VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

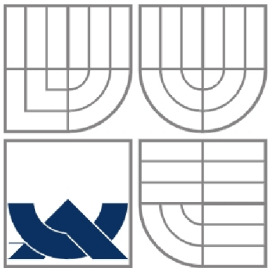
APLIKACE PRO ZPRACOVÁNÍ DAT Z OBLASTI EVOLUČNÍ BIOLOGIE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

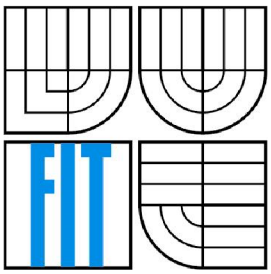
AUTOR PRÁCE
AUTHOR

LUKÁŠ RADAKOVIČ

BRNO 2015



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

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

LUKÁŠ RADAKOVIČ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PAVEL OČENÁŠEK, Ph.D.

BRNO 2015

Abstrakt

Tato bakalářská práce se zabývá návrhem a implementací aplikace, která má za úkol ověřit správnost algoritmu sloužícího na analýzu použitých mechanismů při tvorbě fylogenetického stromu. Aplikace umožňuje uživatelům zadat různé parametry fylogenetického stromu, jeho následné vytvoření a analýzu pomocí algoritmu. Výsledky analýzy se uloží do výstupního souboru, přičemž uživatel má možnost nastavit cestu. Zkoumaný algoritmus správně odhaduje podíl mechanismů na tvorbě stromu. Při odhadu relativního a absolutního podílu na změně počtu chromozomů a velikosti genomu nedosahuje přesné výsledky.

Abstract

This Bachelor's thesis describes the design and implementation of the application that has the task to verify the accuracy of the algorithm. Purpose of the algorithm is to analyze mechanisms used in the creation of the phylogenetic tree. The application allows users to specify different parameters of phylogenetic tree, its generation and subsequent analysis using an algorithm. Results of the analysis are written to the output file, giving the user the option of setting file path. Studied algorithm correctly estimates the participation of specific mechanism in the tree formation. Estimates of the absolute and relative share of changes in chromosome number and genome size are less than accurate results.

Klíčová slova

Fylogenetický strom, aneuploidie, polyploidie, proliferace DNA, odstranění DNA, rozpad chromozomů, spojení chromozomů

Keywords

Phylogenetic tree, aneuploidy, polyploidy, proliferation of DNA, removal of DNA, chromosome fission, chromosome fusion

Citace

Lukáš Radakovič: Aplikace pro zpracování dat z oblasti evoluční biologie, bakalářská práce, Brno, FIT VUT v Brně, 2015

Aplikace pro zpracování dat z oblasti evoluční biologie

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Pavla Očenáška, Ph.D.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Lukáš Radakovič
20. května 2015

Poděkování

Rád bych poděkoval vedoucímu mé práce, Ing. Pavlu Očenáškovi, Ph.D., za pomoc, rady a připomínky.

Obsah

1 Úvod

V súčasnosti čoraz častejšie dochádza k prepojeniu informatiky a biológie. Vďaka tomu vznikol aj relatívne nový odbor bioinformatika, ktorá spája tieto dve vedné disciplíny. V biológii je snaha o organizáciu živých organizmov do spoločnej štruktúry. Takéto štruktúry na nazývajú aj stromy života alebo fylogenetické stromy. Aby sa mohli vytvoriť stromy života, je potrebné určiť podľa akých znakov budeme triediť organizmy, napríklad podľa veľkosti genómu, počtu chromozómov.

Cieľom tento práce je overiť spoľahlivosť algoritmu na veľkej množine dát, ktorý bol vytvorený na Přirodovědeckej fakulte Masarykovej Univerzity v Brne. Algoritmus na základe konečných uzlov v strome získa podiel rôznych mechanizmov, ktoré ovplyvňujú veľkosť genómu a počet chromozómov, v celom strome.

V kapitole 2 sa nachádza teoretický úvod do problematiky. Obsahuje základné pojmy, ktoré sa bežne používajú v molekulárnej biológii. Ďalšom významnou časťou sú štyri mechanizmy, ktoré majú vplyv na počet chromozómov a veľkosť genómu. Tieto mechanizmy berie do úvahy aj skúmaný algoritmus. Potom sa tu popis fylogenetických stromov, ich grafická a textová reprezentácia. Keďže fylogenetický strom je možné reprezentovať ako strom z teórie grafom, sú tu uvedené základné definície. Poslednom časťou je problematika analýzy fylogenetických stromov a stručný popis skúmaného algoritmu.

Kapitola 3 obsahuje popis návrhu aplikácie, jej grafické znázornenie a diagram prípadov užitia.

Kapitola 4 popisuje implementáciu aplikácie. Obsahuje diagram tried, popis grafického užívateľského rozhrania, použitých technológii, popis jednotlivých tried a ich významných metód. Sú tu uvedené algoritmy pre generovanie stromu a jeho analýzu.

Spracovanie výsledkov a diskusia sa nachádza v kapitole 5. V prvej časti je uvedený spôsob spracovania výsledkov, nevyhnuté matematické vzťahy. Druhá časť popisuje niekoľko vybraných prípadov určených na diskusiu. Každý obsahuje zhodnotenie výsledkov

V kapitole 6 je záver práce, stručné zhodnotenie výsledkov a návrh vylepšenia aplikácie a algoritmu.

2 Teoretický úvod

2.1 Úvod do molekulárnej biológie

Jednou zo základných vlastností živých organizmov je ich variabilita alebo premenlivosť. Organizmy sa navzájom odlišujú vlastnosťami, a to napríklad veľkosťou, štruktúrou tela, usporiadanín orgánov, buniek. Tieto vlastnosti sú podmienené podmienkami prostredia, v ktorom organizmy žijú, a stavebným plánom, ktorý je vstavaný v základných štruktúrach organizmu. Tejto plán jedinec dedí od svojich rodičov v prípade pohlavného rozmnožovania, čiže je kombináciou informácií získaných od oboch rodičov. Pri nepohlavnom rozmnožovaní je súčasťou zárodočnej časti, z ktorej jedinec vzniká, a je identický s informáciou rodiča, z ktorého sa zárodočná časť oddelila. Stavebný plán reprezentuje informáciu zdedenú od predchádzajúcej generácie, a to dedičnú informáciu. [3]

Jedinec od svojich rodičov nededí znak samotný, ale len vlohu pre jeho prejavenie. Používa sa pre to označenie gén. Súbor všetkých génov jedinca označujeme ako genotyp. Súbor všetkých znakov jedinca sa nazýva fenotyp. Fenotyp nie je len výsledkom aplikácie genetickej informácie, ale aj vplyvu vonkajšieho prostredia. [3]

Rozdiely vlastností medzi jedincami sú v určitých prípadoch dostatočné na to, aby sme mohli jedince zaradiť do odlišných taxónov rôznych hierarchických úrovní. Taxón alebo taxonomická jednotka je skupina konkrétnych organizmov, ktoré majú spoločné určité znaky, čím sa odlišujú od ostatných taxónov. Konkrétny taxón zvyčajne býva pomenovaný [8]. Základnou taxonomickou kategóriou je druh. Premenlivosťou sa vyznačuje akýkoľvek súbor organizmov nezávisle na úrovni organizácie. Časť tejto premenlivosti spočíva v tom, že súbor pozostáva z kvalitatívne odlišných jedincov. Tento typ premenlivosti sa nazýva ako rozmanitosť alebo diverzita. Čím vyššia je vnútorná rozmanitosť systému, tým vyššia je pravdepodobnosť, že v prípade zásadných zmien prostredia sa v jeho rámci nájde aspoň jeden prvok, ktorý sa s nimi bude vedieť vyrovnať a zabezpečí pokračovanie jeho existencie. Biologická diverzita je teda predpokladom pre udržanie života na Zemi. [3]

Genetika je veda o dedičnosti a premenlivosti živých organizmov. Dedičnosť znamená schopnosť organizmov produkovať potomstvo, ktoré na rovnaké podmienky reaguje rovnakým spôsobom. [3]

Molekulárna genetika rieši otázky spojené s účasťou jednotlivých typov organických makromolekúl na dedičnosti a molekulárnej podstate uloženia a rozmnožovania dedičnej informácie a jej vzťahu k fenotypovým znakom, ktoré ovplyvňuje. Rieši teda otázky vzťahu medzi dedičnou informáciou a štruktúrou organických makromolekúl, ktoré sú jej nositeľmi, a procesov, ktoré sú nutné pre vytváranie produktov dedičnej informácie.

. Predmetom genetiky jedinca sú otázky dedičnosti fenotypových znakov, teda otázka, ako sa vonkajšie vlastnosti dedia, prenášajú z rodičov na potomstvo. [3]

Genomika skúma štruktúry a funkcie genómu, teda súboru všetkých dedičných informácií v rámci živej bunky. To pozostáva z analýzy dedičnej informácie, identifikácie génov a ďalších úsekov v genóme, identifikácia funkcií génov. [3]

Bioinformatika sa zaoberá spracovaním veľkých objemov dát, ktoré vznikajú v rámci genetického a molekulárno-biologického výskumu. [3]

2.1.1 DNA

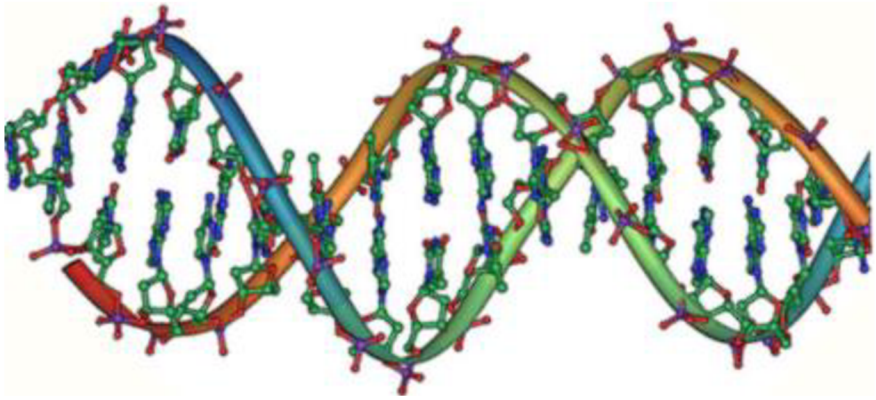
Nositeľom dedičnej informácie u všetkých živých organizmov sú molekuly nukleových kyselín. U väčšej časti organizmov je to deoxyribonukleová kyselina (DNA). V ostatných organizmoch

(napríklad retrovírusy) je nositeľkou ribonukleová kyselina (RNA). V oboch prípadoch je dedičná informácia najprv prenesená do molekuly DNA, potom sa prejaví do fenotypových znakov. [3]

Makromolekula je veľká molekula, ktorá sa skladá z mnohonásobne sa opakujúcich štruktúrnych jednotiek, a to z niekoľko sto až tisíc atómov spojených kovalentnými väzbami. [3]

Nukleové kyseliny sú organické makromolekuly. DNA a RNA sú nerozvetvené molekuly, pozostávajúce z nukleotidov. Nukleotid pozostáva z fosforylovanej pentózy a jednej zo štyroch heterocyklických organických báz (adenín, guanín, cytozín, pri DNA tymín, pri RNA uracil). Z pentózofosfátových molekúl je tvorená kostra reťazca, a na ňu sú naviazané bázy. Ich poradie predstavuje zápis genetickej informácie. Molekula DNA je tvorená dvomi protismerne prebiehajúcimi reťazcami, ktoré sú navzájom previazané prostredníctvom vodíkových mostíkov medzi bázami. Molekula RNA je tvorená len jedným reťazcom. [3]

DNA môže existovať vo viacerých priestorových usporiadaniach, v živých bunkách je zvyčajne prítomná ako pravotočivá špirála (obrázok 2.1), v ktorej sú molekuly fosforylovanej deoxyribózy orientované pozdĺž oboch vlákien a nukleotidy sú usporiadané v kolmej rovine na os špirály. [3].



Obrázok 2.1: Štruktúra DNA. Prevzatý z [9]

2.1.2 Genóm

Celková genetická informácia bunky sa označuje termínom genóm. Je tvorený súborom všetkých génov bunky. Veľkosť genómu ale aj počet génov čiastočne závisí od zložitosti organizmu, ale je čiastočne aj výsledkom evolúcie genómu. Súčasťou genómu sú kódujúce aj nekódujúce sekvencie. Najväčší vplyv na veľkosť genómu majú repetitívne (opakujúce sa) sekvencie. Typicky sa jeho hmotnosť uvádza v pikogramoch alebo ako celkový počet nukleotidových bázových párov, jednotka je mega báza, čo je milión bázových párov. Jeden pikogram je ekvivalentný 978 mega bázam. Hmotnosť genómu zvyčajne označujeme symbolom C. [3]

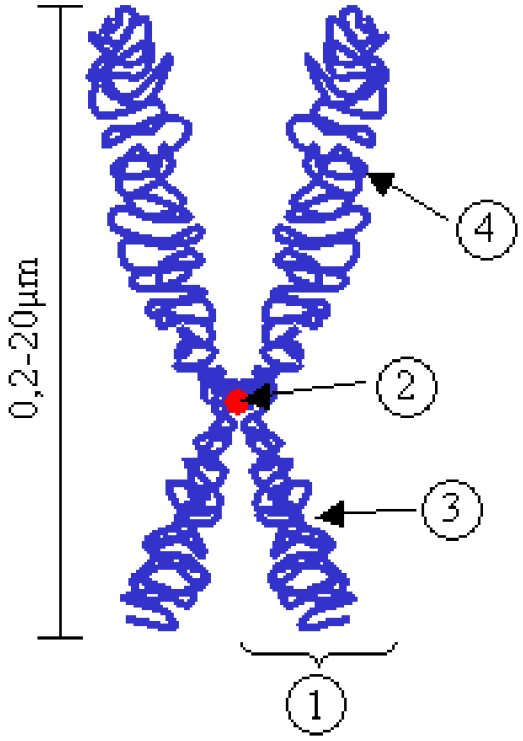
2.1.3 Bunky, chromozóm

Prokaryotická bunka má jednoduchý tvar. Nemá diferencované jadro ani organely, kruhová molekula DNA, plazmidy, je voľne uložená v cytoplazme. [3]

Na rozdiel od prokaryotickej bunky je eukariotická bunka komplikovaný a vysoko organizovaný útvar. Má oddelené jadro, ktoré obsahuje najväčšiu časť genetického materiálu. Bunka je vnútorné rozdelená na kompartmenty a obsahuje množstvo organel, ktoré sú samostatné

vnútrobunkové útvary so špecializovanými funkciami. Niektoré z nich sú samostatné, a teda nesú genetickú informáciu a sú schopné autoreprodukcie. Bunka je zvonku oddelená plazmatickou membránou, na ktorú sa ešte môže nadväzovať bunková stena. DNA v jadre bunky nie je uložená voľne, ale je priestorovo usporiadané do vyššej štruktúry, až po úroveň kondenzovaného chromozómu. Vlákno DNA je navinuté na malé bielkovinové komplexy, nazývané nukleozómy. Nukleozómy sú spakované do chromatínového vlákna. Chromatínové vlákno vytvára slučky a v tejto štruktúre je molekula DNA uložená v jadre v období medzi delením buniek. Pri delení ďalej kondenzuje a ukladá sa do útvaru, ktorý označujeme termínom chromozóm. [3]

Každý chromozóm má určitú štruktúru, ktorá je závislá na štruktúre DNA tvoriacej jeho základ. Na chromozóme možno pozorovať niektoré typické útvary. Konce chromozómov, ktoré sa nazývajú teloméry, sú tvorené tandemovo opakovanými neexprimovanými sekvenciami a chránia chromozóm pred postupným odbúravaním pri každej replikácii. Ďalšia časť chromozómu sa nazýva centroméra, ktorá sa javí ako zúžené miesto. Stavba chromozómu je uvedená na obrázku 2.2. Pri bunkovom delení sa molekula DNA musí replikovať, chromozóm sa v tomto štádiu skladá z dvoch sesterských chromatíd, ktoré zostávajú spojené v centromére až do delenia. Chromozóm má teda väčšinou tvar X. Centroméra delí chromozóm na dve ramená, ktoré sú zvyčajne rovnako veľké, alebo je centroméra umiestnená v blízkosti teloméry, prípadne na konci chromozómu. Holocentrické chromozómy sú také chromozómy, ktoré nemajú centroméru. Pri bunečnom delení celá dĺžka chromozómu nahrádza funkciu centroméry. Počet chromozómov v jadre určuje jej ploidiu. [3]



Obrázok 2.2: Štruktúra chromozómu. 1. chromatida, 2. centroméra, 3. rameno chromatidy, 4. dlhé rameno chromatidy. Prevzatý z [6]

Somatické bunky, ktoré obsahujú za normálnych okolností dve sady homolických chromozómov, sú takzvané diploidné. Ak kompletná sada obsahuje N chromozómov, čo je haploidný

počet, potom počet chromozómov v somatickej bunke je 2N. Označenie N nazývame haploidné číslo. Ľudské diploidné bunky majú 46 chromozómov, a haploidné bunky 23 chromozómov. Polyploidy majú niekoľko chromozómových sád, zvyčajne 3 a viac. [3]

2.2 Mechanizmy ovplyvňujúce veľkosť genómu a počet chromozómov

V tejto podkapitole sú uvedené štyri typy mechanizmov, ktoré ovplyvňujú veľkosť genómu a počet chromozómov.

2.2.1 Spojenie, rozpad chromozómov

Tento mechanizmus patrí do skupiny chromozómových mutácií. Dvojreťazcové zlomy, teda prerušenie oboch reťazcov DNA na rovnakom mieste, vedú k rozpadu chromozómov. To znamená, že chromozóm sa rozdelí na dve samostatné časti. Rozpad spravidla vedie k závažným poruchám funkcie organizmu, pretože len jeden z úsekov má centroméru, a teda normálne funguje pri bunečnom delení. Rozpadom chromozómov sa teda počet chromozómov zväčšuje, pričom veľkosť genómu ostáva nezmenená.

Homologické chromozómy alebo sesterské chromatidy sa môžu aj spájať, čo nazývame aj fúzia chromozómov. Tieto novovzniknuté chromozómy sú schopné normálneho fungovania, a to len vtedy, ak si zachovajú len jednu centroméru. Dicentrické chromozómy by neboli schopné normálneho presunu počas bunečného delenia. Pri spojení chromozómov dochádza k zmenšeniu počtu chromozómov, avšak veľkosť genómu sa nemení. [3]

2.2.2 Repetitívne DNA šírenie, odstránenie

Podstatnú časť genómu tvoria práve repetitívne sekvencie. Procesy zodpovedajúce za ich šírenie v genóme sa označujú ako molekulárny ťah. Zmeny genómu častokrát ovplyvňujú viac jedincov v populácii súčasne. Sú spôsobené napríklad génovou konverziou, translokáciou, nerovným crossing-overom, atď. V tomto prípade sa mení jedine veľkosť genómu, počet chromozómov je zachovaný. Pri šírení veľkosť genómu rastie, pri odstránení naopak klesá. [3]

2.2.3 Polyploidia

Polyploidia patrí medzi genómové mutácie. Tie postihujú celé chromozómy a ich sady, menia ich počet, a tým zásadne menia veľkosť genómu. Vznikajú v dôsledku nepravidelnosti bunkového delenia, ak sa chromozómy správne nerozídu do dcérskych jadier. Pri polyploidii dochádza k znásobeniu celej chromozómovej sady. Polyploidia sa bežne vyskytuje u prevažne rastlinných rodov, ktoré môžu obsahovať rôzne stupne ploidie. U živočíchov sa objavuje len výnimočne. Pri autopolyploidii bunky obsahujú nadbytočné sady plne homologických chromozómov, a to v dôsledku absencie delenia jadra alebo cytokinézy. Autopolyploidy sú zvyčajne sterilné v dôsledku nepravidelného párovania chromozómov počas bunečného delenia, vytvorí sa trojica homologizovaných chromozómov a jeden ostane navyše. Ak sa rozmnožujú nepohlavnou cestou, počet mutácii sa môže zmenšiť a môžu byť opäť fertilné. [3][5]

Polyploidia je stav, pri ktorom sa v jadre bunky vyskytuje troj a viac násobok haploidného počtu chromozómov. Pri meiotickom delení môže dôjsť k poruchám v redukcii počtu chromozómov, na základe čoho vznikajú namiesto haploidných buniek bunky s diploidným (2n) počtom chromozómov. Ak splynú bunky, ktoré obidve majú 2n, vzniknú tetraploidné organizmy s 4n počtom

chromozómov. Polyploidiu, pri ktorej ide o celočíselné násobky základného počtu chromozómov označujeme ako euploidu (2n = k . x; k je celé číslo). V tomto mechanizme rastie počet chromozómov, a zároveň s ním aj veľkosť genómu. Typy polyploidie sú 6x/8x, 4x/6x, 2x/4x, 2x/6x, 2x/8x. Pri polyploidii typu 2x/4x počet chromozómov a veľkosť genómu sa zväčší dvojnásobne. [3][5]

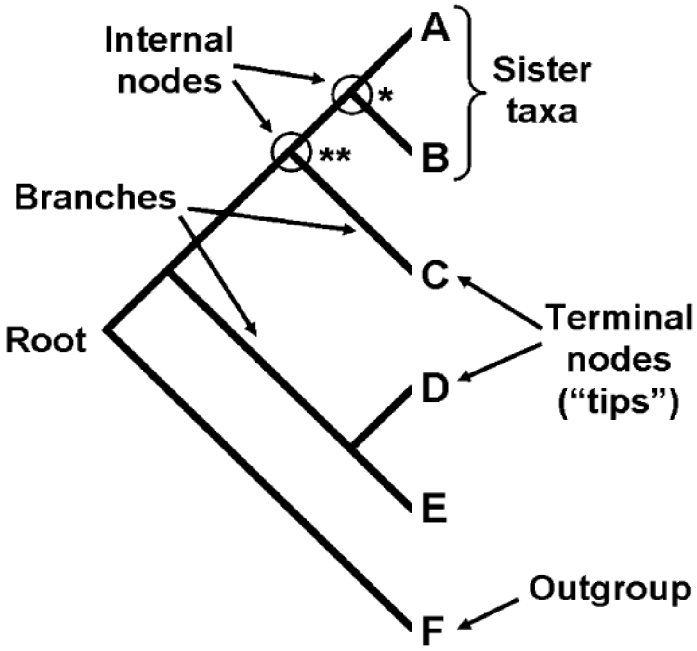
2.2.4 Aneuploidia

Aneuploidia je polyploidia, pri ktorej počet chromozómov v jadre bunky nie je presným násobkom haploidného počtu (2n = k . x; k nie je celé číslo). Aneuploidná bunka teda môže mať jeden alebo niekoľko chromozómov navyše, alebo mu môže jeden alebo niekoľko chromozómov chýbať. U živočíchov je aneuploidia spravidla letálna alebo vedie k závažným vývojovým poruchám. Rastliny majú lepšiu toleranciu pre aneuploidiu a nie vždy má zmena počtu chromozómov dopad na ich fenotyp. Organizmus, ktorý je aneuploidný a má počet chromozómov vyšší ako 2n označujeme ako hyperploidný, v opačnom prípade hypoploidný. Aneuploidy vznikajú pri poruche rozostupovania chromozómov k pólom bunky pri meióze alebo mitóze. Homologické chromozómy sa nemôžu oddeliť pri rozostupovaní, čo má za následok, že tento pár je presunutý len k jednému pólu bunky. Vznikne jedna gaméta s nadbytočných chromozómov, druhej gaméte bude tento chromozóm chýbať. Splývaním gamét so zmeneným počtom chromozómov vznikne aneuploidia. V tomto mechanizme môže počet chromozómov a veľkosť genómu sa zväčšovať, ale aj zmenšovať. [3][5]

2.3 Fylogenetický strom

Fylogenéza je štúdium evolučných vzťahov medzi skupinami organizmov a to na základe podobnosti ich DNA. Evolúcia je proces, kedy sa populácia mení v čase a môže sa rozdeliť do oddelených vetví, ktoré sa môžu prípadne spojiť alebo dokonca vyhynúť. Tento proces môžeme zobraziť napríklad fylogenetickým stromom. [1]

Fylogenetický strom je grafický popis biologických organizmov, ktoré majú spoločného predka, ako druhy alebo taxonomické zoskupenia vyššieho druhu. Veľké množstvo dôkazov podporuje záver, že všetky dnes žijúce, ale aj vyhynuté organizmy, majú spoločné znaky, vďaka ktorým môžeme určiť počiatok života približne na 3,8 miliardy rokov. Teoreticky by malo byť možné vytvoriť jeden strom života z dnešných žijúcich druhov a nájsť spoločného predka, Problém je v tom, že evolučné procesy sú častokrát komplexné. Súvislosti medzi druhmi závisí na genetike, ako aj na histórii a postačujúci dôkaz je, že dokonca aj najvzdialenejšie línie majú významné podobnosti v génoch. Avšak je tu možnosť, že neexistuje len jediný koreň stromu života, ale je ich viacej paralelne vedľa seba bez prepojenia. Podobne génové stromy sa nemusia zhodovať s druhovými stromami, čo spôsobuje komplikácie pri vytváraní fylogenetických stromov. Všetky fylogenetické stromy poskytujú informácie o pôvode a rozmanitosti druhov. Zobrazujú skupiny vetví, ktoré sa spájajú v bodoch, ktoré reprezentujú spoločných predkov, ktorí sú podobne prepojení vo vzdialenejších predkoch. Na obrázku 2.3 vidíme, že druhy A a B majú štyroch spoločných predkov, A a E dvoch spoločných predkov. Sesterské taxóny sú druhy, ktoré majú priameho spoločného predka. Koncovými uzlami označujeme také vrcholy, ktoré nemajú nasledovníkov, v našom prípade A, B, C, D, E, F. Uzly, ktoré majú aj nasledovníkov sa nazývajú vnútorné uzly. Cesta od predka k potomkovi sa nazýva vetva. Outgroup sa používa pre označenie vetvy, ktorá nemá vplyv na skúmanú časť fylogenetického stromu.



Obrázok 2.3 Fylogenetický strom. Prevzatý z [1].

Vnútorné vrcholy reprezentujú vyhynutých predkov dnešných organizmov. Každý vnútorný vrchol stromu reprezentuje speciáciu. Speciácia je udalosť, keď sa spoločná ancestrálna populácia rozdelila na dve alebo viac častí, z ktorých sa neskôr vyvinuli rôzne biologické druhy. Jednotlivým hranám fylogenetického stromu často priraďujeme určitú dĺžku, ktorá zodpovedá evolučnému času medzi dvoma specíáciami, alebo množstvu mutácií, ktoré v sekvencii na tejto hrane nastali. Nezakorenený fylogenetický strom je taký strom, ktorý znázorňuje vzťahy medzi taxonomickými jednotkami bez toho, aby určoval ich spoločného predka. Zakorenený fylogenetický strom je strom, u ktorého je jeden z vnútorných uzlov označený ako koreň. Vďaka tomu hrany stromu získajú orientáciu v smere od koreňa ku koncovým uzlom. Koreň reprezentuje spoločného predchodcu všetkých taxonomických jednotiek. Zo zakoreneného stromu je možné vytvoriť nezakorenený strom, opačný postup je možný len s ďalšími informáciami o priebehu evolúcie.

Pri tvorbe fylogenetických stromov sa vychádza z údajov o podobnosti medzi jednotlivými taxonomickými jednotkami. Je niekoľko spôsobov, ako túto podobnosť definovať. Najčastejšie využívajú znalosti z molekulárnej biológie. Vychádza sa zo sekvencií báz v genómoch jednotlivých biologických organizmov, prípadne sa dajú použiť aj vedomosti o aminokyselinových a proteínových produktoch. Vďaka týmto dátam môžeme určiť genetické vzdialenosti medzi jednotlivými taxonomickými jednotkami. Najprv je potrebné vhodne zarovnať porovnávané DNA sekvencie. Jedná sa o výpočtovo veľmi ťažkú úlohu (NP-úplný problém), v praxi sa preto používajú heuristické metódy na nájdenie suboptimálneho riešenia. Vzájomnú vzdialenosť môžeme určiť napríklad na základe percenta odlišných báz medzi sekvenciami. Lepšie metódy sa snažia odhadnúť počet mutácií, ktoré sú potrebné na prechod z jednej sekvencii do druhej.

2.4 Teória grafov

Fylogenetický strom je možné popísať aj matematicky. V tejto časti sú uvedené nevyhnutné definície potrebné na formuláciu stromu.

Definícia 2.1: Usporiadaná dvojica (u,v) prvkov u, v z množiny V je taká dvojica pri ktorej je určené, ktorý z prvkov u, v je na prvom, a ktorý na druhom mieste. Usporiadaná n-tica prvkov je taká n-tica prvkov $(a_1, a_2, \ldots, a_n)$, pri ktorej je jednoznačne určené poradie prvkov [4].

Definícia 2.2: Množina všetkých neusporiadaných dvojíc z V budeme značiť $V \circ V$ [4].

Definícia 2.3: Grafom nazveme usporiadanú dvojicu $G = (V, H)$, kde V je neprázdna konečná množina a H je množina neusporiadaných dvojíc typu $\{u, v\}$, pre ktoré platí, že $u \in V$, $v \in V$, a $u \neq v$, t. j.

$$H \subseteq \left\{ (u,v) \;\middle|\; u \neq v; u,v \in V \right\} \subset V \times V \tag{2.1}$$

Prvky množiny V nazývame vrcholmi a prvky množiny H hranami grafu G [4].

Definícia 2.4: Hovoríme, že graf $G' = (V', H')$ je podgrafom grafu $G = (V, H)$, ak platí $V' \subseteq V$ a $H' \subseteq H$ [4].

Definícia 2.5: Nech $G = (V, H)$ je graf, $v \in V$, $h \in H$. Vrchol v je incidentný s hranou h, ak je v jedným z vrcholov hrany h [4].

Definícia 2.6: Stupeň $deg(v)$ vrcholu v v grafe $G = (V, H)$ je počet hrán incidentných s vrcholom v [4].

Definícia 2.7: Pravidelný graf stupňa k je taký graf $G = (V, H)$, v ktorom má každý vrchol $v \in V$ stupeň k [4].

Definícia 2.8: Nech $G = (V, H)$ je graf. Sled v grafe G je ľubovoľná alternujúca (striedavá) postupnosť vrcholov a hrán tvaru

$$\mu(v_1, v_k) = (v_1, \{v_1, v_2\}, v_2, \{v_2, v_3\}, v_3, \ldots, \{v_{k-1}, v_k\}, v_k) \text{[4]}. \tag{2.2}$$

Definícia 2.9: Cesta v grafe G je taký sled v grafe G, v ktorom sa žiaden vrchol neopakuje [4].

Definícia 2.10: Hovoríme, že graf $G = (V, H)$ je súvislý, ak pre každú dvojicu vrcholov u, $v \in V$ existuje u-v cesta. Inak hovoríme, že graf G je nesúvislý [4].

Definícia 2.11: Kružnica je pravidelný súvislý graf 2. stupňa [4].

Definícia 2.12: Acyklický graf je taký graf, ktorý neobsahuje ako podgraf kružnicu [4].

Definícia 2.13: Strom je súvislý acyklický graf [4].

Definícia 2.14: Koreňový strom je strom $G = (G, V)$ s pevne vybraným vrcholom $k \in V$, ktorý

nazývame koreň. Koreňový strom sa označuje $G = (V, H, k)$. Úroveň vrcholu u v koreňovom strome $G = (V, H, k)$ je dĺžka k-u cesty. Výška koreňového stromu $G = (V, H, k)$ je maximum z úrovní všetkých vrcholov koreňového stromu G [4].

Definícia 2.15: Binárny koreňový strom je koreňový strom, v ktorom má každý vrchol najviac dvoch bezprostredných nasledovníkov [4].

2.4.1 Prechod stromom

Prechod stromom je postupnosť všetkých vrcholov stromu, v ktorej sa žiadny uzol nevyskytuje dvakrát: Priechod transformuje nelineárnu štruktúru stromu na lineárnu [7]. Najpoužívanejšie sú tieto tri typy prechodov stromom: preOrder, postOrder, inOrder.
PreOrder spracuje strom v tomto poradí: vrchol, ľavý podstrom, pravý podstrom.
PostOrder spracuje strom v tomto poradí: ľavý podstrom, pravý podstrom, vrchol
InOrder spracuje strom v tomto poradí: ľavý podstrom, vrchol, pravý podstrom.
K týmto trom priechodom existujú aj ich inverzné verzie.

Matematický binárny koreňový strom môžeme použiť aj na popis fylogenetického stromu.

2.5 Zložitosť algoritmu

Definícia 2.16: Hovoríme, že algoritmus A ma zložitosť $O(f(n))$, ak pre horný odhad $T(n)$ počtu krokov algoritmu A pre úlohu dĺžky n platí

$$T(n) = O(f(n)). \tag{2.3}$$

Špeciálne ak $f(n) \leq n^k$ pre nejaké konštantné k, hovoríme, že A je polynomiálny algoritmus.

Definícia 2.17: Úloha lineárneho programovania je nájsť také reálne čísla
$x_1, x_2, \ldots, x_n$, pre ktoré je

$$f(x) = c^T \cdot x = c_1x_1 + c_2x_2 + \cdots + c_nx_n \tag{2.4}$$

minimálne za predpokladov

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \\ x_1, x_2, \ldots, x_n &\geq 0 \text{ [4].} \end{aligned} \tag{2.5}$$

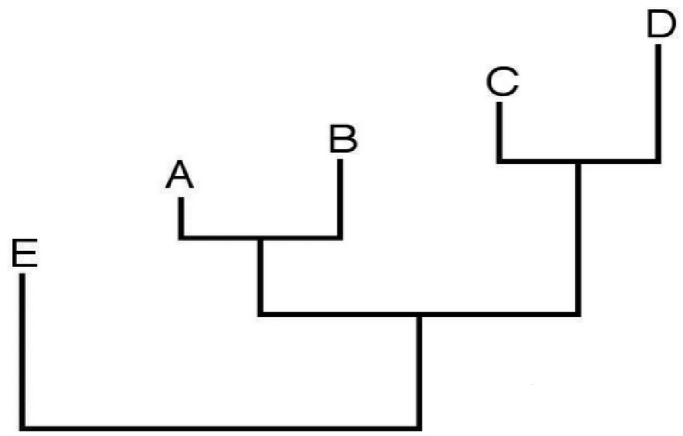
Definícia 2.18: Úloha celočíselného lineárneho programovania (úloha CLP) je nájsť takú n–ticu celých čísel x, pre ktorú je $c^T . x$ minimálne za predpokladov $A . x = b$, $x \geq 0$ [4].

Definícia 2.19: Úloha bivalentného (alebo binárneho) lineárneho programovania je nájsť takú n-ticu celých čísel x, pre ktorú je $c^T . x$ minimálne za predpokladov $A . x = b$, $x_i \in \{0, 1\}$ pre $i = 1, 2, \ldots, n$ [4].

Definícia 2.20: Hovoríme, že problém je NP-ťažký, ak úlohu bivalentného lineárneho programovania možno polynomiálne redukovať na *P*.
Definícia 2.21: Hovoríme, že problém *P* je NP-ľahký, ak problém *P* možno polynomiálne redukovať na úlohu bivalentného lineárneho programovania
Definícia 2.22: Hovoríme, že problém *P* je NP-ekvivalentný, ak je problém *P* polynomiálne ekvivalentný s úlohou bivalentného lineárneho programovania.

2.6 Newickov formát

V matematike, je Newickov formát spôsob ako reprezentovať stromy spoločné s dĺžkami hrán použitím zátvoriek a čiarok. Ak chceme reprezentovať strom, ktorý nemá koreň, ľubovoľný vrchol stromu zvolíme ako jeho koreň. Koreň je zvyčajne vnútorný vrchol stromu, výnimočne sa používa listový vrchol. Každý vnútorný vrchol zapíše ako pár zátvoriek, vnútri ktorého sú jeho potomkovia oddelení čiarkami. Ak je potomok list, uvedie sa názov druhu, ak je to vnútorný vrchol uvedie sa ďalšia zátvorka, v ktorej vnútri sú opäť uvedený potomkovia. Newickov formát je vhodný aj pre fylogenetické stromy. Na obrázku 2.3 je uvedený fylogenetický strom, ktorého Newickov formát je tvaru (((*A*, *B*), (*C*, *D*)), *E*).



Obrázok 2.4 Fylogenetický strom určený na zápis v Newickovom formáte.

2.7 Analýza fylogenetického stromu

2.7.1 Metóda maximálnej úspornosti

Metóda maximálnej úspornosti sa používa najmä na blízke druhy, ktorých sekvencie sa len pomerne málo navzájom líšia. Jej cieľom je nájsť evolučnú históriu druhov, ktorá vysvetľuje dnešné sekvencie z niektorých organizmov. Na vstupe je viacnásobné zarovnanie sekvencií z niekoľkých organizmov. Cieľom je nájsť fylogenetický strom, ktorý bude mať dané organizmy v listoch a tiež nájsť predpokladanú dedičnú sekvenciu dĺžky *n* pre každý vnútorný vrchol stromu. Zo všetkých možných stromov a dedičných sekvencií sa vyberú také, ktoré majú minimálny počet mutácii v evolučnej histórii, a to tak, že prejdeme všetkými hranami stromu a spočítame počet miest, na ktorých sa

sekvencie na koncoch hrany líšia. To je najmenší počet mutácií, ktorý nastal na tejto hrane. Tento problém patrí medzi NP-ťažké problémy.

2.7.2 Metóda spájania susedov

Na rozdiel od metódy maximálnej úspornosti metóda spájania susedov nedostáva na vstupe celé viacnásobné zarovnanie m sekvencií, ale len maticu M s rozmermi m x m, ktorá obsahuje vzdialenosti medzi jednotlivými sekvenciami ($M_{i,j}$ je vzdialenosť sekvencií i a j). Ak je na vstupe strom, kde každá hrana má dĺžku, na určenie vzdialenosti medzi uzlami stačí spočítať dĺžky hrán na nej. Metóda spájania susedov sa snaží pre maticu M nájsť zodpovedajúci strom S s dĺžkami hrán, tak aby $M(S) = M$. Toto má za následok, že pre každú tabuľku M nemusí existovať strom. Matice, pre ktoré existuje, voláme aditívne. Zložitosť je $O(n^3)$.

2.7.3 Metóda maximálnej vierohodnosti

Táto metóda je podobná metóde maximálnej úspornosti s tým rozdielom, že namiesto hľadania stromu, ktorý vyžaduje čo najmenej mutácií, hľadáme strom, ktorý sa bude zdať najvierohodnejší vzhľadom na určitý model evolúcie. Do úvahy berieme aj dĺžky hrán, na ktorých nastalo viac mutácií ako na kratších. Vstupom je opäť zarovnanie m sekvencií, pričom neuvažujeme medzery. Výstupom je strom, ktorý má tieto sekvencie v uzloch a dĺžky hrán stromu. Zo všetkých možných stromov vyberieme tie, ktoré najviac zodpovedajú vstupných sekvenciám. Toto meriame veličinou zvanou vierohodnosť. Fylogenetický strom je potrebné reprezentovať ako pravdepodobnostný model. Zložitosť tejto metódy je NP-ťažký.

Metóda je konzistentná ak v prípade, že dĺžka sekvencií n rastie do nekonečna, pravdepodobnosť, že metóda nájde správny strom sa blíži k jednej. Táto pravdepodobnosť sa počíta cez všetky zarovnania, ktoré mohol model vygenerovať. Jedine algoritmus maximálnej úspornosti nie je konzistentný.

2.8 Generovanie náhodných stromov

Nástroje simulujúce evolúciu znakov pozdĺž fylogenetického stromu sa typicky nevyskytujú, alebo nevyhovujú našim požiadavkám. Z tohto dôvodu využijeme na to metódy používané na generovanie počiatočnej populácie v genetickom programovaní. Stromy sa väčšinou generujú od koreňového vrcholu rekurzívnym algoritmom. Pre generovaný uzol sa náhodne určí, či sa má ďalej rozvíjať alebo je to konečný uzol, Ak je výsledok ďalšie generovanie, náhodne sú vygenerované dva uzly. Ako vstupné parametre na generovanie sa použijú parametre ich predchodcu. Na obmedzenie zložitosti vygenerovaných stromov je vhodné určiť maximálnu hĺbku, čím sa každý vrchol v tejto maximálnej hĺbke označí ako konečný.

2.9 Algoritmus v analyzátore

Počet chromozómov budeme označovať ako $2N$, veľkosť genómu ako $2C$. Úlohou algoritmu, ktorý má byť overený, je určenie podielu mechanizmov uvedených v kapitole 2.2 na veľkosť genómu (jeho hmotnosť) a počtu chromozómov. Algoritmus má k dispozícii iba koncové uzly fylogenetického stromu, a to ich veľkosť genómu a počet chromozómov. Pre každú vetvu je odhadnutý podiel mechanizmu a pre každý vnútorný uzol pôvodný stav $2C$, a $2N$. Algoritmus je zložený z týchto krokov:

1. Nastavenie hodnôt koncových uzlov a dĺžok vetiev,

2. Výpočet relatívneho rozdielu pre *2C* a *2N* medzi sesterskými uzlami,
3. Odhad pravdepodobnosti použitia mechanizmov evolúcie genómu a chromozómov pre vetvy. Možnosti, ktoré môžu nastať na vetvách sú zobrazené na obrázku 2.5,
4. Odhad *2C* a *2N* vnútorných uzlov, a to na základe stavu ich bezprostredných potomkov,
5. Výpočet rozdielov na vetvách, odhad absolútneho a relatívneho zastúpenia mechanizmov pre každú vetvu,
6. Výpočet absolútneho a relatívneho zastúpenia určitého mechanizmu na konkrétnej vetve a celom strome.

Obrázok 2.5 Možné mechanizmy a ich kombinácie na vetvách stromu.

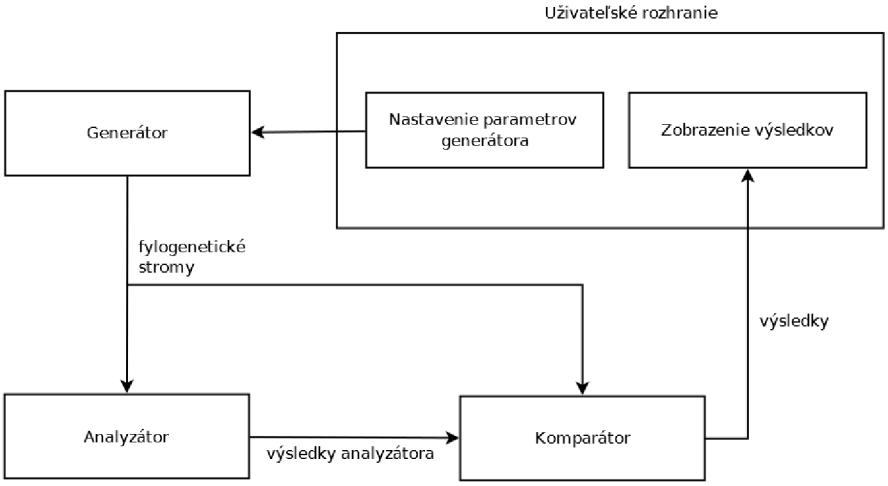
3 Návrh aplikácie

V realite je počet chromozómov celé číslo, pre naše potreby budeme pracovať s reálnymi číslami, pretože v konečnom dôsledku toto nebude mať vplyv na výsledky overenia spoľahlivosti existujúceho analyzátora. Pôvodne analyzátor je napísaný v jazyku R, používa metódu maximálnej úspornosti a na jeho vstupe sa používa fylogenetický strom v Newickovom formáte. Každý vrchol v strome má uvedený počet chromozómov a veľkosť genómu. Keďže do tohto analyzátora nie je možné zadávať viacero stromov súčasne, je potrebné z neho získať algoritmus, ktorý použijeme v našej aplikácii.

Užívateľ bude môcť nastaviť tieto parametre:

- počet vygenerovaných stromov,
- rozsah, v rámci ktorého sa bude náhodne generovať počet uzlov v strome,
- pravdepodobnosť výskytu mechanizmu,
- minimálnu a maximálnu hodnotu koeficientu použitého pri konkrétnom mechanizme,
- pri polyploidii nastavenie pravdepodobnosti výskytu jej podtypov, napríklad 2x/4x,
- nastavenie pravdepodobnosti vygenerovania ľavého alebo pravého potomka. Toto umožní generovanie rôznych tvarom stromov,
- čas medzi dvomi generovanými uzlami,
- počiatočné hodnoty veľkosti genómu a počtu chromozómov,
- minimálne hodnoty pre veľkosti genómu a počet chromozómov,
- neexpandovanie uzla, ak nespĺňa obmedzenia,
- importovanie konfigurácie,
- exportovanie konfigurácie,
- nastavenie cesty k výstupnému súboru.

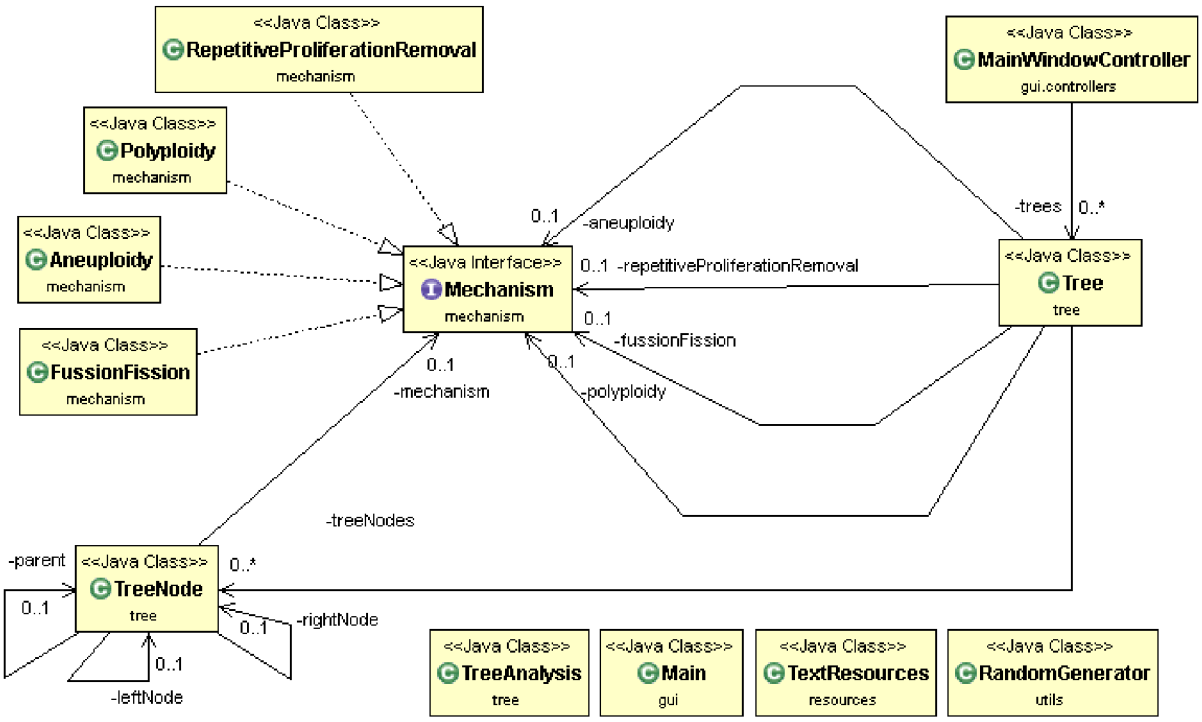
Úlohou aplikácie bude generovanie fylogenetických stromov s parametrami, ktoré zadá užívateľ. Tieto stromy následne pôjdu do analyzátora, ktorý získa rozdelenie použitia mechanizmov. V komparátore sa dáta porovnajú so skutočnými hodnotami, ktoré boli použité pri vytváraní stromov. Výsledky sa budú generovať textového súboru. Na obrázku 3.1 je uvedená schéma aplikácie.



Obrázok 3.1 Návrh aplikácie.

4 Implementácia

Táto časť popisuje implementáciu validátora, použité technológie, spracovanie dát. Na obrázku 4.1 je uvedený diagram tried aplikácie. Na jeho vytvorenie som použil plugin ObjectAid pre vývojové prostredie Eclipse



Obrázok 4.1 Diagram tried aplikácie.

4.1 Užívateľské rozhranie

Užívateľské rozhranie umožňuje používateľovi nastaviť rôzne parametre pre generovanie fylogenetických stromov s náhodnými parametrami. Na vytvorenie užívateľského rozhrania sme si zvolili platformu JavaFX. JavaFX je nástupcom dlhodobo používanej knižnice Swing. JavaFX umožňuje prehľadne oddeliť grafické komponenty a ich umiestnenie v aplikácii od funkčnej časti. V JavaFX sú dve možnosti ako tvoriť užívateľské rozhranie.

Prvý je spôsob umiestniť všetky komponenty spoločne s ich funkciami priamo v kóde. Druhá možnosť je použiť FXML formát. FXML je deklaratívny jazyk založený na XML. V ňom môžeme jednoducho vytvárať aplikácie. Spoločne s FXML je možné používať technológiu JavaFX Scene Builder. JavaFX Scene Builder je vizuálny nástroj pre urýchlenie dizajnu aplikácie bez kódovania. Užívateľ si môže poskladať UI komponenty pomocou techniky zvanej ťahaj a pusti (drag and grop). Ďalšou výhodou Javy FX je oddelenie štýlov komponent do CSS súborov. JavaFX podporuje aj jazykovú lokalizáciu. Môžeme priamo využiť triedu z Javy `ResourceBundle` vytvárajúcu objekty určené pre lokalizáciu.

Ako základný grafický prvok sme si zvolili layout `GridPane`. Tento umožňuje rozdeliť okno na mriežky so zadaným počtom riadkov a stĺpcov. Riadky a stĺpce nemusia byť nutne rovnaké.

Pre každú komponentu je potrebné určiť, v ktorom riadku a stĺpci sa nachádza. V aplikácii

používame 4 typy komponentov, a to `Label`, `TextField`, `CheckBox`, `Button`.

`Label` umožňuje vypísať text pre užívateľa, ale neumožňuje jeho následnú zmenu. `TextField` je jeden zo základných prvok pre vstup od užívateľa. Užívateľ môže zadať do neho text, ktorý môžeme ďalej spracovať.

Prvky typu `TextField` využívame na zadávanie vstupných parametrov od užívateľa, napríklad pravdepodobnosť výskytu určitého mechanizmu.

`CheckBox` je zatrhávacie tlačidlo, a má dve hodnoty: zatrhnuté, nezatrhnuté. Užívateľ si môže zvoliť či prvok, ktorý presiahne určené hodnoty má byť vymazaný zo stromu. Komponenty `Button` sú tlačidla. Tieto majú k sebe priradené udalosti, ktoré sa vykonajú keď užívateľ na ne klikne. Tlačidla používame pre generovanie stromu, exportovanie a importovanie konfigurácie.

Aby mohla byť aplikácia v JaveFX spustená, je potrebné vytvoriť práve jednu triedu, ktorá bude dediť z triedy `javafx.application.Application`. Štart aplikácie začne zavolaním statickej metódy `launch`. To spôsobí vytvorenie nového vlákna pre aplikáciu. JavaFX aplikácia sa ukončí vtedy, keď nie je vykreslené ani jedno okno aplikácie. V triede `Main` vytvoríme jedno okno aplikácie so štandardným dizajnom. V tejto triede sa načíta FXML súbor `MainWindowView.fxml`. Tento vytvorí riadiacu triedu `MainWindowController`, pomocou ktorej ovládame reakcie na akcie užívateľa v užívateľskom rozhraní. Trieda `MainWindowController` implementuje metódu `initializable` rozhrania `Initializable`. Táto metóda sa zavolá bezprostredne po vytvorení všetkých komponentov aplikácie. V nej nastavujeme základné hodnoty vstupných parametrov. Po stlačení tlačidla Generate, spracujem a skontrolujem vstupné parametre.

Aplikácia obsahuje na hornom riadku menu. V prvej položke môže uložiť, prípadne načítať konfiguračné nastavenia. Druhá položka slúži na nastavenie cesty k výstupnému súboru aplikácie. Jeho meno je nastavené na output.txt. V prípade, že táto cesta nebola nastavená , použije sa cesta, odkiaľ sa aplikácia spustila. Na spodnom riadku aplikácie je stavový riadok, ktorý zobrazuje úspech, neúspech užívateľovej akcie. Na obrázku 4.2 je ukážka aplikácie validátora.

4.2 Vstupné parametre

Parametre aplikácie nie sú triviálne, je potrebné po ich zadaní ich skontrolovať. Ak dôjde k chybe, užívateľ je na ňu upozornený chybou, ktorá sa objaví v stavovom riadku aplikácie.
Počet stromov a počet jeho uzlov musí byť celé číslo typu `int`. Všetky ostatné hodnoty môžu byť reálne čísla typu `double`.

Parametre sa overujú v metóde s názvom `checkParameters`, ktorá vráti `true`, ak nenastala žiadna chyba. Pre pravdepodobnosti je potrebné kontrolovať či sú v rozsahu <0, 1>. Súčet pravdepodobností pre mechanizmy musí byť rovný 1. Toto platí aj pre súčet pravdepodobností typu polyploidie. Počet generovaných stromov musí byť kladné číslo. Zadaný počet uzlov musí byť kladný, a maximálna hodnota musí rovná alebo väčšia ako minimálna hodnota. Ak sa tieto dve hodnoty rovnajú, rozsah generovanie bude rovný 1. Čas medzi dvoma delenia uzlov musí byť kladné číslo.
Pre počet chromozómov je potrebné určiť počiatočnú hodnotu, ktorá sa priradí koreňovému uzlu. Táto hodnota musí byť kladná. Ak je minimálna alebo maximálna hodnota počtu chromozómov zadaná ako záporné číslo, potom sa táto hodnota ignoruje a nedochádza ku kontrole u tohto parametru. Použitá je konštanta `Double.NEGATIVE_INFINITY` pre minimálnu hodnotu, respektíve konštanta `Double.POSITIVE_INFINITY` pre maximálnu hodnotu. Tieto dve konštanty reprezentujú kladné, záporné nekonečno. Obe hodnoty môžu byť záporné, jedna z nich alebo žiadna.

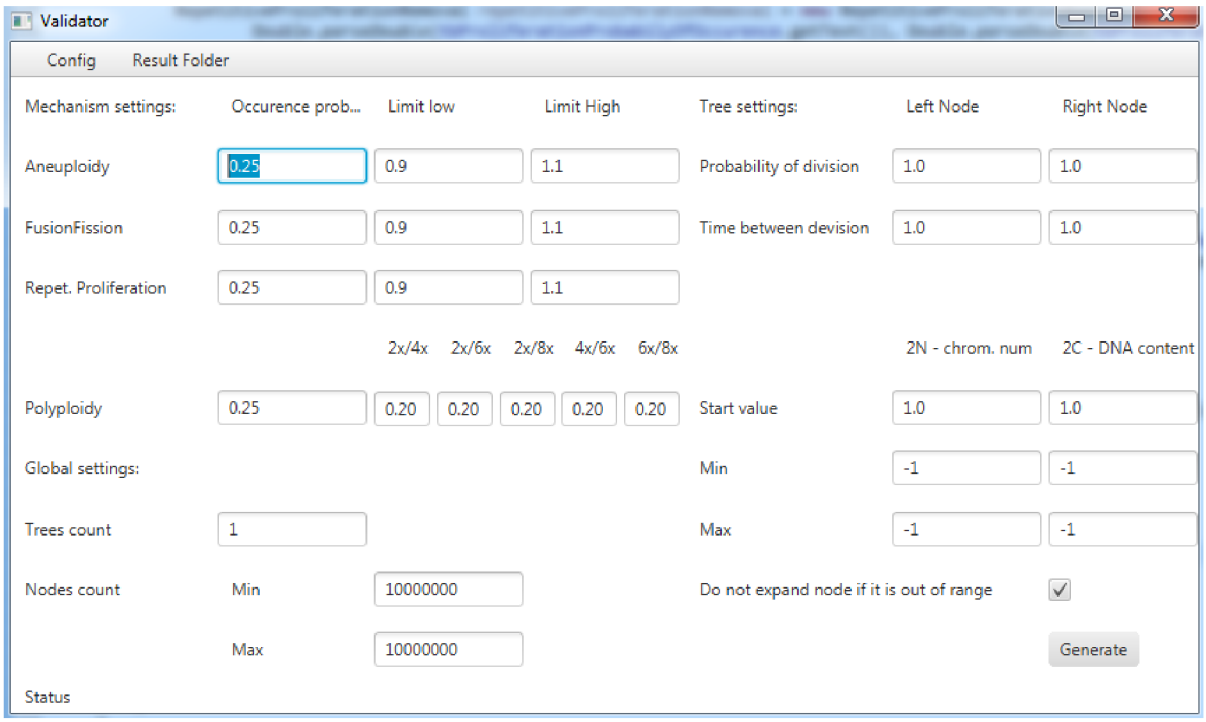
Ak ani jedna nie je záporná, potom musí minimálna hodnota byť menšia ako maximálna. Tento princíp je uplatnený aj pre veľkosť genómu.

Minimálne a maximálne hodnoty sa berú do úvahy jedine vtedy ak je zatrhnutá možnosť neexpandovania uzla, ktorý je mimo hraníc.

Trieda Javy `Random` vráti náhodné číslo typu `double` v rozsahu <0, 1). Tento rozsah je potrebné prepočítať na užívateľom zadaný rozsah. Použitý je vzťah

$$coef = (\max - \min) \cdot n_{rand} + \min , \tag{4.1}$$

kde *coef* je výsledný koeficient, *max* je maximálna hodnota zadaná užívateľom, *min* je minimálna hodnota zadaná užívateľ, n_{rand} je náhodne vygenerované číslo. Toto zabezpečí generovanie hodnôt v rozsahu <minimum, maximum).



Obrázok 4.2 Aplikácia validátora. V hornom riadku sa nachádza menu, kde užívateľ môže uložiť, načítať konfiguračné dáta. Stredná časť obsahuje možnosť nastavenia parametrov. Spodný riadok je stavový riadok, kde sa užívateľovi zobrazí výsledok jeho akcie.

4.3 Výber mechanizmu na základe pravdepodobnosti

Mechanizmus s väčšou pravdepodobnosťou má väčšiu šancu na výber. Tento výber môžeme prirovnať k rulete, kde je obvod rulety rozdelený do oblastí rôznych veľkostí, a to podľa veľkosti pravdepodobnosti. Náhodne generované číslo má teda vyššiu šancu , že bude patriť do časti, ktorá patrí mechanizmu s vyššou pravdepodobnosťou, ako mechanizmu z nižšou pravdepodobnosť, pretože ten má pokrytú menšiu časť obvodu.

4.4 Triedy pre mechanizmy

Všetky štyri typy mechanizmov implementujú rozhranie `Mechanism`. Metóda `getChromosomeCount` vypočíta počet chromozómov pre daný uzol, a to z argumentu metódy, čo je počet chromozómov rodiča. Druhá metóda `getDNAContent` vypočíta veľkosť genómu pre uzol tiež na základe rodiča. Metóda `getProbability` vracia pravdepodobnosť výskytu mechanizmu. Táto hodnota sa nastavuje pred generovaním stromu. Keďže v niektorých mechanizmoch sa menia obe hodnoty súčasne s tým istým koeficientom, je potrebné určiť tento koeficient pred prepočtom. Na toto slúži metóda `refresh`, ktorá musí byť zavolaná pred `getDNAContent` a `getChromosomeCount`.

Každý typ mechanizmu má na vstupe pravdepodobnosť, minimálnu a maximálnu hodnotu koeficientu. Na základe vzťahu 4.1 vypočítame výsledný koeficient na prepočet rodičovských údajov na potomkov.

4.5 Triedy pre generovanie stromu

Uzol stromu reprezentuje trieda `TreeNode`. Táto obsahuje referencie na ľavého, resp. pravého potomka, na rodiča, mechanizmus, ktorý bol použitý pri vzniku uzlu, počet chromozómov, veľkosť genómu. Podobne obsahuje aj pravdepodobnosti výskytu určitého mechanizmu, tieto hodnoty sú nastavované pri analýze stromu. Pre každý parameter sú k dispozícii metódy pre `get` a `set`.

Trieda `Tree` popisuje strom. Metóda `create` vygeneruje nový strom skúmaného algoritmu pomocou algoritmu 4.1. Metóda `printMechanism` vypíše do súboru počet použitých mechanizmov a ďalšie informácie. Metóda `maxDepth` rekurzívne vypočíta hĺbku stromu.

Algoritmus 4.1: Algoritmu generovania stromu:

1. Vytvor koreňový uzol, a vlož ho do listu. Nastav index na 0. Nastav počet uzlov na 1
2. Ak počet uzlov je väčší alebo rovný ako maximálny počet uzlov, ukonči algoritmus. Inak pokračuj na krok 3.
3. Z listu vyber uzol, na ktorý ukazuje index.
4. Ak sa môže uzol expandovať, vytvor jeho oboch potomkov, vlož ich na koniec listu a k počtu uzlov pripočítaj 2.
5. Index zväčši o 1. Pokračuj na krok 2.

Každý generovaný strom inicializujeme s mechanizmami. Novo vytvorené uzly pridávam na koniec kolekcie `ArrayList`, pretože táto umožňuje pristupovať k jednotlivým prvkom v liste pomocou metódy `get`, čo je podobný prístup ako k prvku v poli. Keďže každý uzol musí mat vždy dvoch potomkov, musím v určitých prípadoch zväčšiť maximálny počet generovaných uzlov o 1. Toto závisí od tvaru stromu.

4.6 Trieda pre analýzu stromu

Trieda pre analýzu stromu je `TreeAnalysis`. Je to trieda so statickými metódami.

Algoritmus 4.2: Algoritmus analýzy stromu:

1. Nastav index *i* na index predposledného prvku v liste.
2. Ak je index *i* väčší alebo rovný 1 pokračuj na krok 3. V opačnom prípade ukonči algoritmus.

3. Z listu vyber uzly s indexom *i* a *i* + 1.
4. Na tieto dva uzly aplikuj analýzu. Vypočítaj hodnoty ich rodiča, typ použitého mechanizmu.
5. Index *i* zmenši o 2.
6. Pokračuj na krok 2.

Metóda `analyse` spustí analýzu stromu. Využíva algoritmus 4.2. Algoritmus analyzátora predpokladá v niektorých prípadoch, že ľavý potomok má menšiu veľkosť genómu. Preto pred začiatkom analýzu musí vymeniť ľavého potomka za pravého a opačne. Najprv sa zistí, o aký mechanizmus sa jedná. Druhá časť pozostáva z určenia pravdepodobnosti mechanizmu, a určenie poctu chromozómov a veľkosť genómu pre uzol rodiča. Metóda `computeAbsRelTotals` vypočíta absolútny a relatívny podiel mechanizmov na strome.
V algoritme v prílohe 1 sa častokrát vyskytuje rovnosť medzi dvoma číslami. Z tohto dôvodu sme si vytvorili metódu `eq`, ktorá porovnáva dva čísla s určitou presnosťou.
Čísla *a* a *b* sa rovnajú ak platí vzťah

$$abs(a-b) < \varepsilon . \tag{4.2}$$

V opačnom prípade sa čísla nerovnajú. Ako ε som zvolil hodnotu 0.0001.

5 Spracovanie výsledkov a diskusia

V tejto kapitole sa zaoberám spracovaním a diskusiou výsledkov, ktoré som získal z validátora. Počet vetiev strome je rovný poctu uzlov stromu bez koreňového uzla.

5.1 Percentuálny podiel rozdelenia mechanizmov

Ako prvé sme museli porovnať zastúpenie jednotlivých mechanizmov v strome. Skutočný počet jednotlivého mechanizmu získame už počas generovania fylogenetického stromu. V rámci analýzy stromu ukladám odhadované zastúpenie mechanizmov. Oba tieto výsledky vydelíme sumou všetkých výskytov mechanizmov. Táto suma môže byť pre každý prípad rozdielna. Pre reálne dáta sa rovná počtu vetiev, respektíve uzlov v strome. Celkovú sumu pre dáta z analyzátora získame ako súčet jednotlivých počtov mechanizmom. Celková suma je ešte znížená o uzly, ktoré nebolo možné zatriediť do žiadnej z vetiev algoritmu (príloha 1), nakoľko nepokrýva všetky existujúce možnosti. Výsledky sú zobrazené do histogramu, kde zvislá os je percentuálny podiel daného mechanizmu, vodorovná os označuje typ mechanizmu.

5.2 Relatívny a absolútny podiel mechanizmu

Ďalšou úlohou bolo porovnať relatívny a absolútny podiel mechanizmu na vetve a to podľa vzťahov z prílohy 1:

$$\begin{aligned} d_{abs\_2C} &= abs(2C_P - 2C_R) \\ d_{abs\_2N} &= abs(2N_P - 2N_R) \\ d_{rel\_2C} &= \max(2C_P / 2C_R, 2C_R / 2C_P) - 1 \\ d_{rel\_2N} &= \max(2N_P / 2N_R, 2N_R / 2N_P) - 1 \end{aligned} \tag{5.1}$$

kde d_{abs_2C} obsahuje absolútny rozdiel pre veľkosť genómu, $2C_P$ označuje veľkosť genómu potomka, $2C_R$ veľkosť genómu rodiča. d_{abs_2N} pre počet chromozómov, $2N_P$ označuje počet chromozómov potomka, $2N_R$ označuje počet chromozómov rodiča, d_{rel_2C} je relatívny rozdiel pre veľkosť genómu, d_{rel_2N} je relatívny rozdiel pre veľkosť genómu.

Pre dáta z analyzátora je počítaná pravdepodobnosť, s akou sa daný mechanizmus vyskytuje na vetve. Táto pravdepodobnosť je počítaná podľa vzťahov uvedených v prílohe 1 a môže mať hodnotu z intervalu <0, 1>. Pre skutočné dáta má táto pravdepodobnosť hodnotu buď 0 alebo 1, pretože je jasné, aký mechanizmus sa použil na konkrétnej vetve.

Čiastková absolútna rola r_{abs} mechanizmu x na vetve V je potom daná vzťahom

$$r_{abs} = p_x(V) \cdot d_{abs}(V), \tag{5.2}$$

kde p_x je pravdepodobnosť mechanizmu x na vetve V, d_{abs} je absolútny podiel mechanizmu x na vetve V.

Čiastková relatívna rola r_{rel} mechanizmu x na vetve V je daná vzťahom

$$r_{rel} = p_x(V) \cdot d_{rel}(V), \tag{5.3}$$

kde p_x je pravdepodobnosť mechanizmu x na vetve V, d_{rel} je relatívny podiel mechanizmu na vetve V.

Celkový podiel mechanizmu d_{celk} je daný ako súčet všetkých čiastočných podielov

$$d_{celk\_abs} = \sum_{i=1}^{N} p_x(i) \cdot d_{abs}(i) \tag{5.4}$$

$$d_{celk\_rel} = \sum_{i=1}^{N} p_x(i) \cdot d_{rel}(i) \tag{5.5}$$

kde N je počet vetiev (uzlov), i označuje konkrétny uzol. Tieto hodnoty počítame aj pre skutočné dáta. Keďže odhadovaný výsledný počet chromozómov a takisto veľkosť genómu v celom strome sa líšil od skutočných dát, bolo potrebné upraviť relatívne a absolútne podiely mechanizmov na zmene *2C* a *2N*. Preto sme tieto relatívne a absolútne podiely sčítali a touto sumou vydelil pôvodné podiely. Tak sme získali relatívny (percentuálny) podiel vplyvu mechanizmu na zmenu chromozómov a veľkosti genómu. Toto sme aplikovali pre skutočné, ale aj pre dáta z analyzátora. Na ich porovnanie sme použili známy vzťah pre výpočet relatívnej odchýlky,

$$\varepsilon_r = \frac{\Delta X}{X} = \frac{X - x}{X} = 1 - \frac{x}{X} \tag{5.6}$$

kde X je skutočná hodnota, x je nameraná hodnota, ε je relativna chyba merania.

5.3 Analýza výsledkov a diskusia

V nasledujúcej tabuľke sú uvedené hodnoty, ktoré sme použili pre generovanie stromu.

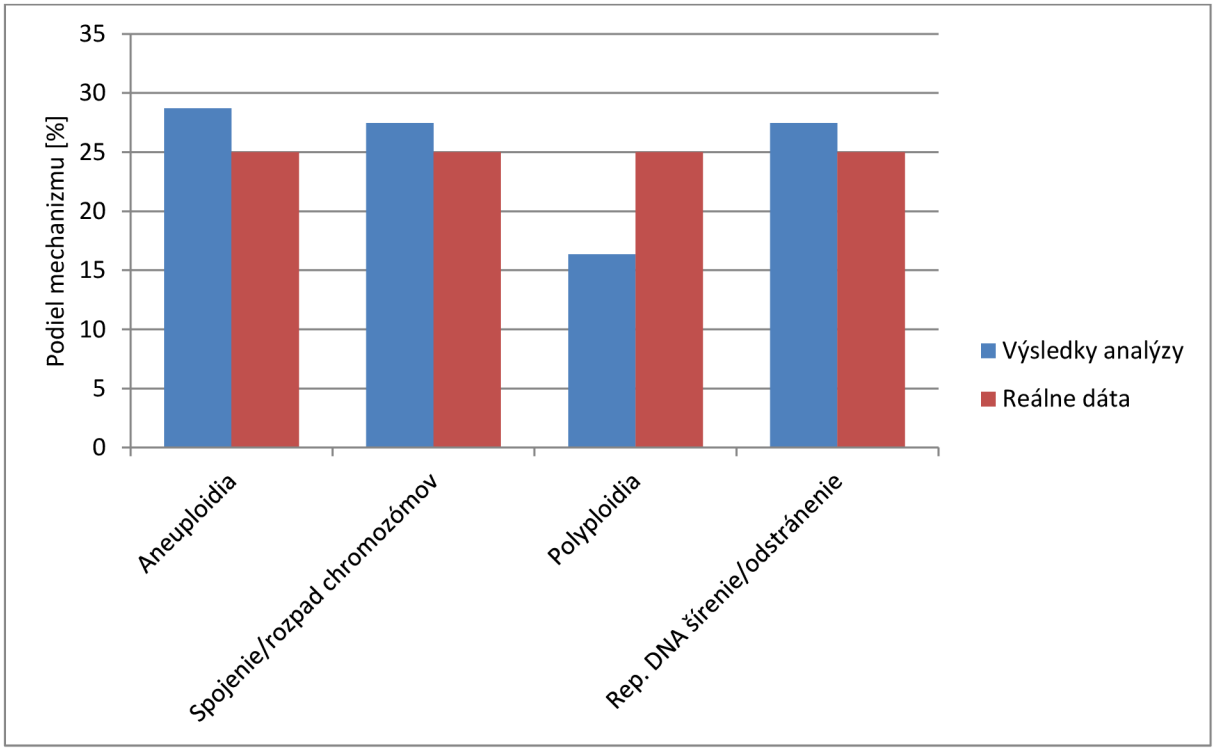
Parametre	**Hodnoty**
Min. hodnota koeficientu	0,25
Max. hodnota koeficientu	4
Pravdepodobnosti podtypov polyploidie	0,2
Počet stromov	1000
Min počet uzlov v strome	5000000
Max. počet uzlov v strome	10000000
Počiatočná hodnota pre 2C	1
Počiatočná hodnota pre 2N	1
Expandovanie všetkých uzlov	áno
Pravdepodobnosť delenia ľavého uzla	1
Pravdepodobnosť delenia pravého uzla	1

Tabuľka 5.1: Použite parametre pre testovanie validátora

5.3.1 Rovnomerné rozdelenie pravdepodobnosti

Ako prvý prípad sme si zvolili rovnomerného rozdelenie pravdepodobnosti výskytu mechanizmov, čiže každý typ mechanizmu sa vyskytuje s pravdepodobnosťou 25%. Z grafu 5.1 vidieť že algoritmus v analyzátora dáva pomerne dobre odhady pre rozdelenie pravdepodobnosti. Jedinú výnimku tvorí polyploidia, ktorá má výrazne nižšiu hodnotu. Uzly, ktoré boli generované s polyploidiou, sa teda rovnomerne rozdelili medzi ostatné mechanizmy.

V tabuľke 5.1 je uvedená relatívna chyba. Napriek tomu, že pomerné zastúpenie vyšlo celkom presne, pomer relatívnych a absolútnych roli sa v niektorých hodnotách líši o takmer 90%, pričom hodnota okolo 3-4% sa väčšinou považuje ako úspešné meranie. Kladná hodnota naznačuje, že odhadnutý podiel bol menší ako bol v skutočnosti. Vidíme teda, že daný algoritmus nie celkom presne odhaduje tieto podiely.



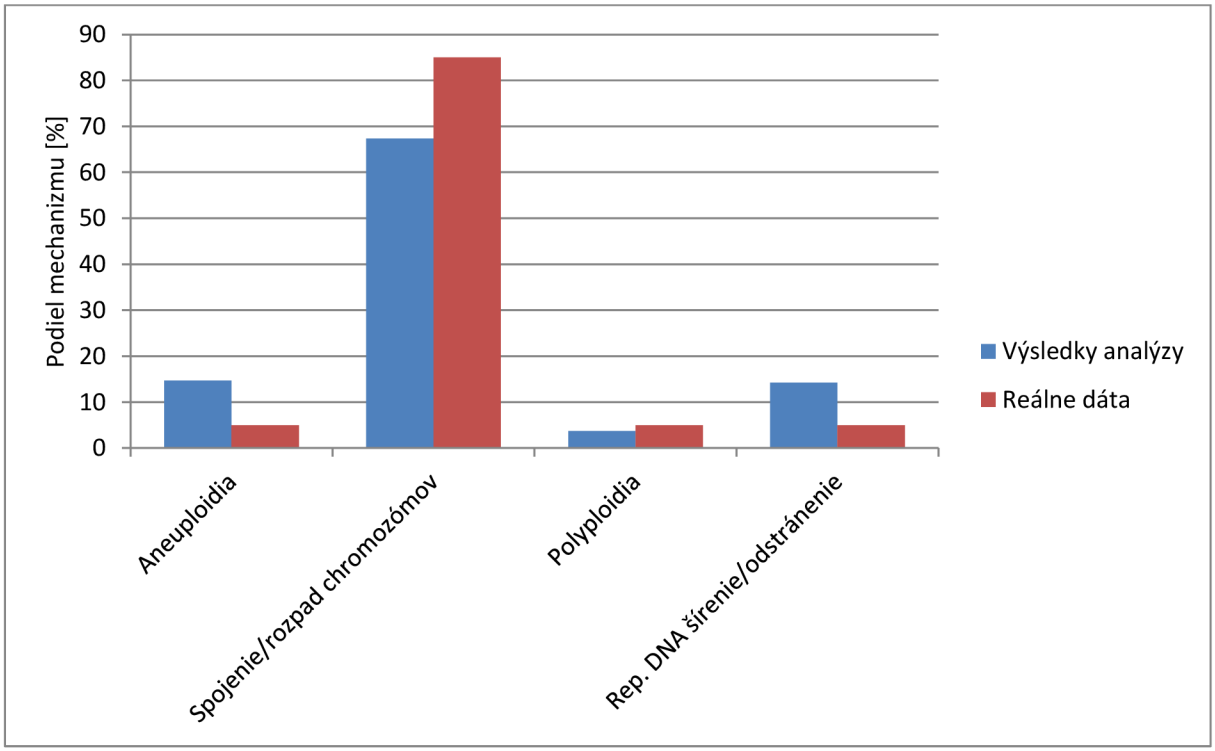
Graf 5.1: Percentuálny podiel mechanizmu získaný z reálnych dát a z výsledkov analýzy.

	Relatívna odchýlka [%]			
	rel. 2C	**abs. 2C**	**rel. 2N**	**abs. 2N**
Aneuploidia	88,20	86,74	87,89	86,38
Polyploidia	-81,08	-66,27	-74,71	-63,71
Spojenie/rozpad chromozómov	0,00	0,00	-44,00	-46,65
Rep. DNA šírenie/odstránenie	-40,60	-43,89	0,00	0,00

Tabuľka 5.1: Výsledky relatívnej odchýlky

5.3.2 Najväčšia pravdepodobnosť pre spojenie, rozpad chromozómov

V tomto prípade sme nastavili pravdepodobnosť použitia mechanizmu rozpadu chromozómov na 85%. Ostatné mechanizmy mali pravdepodobnosť 5%. Ako vidíme z grafu 5.2, algoritmus správne odhadol rozdelenie pravdepodobnosti. Šírenie/rozpad chromozómov má výrazne vyššie zastúpenie. Z tabuľky 5.2 je zrejmé, že relatívna chyba pre absolútny a relatívny podiel pre tento mechanizmus je < 5%, čo môžeme považovať ako správne meranie. Ostatné hodnoty opäť výrazne prekračujú povolené limity.



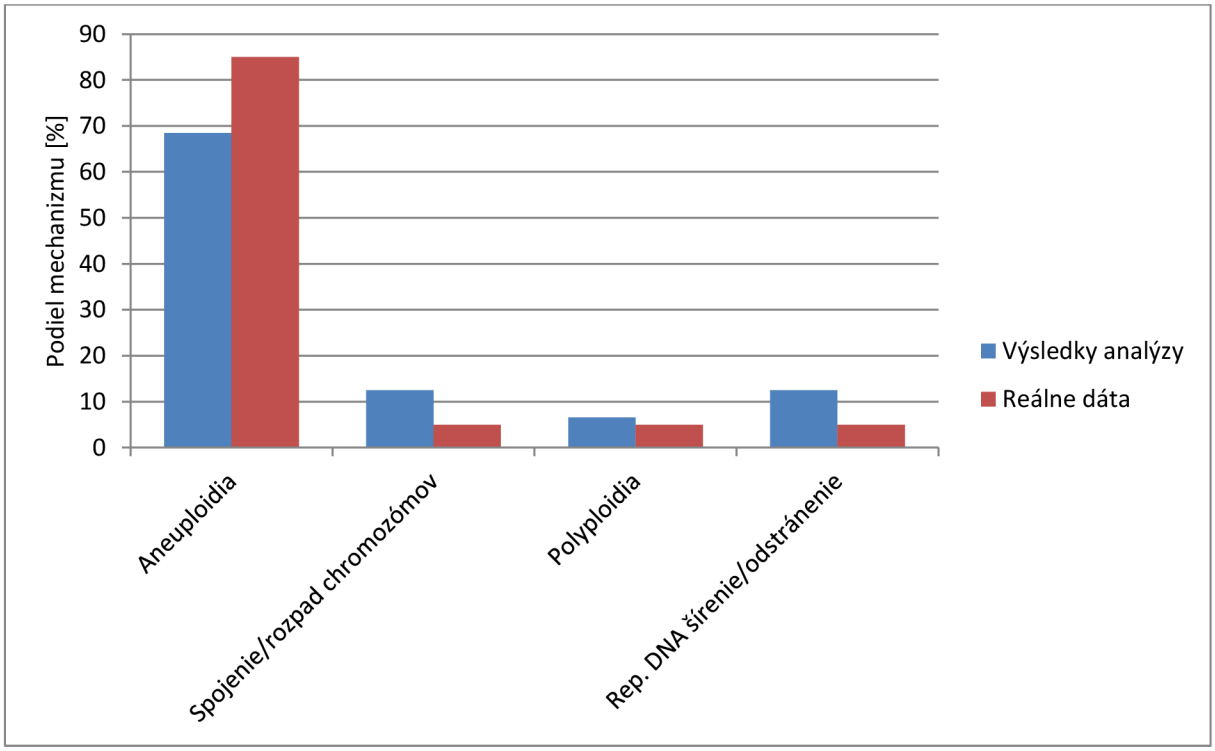
Graf 5.2: Percentuálny podiel mechanizmu získaný z reálnych dát a z výsledkov analýzy.

	Relatívna odchýlka [%]			
	rel. 2C	**abs. 2C**	**rel. 2N**	**abs. 2N**
Aneuploidia	86,17	87,22	59,78	59,68
Polyploidia	-59,69	-30,33	-205,40	-223,15
Spojenie/rozpad chromozómov	0,00	0,00	3,56	4,28
Rep. DNA šírenie/odstránenie	-50,94	-69,93	0,00	0,00

Tabuľka 5.2: Výsledky relatívnej odchýlky

5.3.3 Najväčšia pravdepodobnosť pre aneuploidiu

Aneuploidia ma pravdepodobnosť 85%. Graf 5.2 aj v tomto prípade vyšiel podľa očakávania. Algoritmus je teda účinný pri rozhodovaní typu mechanizmu. Z tabuľky 5.3 vidíme, že relatívna chyba polyploidie je takmer 2400%, teda skutočný podiel polyploidie bol menší. Problém môže byť v tom, že algoritmus nesprávne detekuje rozdiely medzi aneuploidiou a polyploidiou, keďže obidve súčasne mení počet chromozómov a veľkosť genómu.



Graf 5.3: Percentuálny podiel mechanizmu získaný z reálnych dát a z výsledkov analýzy.

	Relatívna odchýlka [%]			
	rel. 2C	**abs. 2C**	**rel. 2N**	**abs. 2N**
Aneuploidia	92,64	90,34	92,52	90,18
Polyploidia	-2383,44	-2182,54	-2373,25	-2207,78
Spojenie/rozpad chromozómov	0,00	0,00	-187,57	-161,24
Rep. DNA šírenie/odstránenie	-183,67	-157,61	0,00	0,00

Tabuľka 5.3: Výsledky relatívnej odchýlky

6 Záver

Táto práca sa problematikou fylogenetických stromov. Cieľom práce bolo overiť správnosť výsledkov, ktoré sa získajú vďaka algoritmu. Vytvorená aplikácia poskytuje grafické užívateľské rozhranie pre nastavenie parametrov na generovanie stromu. Nakoľko užívateľ musí nastaviť väčšie množstvo parametrov, môže si tieto nastavenia exportovať alebo importovať. Aplikácie overuje správnosť zadaných parametrov, užívateľ je o chybe informovaný pomocou stavového riadku. Pri vývoji aplikácie sme sa snažili použiť najnovšie dostupné technológie, a to Java 8 a JavaFX na vytvorenie grafického rozhrania.

Skúmaný algoritmus dosahuje veľmi dobré výsledky v rámci určenia typu mechanizmu. Poradil si aj s extrémami, teda keď jeden mechanizmus mal výrazne vyššiu pravdepodobnosť použitia. Výsledky sú zobrazené v histogramoch. Okrem toho má algoritmus za úlohu zistiť podiel určitého mechanizmu na zmene veľkosti genómu a počte chromozómov. Podobný postup, aký používa algoritmus, sme aplikovali aj na spracovanie skutočných výsledkov. Rozdiel sme interpretovali za pomoci relatívnej odchýlky. Najväčšie odchýlky sa prejavovali pri rozhodovaní medzi aneuploidiou a polyploidiou.

Pre zlepšenie algoritmu by mohla pomôcť bližšia analýza vygenerovaného stromu a to preskúmanie vplyvu mechanizmu na veľkosť genómu a počet chromozómov a následne lepší odhad stavu vnútorných uzlov. V aplikácii by bolo potrebné lepšie spracovanie výsledkov, prípadne ich zobrazenie priamo užívateľovi, napr. v grafe. Takisto algoritmus neumožňuje exportovanie, prípadne importovanie fylogenetických stromov.

Literatúra

[1] GREGORY, Ryan. *Understanding Evolutionary Trees*. Evo Edu Outreach 2008, 1:121-137.

[2] BREJOVÁ, Broňa, VINAŘ, Tomáš. *Metódy v bioinformatike*. Univerzita Komenského, Bratislava 2011.

[3] GÖMÖRY, Dušan. *Genetika.* Technická univerzita vo Zvolene, Zvolen 2014.

[4] PALÚCH, Stanislav. *Algoritmická teória grafov.* Žilinská univerzita, Žilina 2008.

[5] GÁLOVÁ, Eliška, et al. *Vybrané texty a príklady k cvičeniam z genetiky.* Univerzita Komenského, Bratislava 2004.

[6] Wikipédia. *Chromozóm* [online]. [cit. 2015-05-18]. Dostupné z: http://sk.wikipedia.org/wiki/Chromozóm.

[7] HONZÍK, Jan. *Algoritmy.* Vysoké učení technické, Brno 2007.

[8] Wikipédia. *Taxón* [online]. [cit. 2015-05-18]. Dostupné z: http://cs.wikipedia.org/wiki/Taxon.

[9] Wikipédia. *Deoxyribonukleová kyselina* [online]. [cit. 2015-05-18]. Dostupné z: http://sk.wikipedia.org/wiki/Deoxyribonukleová_kyselina.

Zoznam príloh