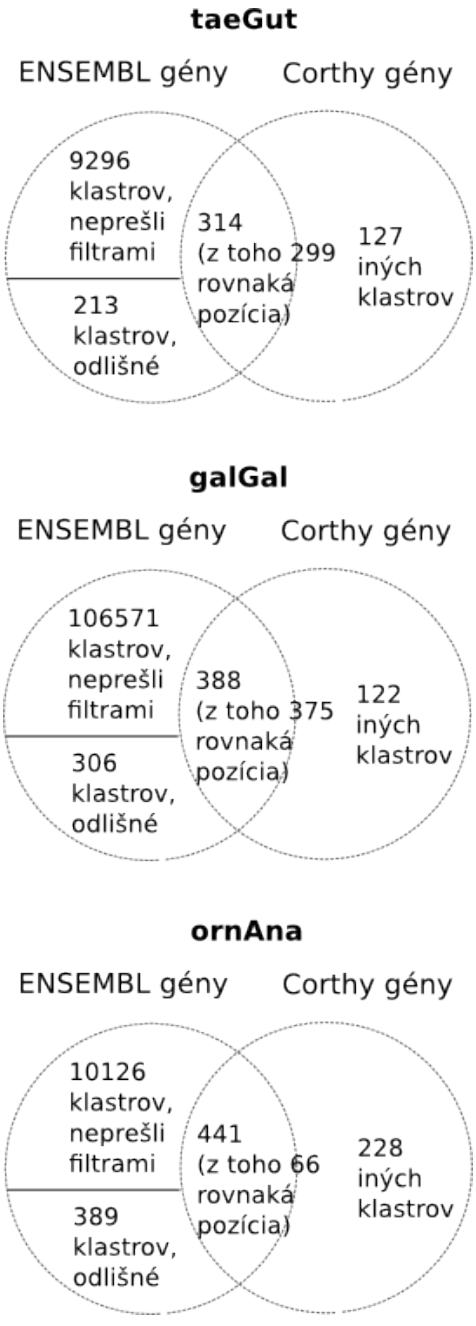
3.4 Porovnanie výsledkov

Ďalej sme naše výsledky porovnali so zoznamom ortológov vytvoreným na základe podobností sekvencií. Množina génov zodpovedala sade ENSEMBL génov, ktoré sme používali ako jeden zo zdrojov anotácií. Porovnanie sme spravili pre trojicu organizmov, pre ktoré boli dáta dostupné - vtákopyska, zebričku a sliepku. Pre vzájomné namapovanie výsledkov sme vytvorili skript v jazyku Perl. Podľa názvu génu sme priradili jednotlivé gény na tie, ktoré sme mali v databáze. Pri malom počte z nich sme nenašli zodpovedajúci gén. Porovnanie je možné vidieť na obrázku 3.3.

Veľká časť z génov neprešla niektorým z testov v našom systéme, jednoznačne najčastejším dôvodom bola chýbajúca časť zarovnania pre daný gén. Naopak z génov, ktorých štruktúra v zarovnaní bola zachovaná, nebola časť na získanom zozname ortológov. Väčšina nami nájdených ortológov v zozname bola. Z týchto spoločných génov sme určili počet génov s rovnakou pozíciou aj chromozómom v obidvoch zdrojoch. Teda vo vstupnom zarovnaní aj v zozname, ku ktorému sme sa porovnávali. Za rovnakú pozíciu sme považovali, ak sa gény na danom chromozóme prekrývali.

Počty génov s rovnakou pozíciou sa značne odlišujú podľa organizmu. Pri vtákopyskovi je ich relatívne málo, čo je spôsobené jeho osekvenovaním. Väčšina génov sa nachádzala na kontigoch - krátkych úsekoch, ktoré sa nepodarilo spojiť pri sekvenovaní do chromozómov. Preto nebolo možné porovnať pozície ani ich vzájomný rozdiel. Pri zebričke boli takmer všetky gény na rovnakej pozícii. Zvyšné gény sa nachádzali výnimkou jediného na rovnakom chromozóme. V týchto prípadoch sa pravdepodobne jedná o homologické úseky v tom istom organizme, ktoré sú navzájom paralógmi.

Komplikácie nastali pri porovnávaní sliepky, kde došlo len k zanedbateľnému prekryvu génov. Príčinou bolo novšie osekvenovanie ako to, s ktorým sme pracovali v zarovnaní. Použili sme preto predošlé dáta z archívu, pri ktorých sme dosiahli výsledky podobné ako u zebričky. Rovnako jediný gén sa líšil aj chromozómom, nejednalo sa ale o rovnaké gény.



Obrázok 3.3: Porovnanie s ENSEMBL databázou ortológov.

Záver

Na rozdiel od štandardných prístupov k porovnávaniu na úrovni báz respektíve aminokyselín sme v tejto práci s génmi pracovali na úrovni exónov. Spomedzi podobných génov sme tak selektovali tie najzachovalejšie. Všetky testované druhy boli relatívne dosť vzdialené od človeka a pri všetkých bola táto vzdialenosť rádovo rovnaká. Preto by možno bolo dobré systém skúsiť aj na iných organizmoch. Pri genóme bližšom človeku možno očakávať nárast zachovaných génov.

Vo výsledkoch sme si všimli značnú variabilitu okrajov génov aj pri inak zachovaných génoch. Bolo by preto vhodné pozrieť sa aj za okraje zarovnaného úseku a nájsť začiatok a koniec takéhoto génu. Koniec je buď posunutý dopredu vo forme nezmyselnej mutácie alebo dozadu za koniec pôvodného génu, v obidvoch prípadoch je ho ľahké lokalizovať. So začiatkom je to komplikovanejšie, pretože triplet, ktorý kóduje, sa môže nachádzať aj vnútri proteínu.

Pri zarovnaní najmä vzdialených organizmov môžu byť intróny nezarovnané z dôvodu príliš veľkých zmien medzi sekvenciami. Dalo by sa to vyriešiť tým, že zo zarovnania iba získame pozíciu génu pre daný organizmus. V prípade zarovnania častí génu k viacerým vzdialeným oblastiam by sme si pämatali všetky takéto oblasti. Následne by bolo potrebné spracovať osekvenovaný genóm porovnávaného organizmu. Vychádzali by sme z oblastí v zarovnaní a na ich miestach overovali štruktúru. Takáto oblasť by bola tvorená jedným súvislým úsekom. Filtre pre zarovnanie a synténiu by už neboli potrebné v aktuálnej podobe.

Literatúra

[1] Acuna, R., Padilla, B. E., Florez-Ramos, C. P., Rubio, J. D., Herrera, J. C., Benavides, P., Lee, S. J., Yeats, T. H., Egan, A. N., Doyle, J. J., and Rose, J. K. Adaptive horizontal transfer of a bacterial gene to an invasive insect pest of coffee. *Proc Natl Acad Sci U S A 109*, 11 (2012), 4197–4202.

[2] Cannon, S. B., and Young, N. D. OrthoParaMap: distinguishing orthologs from paralogs by integrating comparative genome data and gene phylogenies. *BMC Bioinformatics 4* (2003), 35.

[3] Dewey, C. N. Positional orthology: putting genomic evolutionary relationships into context. *Brief Bioinform 12*, 5 (2011), 401–402.

[4] Dubchak, I., and Frazer, K. Multi-species sequence comparison: the next frontier in genome annotation. *Genome Biol 4*, 12 (2003), 122.

[5] Dufayard, J. F., Duret, L., Penel, S., Gouy, M., Rechenmann, F., and Perriere, G. Tree pattern matching in phylogenetic trees: automatic search for orthologs or paralogs in homologous gene sequence databases. *Bioinformatics 21*, 11 (2005), 2596–2603.

[6] Gunduz, M., Gunduz, E., Rivera, R. S., and Nagatsuka, H. The inhibitor of growth (ING) gene family: potential role in cancer therapy. *Curr Cancer Drug Targets 8*, 4 (2008), 275–284.

[7] Komiya, Y., and Habas, R. Wnt signal transduction pathways. *Organogenesis 4*, 2 (2008), 68–75.

[8] Kosiol, C., Vinar, T., da Fonseca, R. R., Hubisz, M. J., Bustamante, C. D., Nielsen, R., and Siepel, A. Patterns of positive selection in six Mammalian genomes. *PLoS Genet 4*, 8 (2008), e1000144.

[9] Kristensen, D. M., Wolf, Y. I., Mushegian, A. R., and Koonin, E. V. Computational methods for Gene Orthology inference. *Brief Bioinform 12*, 5 (2011), 379–381.

[10] Kumar, S., and Filipski, A. Multiple sequence alignment: in pursuit of homologous DNA positions. *Genome Res 17*, 2 (2007), 127–135.

[11] Kuzniar, A., van Ham, R. C., Pongor, S., and Leunissen, J. A. The quest for orthologs: finding the corresponding gene across genomes. *Trends Genet 24*, 11 (2008), 539–541.

[12] Li, L., Stoeckert Jr., C. J., and Roos, D. S. OrthoMCL: identification of ortholog groups for eukaryotic genomes. *Genome Res 13*, 9 (2003), 2178–2179.

[13] Market, E., and Papavasiliou, F. N. V(D)J recombination and the evolution of the adaptive immune system. *PLoS Biol 1*, 1 (2003), E16.

[14] Min-Kyung, K., Young-Joo, S., Hyun-Seok, P., Seung-Hwan, J., Hang-Cheol, S., and Kwang-Hwi, C. A New Approach to Find Orthologous Proteins Using Sequence and Protein-Protein Interaction Similarity. *Genomics Informatics 7*, 3 (2009), 141–147.

[15] O'Brien, K. P., Remm, M., and Sonnhammer, E. L. Inparanoid: a comprehensive database of eukaryotic orthologs. *Nucleic Acids Res 33*, Database issue (2005), D476–480.

[16] Song, G., Riemer, C., Dickins, B., Kim, H. L., Zhang, L., Zhang, Y., Hsu, C. H., Hardison, R. C., r. o. g. r. a. m. Nisc Comparative Sequencing, P., Green, E. D., and Miller, W. Revealing mammalian evolutionary relationships by comparative analysis of gene clusters. *Genome Biol Evol 4*, 4 (2012), 586–601.

[17] Storm, C. E., and Sonnhammer, E. L. Automated ortholog inference from phylogenetic trees and calculation of orthology reliability. *Bioinformatics 18*, 1 (2002), 92–99.

[18] Tatusov, R. L., Fedorova, N. D., Jackson, J. D., Jacobs, A. R., Kiryutin, B., Koonin, E. V., Krylov, D. M., Mazumder, R., Mekhedov, S. L., Nikolskaya, A. N., Rao, B. S., Smirnov, S., Sverdlov, A. V., Vasudevan, S., Wolf, Y. I., Yin, J. J., and Natale, D. A. The COG database: an updated version includes eukaryotes. *BMC Bioinformatics 4* (2003), 41.

[19] Trachtulec, Z., and Forejt, J. Synteny of orthologous genes conserved in mammals, snake, fly, nematode, and fission yeast. *Mamm Genome 12*, 3 (2001), 227–231.

[20] Zmasek, C. M., and Eddy, S. R. RIO: analyzing proteomes by automated phylogenomics using resampled inference of orthologs. *BMC Bioinformatics 3* (2002), 14.

Dodatok A

CD príloha

Členenie súborov na priloženom CD je v priečinkoch podľa nasledovnej štruktúry:

- **solution:** zdrojový kód programu, konfiguračný súbor
- **input:** testovacie dáta
  - **genes:** súbory anotácii génov
  - **mafs:** viacnásobné zarovnanie, obsahuje iba niekoľko chromozómov
- **results:** filtrované gény pre jednotlivé organizmy
- **export:** výsledné exportované gény zo zarovnania
- **compare:** súbory pre porovnanie s ENSEMBL databázou
- **text:** elektronická verzia tohto dokumentu