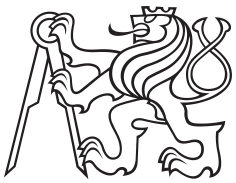
Diplomová práca



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačov

Automatická detekce intronů v genomech hub pomocí pravděpodobnostních modelů

Bc. Marek Zvara

Školiteľ: Doc. Ing. Jiří Kléma, Ph.D.
Odbor: Otvorená informatika
Zameranie: Umelá inteligencia
Máj 2019

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Zvara** Jméno: **Marek** Osobní číslo: **478061**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačů**

Studijní program: **Otevřená informatika**

Studijní obor: **Umělá inteligence**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Automatická detekce intronů v genomech hub pomocí pravděpodobnostních modelů

Název diplomové práce anglicky:

Automatic intron detection in fungal genomes using probabilistic models

Pokyny pro vypracování:

1. Seznamte se se strukturou DNA eukaryot.
2. Vypracujte rešerši stávajících pravděpodobnostních algoritmů automatické segmentace genomu.
3. Vybraný algoritmus implementujte v úpravě vhodné pro detekci intronů hub.
4. Nad daty dodanými vedoucím práce vytvořte modely pro jednotlivé genomy, pokuste se modely zobecnit pro příbuzné čeledě, rody, apod.
5. Vyhodnoťte dosažené výsledky.

Seznam doporučené literatury:

Testa, Alison C., et al. 'CodingQuarry: highly accurate hidden Markov model gene prediction in fungal genomes using RNA-seq transcripts.' BMC genomics 16.1 (2015): 170.
Stanke, Mario, et al. 'Gene prediction in eukaryotes with a generalized hidden Markov model that uses hints from external sources.' BMC bioinformatics 7.1 (2006): 62.
Yu, Ning, et al. 'A Comprehensive Review of Emerging Computational Methods for Gene Identification.' Journal of Information Processing Systems 12.1 (2016).

Jméno a pracoviště vedoucí(ho) diplomové práce:

doc. Ing. Jiří Kléma, Ph.D., Intelligent Data Analysis FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **28.01.2019** Termín odevzdání diplomové práce: **24.05.2019**

Platnost zadání diplomové práce: **20.09.2020**

doc. Ing. Jiří Kléma, Ph.D. podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Ing. Pavel Ripka, CSc. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

.	
Datum převzetí zadání	Podpis studenta

Poďakovanie

Z akademickej pôdy je to predovšetkým vedúci mojej diplomovej práce doc. Ing. Jiří Kléma, Ph.D., ktorému ďakujem za jeho čas, cenné rady a konzultácie pri tvorbe tejto práce.

Taktiež ďakujem svojím rodičom, starým rodičom a všetkým členom rodiny, ktorí pri mne stáli počas štúdia a podporovali ma.

Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval samostatne, a že som uviedol každú použitú literatúru.

V Prahe, 23. mája 2019

Abstrakt

Huby sú rozmanité spoločenstvo a pri veľa druhoch doposiaľ nepoznáme ich genóm a funkcie ich génov. Detekcia intrónových oblastí môže pomôcť k vzájomnému celogenómovému porovnaniu húb a určenie ich vzájomných príbuzností. Práca sa tak zameriava na analýzu genómu húb a jeho špecifiká a vlastnosti, aby bolo možné čo najlepšie určiť úzkalia a problémy, ktorým daná detekcia bude musieť čeliť. Okrem toho sa práca venuje aktuálnym pravdepodnostným prístupom detekcie génových oblastí, rozdeleniu a porovnaniu metód rôznych pravdepodobnostných modelov a nakoniec predstavuje aj state-of-the-art nástroje ako Augustus+ a CodingQuarry, aby sme mali prehľad o najlepších možných prístupoch v danej oblasti.

Vďaka tejto analýze bolo možné navrhnúť a implementovať pravdepodobnostný model, ktorý je založený na generalizovaných skrytých Markovských modeloch a za pomoci Viterbiho algoritmu pre generalizovaný skrytý Markovský model určiť najpravdepodobnejšiu sekvenciu stavov. V práci sa tak venujeme popisu návrhu a trikoch použitých pri samotnej implementácie ako napríklad urýchlenie Viterbiho algoritmu alebo rozdelenie detekcie do dvoch nadvezujúcich modelov, transkript model, ktorý slúži na označenie kódujúcich oblastí neznámeho genómu a genóm model, ktorý využíva tieto označenia a spresňuje detekciu pomocou zložitejšieho GHMM modelu. Výsledkom je teda nástroj, ktorý za pomoci vstupnej sekvencie a prípadných užívateľom špecifikovaných napovedajúcich anotácii dokáže detekovať jednotlivé génove úseky DNA sekvencie, ako promoter, exón, intrón, stop kodón a UTR oblasti. Výsledkom sú práve anotácie tejto detekcie a prípadná voliteľná vizualizácia v podobe HTML výstupu.

Práca sa venuje zovšeobecňovaniu daných modelov za použitia taxonómie húb. Porovnáva presnosti detekcie intrónových oblastí v rámci jednotlivých taxómov húb. Nástroj tak dokáže naučiť všeobecnejšie modely v rámci definovaného taxómu. V práci daný model podrobujeme veľkému množstvu rôznych experimentov, snažíme sa tak nájsť vhodné hyperparametre modelu a prebádať ako funguje daná detekcia naším nástrojom v prípade rôznych taxonomických úrovní. Zmyslom takéhoto zovšeobecňovania modelov je ich následne nasadenie na detekciu v neznámom metagenóme, v ktorom sa nachádza zmes genómov rôznych organizmov, z čoho najväčší podiel tvoria práve huby. Práca poukazuje na slabiny a výhody nasadenia takýchto všeobecnejších modelov na daný metagenóm.

Kľúčové slová: intróny, huby, skryté markovské modely, genóm, DNA

Školiteľ: Doc. Ing. Jiří Kléma, Ph.D.
Katedra počítačů,
Karlovo náměstí 13 ,
Praha 2

Abstract

Fungi are a diverse community and for many species we do not yet know their genome and the functions of their genes. Detection of intron regions may be helpful in cross-genome comparison of fungi and so we can better determine their interrelationships. The paper analyses the fungal genome and its specificities and properties in order to determine the bottlenecks and problems that the detection will have to face. In addition, the work addresses and compares current probability approaches for gene region detection. Moreover the paper introduces state-of-the-art tools such as Augustus+ and CodingQuarry to review the best possible approaches in the field.

With this analysis, it was possible to design and implement a probabilistic model based on generalized hidden Markov model. To determine the most probable sequence of states the model uses Viterbi algorithm for the generalized hidden Markov model. In this paper, we describe the design and the tricks used in the implementation itself, such as accelerating the Viterbi algorithm or splitting the detection into two emerging models, a transcript model that serves to designate the coding regions of the unknown genome and the genome model that uses these detected coding regions and refines the detection with more complex GHMM model. The result is a tool that, using the input sequence and any user-specified optional annotations, can detect individual gene sequences of a DNA sequence, such as promoter, exon, intron, stop codon, and UTR regions. The output is a list of annotations of this detection and optional visualization in the form of HTML output.

The thesis deals with generalization of given models using taxonomy of fungi. It compares the accuracy of intron detection within individual fungal taxa. Thus, the tool can train more general models within a defined taxa. In the paper, we discuss results of many different experiments with the model, trying to find the appropriate hyperparameters of the model and exploring how the detection works with our tool for different taxonomic levels. The purpose of this model generalization is to subsequently deploy them for detection of an unknown metagenome. This metagenom is a mixture of genomes of different organisms. Fungal genome takes the majority in this metagenom. The paper points out the weaknesses and advantages of these general models regarding the detection in metagenome.

Keywords: introns, fungus, hidden markov models, genome, DNA

Title translation: Automatic detection of introns in fungal genome using probabilistic models

Obsah

Obrázky

Tabuľky

Kapitola 1

Úvod

Huby a ich genóm sú v dnešnej dobe stále pomerne málo prebádaná oblasť biológie a nevie sa o nich tak moc v porovnaní s genómami vyšších eukaryot či ľudským genómom. V biológii preto existuje motivácia lepšie porozumieť hubám, aké gény tvoria ich genóm a aká je ich funkcia.

Cieľom práce je vytvoriť nástroj schopný detekovať génové oblasti sekvencií DNA húb. Snahou je dosiahnuť čo najvyššiu mieru presnosti detekcie intervalov pre intróny, tak aby následne bolo možné tieto intrónové časti odstrániť a vytvoriť tak čo najdlhšiu súvislú sekvenciu exónových častí bez chybného posunu čítacieho rámca. Vytvorením takejto sekvencie sa uľahčuje následný problém identifikácie samotného génu a identifikácie vlastností proteínu, ktorý tento gén kóduje. Predikcia pozícii génov a ich génových oblastí umožní celogenómové porovnanie húb, ktoré sa používa pre určenie ich vzájomnej príbuznosti.

Problém detekcie sa ešte komplikuje tým, že detekcia musí prebehnúť na metagenóme. Na vstupe je teda zmes sekvencií DNA viacerých organizmov. V tomto prípade ide o metagenóm z tlejúceho dreva zo Žofínskeho pralesa. Metagenóm môže teda obsahovať sekvencie DNA jak húb, tak aj baktérii, rastlín alebo iných živočíchov, ktoré sa do vzorku mohli primiešať.

Jedným z možných kandidátov na riešenie tohto problému je detekcia pomocou pravdepodobnostných modelov ako sú Markovské reťazce, skryté Markovské modely [5, 6] a generalizované skryté Markovské modely [7]. V bioinformatike je detekcia génových oblastí týmto prístupom dosť obľúbená a vykazuje pomerne dobré výsledky. Cieľom je zistiť či je tomu tak aj pri tak

špecifickom organizme ako sú huby a pri tak špecifickej úlohe ako je správne určenie pozícií intrónov v ich DNA. Okrem toho máme k dispozícii 949 rôznych druhov húb s oanotovanými DNA sekvenciami a ich taxonómiou. Cieľom práce je teda vytvoriť viacero modelov, ktoré budú schopné klasifikovať sekvencie DNA vrátane rozpoznávania kódujúcich a nekódujúcich úsekov DNA. Následne v týchto hubových sekvenciach budú modely schopné anotovať pozície intrónov a exónov, čím sa umožní vystrihnutie čo najdlhšej a najpravdepodobnejšej sekvencie exónov. Vďaka poskytnutej taxonómii je cieľom vytvoriť rôzne modely pre jednotlivé druhy, čeľade či rody a následne zistiť a porovnať výsledky detekcie modelov na poskytnutom metagenóme.

Kapitola 2

Východiská práce

Nasledujúca kapitola sa zameriava na východiská samotnej práce a to biologické a následne aj technické.

V prvej časti tejto kapitoly sa čitateľ zoznámi s biologickými predpokladmi potrebnými pre samotnú prácu. Oboznámi sa s pojmami ako DNA, exón, intrón, splice site a následným biologickým pozadím štruktúry. Druhá časť sa zameriava na biologicé východiská špecifické pre genómy húb. Nakoniec sa kapitola venuje technickým východiskám, popisuje a porovnáva prístupy a metódy anotácie genómov a to hlavne pomocou state of the art prístupov využívajúcich skryté Markovské modely.

2.1 Biologické východiská

Genetická informácia organizmov je uložená vo forme DNA, ktorá pozostáva zo 4 základných nukleových kyselín Adenin (A), Guanin (G), Tymin (T) a Cytozin (C). Tieto nukleové kyseliny vytvárajú dva dlhé reťazce, známe ako DNA reťaze, kde sú jednotlivé nukleové bázy spojené pomocou Hydrogénových väzieb [8]. Dvojice A-T a G-C sú komplementárne a tvoria takzvané Watson-Crick bázove párovanie [9]. Práve vďaka tomuto párovaniu sú dané vlákna komplementárne. Toto bázové párovanie je dôležité pri DNA replikácii, kedy príslušná DNA polymeráza vytvára replikované vlákno spájaním komplementárnych nukleotidov.

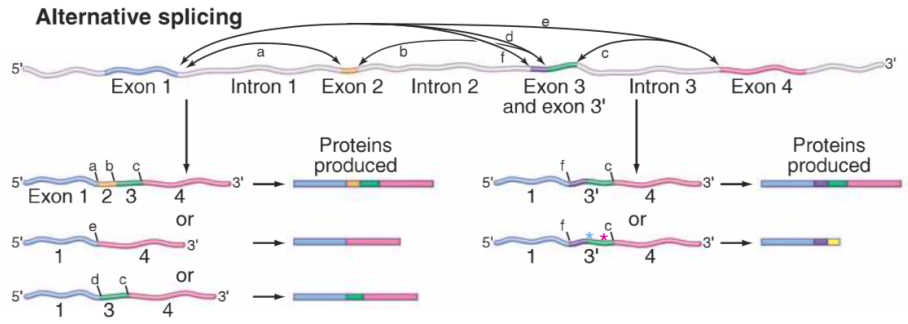
Oblasti DNA sekvencie môžeme rozdeliť na dve základné podkategórie, časti špecifického obsahu a signálne časti [10]. Do prvej kategórie sa dajú zaradiť celky ako exóny, intróny a medzigénové oblasti. Ide o dlhšie časti DNA, ktoré majú špecifický obsah. Do druhej kategórie signálov radíme kratšie úseky DNA. Za signály považujeme celky ako štart a stop kodón, promotér či 5' a 3' splice sites.

Sekvencia DNA pozostáva zo striedania génových a medzigénových oblastí. Génové oblasti sa začínajú takzvaným promotérom. Ide o časť DNA sekvencie, ktorá sa nachádza pred samotným štart kodónom. Tento úsek je rozpoznávaný pomocou RNA polymerázy pri začiatku transkripcie DNA. Najznámejším motívom promotéru je TATA box. Vzdialenosť promotéru a začiatku transkripcie od štart kodónu nie je presne definovaná. Ide väčšinou o niekoľko 100 bp v upstream smere od *core promotéru* [11]. Samotná génová oblasť začína štartovacím kodónom. V prípade eukaryot je touto sekvenciou trojica nukleotidov ATG [12]. Následne je génová oblasť definovaná ako úseky exónov a intrónov, ktorá vždy začína a končí exónom. Intróny je možné chápať ako tie úseky sekvencie, ktoré keď odstránime, tak dostaneme sekvenciu exónov tvoriacu mRNA. Táto výsledná sekvencia musí byť zložená z 3-nukleotidových kodónov [13]. Je preto veľmi dôležitá správna lokácia prechodov exon-intron a intron-exon oblastí. Posun týchto prechodov o 1 alebo 2 pozície môže viesť k fatálnej zmene kódovanej informácie. Došlo by k posunu celého čítacieho rámca, čo by malo za následok vznik úplne inej sekvencie mRNA a kódovanie odlišného proteínu. Lokácie 5' a 3' splice sites sú pre vznik správneho proteínu kľúčové. Väčšina ľudskej mRNA má splice sites definované dinukleotidmi GU na 5' pozícii a AG na 3' pozícii. Existuje ale aj malá skupina intrónov, ktorá má odlišnú dvojicu splice sites. AT dinukleotid na 5' pozícii a AC na 3' pozícii [1]. Celú génovú oblasť DNA sekvencie zakončuje exón, ktorý obsahuje takzvaný stop kodón. DNA eukaryot má väčšinou tieto tri stop kodóny: TAA, TAG, alebo TGA [14].

2.1.1 Alternatívny splicing

Pri sekvenovaní ľudského genómu sa zistilo že počet jeho génov je len 20000, čo je značne menej než sa predpokladalo [1]. Na základe tohto zistenia vznikla myšlienka o alternatívnom splicingu. Zo sekvencie DNA pre jeden gén môže vzniknúť viacej mRNA a tým pádom aj viacej proteínových produktov. Je to možné dosiahnuť buď vynechaním jedného alebo viacerých exónov vo finálnej transkripcii, prípadne zahrnutím jedného či viacerých intrónov [1]. Takéto alternatívne zostrihy môžu vytvoriť kratšie alebo dlhšie variácie mRNA. Alternatívny splicing dáva priestor k vzniku antagonistických proteínov. Alternatívny prepis môže tak zmenšiť alebo zvýšiť produkciu daného proteínu

[1]. Pre lepšiu predstavu sú jednotlivé príklady alternatívneho splicingu vyobrazené na obrázku 2.1.

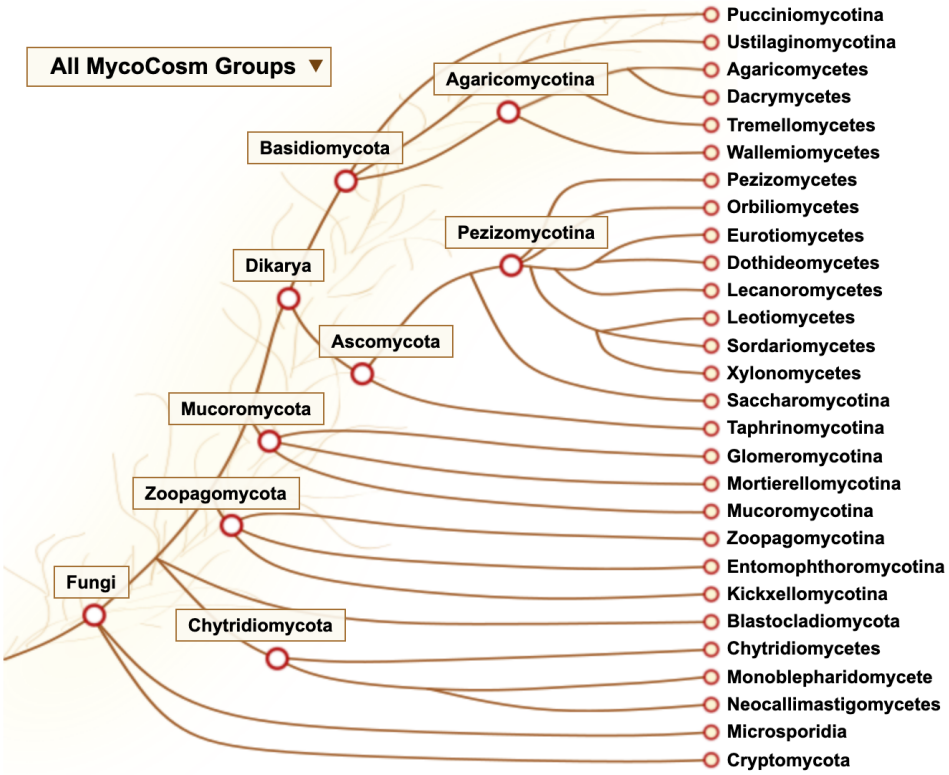


Obrázok 2.1: Alternatívny splicing a jeho rôzne možnosti [1]. Na ľavej strane je vidno ukážky, kedy sa preskočí jeden alebo viacej exónových oblastí, vznikne tak transkriptová oblasť s odlišnou sekvenciou nukleotidov. Na pravej strane môžeme vidieť vplyv alternatívnych splice sites. To môže viesť k zahrnutiu dodatočných aminokyselín do proteínového produktu, prípadne, že exón bude čítaný iným čítacím rámcom (zelená hviezdička), ktorý môže obsahovať stop kodón (červená hviezdička), čím sa vyprodukuje kratší proteínový produkt.

2.2 Biologické východiská genómu húb

Huby patria medzi eukaryota. Ide o veľmi rozmanitú skupinu organizmov a sú jedny z najstarších organizmov žijúcich na Zemi s takmer 1 miliardou evolučnej histórie. Nájdeme ich vo všetkých možných prostrediach, od polárnych oblastí až po trópy [15].

Huby, ako každý organizmus, majú svoju taxonómiu. Delíme ich na desať základných kmeňov (*phylum*): *Ascomycota*, *Basidiomycota*, *Chytridiomycota*, *Monoblepharidomycota*, *Neocallimastigomycota*, *Blastocladiomycota, Glomeromycota*, *Entomophthoromycota*, *Stramenopiles* a *Micorsporidia*. Tieto kmene sa ďalej delia na patričné podkmene, napr. *Kickxellomycotina*, *Kickxellomycotina* alebo *Saccharomycotina*, triedy, rady, čeľade, rody a druhy [16, 17]. Ukážku taxonomického prehľadu húb je možné vidieť na obrázku 2.2.



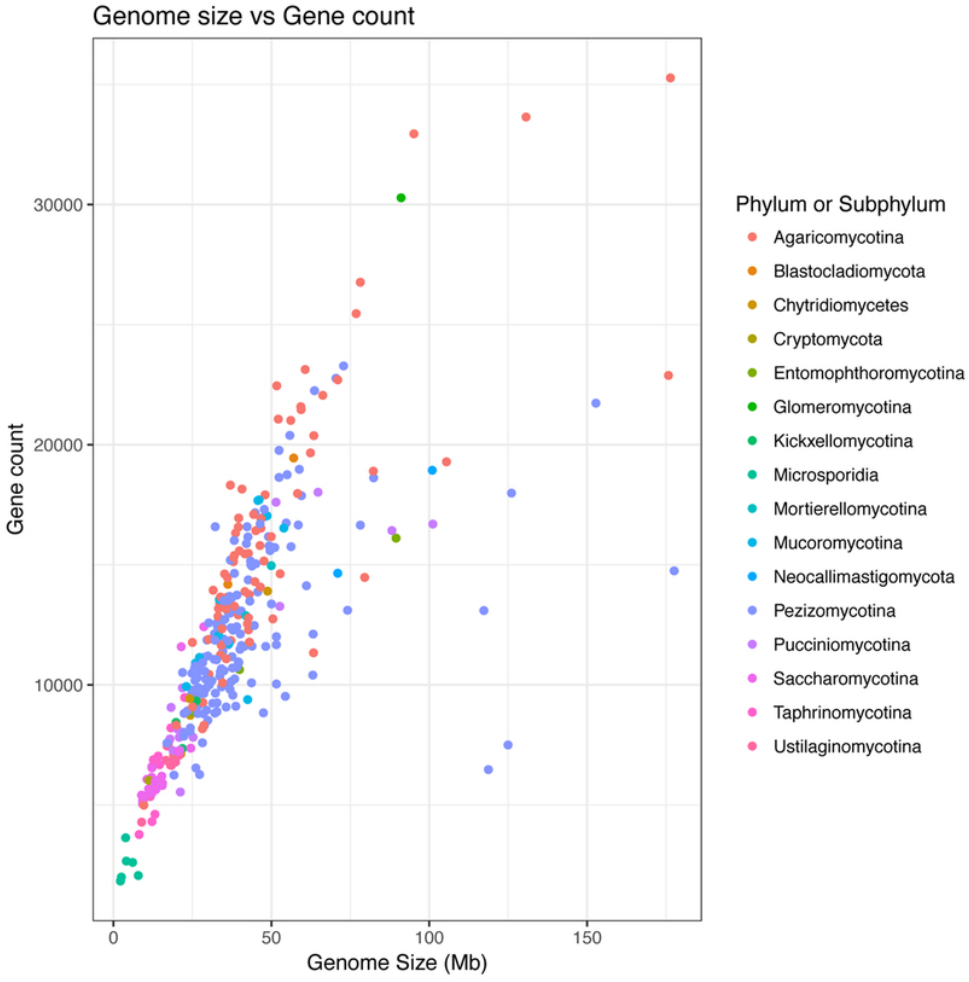
Obrázok 2.2: Ukážka taxonómie húb - kmene, podkmene a triedy [2]

Vďaka tomu, že huby sú tak rozšíreným organizmom naprieč rôznymi ekologickými prostrediami, z pohľadu evolúcie znamenajú veľmi zaujímavú oblasť [4]. Doposiaľ sekvenované a anotované gény húb môžeme nájsť v genómovej databáze *Mycocosm JGI* [2].

Pochopiť fungovanie húb a ich génové dráhy môže mať vplyv nie len v pochopení fungovania samotných húb, ale aj v pochopení istých spoločných fyziologických a genetických aspektov s rastlinnou a živočíšnou bunkou [4]. Ako príklad môžeme uviesť mnohobunečnosť, štruktúru cytoskeletu, bunečný cyklus, cirkadianný rytmus, medzibunkové signály či sexuálnu reprodukciu [18].

Huba *Saccharomyces cerevisae* sa stala modelovou hubou pre všetky huby a pôvodne sa myslelo, že genómy všetkých húb vychádzajú z tohto modelu. Neskôr sa zistilo, že tomu tak nie je a medzi genómami húb sa objavuje genómova a molekulárna diverzita [4]. Huby sa tiež značne líšia vo veľkosti genómu v porovnaní s genómom rastlín a živočíchov, genóm húb je znateľne kratší [19]. Modelová huba *Saccharomyces cerevisae* má dĺžku genómu len

okolo 12 Mb a vo všeobecnosti, až na pár výnimiek (*Cenococcum geophilum* (177.57 Mb)), dĺžka genómov húb nepresahuje 100 Mb [4]. Prehľad závislostí vo veľkostiach genómov húb a počtu génov je vidno na obrázku 2.3. Tabuľka 2.1 uvádza priemerné hodnoty pre veľkosť genómu, počet génov a počet exónov na gén vo vybraných kmeňoch štúdie [4].



Obrázok 2.3: Závislosť počtu génov a veľkosti genómu naprieč kmeňmi a podkmeňmi húb [3]

Kmeň	Priemerná veľkosť genómu (Mbp)	Priemerný počet génov	Priemerný počet exónov na gén
Ascomycota	36.91	11129.45	2.58
Basidiomycota	46.48	15431.51	5.28
Oomycota	74.85	24173.33	2.23
Mucoromycotina	38.777	13306.85	4.25

Tabuľka 2.1: Prehľad priemerných veľkostí, počtu génov a počtu exónov na gén v dostupných kmeňoch [4]

Výskum génomov húb prináša nové a nové sekvenované a anotované genómy. Vďaka tomu sa podarilo objaviť fakt, že huby sú veľmi divergentné. Aj na úrovni rodov (*genus*) dochádza k značným génovým rozdielom medzi jednotlivými zástupcami. Príkladom sú huby *Aspergillus nidulans*, *Aspergillus fumigatus* and *Aspergillus oryzae*, ktoré zdieľajú len 68 % genómu [18].

Vývoj a dĺžka genómu závisí v značnej miere na prostredí a potrebách daného organizmu [20]. Tvrdiť ale, že komplexnejšie organizmy (napr. tie ktoré vytvárajú zložitejšie štruktúry z viacerých buniek) obsahujú viacej génových oblastí v porovnaní s menej komplexnejšími je nepravdivé [4]. Väčšie genómy väčšinou obsahujú viac repetitívnych a nekódových oblastí. U húb môžeme ale vidieť istú zaujímavú koreláciu. Dlhšie genómy obsahujú aj väčší počet génov. Túto závislosť je vidno na obrázku 2.3. Zaujímavý je aj fakt, že huby majú v porovnaní s vyššími eukaryotmi kratšie intrónové úseky [21]. Vyššia hustota génových oblastí zároveň prináša istú mieru prekrytí *UTR* oblastí susedných génov. K týmto prekrytiam môže dochádzať na rovnakom reťazci medzi 3' a 5' koncami alebo na opačných reťazcoch medzi 5' a 5' alebo 3' a 3' koncami susedných génov [21].

Čo sa týka alternatívneho splicingu, v genómoch húb nájdeme omnoho menej alternatívneho splicingu ako v genómoch vyšších eukaryot. Napríklad len 5 % génov je alternatívne zostrihnutých v *Saccharomyces cerevisae* a menej ako 1 % v *Encephalitozoon cuniculi* [22]. Vďaka tomuto faktu vzniká menej priestoru na vznik chyby pri následnej anotácii genómu.

2.3 Technologické východiská

Oblasti génov sa v genóme dajú detekovať viacerými prístupmi. Existujú ab initio metódy, ktoré používajú len genóm ako vstup detekcie, napr. GENSCAN

[23], AUGUSTUS [24], HMMGene [25] alebo GENEID [26]. Ich rozšírením sú skupiny metód využívajúce doplňujúci zdroj informácie ako syntenické sekvencie (N-SCAN [27], SLAM [28], DOUBLESCAN [29], AGenDA [30]) či prehľadávaním databáz proteínových či EST sekvencií. GenomScan [31] rozširujúci GENSCAN pomocou BLAST zarovnaní, Gene Wise [32], ktorý zrovnáva proteínové sekvencie s genómovými. Augustus+ [33] prehľadáva v databázach homológnych zarovnaní (napr. EST) doplňujúce proteínové a genómové informácie, ktoré používa ako *nápovedy (z ang. hints)* redukujúce falošne pozitívne predikcie ab initio modelov a zvyšuje tak špecificitu predikcie. Nová verzia GeneMark-ES [34], ktorá sa volá GeneMark-ET [35] umožňuje tiež zakomponovať RNA sekvencie do trénovacej fázy trénovania modelu. Okrem týchto dvoch typov metód existuje ešte špeciálne programy ako Combiner [36], ktorý kombinuje predikcie niekoľkých jednoduchších génových prediktorov. Taktiež existujú prístupy, ktoré používajú Conditional Random fields [37] alebo Support Vector Machines [38].

Huby majú veľký ekologický význam. Predstavujú patogény rastlín, zvierat, ľudí a zdroj pri výrobe jedla, liečiv, enzýmov či priemyselných chemikálii. Detekcia génových oblastí v hubách má preto bohaté uplatnenie v oblastiach ako poľnohospodárstvo, medicína, spracovanie biomasy či potravinárstvo [21]. Ich význam v týchto oblastiach je podstatný, a preto je snaha lepšie pochopiť ich fungovanie. K tomu prispieva práve kvalitnejšia anotácia genómov húb. Stále objavujeme nové a nové druhy húb a databáza sekvenovaných genómov sa tak rozširuje o nové sekvencie. Vyššie spomenuté metódy sa špecializujú na eukaryota ako také. Huby sa svojím genómom od vyšších eukaryot odlišujú. Majú vyšší počet génov s krátkymi intrónmi [18, 39]. Gény zaberajú pomerne širokú časť genómu, pohybujeme sa zhruba od 27.8 % (Neurospora crassa) do 52.6 % (Botritus cineria) [34]. Oblasti medzi génmi zvyknú byť tak úzke, že neprekladané oblasti (UTR) sa väčšinou prekrývajú [40]. U húb môžeme taktiež pozorovať menej alternatívneho splicingu v porovnaní s inými eukaryotami [41]. Signály genómu využívané pri eukaryotách na započatie a ukončenie translácie a transkripcie sa u húb nevyskytujú a jak fungujú tieto procesy u húb nie je ešte úplne jasné [42, 40]. Pre mnoho húb ich príbuzné druhy ešte buď vôbec nemajú osekvenovaný genóm alebo ho nemajú oanotovaný. To znamená, že množina homológnych proteínov použiteľná pre anotáciu je buďto malá alebo nespoľahlivá. Je možné teda využiť len EST/RNA zarovnania a ab initio metódy. V súčasnej dobe ale prebiehajú veľké projekty ako 1000 Fungal Genomes [43] a Fungal Genome Initiative [44], ktoré usilujú o rozšírenie týchto sekvenovaných sekvencii nových druhov húb.

Na detekciu génových oblastí v hubách sa špecializuje metóda CodingQuarry [21], ktorá zohľadňuje vyššie spomenuté kritéria. Nedávna štúdia [45] ukazuje, že rekonštrukcia kratších intrónov z transkriptu je úspešnejšia ako rekonštrukcia dlhých intrónov. Túto výhodu využíva CodingQuarry, zostavovanie anotácie priamo z transkriptu nie je tak náchylné na chyby ako

pri vyšších eukaryotách.

V roku 2008 Ter-Hovhannisyan et al. vytvorili ab initio metódu založenú na učení bez učiteľa (z ang. unsupervised learning) pomocou Viterbiho a MCMC algoritmov [46]. Daná metóda zohľadňuje dôležitosť branch pointov pri alternatívnom splicingu húb. Podľa analýz u druhov ako D. melanogaster, A. thaliana, C. elegans, a Homo sapiens, tvorí branch point len 5% informácie potrebnej k presnému intrónovému splicingu. Pri kvasinkách (Saccharromyces cerevisiae), ktoré majú asi 200 spliced génov je tento podiel až 40 %. Tento alogritmus využíva fakt, že znalosť konzervovaných branch pointov intrónov je zásadná a využíva ju pri anotácii. K tomu využíva posilnený intron model. Daný algoritmus dosahuje senzitivitu 82.6–83.6% pre pôvodný algoritmus a 89.1-91.8% pre algoritmus s posilneným intrónovým submodulom [34].

Ďalším predstaviteľom detektoru génových oblastí genómov húb je GipsyGene [47]. Je založený na GenScan [23], takže pozostáva z modelov pre jednoexónový gén, počiatočný/prostredný/konečný exón, intrón či medzigénové oblasti. Nižšie huby majú dobre definovateľný branch point signál, vďaka čomu sa zvyšuje senzitivita exónov. Pri testovaní na genóme Neurospora crassa dosahuje senzitivitu intrónov 87% a senzitivitu exónov 74%. V porovnaní s vyššie zmieneným algoritmom od Ter-Hovhannisyana má ale slabšie výsledky, aj po zohľadnení branch pointov. Senzitivita pre model bez branch pointov je 69-75% a s branch pointmi je 75-80% [34].

SnowyOwl [40] je pipline, ktorá bola vytvorená za účelom rýchlejšieho a presnejšieho anotovania nových génov v genómoch húb za pomoci spresnenia predikcie pomocou RNA sekvencií. SnowyOwl využíva viaceré nástroje, ktoré dokážu mapovať RNA-seq ready na daný genóm a výstupy týchto mapovaní používa na učenie a predikciu najlepšej možnej sekvencie z pomedzi nich. SnowyOwl dosahuje 88% správnej predikcie pre gény Neurospora crassa, ktoré majú pre patričné RNA-Seq pokrytie 82% špecificitou [40]. Okrem SnowyOwl existujú aj rôzne ďalšie pipliny, ktoré sa ale nezameriavajú priamo na huby. Ide napríklad o MAKER2 [48], ktorý kombinuje rôzne zdroje vrátane ab initio prediktorov a RNA sekvencií. Pomerne novou piplinou je BRAKER1 [49], ktorý kombinuje nástroje AUGUSTUS [24] a GeneMark-ET [35].

Kapitola 3

Popis pravdepodobnostných modelov

Táto kapitola detailnejšie popisuje aktuálne prístupy a využitia pravdepodobnostných modelov v anotácii DNA sekvencií. Čitateľ by mal v tejto kapitole získať prehľad nad základným rozdelením daných modelov a popis hlavných state of the art nástrojov.

3.1 Základné rozdelenie

Podľa samotnej štruktúry DNA a jej signálov môžeme rozdeliť modely na dve hlavné kategórie [10]:

1. Modely obsahu - modely, ktoré sa zameriavajú na variabilné charakteristiky DNA ako exóny, intróny a medzi génové oblasti.
2. Modely signálov – modely, ktoré sa zameriavajú na detekciu krátkych nukleových signálov špecifických pre DNA ako štart/stop kodón, splice sites, poly A oblasti, promotéry, a pod.

Pre úspešné segmentovanie DNA a predikciu génov sa vo finálnych prístupoch používa práve kombinácia týchto dvoch hlavných kategórii. Zohľadňovanie jednotlivých signálov môže byť veľmi náročné. Genomické sekvencie obsahujú niekoľko tisícok podobných signálov, ktoré môžu byť vo výsledku len

šum imitujúci skutočné signály DNA. Pre príklad GC bohaté oblasti a TATA boxy sú považované za dôležité signály promotérov. Výskum z posledných rokov ukazuje ale, že TATA box sa nenachádza vo všetkých eukaryotách a len 45% promotérov obsahuje TATA Box [50].

Podľa výslednej použitej metódy pre segmentáciu DNA môžeme súčasné modely rozdeliť na 3 skupiny [10]:

1. Ab initio
2. Komparatívne metódy
3. Hybridné metódy

3.1.1 Ab initio

Tieto metódy sa snažia o systematickú analýzu a oddeľovanie jednotlivých signálov DNA a rozličných biologických vzorov DNA. Sú založené na vnútorných informáciach v DNA a vo väčšine prípadov využívajú pravdepodobnostné modely, ktoré popisujú chovanie modelov detekujúcich obsah a signály. Hlavným predstaviteľom tejto kategórie sú teda skryté Markovské modely. Môžeme sem ale zaradiť aj ďalšie prístupy, ktoré využívajú vnútorné informácie o daných sekvenciách DNA ako napríklad metódy založené na Furiorových transformáciách a spracovaní digitálnych procesov, umelé a hlboké neuronové siete či SVM s použitím vhodných kernel funkcií [10].

3.1.2 Komparatívne metódy

Tieto metódy využívajú skutočnosť, že v dnešnej dobe máme dostupnú veľmi veľkú databázu už osekvenovaných druhov a živočíchov. Vďaka tomuto faktu je v predikcii génových častí DNA možné využiť komparatívny prístup založený na homológii – kódujúce sekvencie sú v evolúcii DNA zachovávané skôr ako nekódujúce [10]. Je preto možné použiť metódy lokálneho zarovnania (*Smith-Waterman algoritmus*, *BLAST*, [51, 52]), globálneho zarovnania (*Needleman-Wunsch algoritmus* [53]), prehľadávanie databáz (*BLAST, BLASTX, BLAT* [54, 52]) a iné.

3.1.3 Hybridné metódy

Hybridné metódy využívajú kombináciu Ab initio metód a komparatívnych metód. Snažia sa prevziať výhody z oboch metód a maximalizovať tak úspešnú predikciu génových častí DNA. Dnešné state-of-the-art metódy využívajú práve kombináciu rôznych štatistických modelov s porovnávaním v databázach homologických sekvencií . Výsledná úspešnosť takýchto hybridných modelov je ale závislá na použitom ab initio modeli a aplikovaných obmedzujúcich pravidlách porovnávania[10]. Príkladom hybridnej metódy je napr. nástroj *Augustus* [24], ktorý si detailnejšie popíšeme v ďalšej časti.

3.2 State of the art metódy

Nasledujúce podkapitoly popisujú detailnejšie fungovanie dvoch vybraných state of the art metód. Augustus/Augustus+, ktorý sa nezameriava na žiaden konkrétny organizmus a CodingQuary, ktorý sa špecifikuje len na detekciu génov v genómoch húb.

3.2.1 Augustus/Augustus+

Využíva generalizovaný skrytý Markovský model [55] do ktorého sa snaží vložiť dodatočné informácie, ktoré nie sú zrejmé zo samotnej DNA sekvencie. Takéto jednotlivé kusy vonkajších informácií sú v modeli Augustus označené ako *hint* alebo *nápoveda*[24]. Napríklad, potencionálna pozícia splite site miesta, ktorá sa odvodí zarovnaním sekvencie s EST sekvenciou je hint.

Augustus pracuje so 6 typmi *nápoved*: start, stop, dss, ass, exonpart, exon, Ide o definované intervaly a orientáciu vlákna pre štart/stop kodóny, 5' a 3' splite sites, kódujúca oblasť a samostatný exón. Každá *nápoveda* má pridelený stupeň g, ktorý určuje mieru spoľahlivosti nápovedy vzhľadom na jeho zdroj [24].

Augustus+ následne rozširuje fungovanie GHMM verzie Augustus na pravdepodobnostnú distribúciu trojice $P(\phi, s, h)$, kde ϕ je štruktúra génu, s je vstupná sekvencia a h je kolekcia nápoved. Hľadá sa potom taká štruktúra, ktorá maximalizuje podmienenú pravdepodobnosť $P(\phi|s, h)$. Funguje to tak,

že abecedu modelu $\Sigma = \{a, c, g, t\}$ rozšírime o kolekciu nápoved H, $\Sigma' = \Sigma \bigcup H$ [33].

Nápovedy *start, stop, ass, dss* sú spojené vždy len s jednou konkrétnou pozíciou, keďže je to pre ne prirodzené. Nápovedy *exon* a *exonpart* sú určené intervalovými pozíciami [33]. Z toho dôvodu sa v nástroji Augustus takáto intervalová nápoveda spája len s konečnou pozíciou daného intervalu. na i-tej pozícii vstupného vlákna je teda určený daný nukleotid a dané nápovedy. Daný typ nápovedy môže byť na i-tej pozícii definovaný najviac raz. i-tý znak v sekvencii je teda n-tica $(b_i, h_{i,start}, h_{i,stop}, ..., h_{i,exon})$, kde b_i je nukleotid na i-tej pozícii a $h_{i,type}$ je nápoveda i-tej pozície. Nápoveda $h_{i,type} = (grade, strand)$ pre nápovedy *start, stop, ass, dss*. Nápovedy *exon a exonpart* sú určené $h_{i,exonpart} = (grade, strand, length, readingframe)$, kde *length* určuje dĺžku a *reading frame* určuje jeden z troch možných štartov čítacieho rámca. Ak na i-tej pozícii daná nápoveda nie je definovaná, tak $h_{i,type} = \Psi$, kde Ψ je označenie pre prázdnu nápovedu [33].

Augustus+ zavádza zjednodušujúci predpoklad, že nápovedy na rôznych pozíciach i v sekvencii s sú nezávislé, keď máme danú génovú štruktúru ψ. Taktiež nápovedy rôznych typov t emitovaných na rovnakej pozícii i sú považované za nezávislé. Vďaka tomu môžeme uvažovať:

$$P(\phi, s, h) = P(\phi, s)P(h \mid \phi, s) = P(\phi, s) \prod_{1 \leq i \leq |s \ |, t \in TYPE} P(h_{i,t}|\phi, s) \quad (3.1)$$

Augustus+ zvažuje len tie nápovedy, ktoré sú kompatibilné so vstupnou sekvenciou a sú správne. Napríklad $h_{i,start}$ musí končiť sekvenciou ATG na i-tej pozícii, ináč daná nápoveda nie je správna. Nesprávne nápovedy sa ignorujú. K vyhodnoteniu správnosti nápovedy slúžia tieto tri druhy kompatibility [33]:

1. Nápoveda je kompatibilná s génovou štruktúrou ϕ a tým pádom aj s so sekvenciou s
2. Nápoveda je kompatibilná so sekvenciou s, ale nie je kompatibilná s génovou štruktúrou ϕ
3. Nápoveda je nekompatibilná s génovou štruktúrou ϕ a so sekvenciou s

Výsledná podmienená pravdepodobnosť nápoved je teda závislá na type nápovedy a jeho stupni g [33]. Tieto pravdepodobnosti je možné odhadnúť z dát pomocou MLE[56].

V niektorých prípadoch nápoved je porušená úvaha o nezávislosti. Napríklad, ak vďaka rovnakému EST zarovnaniu získame nápovedy pre exón a aj splice site. V takom prípade sa ponechávajú len tie nápovedy, ktoré najlepšie sumarizujú dodatočné informácie potrebné pre predikciu. Nechá sa nápoveda *exon* a *ass a dss* sa ignorujú [33].

Augustus+ môže mať vo výsledku aj negatívny efekt na predikciu výslednej génovej štruktúry. Takýto negatívny efekt môže nastať, pokiaľ sa rozhodujeme či daná časť DNA je alebo nie je exón. Ab initio model by mohol nadobudnúť mieru istoty, že ide o potencionálny exón. Pri prehľadávaní databáz sa môže stať, že pre danú sekvenciu potencionálneho exónu sa ale žiadna nápoveda nenájde. Pre nepodporovaný falošne pozitívny exón je vo výsledku menšia pravdepodobnosť, že bude súčasťou výslednej génovej štruktúry, lebo jeho pravdepodobnosť je podmienená pravdepodobnosťou nálezu nápovedy [33].

3.2.2 Coding Quarry

Ide o nástroj využívajúci GHMM prediktor zameraný na predikciu génov hub, ktorý využíva zarovnané RNA sekvencie podobne ako systém Augustus. CodingQuery sa vyznačuje tým, že zohľadňuje len RNA sekvencie pri učení a predikcii a ukazuje, že dobrých výsledkov sa dá dosiahnuť aj napriek málo dostupným proteínových homológom z dostupných databáz či trénovacej množiny génov.

CodingQuarry využíva fakt, že huby sú zatiaľ veľmi málo preskúmaná oblasť, hlavne čo sa týka ich genómu. Pre mnoho húb genómy ich príbuzných druhov neboli buď ešte vôbec oanotované alebo ani osekvenované. To znamená, že množina použiteľných proteínových homológov použiteľných pre anotáciu nových homológov je buď veľmi malá alebo nespoľahlivá [21]. Preto predikcia musí spoliehať skôr na samotné ab initio metódy a EST/transkriptové zarovnania. Okrem toho CodingQuarry pracuje s pár ďalšími predpokladmi:

1. Huby majú v svojom genóme signifikantne menší výskyt alternatívneho splicingu ako vyššie eukaryota, ako sme si popísali v časti 2.2.
2. Huby majú kratšie intróny ako vyššie eukaryota [21].
    a. Podľa štúdie sú krátke intróny v transkripte rekonštruované úspešnejšie ako tie dlhé. Proces rekonštrukcie je teda menej náchylný k chybám ako u vyšších eukaryot [57].

3. Intróny húb majú najbohatší zdroj informácie v oblasti 5', 3‘ splice sites a v ich okolí [21].

Predikcia výslednej anotácie pozostáva v kombinácii predikcie dvoch oddelených GHMM modelov. V prvej fáze sa daný GHMM učí len z RNA transkriptov a na základe anotácii z *GFF* súborov sa vytvoria sekvencie osekaných transkriptov, v daných génových sekvenciach sa vynechávajú intrónové časti. Tento GHMM model pozostáva z tri-periodického nehomogénneho Markovského reťazca piateho stupňa pre kódové oblasti. 5' a 3' UTR, medzigénové UTR modely a nekódujúce časti transkriptu sú modelované ako homogénne Markovské reťazce so stupňom päť. Pre promotér a stop kodón sa používa takzvaný WAM model [58] stupňa dva a fixnou dĺžkou jedenásť nukleotidov, kde posledné tri nukleotidy tvoria štart kodón. Trénovanie v tejto fáze má isté obmedzenia, aby sa predišlo falošným pozitívnym nálezom. Pre génový stav sa uvažujú len tie oblasti, ktoré sú dlhšie ako 600 nukleotidov. UTR sekvencie dlhšie ako 300 nukleotidov sú tiež vynechávané z učenia. Pokiaľ existujú gény, ktoré sa prekrývajú, pri učení sa berie do úvahy vždy iba ten dlhší. Výhodou tohto prístupu je, že sa určia hranice intrónov z týchto transkriptov a zarovnajú sa na pôvodný genóm [21].

V druhej fáze predikcie sa použijú tieto predikované hranice intrónov a génov a spustí sa predikcia pomocou ďalšieho GHMM modelu, ktorý už obsahuje intrón model. Ten má 5' a 3' splice sites modelované ako WAM model stupňa jeden s fixnou dĺžkou a homogénny model Markovského reťazca stupňa päť pre intrónové oblasti medzi týmito fixnými dĺžkami. Model pre UTR oblasti nemá obmedzenú dĺžku ako pri stupni jeden a modeluje dlhšie medzigénové oblasti [21].

Kapitola 4

Dáta

Táto kapitola popisuje vstupné dáta, ktoré máme k dispozícii a uvádza určitú štatistickú analýzu vykonanú nad týmito dátami. Tieto informácie slúžia k lepšiemu pochopeniu rozhodnutí vykonaných v nasledujúcej kapitole 5, ktorá sa venuje návrhu pravdepodobnostného modelu.

Kapitola je rozdelená na dve časti. Prvá časť sa venuje popisu genómov húb a druhá kapitola sa zameriava na popis dát k metagenómu.

4.1 Genóm húb

Máme dostupné tri rôzne druhy súborov: *Assembly*, *CDS* a *Genes*. *Assembly* sú *fasta* [59] súbory obsahujúce jednotlivé scaffoldy [60]. Pod *CDS* súbormi označujeme tiež súbory formátu *fasta*, ktoré obsahujú len pre-mRNA daných *Assembly* scaffoldov. Posledný typ *Genes* sú súbory typu *gff* [61], ktoré obsahujú anotácie dostupných scaffoldov. Popisujú, na ktorých pozíciach sa nachádzajú štart a stop kodóny, kde sú exóny a kódové oblasti. Dohromady máme k dispozícii 949 rôznych genómov húb, ktoré pochádzajú z databázy JGI [2].

Nad dátami bola vykonaná určitá vstupná štatistická analýza, aby sme mali lepšiu predstavu aké výsledky zhruba očakávať pre jednotlivé *phylum* húb. Tabuľka 4.1 popisuje vybrané priemerné hodnoty pre jednotlivé *phylum*

húb. Ide o priemerné hodnoty dĺžok intrónu, počtu génov, dĺžok scaffold sekvencií.

Phylum	Dĺžka intrónu	Počet exónov na gén	Dĺžka scaffoldu	Počet génov
Ascomycota	96.68	2.67	666346.94	11009.53
Basidiomycota	76.69	5.44	216730.49	13489.76
Blastocladiomycota	88.42	3.56	183101.02	12296.25
Chytridiomycota	129.43	4.15	134184.63	10901.50
Cryptomycota	48.45	4.11	7199.50	5270.00
Microsporidia	24.19	1.01	106595.04	2197.62
Mucoromycota	94.52	4.35	216903.35	13309.92
Zoopagomycota	132.50	3.05	92791.61	7203.50
Všetky:	90.49	3.72	202981.57	9459.76

Tabuľka 4.1: Vybrané priemerné hodnoty jednotlivých *phylum* húb

Tabuľka 4.2 ukazuje priemerné dĺžky exónových sekvencií rôznych typov. *ExonStart* je prvý exón génovej sekvencie, ktorá obsahuje aspoň jeden intrón. *ExonEnd* je posledný exón takejto sekvencie. *ExonInter* je exón, ktorý sa nachádza medzi nimi, čiže sekvencia má aspoň dva intróny. *ExonSingle* je z génovej sekvencie, ktorá neobsahuje ani jeden intrón.

Phylum	ExonEnd	ExonInter	ExonSingle	ExonStart
Ascomycota	634.61	358.48	1068.90	334.21
Basidiomycota	336.12	195.57	740.41	239.11
Blastocladiomycota	528.93	292.05	936.81	343.00
Chytridiomycota	460.36	253.69	1000.44	309.61
Cryptomycota	292.17	235.37	925.52	197.35
Microsporidia	355.10	173.78	985.27	150.16
Mucoromycota	391.76	225.33	757.14	234.69
Zoopagomycota	596.17	267.93	1061.11	330.95
Všetky:	513.46	290.76	938.93	294.61

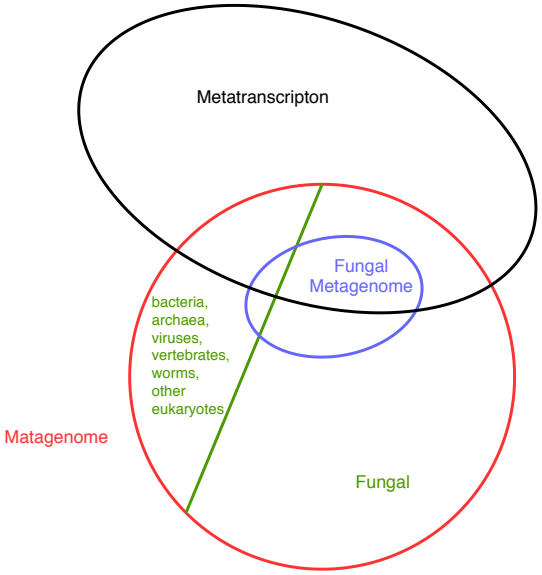
Tabuľka 4.2: Priemerné dĺžky exónových sekvencií rôznych druhov.

Detailnejšie štatistiky sú priložené v prílohe C.

4.2 Metagenóm

K dispozícii máme dáta metagenómu od Mikrobiologického ústavu AV ČR. K dispozícii máme hneď 3 rôzne *fasta* súbory. Všeobecný metagenóm zo vzorku

tlejúceho dreva Žofínskeho pralesa. Tento súbor obsahuje veľké množstvo scaffoldov, kde každý scaffold podľa MBÚ patrí vždy len jednému organizmu, ako napríklad baktérie, archaea, vírusy, stavovce, červy či huby. Práve hubám by malo patriť okolo 60 % všetkých týchto scaffoldov. Ďalší súbor je podmnožinou tohto všeobecného metagenómu. S vysokou pravdepodobnosťou obsahuje hlavne scaffoldy patriace hubám. Je tu ale istá veľmi malá pravdepodobnosť, že daná množina scaffoldov obsahuje malé množstvo iných organizmov. Nakoniec máme k dispozícii metatranskriptom čo je nezávislá množina transkriptov predchádzajúcich súborov. Medzi kontigmi metagenómu a metatranskriptomu nie je udané žiadne priame mapovanie. Jedine čo vieme, že metatranskriptom a metagenóm by sa mal určite prekrývať, no v metagenóme môžu existovať sekvencie, ktoré sa vôbec nevyskytujú v metatranskriptome a naopak. Tento jav je zapríčinený degradáciou buniek, DNA zotrváva v prostredí omnoho dlhšie (až roky) v porovnaní s RNA (rádovo minúty) [62] a preto môže DNA obsahovať informácie pre už neaktívne (spóry a pod.) alebo mŕtve organizmy, pre ktoré RNA už neexistuje. Väčšina RNA sa taktiež zničí počas procesu izolácie. V poslednom rade je dôvodom aj nízka hĺbka sekvenovania vzhľadom k obrovskej diverzite organizmov, to znamená že u prítomných aktívnych organizmov bola prečítaná len informácia z DNA alebo jej RNA, aj keď v danom vzorku boli prítomné obe. Rozdelenie dát metagenómu je vidno na obrázku 4.1.



Obrázok 4.1: Diagram zobrazuje názornú štruktúru a prekrytie metagenómu s metatranskriptom

Kapitola 5

Návrh pravdepodobnostného modelu

Táto kapitola popisuje návrh metódy pre anotáciu génových oblastí a detekcie samotných intrónových úsekov v daných DNA sekvenciach. Pre popis samotného modelu pre huby je potrebné predstaviť najskôr isté základy GHMM modelov, ako fungujú a na čom sú postavené. Následne kapitola popisuje štruktúru a princíp chovania navrhnutého generálneho skrytého Markovského modelu pre genóm húb.

5.1 GHMM model

Navrhovaný model pre detekciu intrónových sekvencií vychádza z technologických a biologických východísk v popisovaných v úvodných kapitolách 2.1, 2.2 a 2.3. Návrh je teda postavený na samotnom základe *ab initio* pravdepodobnostných modelov a to na *Generálnych skrytých Markovských modeloch* [55].

Generálny skrytý Markovský model, jak názov už napovedá, zovšeobecňuje použitie *Skrytých Markovských modelov*. Tento model vytvára istú vyššiu úroveň abstraktnosti, pretože pozostáva z jednotlivých pravdepodobnostných modelov tvoriace stavy. Ide tak o model, ktorý musí pracovať s:

1. Konečnou množinou Γ ,ktorá obsahuje skryté stavy

2. Konečnou množinou Σ reprezentujúcou vstupnú abecedu
3. Iniciálnymi pravdepodobnosti pre štart
4. Pravdepodobnosťami prechodov medzi jednotlivými stavmi
5. Pravdepodobnosťami zotrvania v danom stave

Pre bod 5 musí platiť, že ak máme stav $s, s \in \Gamma$, tak pre danú podsekvenciu $x_{q+1}x_{q+2}...x_{q+d_i}$ vieme spočítať pravdepodobnosť $P(x_{q+1}x_{q+2}...x_{q+d_i}|s_i)$, pričom q je dĺžka sekvencie, ktorá predchádza danú podsekvenciu: $q = \sum_{j=1}^{i-1} d_j$ [55]. Spočítaním takýchto pravdepodobností podsekvencií pre každý model a každú podsekvecniu dokážeme nájsť takú kombináciu, ktorá je najpravdepodobnejšia. Tu sa ponúka možnosť využiť *Viterbiho algoritmus* v mierne upravenej podobe. Tento algoritmus bude detailne popísaný v nasledujúcej kapitole 5.1.4, ktorá vysvetľuje fungovanie Viterbiho algoritmu.

Kľúčový predpoklad k použitiu GHMM je teda mať stavy reprezentované pravdepodobnostnými modelmi, ktoré dokážu vyhodnotiť pravdepodobnosť vstupnej podsekvencie $P(x_{q+1}x_{q+2}...x_{q+d_i}|s_i)$. Pre tieto účely môžeme zvažovať modely ako napríklad *Markovské reťazce*, *WAM*, *HMM* či *periodické Markovské reťazce*.

5.1.1 Markovské reťazce

Ide o stochastický proces $\{X_n\}$, ak pre každé $n \geqslant 0$ a pre všetky stavy $i_0, ..., i, j \in S$ platí [63]

$$P(X_{n+1} = j|X_n = i, X_{n-1} = i_{n-1}, ..., X_0 = i_0) = P(X_{n+1} = j|X_n = i) = P_{ij} \tag{5.1}$$

P_{ij} značí teda pravdepodobnosť prechodu pokiaľ sme v reťazci v stave i a prechádzame do stavu j s jedným krokom. Takýto model sa dá následne zapísať do dvojrozmernej matice, kde riadky označujú počiatočný stav i a stĺpec cieľový stav prechodu j. Pre každý riadok v matici, teda pre každý stav musí platiť podmienka, že suma prechodových pravdepodobností je rovná jednej:

$$\sum_{j \in S} P_{ij} = 1, \forall i \in S \tag{5.2}$$

Pravdepodobnosť vstupnej sekvencie môžeme vyhodnotiť ako produkt pravdepodobností jednotlivých prechodov:

$$P(i_0, i_1, ...i_n) = P(i_0) \prod_{s=1}^{n} P(i_s|i_{s-1}) \tag{5.3}$$

Markovské reťazce sa vyznačujú vlastnosťou, kde všetky nasledujúce stavy aktuálneho stavu sú nezávislé na predchádzajúcich stavoch aktuálneho stavu. Závislosť sa viaže iba na susedné stavy. Pokiaľ túto závislosť predĺžime vytvoríme Markovský reťazec vyššieho stupňa. Hovoríme, že daný model má určitú mieru histórie, s ktorou dokáže pracovať. Takýto Markovský reťazec označujeme ako Markovský reťazec k-tého stupňa, kde k udáva veľkosť histórie. Výpočetná sila takéhoto modelu sa môže zvýšiť pri zvýšení stupňa no treba si uvedomiť, že výskyt daných k-mérov sa exponenciálne zmenšuje pri zväčšovaní histórie modelu [64].

Nehomogénny Markovský model

Pri klasickom homogénnom Markovskom modeli popísanom v predchádzajúcej kapitole 5.1.1 neuvažujeme prechodové pravdepodobnosti vzhľadom na dané pozície. Pri Markovských modeloch popisujúcich fungovanie exónu by sme si priali zaviesť túto vlastnosť nehomogenity, keďže posun čítacieho rámca o jednu alebo dve pozície by mohlo mať fatálne následky na výslednu $mRNA$. Pri nehomogénnom Markovskom modeli máme rôzne prechodové pravdepodobnosti (2D matice) pre rôzne pozície [64]. Výsledná pravdepodobnosť sekvencie je potom daná predpisom:

$$P(X) = p^{(0)}_{x_1} \prod_{i=1}^{L} p^{[(i-1) \mod c],(i \mod c)}_{x_{i-1},x_i} \tag{5.4}$$

kde c je počet pozícii, pre ktoré si pamätáme dané prechodové pravdepodobnosti.

5.1.2 WAM

Weight array model[65] uvažuje závislosti len medzi susednými pozíciami v danej sekvencií. Tento model môžeme chápať ako nehomogénny Markovský reťazec s fixnou dĺžkou. Výsledná pravdepodobnosť sekvencie sa spočíta ako

$$P(X) = p_{x_1}^{(1)} \prod_{i=2}^{n} p_{x_{i-1},x_i}^{i-1,i} \tag{5.5}$$

kde člen $p_{x_{i-1},x_i}^{i-1,i}$ udáva pravdepodobnosť že znak x_i na pozícii i je predchádzaný znakom x_{i-1} na pozícii *i-1*. Daný model pracuje s n pozíciami [10].

5.1.3 Učenie GHMM modelu

Keďže celý model ja postavený na jednotlivých celkoch popísaných pomocou Markovských reťazcov, tak učenie takéhoto modelu bude prebiehať pomerne triviálne. Ako popisuje kapitola 4, okrem vstupných sekvencií máme k dispozícii aj jednotlivé anotácie. Tento fakt nám učenie modelov zjednodušuje. Je teda možné využiť priamočiary prístup metódy *MLE* [56], kde spočítame nálezy daných pozorovaní:

$$p_{st} = \frac{c_{st}}{\sum_{t'} c_{st'}} \tag{5.6}$$

Kde c_{st} označuje koľkokrát znak t nasledoval po znaku s, kde oba znaky patria do abecedy modelu $s, t \in \Sigma$ definovanej v 5.10.

Problém výpočtu 5.6 spočíva práve v počte vstupných pozorovaní. Ak sa stane, že niektorý z prechodov nebudeme pozorovať ani raz, tak výsledná pravdepodobnosť takéhoto prechodu bude rovná nule. Pri evaluovaní pravdepodobnosti vstupnej podsekvencie ϕ z predpisu 5.17 by nám vyšla výsledná pravdepodobnosť daného reťazca na základe predpisu 5.3 rovná nule, lebo niekde cestou narazíme na danú nulovú pravdepodobnosť. Tento jav nie je úplne žiaduci lebo sa tak pripravíme o istú možnosť predikcie. Problém vyriešime pridaním takzvaných pseudopočetností [56] do predpisu 5.6:

$$p_{st} = \frac{c_{st} + 1}{\sum_{t'} c_{st'} + |\Sigma|} \tag{5.7}$$

5.1.4 Najpravdepodobnejšia sekvencia stavov

Pomocou GHMM modelu môžeme spočítať najpravdepodobnejšiu sekvenciu skrytých stavov danej vstupnej sekvencie. Týmto spôsobom dosiahneme výslednú segmentáciu genómu na jednotlivé skryté stavy (exón, intrón, promoter, stop kodón...). GHMM model pri tomto výpočte berie do úvahy pravdepodobnosť dĺžky zotrvania v danom stave. Vďaka tomu vieme nie len najpravdepodobnejšiu postupnosť daných skrytých stavov, ale aj ich dĺžku a teda pozície začiatku a konca vzhľadom na vstupnú sekvenciu.

Podobne ako pri klasickom HMM modeli, v GHMM modeli spočítame najpravdepodobnejšiu sekvenciu skrytých stavov pomocou *Viterbiho algoritmu* [66].

Tento klasický *Viterbiho algoritmus* upravíme podľa nasledujúcej idey od Yuzhe Ye z Indiana University [55]. Cieľom je spočítať objektívnu funkciu

$$argmax_{(s_i,...,s_L)} p(s_1, ...s_L; x_1, ..., x_L) \tag{5.8}$$

Kde $s \in \Gamma$ je skrytý stav GHMM modelu na pozícii i sekvencie dĺžky L a $x_1, ...x_L$ sú dané znaky sekvencie. Pre každú pozíciu $i = 1, ..., L$ a pre každý stav $l, l \in \Gamma$ spočítame hodnotu $v_l(i)$, ktorá bude predstavovať pravdepodobnosť $p(s_1, ..., s_i; x_1, ..., x_i | s_i = l)$, čiže najpravdepodobnejšiu cestu k pozícii i, ktorá končí v stave l.

Následne môžeme využiť túto hodnotu $v_l(i)$ a pre každú pozíciu $i = 1, ..., L$ a pre každý stav $l, l \in \Gamma$ môžeme spočítať:

$$v_l(i) = \max \begin{cases} \max\limits_{\substack{1 \le q < i \\ 1 \le k \le m, k \ne l}} p(x_{q+1}x_{q+2}...x_i|s_l)v_k(q)a_{kl} \\ p(x_1x_2...x_i|s_l)a_{0l} \end{cases} \tag{5.9}$$

kde a_{kl} je pravdepodobnosť prechodu zo stavu k do stavu l. Vrchnú časť maxima funkcie je možné chápať ako hľadanie najpravdepodobnejšieho miesta prechodu zo stavu k do stavu l, ktorý sa nachádza na pozícii q vzhľadom

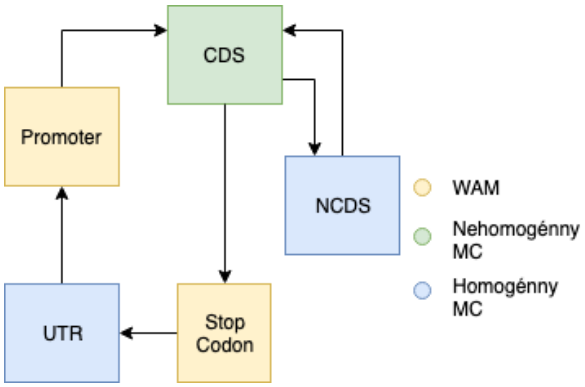
na pravdepodobnosť dĺžky zotrvania $i - q$ v stave l. Hľadáme najpravdepodobnejší bod prechodu, preto maximalizujeme skrz všetky stavy $k, k \in \Gamma$ a skrz všetky pozície $q, 1 \leq q < i$. Spodnú vetvu maxima chápeme ako hľadanie najpravdepodobnejšieho počiatočného stavu, kde a_{0l} je iniciálna pravdepodobnosť, že začíname v stave l. Zložitosť takéhoto výpočtu je preto $O(m^2L^2)$, kde $m \in \Gamma$ a L je dĺžka vstupnej sekvencie [55].

5.2 GHMM model húb

Pre detekciu jednotlivých oblastí v DNA sekvenciach húb boli vytvorené dva GHMM modely, *transkript* a *genóm* model, kde *transkript* model slúži ako podporný model pre *genóm* model. Dané modely si popíšeme v nasledujúcich kapitolách.

5.2.1 Transkript model

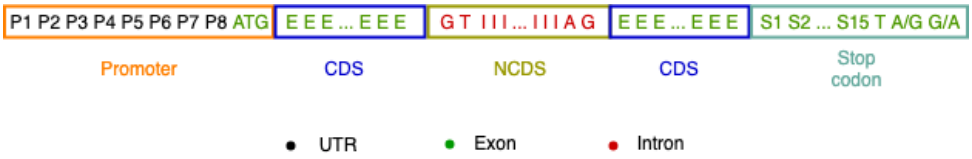
Transkript model pozostáva z piatich stavov: *UTR*, *CDS*, *NCDS*, *StopCodon* a *Promoter*. *CDS* stav modeluje exónové úseky v genóme, čiže kódujúce úseky a komplementárne stav *NCDS* modeluje nekódové úseky génových oblastí, čiže intróny. Zmyslom tohto modelu je hlavne orientačne určiť oblasti, ktoré sú potenciálne kódujúce a nekódujúce úseky v danej DNA sekvencii nukleotidov. Jeho učenie je teda založené hlavne na exónových a intrónových oblastiach bez signálnych informácii. Daný model preto nepočíta so splice site úsekmi, snaží sa byť čo najjednoduchší. Ako bolo vysvetlené v kapitole 3.2.2, state-of-the-art nástroj CodingQuarry využíva obdobný princíp pre detekciu génových oblastí.



Obrázok 5.1: Graf prechodov GHMM modelu pre transkript huby

Stav *CDS* je modelovaný ako nehomogénny Markovský reťazec s periódou tri. *NCDS* a *UTR* stav modelujeme ako homogénny Markovský reťazec a stavy *Promoter* a *StopCodon* ako WAM modely. Pre *Promoter* stav je nastavená fixná dĺžka jedenásť nukleotidov, kde prvých osem tvorí *Kozaková sekvencia* [67] a zvyšné tri sú rezervované pre štart kodón. Stav *StopCodon* má fixnú dĺžku osemnásť nukleotidov, kde posledné tri sú rezervované pre samotný stop kodón.

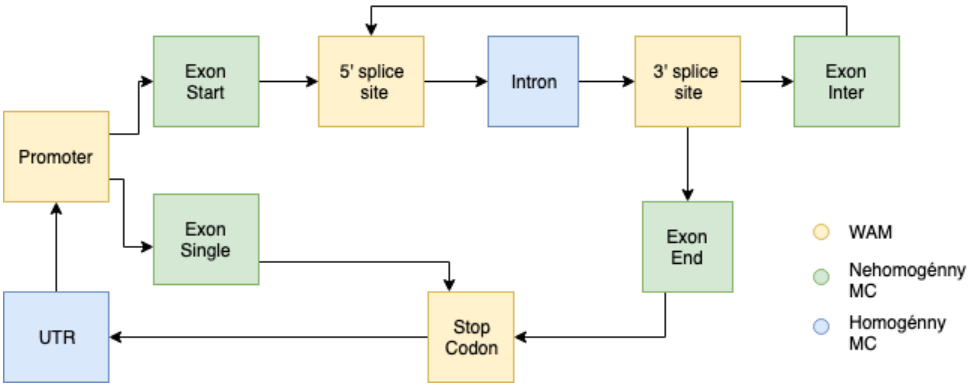
Na obrázku 5.2 je možné vidieť ukážku mapovania jednotlivých úsekov DNA sekvencie na stavy daného GHMM transkript modelu.



Obrázok 5.2: Ukážka mapovania vstupnej sekvencie na jednotlivé vstupy daných modelov pre transkript model

5.2.2 Genóm model

Druhým GHMM modelom je *genóm* model, ktorý obsahuje desať pravdepodobnostných modelov, ktoré tvoria jednotlivé stavy GHMM modelu. Ide o nasledujúce skryté stavy: *Promoter*, *ExonSingle*, *ExonStart*, *5' splice site*, *Intron*, *3' splice site*, *Exon Inter*, *Exon End*, *Stop Codon* a *UTR*. Prechodový graf medzi jednotlivými stavmi môžeme vidieť na obrázku 5.3. Daný model už uvažuje so signálmi v oblastiach splice sites.



Obrázok 5.3: Graf prechodov GHMM modelu pre genóm huby

Modely *Promoter*, *5' splice site*, *3' splice site* a *Stop Codon* sú modelované ako WAM modely s fixnou dĺžkou. WAM modely sa používajú vo veľkej miere hlavne pre detekciu hraničných oblastí medzi exónmi a intrónmi [58, 68, 69]. Nátroj *Coding Quarry* okrem toho využíva WAM model aj pre oblasti *promotérov* a *stop kodónov* [21]. Promotér je modelovaný s dĺžkou jedenásť nukleotidov vrátane štart kodónu na posledných troch miestach. Prvých osem nukleotidov modeluje *Kozakovú sekvenciu*, ktorá je dôležitým iniciálnym signálom promoteru v hubách[67]. 5' a 3' splice sites sú fixované na dĺžku šesť nukleotidov. Pri 5' splice site prvé tri nukleotidy predstavujú posledné tri nuklotidy exónu a zvyšné tri nukleotidy predstavujú zas prvé tri nukleotidy nasledujúceho intrónu. Miesto rozdelenia sa tak nachádza uprostred. Obdobne to funguje pri 3' splice site. Stop kodón model má dĺžku osemnásť nukleotidov, kde posledné tri predstavujú samotný stop kodón. Dĺžka modelu bola natiahnutá, aby mal model šancu sa vysporiadať s takzvaným javom zvaným *Stop codon readthrough*[70], kedy translácia proteinu môže pokračovať aj za stop kodónom až do nálezu nasledujúceho stop kodónu. Vizualizáciu vstupov jednotlivých modelov voči pôvodnej genómovej sekvencii zobrazuje obrázok 5.4.

Obrázok 5.4: Ukážka mapovania vstupnej sekvencie na jednotlivé vstupy daných modelov pre genóm model

Pri modeloch *UTR* a *Intron* je použitý homogénny Markovský reťazec ako pravdepodobnostný model. Modelovať tieto úseky nehomogénne, teda s určitou periodicitou nie je potrebné. Pod *UTR* sekvenciami sa rozumejú oblasť genómu, ktoré sa nachádzajú medzi stop kodónom a promotérom susedných génov.

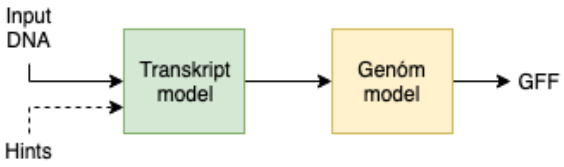
Zvyšné štyri exón modely sú reprezentované nehomogénnym Markovským reťazcom s periodicitou tri. To znamená že model si pamätá pravdepodobnosti prechodov vzhľadom na relatívnu pozíciu v reťazci, viď 5.1.1.

Vyššie uvedené modely sú všetky modelované ako modely vyšších stupňov. Stupne jednotlivých modelov boli vybrané na základe poznatkov fungovania nástroja *Coding Quarry* a všeobecne používaných a osvedčených stupňov v praxi [56]. *Promoter*, *3', 5' splice site* a *Stop codon* sú modelované ako modely stupňa dva. Všetky ostatné modely sú modelované so stupňom päť.

Model je rozdelený na dve cesty, po ktorých sa detekcia môže uberať. Spoločný počiatočný stav týchto ciest je model *Promoteru*. Po jeho nájdení sa môže model vybrať dvomi spomínanými cestami. Buď sa rozhodne, že pôjde do modelu *Exon Single*, ktorý názvom napovedá, že výsledná génová oblasť bude obsahovať len jeden exón bez intrónov. Druhou alternatívou je prechod do *Exon Start*, kedy sa model rozhodne nasledujúcu skevenciu anotovať ako oblasť s výskytom intrónov a viacerých exónov. V tejto vetve môže buď cykliť s využitím modelu *Exon Inter*, čo je v podstate exón, ktorý sa vždy nachádza medzi vnútri génovej oblasti a musí mu vždy predchádzať a za ním nasledovať intrón. Návšteva tohto stavu je ale veľmi málo pravdepodobná, kvôli priemernému počtu exónov na gén húb, viď tabuľka 2.1 alebo časť o štatistických vlastnostiach vstupných dát v 4. Pravdepodobnejšie prechody budú preto do modelu *Exon End*, čo je posledný exón v danej génovej oblasti genómu. Obidve cesty sú následne zakončené spoločným modelom *Stop Codon* a cez model *UTR* sa celý cyklus môže znovu zopakovať.

5.2.3 Fungi model

Transkript a *Genóm* GHMM modely sú spojené do samostatného modelu, ktorý je vyobrazený na obrázku 5.5. Do *transkript* modelu vstupuje vstupná DNA sekvencia, prípadne na vstupe definované voliteľné nápovedy v podobe GFF anotácii. Výstupom *transkript* modelu sú detekované intervaly pre kódujúce a nekódujúce úseky, promoter a stop kodón úseky a úseky UTR. Tieto intervaly sa posielajú ďalej do *genóm* modelu, ktorý ich využíva ako nápovedy. Hľadanie výsledných intervalov je tak spresnené na intervaly z predchádzajúcej predikcie a keďže sa využívajú aj modely pre signály splice sites, malo by dochádzať k spresneniu predošlej predikcie.



Obrázok 5.5: Graf prechodov hlavného GHMM modelu pre genóm huby

5.2.4 Formálna definícia GHMM modelu pre huby

Pre náš GHMM model neformálne definovaný v predchádzajúcej časti 5.2.3 máme teda definovanú vstupnú abecedu

$$\Sigma = \{a, c, g, t, n\} \tag{5.10}$$

a, c, g, t reprezentujú jednotlivé nukleové kyseliny a znak n označuje neznámu nukleovú kyselinu. Ďalej majme definované stavy Γ_t pre transkript model:

$$\Gamma_t = \{promoter, cds, ncds, stopCodon, utr\} \tag{5.11}$$

a pre genóm model majme Γ_g:

$$\begin{aligned}\Gamma_g = \{&promoter, exonSingle, 5SpliteSite, 3SpliteSite, exonStart,\\ &intron, exonEnd, exonInter, stopCodon, utr\}\end{aligned} \tag{5.12}$$

Všetky ďalšie označenia stavovej množiny sa týkajú jak Γ_t tak Γ_g, preto tieto dve množiny budeme reprezentovať spoločným označením Γ_n, kde $n \in \{g, t\}$.

Následne definujme prechodové pravdepodobnosti pre každý stav:

$$P(i|j); i, j \in \Gamma_n \tag{5.13}$$

$$\sum_{i \in \Gamma_n} P(i|j) = 1, \forall j \in \Gamma_n \tag{5.14}$$

čo nám hovorí aká je pravdepodobnosť prechodu zo stavu j do stavu i.

Okrem toho potrebujeme poznať počiatočné pravdepodobnosti daných stavov. K tomu si môžeme zaviesť akýsi tichý bezvýznamný stav *Begin*, ktorý pridáme do množiny stavov Γ_n

$$\Gamma'_n = \Gamma_n \cup \{Begin\} \tag{5.15}$$

$$\sum_{i \in \Gamma'_n} P(i|j) = 1, \forall j \in \Gamma'_n \tag{5.16}$$

Nakoniec potrebujeme mať spôsob jak spočítať pravdepodobnosti jednotlivých podsekvencií ϕ:

$$P(\phi|i), \forall i \in \Gamma'_n \tag{5.17}$$

Jednotlivé modely $i, i \in \Gamma'_n$ sú založené na Markovských reťazcoch, takže učenie týchto pravdepodobností a evaluácia je pomerne jednoduchá záležitosť, ktorú sme si popísali v kapitole 5.1.3.

5.3 Taxonomické zovšeobecňovanie modelu

V kapitole o biologických východiskách húb 2.2 sme si popísali, že huby majú isté taxonomické rozdelenie na kmene, podkmene, trieda, rad, čeľaď, rod a druh. Toto taxonomické rozdelenie môžeme využiť pri učení a vytvoriť tak špecifické modely, ktoré pokrývajú rôzne taxonomické úrovne. Výsledkom bude napríklad desať rôznych modelov reprezentujúcich daný kmeň húb. Daný GHMM model zostáva v nezmenenej podobe, no okrem jedného takého modelu budeme disponovať viacerými modelmi reprezentujúcimi daný taxón. Tieto modely sa teda budú učiť z genómov viacerých húb súčasne na základe spoločného taxónu.

Takto naučené modely na príslušných taxonomických úrovniach je možné znovu použiť. Vďaka tomu, že naučené GHMM modely sú zas len pravdepodobnostné modely, ktoré dokážu evaluovať pravdepodobnosť vstupnej sekvencie, môžeme tieto modely použiť ako stavy nového, abstraktnejšieho GHMM modelu. Ten by mal teoreticky vedieť segmentovať príslušný metagenóm na jednotlivé celky, kde skryté stavy sú jednotlivý zástupcovia daného taxónu, pokiaľ by sme v daný vstupný scaffold metagenómu patril viacerým organizmom/hubám súčasne.

Zložitosť evaluácie rastie kvadraticky s počtom stavov a kvadraticky s dĺžkou vstupnej sekvencie. Prečo je tomu tak sme si vysvetlili v kapitole 5.1.4. Ak teda budeme uvažovať, že každý GHMM model má počet stavov $m_g = |\Gamma_g| = 10$ a $m_t = |\Gamma_t| = 5$, a pre daný taxón, napríklad kmeň, budeme mať $T = 10$ takýchto GHMM modelov, výsledná zložitosť bude $O(T(m_t^2 l^2 + m_g^2 l^2))$, kde l je dĺžka vstupnej sekvencie, keďže ako popisuje kapitola 4.2, metagenóm obsahuje scaffoldy, kde jeden scaffold patrí vždy len jednému organizmu. Môžeme zvoliť prístup postupnej evaluácia všetkými naučenými GHMM modelmi na danej taxonomickej úrovni a následne vybrať tú segmentáciu, ktorá je najpravdepodobnejšia.

5.4 GHMM model pre metagenóm

Model detekujúci intrónove oblasti v metagenóme musí zohľadňovať fakt, že daný vstupný metagenóm neobsahuje iba jeden organizmus. Preto sa naskytuje možnosť hľadania týchto intrónových častí pomocou všeobecnejších GHMM modelov založených na taxonomickom učení, viď 5.3.

Taktiež je potrebné brať do úvahy dĺžky sekvencií vstupného metagenómu, ktoré boli popísané v 4. Ide o krátke úseky DNA, takže so sekvenciami nemôžeme pracovať rovnako ako pri segmentácii dlhého DNA reťazca. Pri učení GHMM modelu popísaného v kapitole 5.2.4 sa učíme počiatočné pravdepodobnosti prechodov z genómov húb. To znamená, že pravdepodobnosti $p(UTR|Begin)$ a prípadne $p(Promoter|Begin)$ budú veľmi vysoké a iniciálne pravdepodobnosti zvyšných stavov budú blízke nule. Tento fakt by nám uprednostňoval štart segmentácie hlavne v UTR modeli. Z toho dôvodu je vhodnejší prístup nastaviť všetky iniciálne pravdepodobnosti na rovnocennú hodnotu $\frac{1}{|\Gamma_n|}$, kde Γ_n je množina stavov GHMM modelu definovaná v 5.12 a 5.11.

Kapitola 6

Implementácia pravdepodobnostného modelu

Kapitola popisuje samotnú implementáciu GHMM modelu popísaného v predchádzajúcej kapitole 5, kde sme sa venovali jeho návrhu. Rozoberá štruktúru a organizáciu kódu a popisuje spôsob implementácie istých vybraných častí ako napríklad *Viterbiho algoritmus* pre GHMM model z podkapitoly 5.1.4. Čitateľovi by nasledujúca kapitola mala objasniť fungovanie kódu, umožniť mu jednoduchšie sa v ňom orientovať a vyznať.

6.1 Základný popis

Program je implementovaný v jazyku *Python 3.6*, vďaka čomu je interpretovaľný a jednoducho prenositeľný na rôzne systémy. Kód je objektovo orientovaný, aby bolo možné čo najlepšie reprezentovať finálny GHMM model za pomoci dedičnosti tried, čím sa redukuje množstvo nadbytočného kódu a prispieva k robustnosti celej aplikácie. Vďaka tomu, že kód je písaný v jazyku Python, je kód rozdelený do jednotlivých modulov, ktoré sú tak jednoducho použiteľné aj v iných aplikáciach či v budúcich nadstavbách.

Štruktúru kódu môžeme rozdeliť na tieto základné moduly:

- logger - modul umožňujúci logovanie behu aplikácie

- configuration - modul s konfiguračnými súbormi pre GHMM modely
- inout - modul na čítanie vstupu a zápis výstupu
- bio - modul umožňujúci prácu s génovými anotáciami
- probModels - modul obsahujúci jadro celého GHMM modelu pre huby
- tests - modul obsahujúci evaluačné testy GHMM modelu
- app - obsahuje hlavný program, natrénované modely a konfigurácie

Modul *logger* nie je z pohľadu behu aplikácie zas až tak dôležitý. Umožňuje logovanie správ rôznych úrovní: chyba, upozornenie či verbose. Okrem toho zabezpečuje percentuálny výpis aktuálneho stavu segmentácie. Vďaka tomu, že je to samostatný modul, je možné tieto rôzne úrovne výpisov kedykoľvek zakázať alebo povoliť.

Inout modul obsahuje dva základné submoduly, *dataParser* a *genomeOutput*. Submodul *dataParser* slúži na čítanie vstupných scaffold súborov a na čítanie anotácii v *GFF* súboroch. Tento submodul dokáže čítať dáta zo zdroja definovaného priamo pri spúšťaní programu. Tento proces si detailnejšie popíšeme v nasledujúcej sekcii 6.2. Čítanie dát prebieha s využitím iterátora, pri definovaní viacerých vstupných scaffoldov sa čítajú jednotlivé scaffoldy postupne a ďalší scaffold sa začne čítať až keď je to potrebné. Vďaka tomu nie je potrebné držať v pamäti veľké množstvo dát a medzi jednotlivými čítaniami čiastočne trénovať model. Pri vyžiadaní scaffoldu sa vrátia dve štruktúry typu slovník, kde je kľúčom vždy názov scaffoldu a hodnotou je buď daný scaffold reťazec alebo pole anotácii typu *Gene* z modulu *bio*. Pri čítaní scaffoldov a ich anotácii sa pozerá aj na orientáciu daných reťazcov. Vstupný scaffold nám predstavuje len jeden DNA reťazec. Niektoré gény sú ale anotované aj na opačne orientovanom reťazci, preto sa daný vstupný scaffold pri tomto čítaní prevedie na komplementárny a opačne orientovaný reťazec. K názvu scaffoldu sa tak pridá na koniec ešte znamienko + pre pôvodnú orientáciu a znamienko - pre komplementárny reťazec. Parametrom *strandType* sa dá zvoliť akej orientácie sa vrátia reťazce, +, - alebo oboje.

Submodul *genomeOutput* pracuje tiež s anotáciami typu *Gene* z modulu *bio* a slúži na zápis dát segmentácie. Jeho cieľom je zápis týchto anotácii do GFF výstupu. Okrem toho daný submodul zvláda zápis anotácii do HTML výstupu, čím je umožnená vizualizácia daných anotácii v danej vstupnej sekvencii. Anotácie sú mapované na farbu, čím sa v danej sekvencii jasne oddelia exónové, intrónové a UTR úseky.

Ako už bolo vyššie spomenuté, modul *bio* definuje triedu *Gene* pre anotácie scaffoldov. Objekty tejto triedy si pamätajú intervaly štart a stop kodónov,

exónov, intrónov, splice sites. Okrem toho je k dispozícii aj názov génu a *ID* proteinu, pokiaľ je daný objekt inicializovaný dátami z *GFF* súboru. Objekt si nedrží daný scaffold reťazec, iba niektoré jeho metódy ho očakávajú ako vstupný argument, napríklad metóda, ktorá vráti reťazec transkriptu.

Modulu *tests* sa budeme venovať v samostatnej kapitole 7, ktorá sa zameriava na vyhodnotenie kvality segmentácie.

Poslednými modulmi aplikácie sú modul *probModels*, ktorý sa skladá z viacerých submodulov a tvorí jadro samotnej aplikácie, modul *configuration* definujúci konfigurácie GHMM modelov a modul *app*. *ProbModels* si detailnejšie popíšeme v nasledujúcej kapitole 6.1.1, modul *configuration* zas v kapitole 6.1.2. Posledný spomenutý modul *app* tvorí aplikačné rozhranie, ktorý je možné spustiť trénovanie či predikciu s požadovanými parametrami. Predstavíme si ho v kapitole 6.2.

6.1.1 Modul probModels

Ide o komplexnejší modul aplikácie, ktorý pozostáva z týchto troch submodulov, ktoré sú radené v poradí, akom sú na sebe závislé: *generic*, *bioModels* a *fungiModels*.

Submodul generic

generic modul implementuje abstraktnú triedu *Model*, ktorá obsahuje štyri základné metódy:

- `prepareProbabilities`
  - Metóda slúži k príprave počítadiel frekvencií potrebných pre spočítanie *MLE* rovnice 5.6.
- `train`
  - Metóda volá predchádzajúcu metódu v prípade, že dané počítadlá nie sú ešte pripravené. Následne spočíta pravdepodobnosti stavov a prechodov pomocou *MLE* rovnice 5.6.
- `evaluateSequence -> float`

  - Metóda spočíta pravdepodobnosť vstupnej sekvencie na základe naučených pravdepodobností predchádzajúcej metódy `train`.

- `lazyEvaluateSequence -> (float, float)`
  - Metóda spočíta pravdepodobnosť vstupnej sekvencie no namiesto počítania všetkých prechodov reťazca sa počíta len pravdepodobnosť posledného prechodu.

Ide o abstraktnú triedu a abstraktné metódy, ktoré nadobúdajú konkrétnu implementáciu až pri ďalších triedach, ktoré dedia od tejto triedy *Model*.

Trieda *periodicMC* pre nehomogénny Markovský reťazec definovaný v 5.1.1 dedí práve od tejto abstraktnej triedy *Model*. Metóda `evaluateSequence -> float` tak implementuje chovanie popísané rovnicou 5.4. Trieda *WAM* dedí od triedy *periodicMC*, keďže chovanie v evaluačnej metóde môže byť simulované práve nehomogénnym modelom Markovského reťazca. Pred evaluáciou sa ale kontroluje, že vstupná sekvencia má požadovanú fixnú dĺžku. Ak tomu tak nie je, pravdepodobnosť takejto sekvencie je rovná nule. Implementácia pre nehomogénny Markovský reťazec obsahuje parameter `period`, ktorým je možné nastaviť počet pozícii, pre ktoré si model bude pamätať prechodové pravdepodobnosti. Nastavením tohto parametra na hodnotu jedna môžeme zas docieliť simuláciu chovania homogénneho Markovského reťazca popísaného v 5.1.1.

Trieda *GHMM* implementuje všeobecné chovanie generálneho skrytého Markovského modelu. Rozširuje pôvodnú abstraktnú triedu o ďalšiu dôležitú metódu `segmentSequence`, ktorá je zodpovedná za výslednú segmentáciu vstupnej sekvencie. Táto metóda využíva implementáciu *Viterbiho algoritmu*, ktorý popisovala kapitola 5.1.4. Implementácia tohto algoritmu pracuje so špeciálnym objektom `ViterbiItem`, ktorý reprezentuje premennú $v_l(i)$ definovanú v časti 5.1.4. Vďaka tomuto objektu si nepamätáme len najlepšiu pravdepodobnosť sekvencie do indexu i vzhľadom na aktuálny stav l, ale aj predchádzajúci stav, z ktorého sme prišli, pozíciu v danej segmentovanej sekvencii, kde aktuálny stav l začína, dĺžku aktuálneho transkriptu, a medzivýpočty pravdepodobností $p(x_{q+1}x_{q+2}...x_i|s_l)$ a $p(x_1x_2...x_i|s_l)$ z predpisu 5.9. Treba si uvedomiť, že pri počítaní novej hodnoty $v_l(i+1)$ sa v danej sekvencii posunieme len o jednu pozíciu doprava a znovu hľadáme nový bod zlomu. V každej novej iterácii potrebujeme len evaluovať prechod $p(x_ix_{i+1}|s_l)$ a pravdepodobnosť danej podsekvencie $p(x_{q+1}x_{q+2}...x_ix_{i+1}|s_l)$ tak spočítame ako $p(x_{q+1}x_{q+2}...x_i|s_l) * p(x_ix_{i+1}|s_l)$. Práve tu sa dostáva do role doposiaľ moc nespomínaná metóda `lazyEvaluateSequence`. Tá na vstupe očakáva predpočítanú hodnotu pravdepodobnosti vstupnej sekvencie do pozície i a následne dopočíta pravdepodobnosť $p(x_ix_{i+1}|s_l)$. Keďže dané pravdepodobnostné modely vedia počítať ešte pravdepodobnosti pre tichý konečný

stav, táto metóda vracia dve hodnoty. Novú hodnotu $p(x_{q+1}x_{q+2}...x_i|s_l) * p(x_ix_{i+1}|s_l)$, ktorú si budeme pamätať pre nasledujúcu iteráciu. Druhú hodnotu $p(x_{q+1}x_{q+2}...x_i|s_l) * p(x_ix_{i+1}|s_l) * p(end|x_{i+1})$ si uložíme ako skutočnú hodnotu a následne z nej hľadáme dané maximum. Veľmi obdobný prístup aplikujeme v druhej vetve maxima *Viterbiho algoritmu* pre počítanie počiatočného stavu $p(x_1x_2...x_i|s_l)$. Pri počiatočných indexoch, keď je sekvencia ešte dostatočne krátka, väčšinou kým sa dĺžka sekvencie nerovná stupňu modelu, tak sa iniciálne medzivýpočty pre metódu `lazyEvaluateSequence` spočítajú metódou `evaluateSequence`, ktorá vyhodnotí pravdepodobnosť celej sekvencie.

Algoritmus pracuje ešte s ďalším typom optimalizácie. Pri modeloch WAM, ktoré majú fixnú dĺžku nie je potrebné testovať všetky subsekvencie $p(x_{q+1}x_{q+2}...x_i|s_l)$, kde $1 \leq q < i$. Namiesto testovania celého rozsahu hodnôt **q** nám postačí otestovať poslednú subsekvenciu kde $q = i - c_{s_l}$ a c_{s_l} je fixná dĺžka WAM modelu s_l a zároveň nech platí $i > c_{s_l}$, aby sa nám nestalo, že použijeme záporný index **q**. Túto optimalizáciu oceníme hlavne pri sekvenciach s väčšou dĺžkou.

Nakoniec bol algoritmus ešte optimalizovaný za použitia paralelizácie výpočtu pomocou vlákien. Hodnota $v_l(i)$ sa musí spočítať pre každý stav l na danej pozícii i. Tento krok je teda možné paralelizovať a hodnotu $v_l(i)$ počítať pre každý stav v samostatných vláknach. K veľkému zlepšeniu ale nedošlo, porovnanie prístupov je popísané v ďalšej kapitole 7 zameranej na testovanie.

Okrem toho sa v objekte `ViterbiItem` uchováva dĺžka aktuálne objaveného transkriptu. Idea je, že pri počítaní novej položky *ViterbiItem* sa volá funkcia *getmRNALen*. Tá na základe aktuálneho objektu *ViterbiItem* zistí z akého stavu sa prišlo a aká v ňom bola doposiaľ dĺžka nájdeného transkriptu. Táto hodnota sa ďalej posiela do evaluačnej metódy každého modelu a pokiaľ ju obdrží niektorý z exon modelov, tak ju dokáže využiť pri evaluácii. *ghmm* trieda ale nevie nič o stavoch, v akom sú biologickom pomere a nepozná ani jednotlivé inštancie stavov. Ide o triedu, ktorá musí byť ďalej dedená a až v tomto potomkovi sa vytvára konkrétny popis GHMM modelu pre huby z časti 5.2.3. Preto je daná metóda len abstraktnou metódou a konkrétna implementácia je ponechaná na potomkovi triedy.

Submodul bioModels

Tento modul obsahuje triedy reprezentujúce stavy GHMM modelu. Ide o triedy, ktoré dedia od tried z modulu *generic*. Nájdeme tu šesť základných tried, ktoré reprezentujú konkrétne genómové sekvencie, čiže *exonModel*, *intronModel*, *utrModel*, ktoré dedia od nehomogénneho modelu Markovských reťazcov a *promoter*, *spliceModel*, *stopCodonModel*, ktoré dedia od WAM modelu. Pri inicializáciach sa nastavujú hyperparametre modelov, ako perióda pre nehomogénny Markovský reťazec a fixná dĺžka pre WAM model. Okrem toho dané modely majú nastavený svoj stupeň histórie, ktoré boli popísané v častiach 5.2.1 a 5.2.2.

Okrem toho má *exonModel* ešte špeciálny atribut `isTerminal`, ktorý sa nastavuje pri inicializácii. Tento atribút zapne špeciálnu kontrolu dĺžky doposiaľ získaného transkriptu spolu s dĺžkou vstupnej sekvencie počas evaluácie. Ak suma týchto dĺžok nie je deliteľná tromi, je jasné, že daný transkript nemôže existovať, keďže finálny transkript je tvorený kodónmi dĺžky tri [13]. Tento atribút na kontrolu dĺžky transkriptu je teda zapnutý pri exonových modeloch, ktoré sú na konci a génových oblastí a obsahujú stop kodón. Zapína sa teda len pri modeloch *exonEnd* a *exonSingle*.

Okrem týchto šiestich genómových modelov obsahuje modul ešte jeden špeciálnu triedu *dummyModel*. Trieda slúži pre reprezentáciu tichých stavov ako je napríklad stav *Begin* popísaný výrazom 5.15.

Submodul fungiModels

Modul má za cieľ zhromažďovať pohromade prípadné viaceré variácie potomkov GHMM modelu z modulu *generic*. Obsahuje akúsi abstraktnú triedu *FungiGenericModel*, ktorá dedí od triedy *GHMM*. Ide o štandardizáciu typu, aby sa pri evaluácii mohlo dynamicky voliť hociktorý model, ktorý dedí od tejto triedy. Momentálne obsahuje teda finálne implementácie GHMM modelov a to triedu **TranscriptModel** z časti 5.2.1, **GenomeModel** z časti 5.2.2 a spojený model v triede **FungiMainModel** popísaný v časti 5.2.3. Dané triedy sú potomkami práve spomínanej triedy *FungiGenericModel*. Tieto triedy využívajú triedy z modulu *bioModels* a definuje svoje stavy 5.11 a 5.12 a k tomu vždy ešte tichý stav *Begin*, ktorý bol popísaný v návrhu 5.2.4.

Dané tri triedy majú vždy vlastnú implementáciu metódy `prepareProbabilities`, kde si pripravia jednotlivé subsekvencie potrebné pre natrénovanie svojich

stavov, rovnako ako ukazuje obrázok 5.4 pre genóm model a obrázok 5.2 pre transkript model. Tieto sekvencie sú ohraničené dĺžkami špecifikovanými v konfiguračných súboroch. Po príprave dát pre jednotlivé stavy metóda volá jednotlivé metódy `prepareProbabilities` daných stavov. Aby sa celé učenie zrýchlilo, volanie týchto metód prebieha paralelne vo vláknach, kde každý stav má svoje vlákno. Pre spojený model **FungiMainModel** sa najskôr volá metóda `prepareProbabilities` pre transkript model a potom pre genóm model.

Okrem toho má trieda ešte vlastnú implementáciu zdedenej metódy `getmRNALen`. Pri počítaní najpravdepodobnejšej cesty cez skryté stavy chceme zaručiť, že nájdené transkripty budú mať dĺžku s periódou tri. Hodnota dĺžky aktuálneho transkriptu sa teda musí v promoteri rovnať nule a pokiaľ je aktuálnym stavom exón, tak sa táto hodnota zvýši o jeho dĺžku. Pri ostatných stavoch sa hodnota musí posunúť z aktuálneho stavu so nasledujúceho, aby sme mali vždy danú hodnotu k dispozícii. Všeobecnejšia trieda *GHMM*, od ktorej sa dedí, netuší o rozdelení stavov a biologickými závislosťami medzi nimi. Túto logiku musí preto implementovať až potomok triedy `FungiGenericModel`.

Každý z potomkov triedy `FungiGenericModel` musí implementovať aj metódu `getHintsFor`. Ako argument tejto funkcie je vždy sekvencia, ktorá sa má evaluovať. Táto metóda mapuje nápovedy pre predikciu na pozície vstupnej sekvencie. Princíp tohto mapovania je popísaný v časti 6.1.3. Každý GHMM model má ale svoje stavy, s ktorými pracuje, preto je potrebné mapovať tieto nápovedy vzhľadom na tieto jeho stavy. Práve vďaka tejto metóde sa mapuje výstup predikcie transkript modelu na vstup genóm modelu v spojenom modeli *FungiMainModel*.

6.1.2 Modul configuration

Tento modul definuje triedy pre konfigurácie GHMM modelov vďaka ktorým je možné nastavovať isté hyper parametre. Ide o minimálne dĺžky a maximálne dĺžky pre všetky modely s variabilnou dĺžkou a o fixné dĺžky pre WAM modely. Tieto hodnoty sa používajú na odfiltrovanie nevyhovujúcich sekvencií pred trénovaním samotného modelu a taktiež pri predikcii. Pri WAM modeloch sa pozerá aj na presah do susedných oblastí. Napríklad ak máme 5' splice site sekvenciu a dĺžka tejto sekvencie by presahovala dĺžku intrónu alebo exónu okolo daného splice site, tak sa takáto sekvencia zahodí. Sekvencie pre WAM modely môžu byť na rozmedzí maximálne dvoch okolitých oblastí. Pokiaľ má predikovaná sekvencia inú dĺžku ako určený interval v tejto konfigurácii, bude pravdepodobnosť takejto sekvencie rovná nule. Okrem hraníc dĺžok sekvencií jednotlivých stavov modelu, je možné špecifikovať aj stupeň histórie daného

modelu.

Pre WAM modely *3' a 5' spliceSite, promoter a stopCodon* je možné nastaviť vždy dve dĺžky a to dĺžku pre ľavú a pre pravú časť sekvencie. Suma dĺžok pre tieto časti dáva celkový počet stavov a teda fixnú dĺžku WAM modelu. Pre lepšiu predstavu si to popíšeme za pomoci obrázku 5.2 pre transkript model. Dĺžka ľavej časti promotéru je počet nukleotidov na ľavo od posledného nukleotidu štart kodónu vrátane, dĺžka pravej časti je zase presah do nasledujúceho exónu od posledného nukleotidu štart kodónu. Pri stop kodóne je podobná situácia, dĺžka ľavej časti je počet nukleotidov od posledného nukleotidu stop kodónu vrátane a dĺžka pravej časti je zas presah do nasledujúcej nekódovej oblasti UTR. Pri genóm modely je chovanie dĺžok ľavej a pravej časti pre promotér a stop kodón rovnaké. Čo sa týka splice site mapovania z obrázku 5.4, pre 5' splice site je ľavá časť určená počtom nukleotidov od posledného exónového nukleotidu vrátane a pravá časť je počet intrónových nukleotidov od prvého nukleotidu daného intrónu. Komplementárne je tomu u 3' splice site. Presah do exónovej časti musí byť vždy deliteľný tromi, aby sa redukovala potencionálna chyba rozbehnutia čítacieho rámca.

Okrem definícii minimálnych a maximálnych dĺžok konfigurácia špecifikuje aj vstupnú abecedu znakov pre GHMM model, ktorá bola definovaná výrazom 5.10. V prípade potreby rozšírenia abecedy o nové znaky je tak len potrebné zmeniť túto definíciu na tomto mieste.

Vďaka štatistikám pre jednotlivé taxóny húb (viď príloha C), je napríklad možné vytvoriť špecifickú konfiguráciu pre dané phylum. Môže sa stať, že ak nastavíme intervaly moc striktne, v danej sekvencii neodhalíme požadovaný relevantný sled stavov, prípadne sa začne daný génový úsek rozpadať na viaceré kratšie gény. Taktiež si treba uvedomiť, že dané rozsahy sa používajú pre učenie modelu. Moc striktné obmedzenia môžu vyradiť nevyhovujúce úseky sekvencií pre trénovanie, čím sa redukuje počet vstupných dát a aj robustnosť celého modelu.

Konfiguráciu pre daný model je možné prepísať vlastným konfiguračným súborom v JSON formáte [71]. Ukážka formátu takéhoto súboru je v prílohe D.1. Následne sa priloží cesta k súboru ako parameter pri spúšťaní trénovania či predikcie, viď nasledujúca kapitola 6.2. Vďaka tomu je možné napríklad upraviť rozsahy dĺžok už pre natrénované modely, ktoré sú dostatočne robustné, lebo počítajú s menej striktnými úsekmi sekvencií a boli potencionálne natrénované na viacerých dátach, no vďaka tejto rekonfigurácií môžeme docieliť väčšiu rýchlosť prípadne inú presnosť predikcie. Je možné rekonfigurovať iba stavy s variabilnou dĺžkou, WAM modely nie je možné rekonfigurovať, lebo majú fixnú dĺžku a bolo by nutné tak znovu natrénovať daný stav.

6.1.3 Nápovedy pre predikciu

Samotný objekt triedy *GHMM* a teda aj potomci triedy *FungiGenericModel* majú špeciálny atribút *hints*, teda nápovedy. Pri spúšťaní predikcie je možné pridať vlastné anotácie pre vstupné sekvencie, napríklad pokiaľ sú známe pozície štart či stop kodónov. Ide o *GFF* súbor rovnakého formátu ako pri učení modelu. Daný model si tak vytvorí na základe nápovedy pravdepodobnosť daného stavu pre danú pozíciu v vstupnej sekvencii. Pre každú pozíciu sekvencie sa vytvorí slovník, kde pre dané stavy modelu je definovaná pravdepodobnosť výskytu daného stavu a aj prípadný rozsah, kde daná pozícia musí byť vždy súčasťou tohto rozsahu, ak je definovaný. Najskôr sa pre každú pozíciu sekvencie nastaví, že každý stav je pravdepodobný s hodnotou 100 %. Následne sa prechádzajú priložené anotácie v podobe nápoved a pre každú pozíciu z daného intervalu sa nastaví ktoré stavy sú ako moc pravdepodobné a v akých rozsahoch.

V genóm modeli sa pre nápovedy intrónu nastavia stavy *intron, 5' a 3' spliceSite, a stopCodon* na hodnotu 100 % a zvyšné stavy sa úplne nevypnú, ale zníži sa ich pravdepodobnosť výskytu na 90 %. Hodnota je zvolená tak, aby nápovedy na vstupe stále plnili účel, teda uprednostňovali definované stavy na daných pozíciach. Na druhej strane, nastavením pravdepodobnosti pre nežiaduce stavy na nulu by bola predikcia až moc prísna a pri nesprávnej nápovede z transkript modelu by sme tak už nemali možnosť nájdenia správneho stavu. Pri niektorých nápovedách sa nastavia viaceré stavy na 100 %, napríklad pri intróne sa zapína tiež *stopCodon*, ak by sa napríklad pri použití výsledku detekcie transkript modelu označil ako intrón aj celý ďalší exón. Vypnutím stop kodónu by sme stratili možnosť nájsť a opraviť takýto nález. Na druhej strane, ak je daná sekvencia naozaj intrónom, tak by mala mať väčšiu výstupnú pravdepodobnosť pre stav *intrón* ako pre stav *stopCodon*, takže k falošnej zámene by nemalo dochádzať. Taktiež je v intrónových oblastiach možnosť nálezu splice site oblastí. To isté platí aj pre exónove oblasti. Pre daný stav sa tak zapínajú vždy aj všetky stavy, ktoré do daného stavu smerujú prípadne tie, kam z daného stavu môžeme vyraziť. Cieľom nápovedy v genómovom modely je spresniť predikciu a to hlavne za pomoci predikcie transkript modelu, pričom nechceme byť moc prísni v tomto spresňovaní. Na druhej strane, transkript model je prísnejší k nápovedám. Nápovedy sú definované na vstupe užívateľom a predpokladá sa tak ich väčšia miera správnosti. Nápovedy sa používajú pre všetky stavy, okrem UTR oblastí.

Vo Viterbiho algoritme sa následne skontroluje či pre danú pozíciu existuje nejaká nápoveda a ak áno, tak sa použije a pripočíta sa k daným pravdepodobnostiam prechodov. Dá sa tak penalizovať daný nežiadúci prechod a uprednostniť prechod do stavu, ktorý bol definovaný v *GFF* súbore nápoved. V prípade budúcej potreby je možné tieto nápovedy rozšíriť o iné druhy, ako

je tomu v nástroji Augustus.

6.2 Spustenie programu

Program je spustiteľný z príkazovej riadky. Je možné spustiť trénovaciu verziu, a to nasledovne:

```
python3 fungiGHMM -t [--fasta folder] [--gff folder] [--model name]
```

kde *-t* hovorí, že ide o trénovací mód, argument *–fasta* udáva cestu k priečinku so scaffold súbormi, z ktorých sa má učiť. Argument *–gff* udáva cestu k priečinku s anotáciami daných scaffoldov. Je dôležité, aby počet súborov v týchto priečinkoch bol zhodný a začiatky názvov súborov mali *fasta* a *gff* súborov mali zhodný prefix. Posledný argument *–name* udáva meno výsledného modelu.

Druhou možnosťou je použiť už natrénovaný model a spustiť:

```
python3 fungiGHMM -p [--fasta folder] [--model name]
[[--strand +/-/0]] [[--hints folder]] [[--html int]] [[--conf file]]
```

kde *-p* hovorí, že pôjde o predikciu. *–fasta* argument udáva cestu ku priečinku obsahujúci všetky fasta súbory so scaffoldmi, ktoré sa majú použiť v predikcii. Argument *–name* hovorí, ktorý natrénovaný model sa má použiť pre predikciu. Posledný prepínač je voliteľný, defaultne sa nastaví na *0*. Udáva či sa bude evaluovať len na vstupnom reťazci *+*, na jeho komplementárnom opačnom doplnku - alebo na oboch *0*. Posledný voliteľný argument *–hints* udáva cestu ku GFF súborom, ktoré sa môžu použiť pri segmentácii. Napovedia algoritmu, kde sa nachádzajú dané známe intervaly pre intróny, exóny, štart/stop kodón. Názov týchto súborov sa musí zhodovať s názvom FASTA súboru. Výstupom tejto predikcie sú vždy dva súbory, *gff* súbor s anotáciami a *html* súbor s farebnou vizualizáciou segmentovaných scaffoldov a transkriptov. Voliteľný argument *–html* udáva koľko takýchto predikcií sa má vizualizovať. Nadmerné množstvo vstupov môže viesť k nečitateľnému výstupu kvôli jeho veľkosti. Argument *–conf* špecifikuje cestu ku konfiguračnému súboru pre maximálne a minimálne dĺžky sekvencií jednotlivých stavov.

Poslednou možnosťou je voľba predikcie za použitia viacerých natrénovaných modelov. Táto možnosť je určená hlavne pre predikciu na metagenóme:

```
python3 fungiGHMM -m [--fasta folder] [--models folder]
[[--strand +/-/0]] [[--hints folder]] [[--html int]] [[--conf file]]
```

kde argument *–fasta* udáva cestu k priečinku s fasta súbormi metagenómu, *–models* udáva cestu k priečinku s natrénovanými modelmi, ktoré sa majú použiť pri predikcii. *–hints*, *–strand*, *–html* a *–conf* má rovnakú funkciu ako v predchádzajúcom prípade.

Kapitola 7

Testovanie pravdepodobnostného modelu

Kapitola popisuje experimenty vykonané s implementovaným modelom, aká je presnosť modelu v rámci jeho taxonomického zovšeobecňovania a pri testovaní na metagenóme. Experimenty boli vykonávané pomocou skriptov v jazyku Python a sú súčasťou zdrojového kódu v module *tests*.

7.1 Porovnanie s CodingQuarry a s Augustus+

Výsledky experimentov sa nepodarilo porovnať s nástrojmi *CodingQuarry* a ani s *Augustus+*. Podľa stránky [1] s dostupnou distribúciou je nástroj CodingQuarry v dobe vzniku nášho nástroja zhruba tri roky starý projekt bez žiadnej ďalšej známky vývoja a údržby. Nástroj má nekvalitnú a chabú dokumentáciu. Nástroj nekontroluje formát vstupných dát. Očakáva vstup v podobe *FASTA* súboru a namiesto *GFF* anotácii očakáva anotácie vo formáte *GFF3*. Nástroj obsahuje skript v jazyku Python *CufflinksGTF_to_CodingQuarryGFF3.py*, ktorý slúži na prevod *GTF* anotácii do formátu *GFF3*, kde *GTF* formát je zhodným formátom s *GFF* formátom [72]. Skript sme použili na prevod dát to požadovaného formátu no pokus o natrénovanie modelov z našich dát končí chybou *Abort trap* a pádom aplikácie:

[1] `https://sourceforge.net/projects/codingquarry/`

```
terminate called after throwing an instance
of 'std::out_of_range'
    what():  basic_string::substr: __pos (which is 3) >
    this->size() (which is 0)
Abort trap: 6
```

Keďže aplikácia je dostupná so zdrojovým kódom v jazyku *C++*, je možné do kódu nahliadnuť. Pád aplikácie bol spôsobený Kód sme sa pokúsili opraviť doplnením kontroly pre dĺžku sekvencie, aby sa nepristupovalo mimo pole znakov. Segmentačná chyba sa tak síce odstránila, no pri predikcií daný nástroj vráti len prázdny súbor a žiadne anotácie sa nenájdu. Vývojárov aplikácie sme sa pokúsili kontaktovať [2] pre lepšie pochopenie problému s nástrojom, no odpoveď neprišla.

Pri trénovaní modelu pre nástroj *Augustus+* sme postupovali podľa dodaného postupu [3]. Nástroj vyžaduje pre trénovanie vstupné data vo formáte *GeneBank* [73]. V repozitári sa nachádza skript *gff2gbSmallDNA.pl*, ktorý zo vstupného *FASTA* a *GFF* súboru dokáže vytvoriť požadovaný *GebeBank* súbor. Následne sme pomocou skriptu *randomSplit.pl* rozdelili vytvorený *GeneBank* súbor na trénovaciu a testovaciu množinu a špecifikovali nový druh pomocou skriptu *new_species.pl*. Po spustení trénovania pomocou programu *etraining* pre daný nový druh a trénovaciu množinu dát program spadne a skončí segmentačnou chybou. Podľa špecifikácie v postupe je doporučený počet trénovacích génových štruktúr v intervale od 200 do 1000. My sme pri tvorbe súboru *GeneBank* použili dáta huby *Linpe1*, kde testovacia množina obsahovala 100 génových štruktúr a trénovacia množina 995 génových štruktúr. Príčina spadnutia aplikácie pri pokuse o natrénovanie je tak neznáma, čo je dôvodom nemožnosti použiť daný nástroj.

7.2 Typické chyby predikcie

V tejto kapitole si predstavíme vybrané chyby, ktoré sa vyskytujú pri predikcii implementovaným modelom. Ide hlavne o chyby špecifické pre detekciu intrónových úsekov, keďže tieto oblasti sú pre nás dôležité.

[2] theMiningSuite@gmail.com

[3] https://vcru.wisc.edu/simonlab/bioinformatics/programs/augustus/docs/tutorial2015/training.html

7.2.1 Nenájdenie intrónu

Model má problém nájsť intrónove oblasti v istých genómoch húb. Ide hlavne o problém, kedy má génová oblasť napríklad štyri intróny, no model objaví len jeden alebo žiaden intrón. Pokiaľ sa neobjaví ani jeden intrón, je daná génová oblasť označená za *exonSingle* stav. Úroveň odhalenia intrónových oblastí je teda nižšia, no ako ukazuje tabuľka testu taxonómie v ďalšej časti 7.3.10, presnosť určenia intervalov týchto intrónov dosahuje až 100 % pre isté taxóny.

Tento problém sa týka hlavne krátkych intrónových úsekov, ktoré majú okolo 30 nukleotidov. Príklad takejto predikcie je možné vidieť v prílohe B.1.

7.2.2 Falošné nálezy

Génové sekvencie môžu obsahovať viaceré pozície, ktoré pripomínajú *donor GT* a *akceptor AG*, a to aj napriek dostatočnému množstvu trénovacích dát. Výsledkom je následne nájdenie falošne pozitívneho intervalu intrónu. Špeciálnym ojedinelým prípadom je, keď je výsledkom predikcie intrónový interval s *donorom GT* a *akceptorom AG*, no skutočná pozícia intrónu je posunutá a začína *donorom CT* a končí *akceptorom TT*. Ukážka takéhoto prípadu je v prílohe B.2.

7.2.3 Readthrough jav

V génových oblastiach môže existovať ďalší stop kodón, ktorý sa nachádza niekde v priebehu génovej oblasti. Ide o takzvaný readthrough jav, kedy sa môže transkripcia nemusí zastaviť na tomto skoršom stop kodóne, ale môže pokračovať ďalej [70]. Nález takéhoto skoršieho stop kodónu môže viesť k tomu, že sa nenájdu všetky intrónové intervaly, lebo niektoré ležia za pozíciou prvého stop kodónu. Ďalšou možnosťou je, že sa celá segmentácia nálezom tohto skoršieho stop kodónu môže rozhodiť a nájdený intrón bude mať kratšiu dĺžku, ako v príklade v prílohy B.3.

7.2.4 Neukončenie génovej oblasti stop kodónom

Problém, ktorý sa vyskytoval pri evaluácii na génových oblastiach bolo, že niektoré predikcie neboli schopné nájsť stop kodón a daná segmentácia končila intrónom alebo exónom, aj keď sa v jeho intervale už nachádzal stop kodón. V tomto prípade je problémom nie len prehliadnutie stop kodónu ale aj nesprávne označenie poslednej intrónovej či exónovej sekvencie a to hlavne pravej strany tohto intervalu. Ukážka takéhoto príkladu je v prílohe B.4.

7.2.5 Označenie sekvencie za UTR

UTR model nie je obmedzený, čo sa týka jeho maximálnej dĺžky pri učení a predikcii. Je tu preto priestor pre chyby, keď celá vstupná sekvencia označí ako UTR segment, lebo to bola najpravdepodobnejšia možnosť a žiaden štart kodón sa neobjaví. Ukážka takejto chyby je dostupná v prílohe B.1.

7.3 Experimenty s modelom

V tejto časti sa zameriame na experimentovanie s modelom a skúsime zlepšiť výsledky predikcie nastavovaním jednotlivých parametrov modelu. Testy sú zamerané hlavne na genómy húb *Rhizoclosmatium globosum* a *Phascolomyces articulosus*, ktoré dosahujú nízke hodnoty presnosti predikcie.

Ako sme si už vysvetlili v predchádzajúcich kapitolách (napr. 5.1.4) o Viterbiho algoritme, zložitosť nájdenia najpravdepodobnejšej sekvencie stavov kvadraticky stúpa s počtom použitých stavov a s dĺžkou sekvencie. Ak uvážime, že napríklad pri taxonomickom teste 7.14 máme osem zástupcov a pre každého sedem taxónov, tak by sa jednalo o 56 nezávislých testov. Čas strávený takýto testovaním by bol neúprosne vysoký, ak by sme testovali celé úseky scaffoldov. Pre získanie presnosti fungovania implementovaného nástroja bolo preto potrebné redukovať dĺžky scaffoldov a evaluovať kratšie úseky. Z týchto dôvodov sme teda ako vstupné sekvencie zobrali len génové úseky sekvencií, ktoré majú dĺžku 100 – 2000 nukleotidov. V reálnom nasadení ale v metagenóme nebude takýto ideálny stav a je vysoko pravdepodobné, že dané scaffoldy metagenómu budú obsahovať UTR a medzigénové oblasti. Preto boli vstupné génové sekvencie rozšírené o prefix a sufix UTR oblasti.

Prefix UTR oblasť má dĺžku 40 nukleotidov a suffix UTR oblasť 200 nukleotidov. Voľba dĺžok týchto oblastí je viacmenej náhodná, dlhší UTR sufix má ale za cieľ trochu viac zmiasť detekciu stop kodónu. V praxi budú dané prefix a sufix oblasti vždy ináč dlhé, čo by sme mohli vynútiť aj u nás. Z dôvodu, že dané výsledky chceme ale porovnávať pre rôzne nastavenia parametrov, je vhodnejšie, aby boli vždy vstupné sekvencie rovnaké.

Čo sa týka učenia daného modelu, model sa pre každú evaluáciu segmentácie učí vždy z 80 % všetkých dostupných scaffoldov danej huby a zo zvyšných 20 %, ktoré model nevidel, sa vyberajú génové oblasti na evaluáciu. Pri učení všeobecnejších taxonomických modelov je okrem 80 % dostupných scaffoldov určenej huby evaluácie použitých vždy päť ďalších genómov rovnakého taxónu, ktoré sú vybrané podľa abecedného zoznamu dodaného taxonomického rozdelenia. Z týchto piatich genómov sa pre učenie použije vždy maximálne 300 scaffoldov, keďže učenie z piatich genómov pri väčšom počte taxonomických testov je pomerne časovo náročná záležitosť. Tabuľka 7.1 ukazuje počet použitých dostupných génových oblastí pre trénovanie na úrovni organizmov.

Huba	Počet génových oblastí trénovacej fázy (taxón Organism)
Schizosaccharomyces pombe	5128
Rhodotorula sp.	13150
Allomyces macrogynus	19273
Rhizoclosmatium globosum	20270
Rozella allomycis	10540
Encephalitozoon cuniculi	1994
Phascolomyces articulosus	13700
Linderina pennispora	8508

Tabuľka 7.1: Počet dostupných vstupných génových oblastí pre trénovaciu fázu na úrovni organizmov

Pri evaluácii pravdepodobnosti vstupnej sekvencie danými modelmi bola použitá konverzia do *log-odds* [6] formátu. Keďže násobenie veľkého počtu pravdepodobností z intervalu nula až jeden vedie k vzniku veľmi malého čísla, na výpočetnom stroji by došlo k vytvoreniu aritmetickej chyby a dané výsledky pravdepodobností vstupnej sekvencie by boli neporovnateľné. Z toho dôvodu sa používa funkcia logaritmu, ktorá dané nepresnosti odstráni. Prirodzený logaritmus je stúpajúca funkcia na celom definičnom obore a preto zachováva významovú hodnotu pôvodnej pravdepodobnosti. Namiesto produktov pravdepodobností sa tak snažíme maximalizovať sumu prirodze-

ných logaritmov týchto pravdepodobností. Takáto suma logaritmov ale trpí závislosťou na dĺžke vstupnej sekvencie. Pokiaľ chceme porovnávať pravdepodobnosti rôzne dlhých sekvencií je vhodné ich znormalizovať pomocou nulového modelu, čo je v našom prípade taký model, ktorý dáva rovnocenné pravdepodobnosti pre všetky prechody. Podiel pravdepodobnosti sekvencie z nášho modelu a z nulového modelu, resp. rozdiel logaritmov týchto dvoch pravdepodobností nám dá hodnotu *log-odds* [6]. Pokiaľ je hodnota *log-odds* kladná, je generovaná naším modelom, ak je záporná je generovaná nulovým modelom. Stále nám teda ide o maximalizáciu sumy *log-odds* hodnôt.

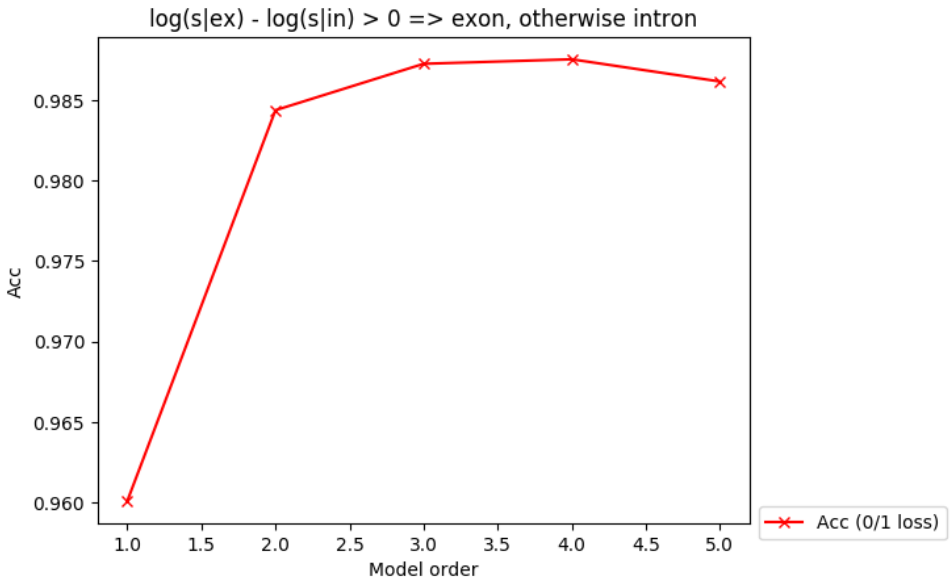
Pre vyhodnotenie presnosti predikcie intervalov sa používajú hodnoty *recall* a *precision* [74]. *Recall* je senzitivita, čiže časť odhalených relevantných prvkov z celkového počtu relevantných prvkov. *Precision* udáva koľko prvkov je relevantných z pomedzi všetkých odhalených prvkov. Pri intervaloch musia súhlasiť obidve hraničné hodnoty, aby sa jednalo o relevantný nález.

Okrem toho je pri evaluácii použitá ešte pozičná chyba, ktorá určuje časť nesprávnych nukleotidov vzhľadom na konkrétnu pozíciu v danej sekvencii. Je použitá 0/1 stratová funkcia [75], prideľujúca hodnotu jeden nesprávnemu nukleotidu na danej pozícii a hodnotu nula správnemu nukleotidu. Výsledkom je suma výsledkov stratovej funkcie delená dĺžkou sekvencie. Ide o emprickú chybu [76]. Táto chyba je merané jak pre celú sekvenciu, tak aj pre intrónové subsekvencie, kedy sa hranice podsekvencie určia ako maximálna hodnota z predikovanej a skutočnej hodnoty pravej strany intervalu a minimálna hodnota z predikovanej a skutočnej hodnoty ľavej strany intervalu. Doplnok tejto chyby do jednej udáva pozičnú presnosť.

7.3.1 Test klasifikácie intrónov a exónov

Ako prvý z experimentov bol vykonaný test správneho fungovania klasifikačnej úlohy oddeliť exóny od intrónov. Pri teste bolo pozorované chovanie presnosti klasifikácie v závislosti od stupňa daného modelu. Modely boli natrénované vždy s rovnakým stupňom histórie a následne evaluované. Test prebehol pomocou cross validácie [77]. Vstupné dáta boli rozdelené do štyroch rovnako veľkých celkov, kde tri boli použité na trénovaciu fázu a zvyšná množina bola použitá na testovanie. Výsledkom je krivka na obrázku 7.1, kde je vidno zlepšovanie presnosti testu po štvrtý stupeň modelov a mierny pokles pri stupni päť. Množinu vstupných dát predstavoval súbor scaffoldov huby *Armillaria solidipes*. Počet trénovacích sekvencií sa pohyboval v priemere 35000 exón a 35000 intrón sekvencií a test prebiehal v priemere na 15000 exón a 15000 intrón sekvenciach. Presnosť bola meraná pomocou 0/1 stratovej funkcie, ktorá prideľuje hodnotu nula správnej predikcii a hodnotu jeden

nesprávnej predikcii [75]. Podiel sumy výsledkov stratovej funkcie a počtu prvkov testovacej množiny nám dá empirickú chybu [76]. Doplnok tejto chyby je vyobrazený ako *Acc* na ose y v grafe 7.1. Ku klasifikácii danej sekvencie bol použitý rozdiel *log-odds* hodnôt exón a intrón modelu. Ak je daná hodnota pre exón model väčšia ako pre intrón model, tak sa daná sekvencia klasifikuje ako exón, v opačnom prípade ako intrón.

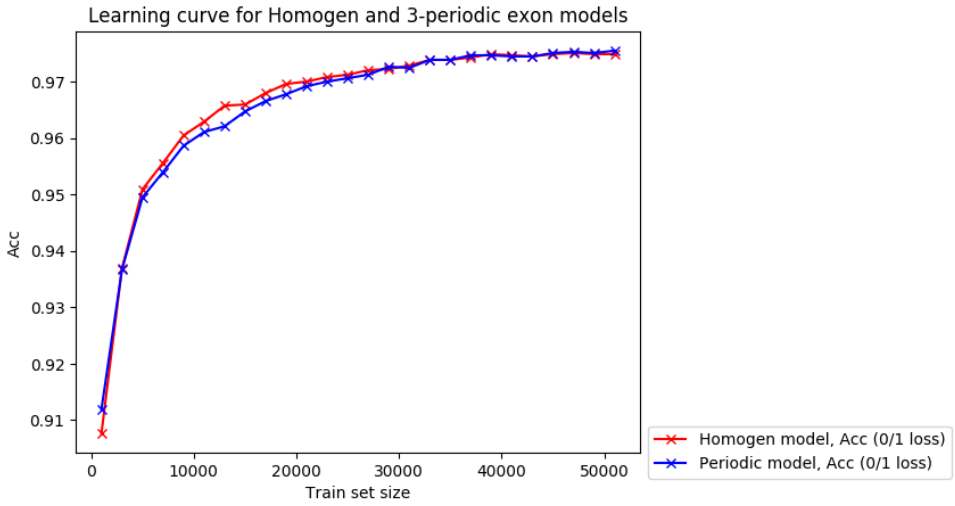


Obrázok 7.1: Krivka presnosti klasifikácie exónov od intrónov homogénnym modelom Markovských reťazcov. Presnosť *Acc* je v rozsahu nula až jeden.

Krivka nám ukazuje, že má zmysel používať modely vyšších stupňov. Pokiaľ odhliadneme od posledného poklesu pre stupeň päť, tak došlo vždy k istému zlepšeniu presnosti. Pri modeloch s najvyšším stupňom päť je presnosť o čosi nižšia. Dôvodom môže byť práve daný stupeň modelov. V dátach sa občas objavujú sekvencie kratšie ako je stupeň daného modelu. V kom prípade je pravdepodobnosť tejto vstupnej sekvencie rovná nule. Ak nám pre obidva modely vychádza nulová *log-odds* hodnota, tak sa evaluácia nevie správne rozhodnúť a vyhodnotí podmienku $logodds(s|ex) - logodds(s|in) > 0$ ako nepravdivú a prikloní sa ku intrón modelu, aj keď to nemusí byť pravda. Výskyt takýchto zmätení dokáže spôsobiť pokles presnosti. Druhou možnosťou je, že model vyššieho stupňa má viac parametrov učenia a videl málo vstupných sekvencií.

7.3.2 Test homogénneho a nehomogénneho Markovského reťazca

Test mal za cieľ overiť prínos nehomogénneho modelu Markovských reťazcov v porovnaní s homogénnym modelom. Test prebehol na klasifikačnej úlohe rozpoznať exóny od intrónov, pričom vznikla krivka učenia zobrazujúca závislosť počtu trénovacej množiny modelov a presnosti evaluácie. Test prebehol na scaffold sekvenciach huby *Saccharomyces cerevisiae*. Ako model pre intrón bol vždy zvolený homogénny model Markovského reťazca stupňa päť, modely pre exón sekvencie mali tiež vždy stupeň päť. Vstupné dáta boli rozdelené vždy v pomere 95 % pre trénovanie a 5 % pre testovanie. Test ukazuje, že nehomogénny model prináša len zlomkové zlepšenie, pokiaľ máme vstupnú množinu trénovacej fázy viac ako 40000 exónových sekvencií. Presnosť rozpoznania exónov od intrónov sa ustáli na úrovni 97 %. Túto závislosť vyjadruje obrázok 7.2. Presnosť bola meraná pomocou 0/1 stratovej funkcie. Na ose y je doplnok empirickej chyby predikcie.



Obrázok 7.2: Porovnanie presnosti klasifikácie exónov od intrónov pomocou nehomogénneho (3-periodického) a nehomogénneho modelu Markovských reťazcov. Na ose y je presnosť z intervalu [0,1]

Pre klasifikačnú úlohu sa nezdá byť periodický model veľmi prínosný. Pre prekonanie presnosti homogénneho modelu by sme museli mať nesmierne množstvo trénovacích sekvencií. Na druhej strane periodický model má tú výhodu, že má nastaviteľný parameter periódy. Je preto možné s ním simulovať chovanie homogénneho modelu, ktorý má periódu jeden. V kóde nám tak stačí implementácia jedného modelu, ktorý môžeme takto využiť v oblastiach, ktoré majú byť modelované ako homogénne.

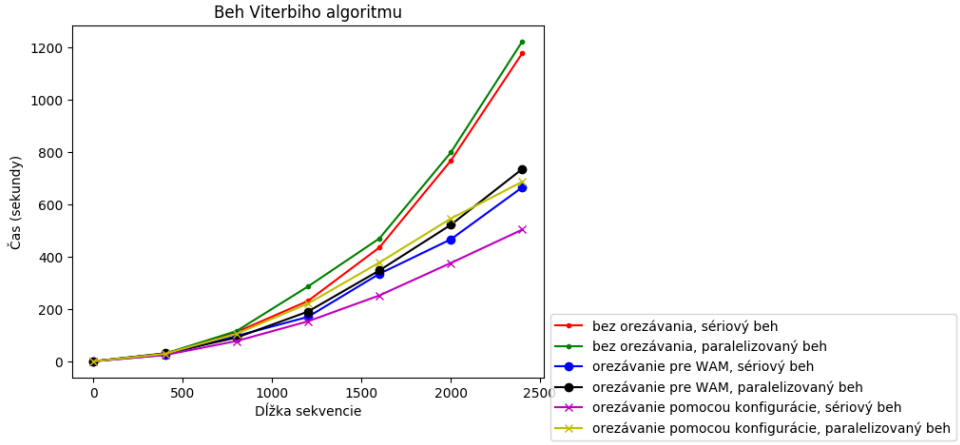
7.3.3 Test rýchlosti Viterbiho algoritmu

Viterbiho algoritmus pre GHMM bol formálne popísaný v časti 5.1.4 a v časti 6.1.1 bola popísaná jeho implementácia. Vďaka štatistikám získaným z dát genómov húb (viď prílohu C) je možné stanoviť isté horné a prípadne aj spodné hranice dĺžok vstupných sekvencií pre jednotlivé stavy modelu. Napríklad z grafu v prílohe C.5 je zrejmé aké sú horné hranice dĺžok intrónových sekvencií pre dané *phylum*. Jak popisuje časť C, konfigurácia s týmito hraničnými hodnotami dokáže urýchliť beh Viterbiho algoritmu, lebo sa netestujú podsekvencie, ktoré nespadajú do daných hodnôt dĺžok. Graf 7.3 ukazuje vplyv tohto orezávaného prehľadávania len pre WAM modely s fixnou dĺžkou a pre všetky modely za pomoci konfigurácie.

Pre WAM modely boli tak nastavené orezávania vzhľadom na ich fixné dĺžky, pre tento test to bolo jedenásť nukleotidov pre promoter, šesť nukleotidov pre splice sites, osemnásť nukleotidov pre stop kodón.

Test prebiehal na genómovom modeli a pre stavy s variabilnou dĺžkou bola zvolená konfigurácia nasledovne: *intron* <10, 980>, *exonSingle* <10, 2800>, *exonInter* <10, 980>, *exonStart* <10, 1222>, *exonEnd* <10, 1950> a *UTR* <1, inf). Tieto hodnoty vznikli zo štatistík v prílohe C. Pre horné hranice ide vždy o maximálnu hodnotu v daných priemerných hodnotách z pomedzi všetkých phylum. K tejto hodnote je pripočítaný ešte 10 % nukleotidov danej hodnoty, aby daná horná hranica bola viac benevolentná k prípadným novým outlierom. Ide tak o konfiguráciu, ktorá by mala byť dostatočne univerzálna naprieč phylum a namala by tak ovplyvniť výsledky presnosti žiadnej predikcie. Pri použití na metagenóme takáto konfigurácia bude iba prospešná. Prípadné dlhé UTR oblasti sa z vrchu neohraničia a podľa dostupných štatistík sú horné hranice ostatných stavov nastavené s dostatočnou rezervou. V praxi je tak možné rýchlejšie detekovať intrónove oblasti aj na dlhších vstupných scaffold sekvenciach.

Z grafu 7.3 môžeme vidieť markantné zlepšenie pri použití orezávania dĺžok podsekvencií. V praxi tak môžeme pre daný taxón v konfigurácii určiť minimálne a maximálne dĺžky vstupných sekvencií daných stavov a urýchliť tak predikciu pre dlhšie vstupné sekvencie. Hodnoty horných hraníc musia byť ale volené s rozvahou a vzhľadom na daný taxón huby. Ak nastavíme horné hranice moc nízke budeme ukončovať sekvencie moc skoro, čo nemusí byť v súlade so skutočným rozdelením v genóme danej huby. Ako ukazuje graf, paralelizácia pomocou vlákien v jazyku Python neprispieva k zrýchleniu predikcie. Je možné, že réžijné nároky tvorby samostatných vlákien pre každý stav v každej pozícii vstupnej sekvencie sú moc vysoké v porovnaní so sériovým behom segmentácie.



Obrázok 7.3: Graf rýchlosti behu Viterbiho algoritmu pre GHMM model v závislosti na dĺžke vstupnej sekvencie. Graf zobrazuje rýchlosť bez orezávania

7.3.4 Test predikcie génových úsekov bez intrónov

Aj keď cieľom práce je primárne správne určenie intrónových oblastí génov, môže byť prospešné vedieť, ako sa chovajú dané modely pre gény, ktoré neobsahujú intróny. Takýto test je vhodný napríklad pre huby, kde sa ukazuje nízky počet intrónov, napríklad huby z phylum *Microsporidia*, viď príloha C.6. Test prebehol pre 20 génových oblastí húb *Schizosaccharomyces pombe, Rhodotorula sp., Allomyces macrogynus, Rhizoclosmatium globosum, Rozella allomycis, Encephalitozoon cuniculi, Phascolomyces articulosus, Linderina pennispora* a to pre spojený Fungi model. Boli použité vstupné génové sekvencie vytvorené prístupom popísaným v úvode tejto kapitoly, viď strana 48.

Pozičná presnosť pre celé sekvencie je pomerne vysoká. Úroveň presností detekcie exónov sú v intervale 10 - 40 %. Keď sa pozrieme na pozičnú presnosť intrónov, vidíme, že okrem huby *Encephalitozoon cuniculi* je všade hodnota 0 %. To indikuje, že v predikcii boli falošne pozitívne nálezy, keďže dané oblasti neobsahujú intrónové sekvencie. Model podáva ale slušnú presnosť pri detekcii štart kodónov. Je vidno, že transkript model pridáva falošné nálezy nekódových oblastí, čo znižuje detekciu jednoexónových génov. Bolo by teda na uvážení, či nie je vhodnejšie použiť len genómový model, ktorý je komplexnejší a má špeciálne stavy pre posledný exón, ktoré lepšie vyjadrujú zakončenie génovej oblasti.

Huba	Recall (exón interval)	Precision (exón interval)	Pozičná presnosť	Pozičná presnosť (intrón)	Recall (štart/stop kodón)	Precision (štart/stop kodón)
Schizosaccharomyces pombe	35.0	33.3	87.3	0	75 / 35	72 / 33
Rhodotorula sp.	30	27.5	82.7	0	85 / 35	80 / 30
Allomyces macrogynus	30	30.6	71.5	0	60 / 35	63.9 / 33.3
Rhizoclosmatium globosum	15	16.7	73.8	0	80 / 20	88.9 / 22.2
Rozella allomycis	30	18.3	75.9	0	75 / 35	65 / 22.5
Encephalitozoon cuniculi	40	53.3	75.6	nan	55 / 50	73.3 / 63.3
Phascolomyces articulosus	10	11.1	67.8	0	70 / 5	77.8 / 5.6
Linderina pennispora	20	21.1	59	0	55 / 25	57.9 / 26.3

Tabuľka 7.2: Výsledky presností pre jednoexónové génové oblasti za použitia spojeného Fungi modelu.

7.3.5 Testy na presnosť intrónových intervalov génového modelu

Model umožňuje nastavovať rôzne parametre pre jednotlivé podmodely pre 3' a 5' splice site a podmodel samotného intrónu. V tejto časti sa zameráme na experimenty s týmito parametrami a porovnanie dosiahnutých výsledkov.

Testy s parametrami splice site modelu

Modely pre splice site 5' a 3' genómového modelu sú vytvorené za pomoci WAM modelu, ktorý obsahuje konfigurovateľné parametre ako fixná dĺžka a stupeň histórie. Z návrhu modelu popísaného v kapitole 5.2.2 je známe, že tieto parametre boli nastavené na hodnoty šesť pre dĺžku, kde tri nukleotidy sú z exónovej časti a tri z intrónovej a hodnotu dva pre stupeň histórie. Voľba iniciálnych hodnôt pre parametre bola inšpirovaná hodnotami z nástroja CodingQuarry [21]. Pozícia *donora GT* a *akceptora AG* je vždy v strede, to znamená štvrtá a piata pozícia sekvencie od konca pre *donor GT* a štvrtá a piata pozícia sekvencie od začiatku pre *akceptor AG*, tak aby boli vždy tri hraničné nukleotidy zo susedného exónu. Vyobrazenie je na obrázku 5.4.

Pri predikcii modelu na niektorých genómoch húb sa ale dosahuje markantne nižší výsledok presnosti správnej predikcie pozícii intrónov ako pri iných genómoch. Napríklad pri evaluácii na dvadsiatich génových oblastiach, ktoré

vznikli postupom popísaným v úvode kapitly na strane 48. Tieto génové oblasti vždy obsahujú intrónové sekvencie. Predikcia pre hubu *Rhizoclosmatium globosum (Rhihy1)* dosahuje pri týchto parametroch presnosť len *10 %* recall a *16.6 %* precision. Preto sa vykonali experimenty s evaluáciou na týchto testovacích dátach s rôznymi nastaveniami parametrov. Výsledky zobrazuje tabuľka 7.3.

Dĺžka splice site	Recall (intron interval)	Precision (intron interval)	Pozičná presnosť	Pozičná presnosť (intrón)
6	10 %	16.6 %	54.1 %	27.74 %
12	6.6 %	16.6 %	47.87 %	20.8 %
18	5.0 %	11.0 %	48.24 %	16.74 %

Tabuľka 7.3: Výsledky testu rovnomerného predlžovania splice site modelu do obidvoch strán pri umiestnení donora a akceptora so zarovnaním na stred sekvencie

Výsledok experimentu neukazuje zlepšenie, preto boli vykonané ďalšie experimenty s nerovnomerným predlžovaním. Umiestnenie *donora GT* je stále na štvrtej a piatej pozícii od začiatku sekvencie a umiestnenie *akceptora AG* je na štvrtej a piatej pozícii od konca sekvencie. Predlžovanie fixnej dĺžky sa tak vykonáva smerom do intrónovej oblasti. Výsledky testu zobrazuje tabuľka 7.4.

Dĺžka splice site	Recall (intron interval)	Precision (intron interval)	Pozičná presnosť	Pozičná presnosť (intrón)
6	10 %	16.6 %	54.1 %	27.74 %
12	5 %	9.09 %	50.74 %	18.51 %
18	6.6 %	18.18 %	51.78 %	28.3 %

Tabuľka 7.4: Výsledky testu predlžovania splice site modelu do intrónových strán pri umiestnení donora na štvrtej a piatej pozícii od začiatku sekvencie a umiestnení akceptora na štvrtej a piatej pozícii od konca sekvencie

Presnosť síce stúpla pri dĺžke 18 nukleotidov, no recall klesol. Je to zrejme spôsobené tým, že pokiaľ má huba krátke intróny, napr. s dĺžkou 30 nukleotidov, tak predlžovaním trénovacích sekvencií splice site modelu v smere do intrónov sa oberáme o trénovacie dáta intrón sekvencií a takéto krátke sekvencie intrónov sa model nenaučí. Pri predikcií model krátke sekvencie intrónov vynechá a predikuje skôr tie dlhšie. Ukážka výsledku takejto predikcie vyobrazená v prílohe B.1.

Ponúka sa teda ešte posledná alternatíva, predlžovanie splice site sekvencií v smere od intrónu, teda do exónových oblastí. Ide o variáciu k predchádzajúcemu testu, *donor* je na štvrtej a piatej pozícii od konca sekvencie a *akceptor* na štvrtej a piatej od začiatku sekvencie. Výsledky zobrazuje tabuľka 7.5.

Dĺžka splice site	Recall (intron interval)	Precision (intron interval)	Pozičná presnosť	Pozičná presnosť (intrón)
3	%	%	%	%
6	10 %	16.6 %	54.1 %	27.74 %
12	5 %	11.0 %	47.96 %	14.99 %
18	14.16 %	31.82 %	55.46 %	31.54 %

Tabuľka 7.5: Výsledky testu predlžovania splice site modelu do exónových strán pri umiestnení akceptora na štvrtej a piatej pozícii od začiatku sekvencie a umiestnení donora na štvrtej a piatej pozícii od konca sekvencie

Pri dĺžke 18 nukleotidov pre splice site sekvenciu a zarovnaní na okraje smerom k intrónom je vidno pomerne dobré zlepšenie v porovnaní s prvotným návrhom s dĺžkou sekvencie šesť nukleotidov a zarovnaním na stred. Presnosť sa zdvojnásobila a recall narástol o 4 %. Predlžovanie smerov do exónovej časti nespôsobuje stratu dát intrónových sekvencií. Ochudobňujeme sa zas ale o nukleotidy v exónoch. To nie je až taký problém, keďže exónové oblasti sú o dosť dlhšie v porovnaní s intrónovými, viď príloha C.

Vplyv dĺžky UTR sekvencie na správnosť predikcie intrónov

O genómoch húb vieme, že pri určitých druhoch je ich genóm pomerne hustý na génové oblasti, viď kapitola 2.2. CodingQuarry používa maximálnu dĺžku 300 nukleotidov pre UTR oblasti medzi génmi. Pri obmedzení UTR sekvencií na túto dĺžku vo fáze trénovania a predikcie v spojení s modelom pre splice site s fixnou dĺžkou nukleotidov s predĺžením do strany exónov sa ukazuje, že výsledok môže byť ešte o čosi presnejší. Test prebehol na rovnakých 20 génoch ako v predchádzajúcom teste 7.3.5 a to za pomoci genóm modelu. Tabuľka 7.6 zobrazuje dosiahnuté výsledky.

Dĺžka splice site	Recall (intron interval)	Precision (intron interval)	Pozičná presnosť	Pozičná presnosť (intrón)
3	18.33 %	31.37 %	75.36 %	42.89 %
6	11.66 %	17.65 %	73.83 %	33.88 %
12	12.66 %	23.53 %	73.81 %	33.48 %
18	16.83 %	32.35 %	74.87 %	38.6 %

Tabuľka 7.6: Výsledky testu predlžovania splice site modelu do exónových strán pri umiestnení akceptora na štvrtej a piatej pozícii od začiatku sekvencie a umiestnení donora na štvrtej a piatej pozícii od konca sekvencie a obmedzení maximálnej dĺžky UTR oblastí na 300 nukleotidov

Je vidno, že ak ponecháme dĺžku splice site na troch nukleotidoch a neberieme do úvahy susedné nukleotidy z exónov, tak stúpne recall na *18.33 %*, čo je takmer o dve percenta viac ako pri fixnej dĺžke 18 nukleotidov. Splice site model s tak krátkou fixnou dĺžkou má ale o čosi horšiu presnosť v porovnaní s modelom s dĺžkou 18 nukleotidov. Zdá sa, že kratší model tak dokáže odhaliť viac intrónových úsekov za cenu väčšieho počtu falošne pozitívnych nálezov.

V praxi ale môže byť problém s vynucovaním UTR sekvencie na maximálny počet 300 nukleotidov a to hlavne pri hubách, ktoré nemajú tak husté oblasti génov. V genómoch húb s dlhšími UTR oblasťami by sa tak mohli vyskytovať falošné nálezy génových oblastí, keďže by sme vynútili dĺžku UTR oblastí len na 300 nukleotidov, no skutočný rozsah by bol ďaleko väčší.

Test stupňa modelu intrónu

Pri jednotlivých modeloch môžeme nastavovať stupeň histórie daného modelu (*order*). Pre daný parameter bol vykonaný test na rovnakom genóme huby *Rhizoclosmatium globosum* s rovnakými génovými sekvenciami ako v teste 7.3.5. Pre splice site model boli použité parametre s najlepším výsledkom prechádzajúceho testu z tabuľky 7.6. Pre intrón model boli otestované stupne z intervalu dva až päť. Výsledky zobrazuje tabuľka 7.7.

Stupeň intrón modelu	Recall (intron interval)	Precision (intron interval)	Pozičná presnosť	Pozičná presnosť (intrón)
2	11.83 %	19.67 %	73.21 %	26.77 %
3	14.33 %	24.51 %	74.02 %	28.31 %
4	15.58 %	26.47 %	73.41 %	32.55 %
5	16.83 %	32.35 %	74.87 %	38.6 %

Tabuľka 7.7: Výsledky testu pre rôzne stupne histórie modelu pre intrón. Splice site model má fixnú dĺžku 18 nukleotidov s predĺžením do exónových oblastí. UTR sekvencie sú obmedzené na maximálnu dĺžku 300 nukleotidov.

Na výsledkoch je vidno mieru závislosti výslednej presnosti a stupňa modelu. Čím vyšší stupeň modelu, tým lepšia presnosť predikcie. Voľba piateho stupňa pre model intrónu je teda správna. Vedie k lepšej presnosti určenia intervalu daného intrónu a menším výskytom falošne pozitívnych nálezov. V metagenóme tak budeme schopný určovať viac relevantných intrónových oblastí s vyššou presnosťou a nižším výskytom falošných nálezov.

Test periodicity intrón modelu

Periodickosť genómu sa prejavuje hlavne v kódujúcich častiach. Pramení to so štruktúry genetického kódu, ktorý je vždy tvorený trojicou nukleotidov. Má preto opodstatnenie modelovať exónové sekvencie ako nehomogénne Markovské reťazce s periódou tri. Pre zaujímavosť bol vykonaný experiment, kde je aj intrón modelovaný ako nehomogénny Markovský reťazec s rôznymi variáciami periódy. Výsledky zobrazuje tabuľka 7.8.

Perióda intrón modelu	Recall (intron interval)	Precision (intrón interval)	Pozičná presnosť	Pozičná presnosť (intrón)
1	16.83 %	32.35 %	74.87 %	38.6 %
2	16.83 %	32.35 %	74.87 %	38.6 %
3	15.41 %	26.47 %	74.03 %	37.86 %
4	12.66 %	23.53 %	74.28 %	32.6 %

Tabuľka 7.8: Výsledky testu pre rôzne dĺžky periódy nehomogénneho modelu pre intrón. Splice site model má fixnú dĺžku 18 nukleotidov s predĺžením do exónových oblastí. UTR sekvencie sú obmedzené na maximálnu dĺžku 300 nukleotidov.

Nehomogénny model s rôznymi periódami pre intrón nemá zmysel, je jasné z výsledkov, že pri zavedení periódy začnú jednotlivé hodnoty presností pre intervaly a pozície klesať. Zmysel má používať teda len homogénny model.

7.3.6 Test promoter modelu génového modelu

Promoter model je modelovaný ako WAM model s fixnou dĺžkou jedenásť nukleotidov, kde prvých osem reprezentuje kozakovú sekvenciu a zvyšné tri štart kodón. Rovnako ako pri predchádzajúcom teste so splice site modelom boli vykonané experimenty s rôznou fixnou dĺžkou a umiestnením štart kodónu. Test prebiehal na rovnakých trénovacích a testovacích dátach ako v prípade predchádzajúcich testov a za pomoci génového modelu.

Tabuľka 7.9 ukazuje výsledky dosiahnuté pri predlžovaní dĺžky promoteru smerom do UTR oblastí. Je vidno, že predlžovanie promoteru nemá žiaden vplyv na zmenu výsledkov pre intervaly intrónov. Pre presnosti detekcie intervalu štart kodónu ale dochádza k nárastu hodnôt. Model s fixnou dĺžkou 25 nukleotidov dokáže odhaliť 70 % správnych štart kodónov s presnosťou 56.6 %.

Dĺžka promoter sekvencie	Recall (intron interval)	Precision (intrón interval)	Pozičná presnosť	Pozičná presnosť (intrón)	Recall (štart/stop kodón)	Precision (štart/stop kodón)
11	16.83 %	32.35 %	74.87 %	38.6 %	60/40 %	52.5/30 %
15	16.83 %	32.35 %	74.83 %	38.6 %	60/40 %	52.5/30 %
20	16.83 %	32.35 %	74.6 %	38.6 %	60/40 %	52.5/30 %
25	16.83 %	32.35 %	74.71 %	38.6 %	70/40 %	56.6/29 %

Tabuľka 7.9: Výsledky testu pre rôzne fixné dĺžky WAM modelu pre promoter, kde štart kodón je zarovnaný na posledné tri pozície sekvencie. Splice site model má fixnú dĺžku 18 nukleotidov s predĺžením do exónových oblastí. UTR sekvencie sú obmedzené na maximálnu dĺžku 300 nukleotidov.

Druhou alternatívou testu je rozširovať fixnú dĺžku promoteru smerom do exónovej časti, tak že prvých osem nukleotidov tvorí Kozaková sekvencia, následne tri reprezentujú štart kodón a ďalej sa rozširuje vždy len o násobky troch, aby sa zachovala periodicita. Výsledky experimentu zobrazuje tabuľka 7.10.

Dĺžka promoter sekvencie	Recall (intron interval)	Precision (intrón interval)	Pozičná presnosť	Pozičná presnosť (intrón)	Recall (štart/stop kodón)	Precision (štart/stop kodón)
11	16.83 %	32.35 %	74.87 %	38.6 %	60/40 %	52.5/30 %
14	16.83 %	32.35 %	74.2 %	36.02 %	60/35 %	52.5/27.5 %
17	16.83 %	32.35 %	74.2 %	36.02 %	60/35 %	52.5/27.5 %
20	16.83 %	35.29 %	75.37 %	43.1 %	60/45 %	52.6/32.5 %

Tabuľka 7.10: Výsledky testu pre rôzne fixné dĺžky WAM modelu pre promoter, kde štart kodón je zarovnaný na deviatu až jedenástu pozíciu sekvencie. Splice site model má fixnú dĺžku 18 nukleotidov s predĺžením do exónových oblastí. UTR sekvencie sú obmedzené na maximálnu dĺžku 300 nukleotidov.

Pri dĺžke promoteru 20 nukleotidov došlo k zlepšeniu presnosti intrónových intervalov. Prekvapivo došlo aj k zlepšeniu presnosti detekcie intervalov stop kodónov a to o 5 % recall a 2.5 % pre precision. Keďže detektor sa zameriava hlavne na správnu detekciu intrónových intervalov, je vhodné sa prikloniť k rozšíreniu do exónových oblastí s fixnou dĺžkou 20 nukleotidov. Predlžovaním promoter sekvencie smerom do UTR oblastí zvyšujeme recall a precision detekcie štart kodónových oblastí génov. V praxi to pre nás znamená pomerne slušnú šancu na detekciu začiatku génovej oblasti v neznámych DNA sekvenciach. Ak dokážeme odhaliť 65 % štart kodónov z pomedzi všetkých testovaných génových oblastí, dáva nám to istotu, že začiatky génových oblastí aj napriek UTR prefixu odhalíme.

7.3.7 Test stop kodón modelu génoveho modelu

Obdobne ako pri modeli promoteru v predchádzajúcom teste, aj model stop kodónu je tvorený WAM modelom. Boli vykonané experimenty s naťahovaním fixnej dĺžky smerom do exónovej časti, tak aby posledné tri pozície boli vždy zastúpené daným stop kodónom. Výsledky zobrazuje tabuľka 7.11.

Dĺžka stop kodón sekvencie	Recall (intron interval)	Precision (intrón interval)	Pozičná presnosť	Pozičná presnosť (intrón)	Recall (štart/stop kodón)	Precision (štart/stop kodón)
6	12.7 %	23.5 %	75.1 %	36.1 %	60 / 45 %	52.5 / 32.5 %
9	15.16 %	27.8 %	73.2 %	39.7 %	60 / 35 %	55 / 25 %
15	15.16 %	27.8 %	75 %	39.7 %	60 / 40 %	55 / 30 %
18	16.83 %	35.29 %	75.37 %	43.1 %	60 / 45 %	52.5 / 32 %

Tabuľka 7.11: Výsledky testu pre stop kodón model pri predlžovaní do exónovej časti. Splice site model má fixnú dĺžku 18 nukleotidov s predĺžením do exónových oblastí. UTR sekvencie sú obmedzené na maximálnu dĺžku 300 nukleotidov. Promoter má dĺžku 20 nukleotidov s predĺžením do exónovej časti.

Pri stop kodóne sa stretávame hneď s dvomi typmi chyby. Buď sa daný stop kodón nenájde a niektorý zo stavov pre exón alebo intrón pokryje miesto stop kodónu, čím nám potencionálne vzniká priestor na vyššie chyby intervalových a nukleotidových pozícii. Segmentácia sa tak nezastaví niekde na stop kodóne, ale predĺži poslednú exónovú sekvenciu, prípadne vloží ďalšiu intrónovú sekvenciu, ktorá tak vytvorí falošne pozitívny nález intrónu. Detekcia správneho miesta stop kodónu je v tomto prípade zásadná. Druhým prípadom je takzvaný *readthrough* jav, ktorý sme si popísali v časti 7.2.3. Detekuje sa skorší stop kodón, ako by sa malo. Pokiaľ je takýto skorší stop kodón súčasťou exónovej sekvencie, tak nám vzniká priestor na menší recall intrónových intervalov, keďže prípadný nasledujúci intrón sa tak neodhalí.

To v praxi pri metagenóme nie je až taký problém, je hlavne dôležité nájsť čo najmenej falošne pozitívnych intrónov, aj keby ich malo byť menej, je teda uprednostňovaný precision pred recall. Horšou alternatívou by bolo nájdenie stop kodónu v takej oblasti, ktorá by bola v skutočnosti intrónom. Vtedy by sa daný intrón mohol skrátiť na pravej strane a táto oblasť by bola segmentovaná ako postupnosť stavov *intron -> exonEnd -> stopCodon*, prípadne by sa mohlo stať, že daný intrón sa celý označí za *exonEnd* a predĺži sa tak exónová oblasť pred týmto intrónom. Druhý prípad je pre nás zas o niečo prijateľnejší ako ten prvý. Je lepšie, ak sa daný intrón nepredikuje vôbec, ako keby sme netrafili pravú stranu jeho intervalu. Vznikla by tak nepresnosť, ktorá by rozbila čítací rámec transkriptu.

7.3.8 Porovnanie predikcie génovým, transkriptovým a spojeným modelom

Z predchádzajúcich testov je vidno, že samostatný génový model má svoje nedostatky a nedosahuje zlepšenia bez nastavovania daných parametrov. Na nasledujúcich výsledkoch je možné vidieť porovnanie výsledkov pre predikciu na dvadsiatich génových úsekoch genómu huby *Rhizoclosmatium globosum* za pomoci samostatných *genóm* a *transkript* modelov a nakoniec za pomoci spojeného modelu, kde sa v *genóm* modeli využívajú výsledky predikcie *transkript* modelu. Výsledky sú v tabuľke 7.12.

model	Recall (intrón interval)	Precision (intrón interval)	Pozičná presnosť	Pozičná presnosť (intrón)	Recall (štart/stop kodón)	Precision (štart/stop kodón)
trankript model	25.6 %	28.9 %	74.8 %	36.9 %	75 / 5 %	75 / 5 %
genóm model	12.7 %	23.5 %	73.8 %	33.5 %	60 / 40 %	52.5 / 30 %
spojený Fungi model	26.8 %	39.7 %	68.2 %	46.7 %	70 / 15 %	82.4 / 17.6 %

Tabuľka 7.12: Výsledky testu pre génový, transkriptový a spojený model. Splice site model má fixnú dĺžku 12 nukleotidov s predĺžením do exónových oblastí. UTR sekvencie sú obmedzené na maximálnu dĺžku 300 nukleotidov. Promotér má dĺžku 11 nukleotidov s predĺžením do exónovej časti. Stop kodón má dĺžku 18 nukleotidov.

Keďže *transkript* model nemá modelované špeciálne stavy pre poslednú exónovu oblasť, ako je tomu pri genómovom modeli, kde je samostatný stav *exonEnd*, tak *transkript* model má problém s určovaním stop kodónu. Ak má napríklad génová oblasť štyri exóny, tak pravdepodobnosť prechodu z *CDS* do *NCDS* je 75 % a prechod z *CDS* do *stopKodon* stavu len 15 %. Model preto vo väčšine prípadov uprednostní prechod do ďalšieho *CDS* stavu. *Genóm* model tento jav mierne vyrovnáva pri použití spojeného modelu, no ako

vidno na výsledkoch, dostávame sa len na hodnotu 15 % pre recall *stopKodon*. Je teda otázne či v praxi budeme voliť medzi lepšou presnosťou detekcie génových oblastí, na čo sa viacej hodí genómový model, alebo zvolíme vyššiu mieru detekcie intrónových oblastí. V našich testovaných génových oblastiach je dĺžka sufix UTR oblasti 200 nukleotidov. V prípade ale, že by bola sufix UTR sekvencia dlhšia je otázne či transkript model nebude skákať medzi *CDS* a *NCDS* oblasťami aj v tejto UTR oblasti. Miera falošne pozitívnych nálezov intrónov by tak mohla byť veľmi vysoká. Z toho dôvodu sa v oblastiach *CDS* a *NCDS* z detekcie transkript modelu, používaných ako nápovedy pre genóm model, povoľuje možnosť výskytu stop kodónu a v určitej miere aj možnosť výskytu UTR oblastí (viď kapitola 6.1.3).

7.3.9 Test skrytých semi-Markovských modelov

Aj keď Viterbiho algoritmus simuluje dĺžku zotrvania v danom stave (viď kapitolu 5.1.4), bol vykonaný experiment, kedy stavy ako *intron, exonSingle, exonStart, exonInter, exonEnd* boli modelované ako skryté semi-Markovské modely [78], kde pravdepodobnosť zotrvania v danom stave závisí aj na čase strávenom v danom stave.

Pre tento účel sa spomenuté stavy učia frekvencie výskytov dĺžok vstupných sekvencií pri trénovaní. Z týchto vektorov frekvencií je možné sa naučiť pravdepodobnostné rozdelenie daných dĺžok pre daný stav. Pri predikcii vo Viterbiho algoritme sa tak penalizujú sekvencie, ktoré nevyhovujú danej distribúcii a naopak, cesty kde je daná dĺžka sekvencie pravdepodobnejšia sú uprednostnené. Výsledok tohto testu je vidno na tabuľke 7.13.

s uprednostňovaním génov s aspoň jedným intrónom. Test prebehol na genóme huby *Schizosaccharomyces pombe* pomocou genómového modelu.

Typ testu	Recall (intrón interval)	Precision (intrón interval)	Pozičná presnosť	Pozičná presnosť (intrón)	Recall (štart/stop kodón)	Precision (štart/stop kodón)
GHMM	35.8	61.5	87.2	79.6	45 / 80	35 / 70
GHSMM	0	nan	19.2	nan	0 / 0	nan / nan

Tabuľka 7.13: Výsledky testu pre semi-Markovské modely. Splice site model má fixnú dĺžku 18 nukleotidov s predĺžením do exónových oblastí. UTR sekvencie sú neobmedzené. Promotér má dĺžku 20 nukleotidov s predĺžením do exónovej časti. Stop kodón má dĺžku 18 nukleotidov s predĺžením do exónovej časti. Vstupné data tvorilo 20 génových oblastí vytvorených postupom popísaným na strane 7.3

Problém so semi-Markovskými modelmi je ten, že až moc penalizujú dané

pravdepodobnosti stavov. Ak by sme chceli zachovať konzistentnosť, mali by byť všetky stavy semi-Markovské. V praxi pri metagenóme pre UTR oblasti ale nedáva zmysel, aby boli semi-Markovské. Ak sa z dát naučíme že najpravdepodobnejšie dĺžky UTR oblastí sú okolo 2000 nukleotidov a potom dostaneme v metagenóme sekvenciu s krátkym UTR prefixom, napr. 40 nukleotidov, tak takýto prefix budeme penalizovať pravdepodobnosťou blízkou nule a daná oblasť sa nikdy neoznačí za UTR sekvenciu. Ak na druhej strane nastavíme ako semi-Markovské modely iba všetky exónové modely a intrónové modely a UTR model bude nesemi-Markovský, tak budeme uprednostňovať UTR stavy pred inými stavmi. Semi-Markovské modely penalizujú danú sekvenciu na základe jej dĺžky. Ak by UTR stav nebol semi-Markovský, k žiadnej penalizácii na základe dĺžky by nedochádzalo a pravdepodobnosť UTR oblastí tak bude vyššia. To sa prejavilo aj v našom teste. Výsledok všetkých predikcií bol, že vstupná sekvencia je UTR oblasť, keďže UTR nebolo penalizované pravdepodobnosťou zotrvania v danom stave na rozdiel od zvyšných homogénnych a nehomogénnych Markovských stavov.

7.3.10 Test taxonomickej segmentácie

Okrem scaffold sekvencií a anotácii máme k dispozícii aj taxonómiu dát. Je preto možné naučiť sa modely z viacerých genómov pre dané taxonomické úrovne a vyhodnotiť či zovšeobecňovanie modelov je prospešné pre danú predikciu. Pre tento test boli vybrané nasledujúce huby: *Schizosaccharomyces pombe, Rhodotorula sp., Allomyces macrogynus, Rhizoclosmatium globosum, Rozella allomycis, Encephalitozoon cuniculi, Phascolomyces articulosus, Linderina pennispora.* Zástupcovia boli volení tak, aby každý spadal do samostatného *phylum* a zároveň aby v ich iných taxonomických úrovniach bolo vždy dostatok ďalších organizmov k učeniu. Ide o strašne veľa testov, kde sa daný model musí vždy natrénovať pre daný taxón a následne evaluovať.

Vznikla teda tabuľka určujúca presnosti precision a recall pre intrónové oblasti, pozičnú presnosť celej predikcie a pozičnú presnosť intrónových úsekov. Vstupné dáta boli vytvorené postupom popísaným v úvode kapitoly na strane 48. Použili sa teda úseky génových oblastí s prefix a sufix UTR oblasťami. Trénovanie prebehlo tiež postupom popísaným v tomto úvode kapitoly. Pre evaluáciu bol použitý genómový model. V každom taxonomickom teste sa tak model učil maximálne z piatich *FASTA* súborov z rovnakého taxónu danej hub a z 80 % percent scaffold sekvencií danej huby. Test prebehol na dvadsiatich génových oblastiach zo zvyšných 20 % scaffoldov danej huby, ktoré model pri trénovaní nevidel. Pre test boli uprednostňované tie génové oblasti, ktoré obsahovali aspoň jeden intrón. Ak takéto gény pri testovaní neboli dostupné, napríklad huba *Encephalitozoon cuniculi* pochádza z phylum, kde nemáme dostupné génové oblasti s intrónmi (viď príloha C.6), tak sa

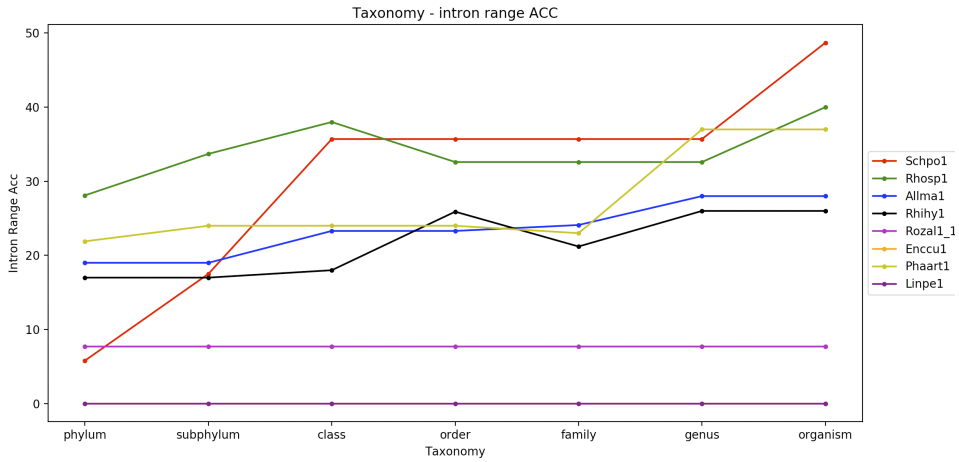
do evaluácie zahrnuli aj gény s jedným exónom. Vzorkovanie testovaných dát na oblasti s aspoň jedným intrónom bolo zvolené za účelom merania primárne intervalových a pozičných presností pre intróny, keďže sa náš nástroj zameriava hlavne na detekciu intrónových oblastí. Stále ale existuje možnosť, že sa do testovania predikcie zahrnie aj gén s jedným exónom. Vďaka tomu majú testované dáta stále istú mieru diverzity, takže sa môžu stále prejaviť potenciálne chyby, kde by gén s jedným exónom bol segmentovaný ako gén s viacerými intrónmi a vznikal by tak priestor pre falošne pozitívne nálezy. Prioritizovaním výberu testovaných génových oblastí s aspoň jedným intrónom klesá možnosť takejto zámeny. Preto prebehol ešte samostatný test 7.3.4, ktorý testuje fungovanie modelu na génových oblastiach s jedným exónom. Pre test bola použitá rovnaká konfigurácia ako sme si popísali v kapitole 7.3.3. WAM modely boli nastavené tak, že promotér mal dĺžu 20 nukleotidov, kde štart kodón začínal na deviatej pozícii. Stop kodón model mal dĺžku 18 nukleotidov, kde samotný stop kodón bol umiestnený na posledných troch pozíciach. Splice site model mal dĺžku 18 nukleotidov, kde tri boli vždy z intrónovej časti a zvyšných 15 z exónovej časti.

Fungi	Phylum	Subphylum	Class	Order	Family	Genus	Organism
Schizosaccharomyces pombe	2.5 / 9.1 68.9 / 35.8	22.5 / 12.5 71.5 / 24.9	30.8 / 40.6 80.7 / 55.6	30.8 / 40.6 80.7 / 55.6	30.8 / 40.6 80.7 / 55.6	30.8 / 40.6 80.7 / 55.6	35.8 / 61.5 79.6 / 87.2
Rhodotorula sp.	19.2 / 37 69.8 / 49.3	20.1 / 47.4 70.4 / 65.7	28.6 / 47.5 69.8 / 56.4	29.9 / 35.3 75.8 / 47.2	29.9 / 35.3 75.8 / 47.2	29.9 / 35.3 75.8 / 47.2	34.6 / 45.4 78.1 / 58.5
Allomyces macrogynus	15.8 / 22.2 69.6 / 38.1	15.8 / 22.2 69.6 / 38.1	25 / 21.7 73.7 / 37.8	25 / 21.7 73.7 / 37.8	24.2 / 24.1 74.5 / 36.4	29.2 / 26.9 76 / 39.7	29.2 / 26.9 76 / 39.7
Rhizoclosmatium globosum	18.3 / 15.8 65.6 / 21.1	18.3 / 15.8 65.6 / 21.1	18.3 / 17.5 65.4 / 25.2	18.5 / 33.3 71.1 / 43.2	15.8 / 26.5 72.4 / 40.5	16.8 / 35.3 75.4 / 43.1	16.8 / 35.3 75.4 / 43.1
Rozella allomycis	3.4 / 11.9 62.3 / 38.1	3.4 / 11.9 62.3 / 38.1	3.4 / 11.9 62.3 / 38.1	3.4 / 11.9 62.3 / 38.1	3.4 / 11.9 62.3 / 38.1	3.4 / 11.9 62.3 / 38.1	3.4 / 11.9 62.3 / 38.1
Encephalitozoon cuniculi	0 / nan 95.6 / nan	0 / nan 95.6 / nan	0 / nan 95.6 / nan	0 / nan 95.6 / nan	0 / nan 94.4 / nan	0 / nan 94.4 / nan	0 / nan 96.6 / nan
Phascolomyces articulosus	22.5 / 21.3 70 / 27.1	23.8 / 24.2 70.7 / 27.5	23.8 / 24.2 70.2 / 27.4	23.8 / 24.2 70.2 / 27.4	23.8 / 24.2 70.5 / 28	37.5 / 36.4 73 / 38.5	37.5 / 36.4 73 / 38.5
Linderina pennispora	0 / 0 78.2 / 5.8	0 / 0 57.7 / 4	0 / 0 59.1 / 4.9	0 / 0 59.1 / 4.9	0 / 0 59.1 / 4.9	0 / 0 62.9 / 5.6	0 / 0 62.9 / 5.6

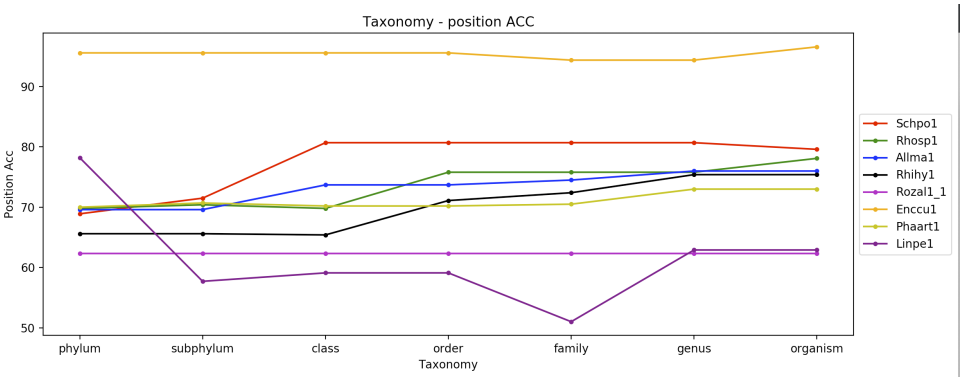
Tabuľka 7.14: Výsledky taxonomickej predikcie. Prvý riadok pre danú hubu udáva hodnoty recall / precision pre intervaly intrónov. Druhý riadok danej huby udáva vždy pozičnú presnosť celej sekvencie / pozičnú presnosť intrónových úsekov. Výsledky sú dosiahnuté použitím genómového modelu.

Na výsledkoch je vidno, že modely pre phylum sú až moc všeobecné a zrejme diverzita húb v tomto taxóne je tak vysoká, že daný model podáva pomerne nízke presnosti predikcie. Keď prechádzame k špecifickejším taxónom, tak sa presnosť vo všeobecnosti zvyšuje a najviac dosahuje presnosť pre daný organizmus. Pridávaním ďalších genómov do učenia vrámci daného taxónu vedie k zníženiu sily modelu, no na druhej strane získame model, ktorý nie je tak špecifický a úzko zameraný na jednu hubu. To má praktický význam pre predikciu na metagenóme, kde kvôli zložitosti samotného Viterbi algoritmu (viď 5.1.4) si nemôžeme dovoliť segmentovať sekvencie pomocou 950 modelov, kde každý model je naučený len na jednej hube. Použitie

všeobecnejších modelov nám redukuje počet daných modelov, ktoré musíme použiť. Z tabuľky 7.14 vznikli dva grafy, 7.4, kde vidíme vývoj presnosti detekcie intervalov intrónov a graf 7.5, kde vidíme vývoj pozičnej presnosti vrámci daných taxónov.



Obrázok 7.4: Graf vývoja intervalovej presnosti pre intróny vzhľadom na daný taxón pre všetkých osem húb taxonomickej evaluácie. Hodnoty na ose y vznikli ako priemer hodnôt recall a precision pre intervaly intrónov z tabuľky 7.14



Obrázok 7.5: Graf vývoja pozičnej presnosti pre celú predikovanú sekvenciu vzhľadom na daný taxón pre všetkých osem húb taxonomickej evaluácie. Hodnoty na ose y sú pozičné presnosti celých sekvencií z tabuľky 7.14

Z grafu 7.4 vidíme, že posúvaním k špecifickejším taxónom danej huby dosahujeme vyššiu presnosť predikcie intervalov intrónov. Modely pre phylum sú na tom najhoršie, preto by bolo vhodné v praxi použiť iný taxón pre segmentáciu metagenómu. Najlepšou voľbou sa zdá byť taxón *trieda*, kde je pomerne vysoký skok presnosti v porovnaní s predchádzajúcimi úrovňami. Na druhej strane máme 45 rôznych tried, čo vedie k dosť vysokému počtu samostatných modelov. V praxi bude nutné vybrať iba 4 — 5 majoritných tried a to za pomoci odborníkov v molekulárnej biológii pre huby.

Zaujímavou hubou z grafu vyššie je huba *Enccu1*. V grafe 7.4 dosahuje hodnotu presnosti 0 % pre intervaly intrónov. V pozičnej presnosti má ale výsledok okolo 95 %. Táto huba teda má veľmi nízky počet intrónov (viď príloha C.6) a gény s jedným exónom tak boli predikované vo všeobecnosti správne.

Pre huby *Linpe1* a *Rozal1_1* sú úrovne presností jak intervalov intrónov, tak aj pozičné presnosti blízke nule. V nasledujúcej tabuľke 7.15 ale vidíme zlepšenie predikce za pomoci použitie spojeného Fungi modelu na úrovni oragnizmu. Pre zjednodušenie testu boli evaluované len taxóny tried a samotné organizmy.

Fungi	Class (Genom model)	Organism (Genom model)	Class (Fungi model)	Organism (Fungi model)
Schizosaccharomyces pombe	30.8 / 40.6 80.7 / 55.6	35.8 / 61.5 79.6 / 87.2	53.3 / 47.4 79.7 / 54.1	64.2 / 53.3 84.4 / 65.8
Rhodotorula sp.	28.6 / 47.5 69.8 / 56.4	34.6 / 45.4 78.1 / 58.5	36.4 / 33.7 72.3 / 49.5	43.3 / 38.1 68.6 / 48.6
Allomyces macrogynus	25 / 21.7 73.7 / 37.8	29.2 / 26.9 76 / 39.7	22.5 / 17.9 71.6 / 36.5	40 / 34.2 75 / 38.4
Rhizoclosmatium globosum	18.3 / 17.5 65.4 / 25.2	16.8 / 35.3 75.4 / 43.1	24.5 / 20.8 69.7 / 28	21.4 / 22.2 68.2 / 36.4
Rozella allomycis	3.4 / 11.9 62.3 / 38.1	3.4 / 11.9 62.3 / 38.1	10 / 19.7 67.5 / 54.1	10 / 19.7 67.5 / 54
Encephalitozoon cuniculi	0 / nan 95.6 / nan	0 / nan 96.6 / nan	0 / nan 69.6 / nan	0 / nan 75.2 / nan
Phascolomyces articulosus	23.8 / 24.2 70.2 / 27.4	37.5 / 36.4 73 / 38.5	50.8 / 35.9 74.7 / 41.7	53.8 / 36.2 77.8 / 45.6
Linderina pennispora	0 / 0 59.1 / 4.9	0 / 0 62.9 / 5.6	0 / 0 57.6 / 3.7	3.3 / 2.8 59.1 / 3.2

Tabuľka 7.15: Výsledky taxonomickej predikcie. Prvý riadok pre danú hubu udáva hodnoty recall / precision pre intervaly intrónov. Druhý riadok danej huby udáva vždy pozičnú presnosť celej sekvencie / pozičnú presnosť intrónových úsekov. Výsledky sú dosiahnuté použitím spojeného Fungi modelu.

Na úrovni organizmov vidíme nárast recall pri použití spojeného Fungi modelu v porovnaní so samostatným genómovým modelom. Transkript model tak odhalí viacej kódových a nekódových častí z relevantných oblastí. Vytvára ale aj priestor pre vyšší počet falošne pozitívnych nálezov, precesion na úrovni organizmov je nižšia pre Fungi model ako pre samostatný genóm model. To môže byť spôsobené práve neodhalením stop kodónu a zamenením tejto oblasti za ďalší intrón.

Na úrovni tried sa tiež prejavil podobný problém, no v menšej miere. Huby *Schizosaccharomyces pombe, Rhizoclosmatium globosum, Rozella allomycis a Phascolomyces articulosus* dosahujú omnohe lepšie výsledky pre Fungi model

ako pre samostatný genóm model jak pre recall tak aj pre precision. Pri hube *Encephalitozoon cuniculi*, ktorá nemá intrónove oblasti, je možné pozorovať markantné zhoršenie detekcie z 95.6 % na 69.6 %. Pre génove oblasti bez intrónov bude teda lepšie voliť samostatný genóm model namiesto spojeného transkript modelu. Posledná huba *Linderina pennispora* začala vykazovať aspoň nejaké miery detekcie intrónov na úrovni organizmu. Pri triede ale hodnoty recall a precision zas klesli na nulu. Zovšeobecňovaním modelu tejto huby vzniká zrejme moc vysoká diverzita, ktorá model mýli, prípadne je otázne správne nastavenie parametrov daného modelu.

7.3.11 Test segmentácie metagenómu

Cieľom práce je finálne nasadenie nástroja na detekciu génových oblastí v neznámom metagenóme, ktorý obsahuje viacero organizmov, v našom prípade vo väčšine prípadov ide hlavne o huby, no metagenóm môže obsahovať aj iné organizmy ako napríklad baktérie či vyššie eukaryotá. Štruktúra metagenómu bola popísaná v kapitole 4.2. Ak máme zaručené, že každý scaffold vo vstupnom *FASTA* súbore reprezentuje len jeden neznámy organizmus, tak môžeme daný metagenóm segmentovať T nezávislými GHMM modelmi a časová zložitosť takejto segmentácie pre jeden scaffold bude $O(T(m_t^2 l^2 + m_g^2 l^2))$, kde m_t je počet stavov transkriptového modelu a m_g je počet stavov genómového modelu, l je dĺžka scaffoldu. Je preto výhodnejšie, ak vstupné scaffoldy sú čo najkratšie, keďže časová náročnosť rastie kvadraticky s dĺžkou. Za predpokladu, že by sme vedeli dopredu určiť aspoň hranice génových oblastí ako štart a stop kodón a orezali dané scaffoldy na tieto podreťazce, tak by sa rýchlosť predikcie značne zväčšila. Je pravdepodobné, že v praxi ale takým luxusom moc disponovať nebudeme a budeme musieť pracovať buďto na stroji, ktorý má veľký výpočetný výkon alebo zaplatíme väčším časom stráveným na segmentácii. Aj keď môžeme vylepšiť rýchlosť behu segmentácie pomocou konfiguračného súboru použitého modelu, tak rýchlosť Viterbiho algoritmu je jednou zo slabín daného nástroja. Implementácia behu tohto algoritmu v inom jazyku, napríklad *C++* by rýchlosť segmentácie určite zlepšilo.

Pri reálnom nasadení nástroja na metagenóm je otázne ako správne inicializovať apriórne pravdepodobnosti (*prior probabilities*). Pri scaffold sekvencii metagenómu sa na začiatku môžeme nachádzať v hociktorom stave GHMM modelu. Ak by sme použili naučené apriórne pravdepodobnosti z našich dát, tak by sme vynucovali začiatok segmentácie v stavoch *UTR* a *promoter*, keďže naše trénovacie scaffold sekvencie začínajú iba týmito stavmi. To by možno nebolo na škodu pri niektorých hubách, ktoré majú veľa UTR skvencií a menej husté génové oblasti, čo by sa dalo odvodiť zo štatistík o počte génových oblastí v prílohe C. Na druhej strane, pri hubách s genómom bohatým na génove oblasti bude väčšia pravdepodobnosť, že scaffold sekvencie metagenómu

budú pochádzať práve z týchto génových oblastí a pri použití apriórnych pravdepodobností z našich dát by sme tak dané sekvencie mohli segmentovať s väčšou chybou. V výsledných modeloch pre predikciu na metagenóme sme preto zvolili rovnocenné nastavenie apriórnych pravdepodobností na hodnotu $\frac{1}{|m_g|}$ pre všetky stavy genóm modelu a na hodnotu $\frac{1}{|m_t|}$ pre transkript model. Keďže každý stav má rovnakú pravdepodobnosť byť ako prvý, tak vzniká aj potenciál na väčšiu nepresnosť segmentácie. Pri našich predošlých testoch sme uvažovali vždy apriórne pravdepodobnosti naučené zo vstupných dát.

Pri teste na metagenóme si musíme stanoviť počet modelov T, ktoré použijeme na segmentáciu. Je prakticky nemožné mať pre každú hubu samostatný model, keďže len v našich trénovacích dátach je okolo 950 rôznych húb. Zložitosť segmentácie pre jeden scaffold by tak bola $O(950(m_t^2 l^2 + m_g^2 l^2))$. Ako ukazuje výsledok taxonomickej evaluácie v teste 7.3.10 ani použitie samostatných ôsmych modelov pre dané *phylum* nie je vhodnou voľbou. Mali by sme síce len osem samostatných modelov, zložitosť by tak bola $O(8(m_t^2 l^2 + m_g^2 l^2))$, no úroveň generalizácie týchto modelov je moc vysoká a presnosť segmentácie je dosť nízka pre intrónové oblasti v tak generalizovaných modeloch. Lepšou voľbou je zvoliť iný taxón pre tvorbu modelov, napríklad taxón triedy a natrénovať modely pre 4 — 5 majoritných tried určených po konzultácii s molekulárnym biológom, keďže počet tried našich dát je 45. Počet unikátnych hodnôt iných taxónov stúpa smerom nadol k daným organizmom. Napríklad taxón rad *(order)* má 123 unikátnych kategórii, taxón čeľaď (*family*) má až 306 unikátnych kategórii.

Test nad metagenómom je problematické vykonať, nakoľko metatranskript nemá mapované transkript sekvencie voči metagenómu a pre každý predikovaný transkript by bolo nutné prehľadať celý metatranskript a použiť algoritmus lokálneho zarovnania (*Smith- Waterman algoritmus*, *BLAST*, [51, 52]). Test bude predmetom ďalšieho výskumu a pokračovania práce.

Kapitola 8

Záver

Vznikol komplexný nástroj pre detekciu intrónových úsekov v génomových segmentoch, ktorý je založený na generalizovanom skrytom Markovskom modeli. Model si berie to najlepšie z nástroju Augustus, a to je možnosť využiť nápovedy a upresniť tak výslednú detekciu intrónových segmentov. Nápovedy nie sú využívané v plnom rozsahu ako je to pri nástroji Augustus, čo ale ani nie je potrebné pre riešenie daného problému. Využívať nápovedy získané z databáz proteínov či BLAT / BLAST nemá zmysel, nakoľko oblasť genómu húb nie je tak moc prebádaná a počet dostupných informácii z týchto databáz by mohol modelu skôr priškodiť ako pomôcť [33], keďže ich je málo. Model preto využíva a implementuje skôt poznatky nástroja CodingQuarry, jeho štýl predikcie pomocou RNA transkriptov a ako sú modelované jednotlivé stavy GHMM modelu.

Okrem toho je ale model špecifický pre oblasť húb a vychádza zo štatistík spočítaných z dát (viď príloha C). Daný GHMM model teda obsahuje špecifické stavy ako *exonSingle, exonInter, exonEnd, exonStart*, ktoré lepšie odrážajú štruktúru genómov. Model využíva triky na zrýchlenie výpočtu Viterbiho algoritmu a to za pomoci konfiguračného súboru, ktorý je možné špecifikovať na vstupe. Nástroje Augustus a CodingQuarry takú možnosť nemajú a hodnoty sa musia meniť priamo v kóde. Taktiež sú k dispozícii prednastavené konfiguračné hodnoty pre dané phylum. Hodnoty je možné meniť podľa potreby.

Náš nástroj pre predikciu sa je taktiež schopný učiť súčasne z viacerých trénovacích FASTA súborov, čím je možné trénovanie na základe taxonomického rozdelenia a vytvorenie tak všeobecnejších modelov pre dané huby. Zmyslom

tohto prístupu je ušetriť čas predikcie, keďže pri množstve 950 organizmov húb je v praxi nemožné vytvoriť model pre každý organizmus a následne spustiť predikciu nad neznámou hubovou DNA sekvenciou, prípadne nad neznámymi DNA sekvenciami metagenómu, aby dobehla v rozumnom čase.

S nástrojom bolo vykonaných nespočetne veľa rôznych testov a pokusov. Len samotné taxonomické testovanie pozostáva z 56 nezávislých testovaní a učení modelu. Navyše boli vykonané rôzne experimenty s parametrami daného modelu, aby sa maximalizoval výsledok predikcie intrónových oblastí.

Nástroj je schopný výsledné predikcie vypísať do *GFF* súboru, čo môže byť praktické pre ďalšie použitie, napríklad pri hľadaní BLAST zarovnaní daných kódujúcich častí v databázach. Okrem toho dokáže nástroj vizualizovať dané predikcie v *html* výstupe a je tak pomerne jednoduché získať predstavu o pozíciach daných intrónov či exónov.

Okrem toho, že sa nástroj dokáže učiť na základe taxonómie, tak dokáže aj predikovať vstupnú sekvenciu s využitím práve viacerých taxonomických modelov. Výsledkom je potom predikcia toho modelu, ktorý ma najväčšiu pravdepodobnosť.

Nástroj bol implementovaný v jazyku Python, čo má svoje úskalia ako napríklad rýchlosť učenia či segmentácie. Kód je ale štrukturovaný, treba podotknúť že ide o úplne samostatnú implementáciu GHMM modelu a modelov WAM, homogénnych a nehomogénnych Markovských modelov. Kód je teda využiteľný nie len pri segmentácii genómu húb, ale pri hocijakej inej úlohe, ktorá by sa dala vhodne modelovať pomocou GHMM modelu.

Výkonnosť modelu nie je až tak vysoká, ako by sa zrejme očakávalo. Presnosť sa mení v závislosti na vstupných genómoch húb, pre niektoré organizmy dosahujeme 50 % presnosť intrónových intervalov, pri iných zas nedokážeme detekovať žiadne intróny. Je tu teda priestor na ďalšie zlepšovanie a experimentovanie. Model je pomerne silný pri detekcii začiatku génovej oblasti a štart kodónu (až 84 % presnosť pre *Rhizoclosmatium globosum*). Ponúka sa teda možnosť fúzie tohto GHMM nástroja s SVM prístupom, keďže daný GHMM model má pomerne nízky prah falošne pozitívnych nálezov pre určité taxóny a genómy. Táto fúzia by mohla byť predmetom ďalšieho pokračovania výskumu segmentácie.

Dodatok A

Literatúra

[1] Thomas D Pollard, William C Earnshaw, and Jennifer Lippincott-Schwartz. *Cell Biology E-Book*. Elsevier Health Sciences, 2007.

[2] The Genome Portal of the Department of Energy Joint Genome Institute. Mycocosm fungi genome database, 2012. `http://jgi.doe.gov/fungi`.

[3] Jason E. Stajich. Fungal genomes and insights into the evolution of the kingdom. *Microbiol Spectr*, 5(4):10.1128/microbiolspec.FUNK–0055–2016, Jul 2017. 28820125[pmid].

[4] Tapan Kumar Mohanta and Hanhong Bae. The diversity of fungal genome. *Biological Procedures Online*, 17(1):8, Apr 2015.

[5] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb 1989.

[6] Anders Krogh. An introduction to hidden markov models for biological sequences. *New Comprehensive Biochemistry*, 32, 12 1998.

[7] David Kulp David Haussler and Martin G Reese Frank H Eeckman. A generalized hidden markov model for the recognition of human genes in dna. In *Proc. int. conf. on intelligent systems for molecular biology, st. louis*, pages 134–142, 1996.

[8] B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J.D. Watson. *Molecular Biology of the Cell*. Garland, 4th edition, 2002.

[9] Henderson James Cleaves. *Watson–Crick Pairing*, pages 1775–1776. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[10] Ning Yu, Zeng Yu, Bing Li, Feng Gu, and Yi Pan. A comprehensive review of emerging computational methods for gene identification. *Journal of Information Processing Systems*, 12(1), 2016.

[11] Antonio Mora, Geir Kjetil Sandve, Odd Stokke Gabrielsen, and Ragnhild Eskeland. In the loop: promoter-enhancer interactions and bioinformatics. *Brief Bioinform*, 17(6):980–995, Nov 2016. 26586731[pmid].

[12] Katsura Asano. Why is start codon selection so precise in eukaryotes? *Translation (Austin)*, 2(1):e28387–e28387, Mar 2014. 26779403[pmid].

[13] Stephen T. Eskesen, Frank N. Eskesen, Brian Kinghorn, and Anatoly Ruvinsky. Periodicity of dna in exons. *BMC Mol Biol*, 5:12–12, Aug 2004. 15315715[pmid].

[14] Laurent Chavatte, Stéphanie Kervestin, Alain Favre, and Olivier Jean-Jean. Stop codon selection in eukaryotic translation termination: comparison of the discriminating potential between human and ciliate erf1s. *EMBO J*, 22(7):1644–1653, Apr 2003. 12660170[pmid].

[15] David L Hawksworth. The magnitude of fungal diversity: the 1· 5 million species estimate revisited. *Mycological research*, 105(12):1422–1432, 2001.

[16] Richard A Humber. Entomophthoromycota: a new phylum and reclassification for entomophthoroid fungi. *Mycotaxon*, 120(1):477–492, 2012.

[17] David S Hibbett, Manfred Binder, Joseph F Bischoff, Meredith Blackwell, Paul F Cannon, Ove E Eriksson, Sabine Huhndorf, Timothy James, Paul M Kirk, Robert Lücking, et al. A higher-level phylogenetic classification of the fungi. *Mycological research*, 111(5):509–547, 2007.

[18] J. E. Galagan, M. R. Henn, L. . J. Ma, C. Cuomo, and B. Birren. Genomics of the fungal kingdom: insights into eukaryotic biology. *Genome Res*, 15, 2005.

[19] Anders Tunlid and Nicholas J Talbot. Genomics of parasitic and symbiotic fungi. *Current opinion in microbiology*, 5(5):513–519, 2002.

[20] Dmitri A Petrov. Evolution of genome size: new approaches to an old problem. *TRENDS in Genetics*, 17(1):23–28, 2001.

[21] Alison C. Testa, James K. Hane, Simon R. Ellwood, and Richard P. Oliver. Codingquarry: highly accurate hidden markov model gene prediction in fungal genomes using rna-seq transcripts. *BMC Genomics*, 16(1):170, Mar 2015.

[22] Brian J. Haas, Qiandong Zeng, Matthew D. Pearson, Christina A. Cuomo, and Jennifer R. Wortman. Approaches to fungal genome annotation. *Mycology*, 2(3):118–141, 2011.

[23] Christopher Burge. *Identification of Genes in Human Genomic DNA*. PhD thesis, 1997.

[24] Waack S Stanke M. Gene prediction with a hidden markov model and new intron submodel. *Bioinformatics*, 19(Suppl2):ii2|5–ii225, 2003.

[25] Krogh A. Two methods for improving performance of an hmm and their application for gene finding. *Proc Fifth Int Conf Intelligent Systems for Molecular Biology*, pages 179–186, 1997.

[26] Guigó R Parra G, Enrique B. Geneld in drosophila. *Genome Research*, 10:511–515, 2000.

[27] Brent MRR Gross SS. Using multiple alignments to improve gene prediction. 2005.

[28] Pachter LR Alexandersson M, Cawley S. Slam: Cross-species gene finding and alignment with a generalized pair hidden markov model. *Genome Research*, I3:496–502, 2003.

[29] Durbin RR Meyer IM. Comparative ab initio prediction of gene structures using pair hmms. *Bioinformatics*, I8(I0):I309–I318, 2002.

[30] Gargh S Sczyrba A Morgenstern B Taher L, Rinner O. Agenda: gene prediction by cross-species sequence comparison. *Nucleic Acids Research*, 32:W305–W308, 2004.

[31] Burge C Yeh RF, Lim LP. Computational inference of homolo- gous gene structures in the human genome. *Genome Research*, II:803–816, 2001.

[32] Durbin R Birney E, Clamp M. Gene wise and genomewise. *Genome Research*, I4:988–995, 2004.

[33] Mario Stanke, Oliver Schöffmann, Burkhard Morgenstern, and Stephan Waack. Gene prediction in eukaryotes with a generalized hidden markov model that uses hints from external sources. *BMC Bioinformatics*, 7(1):62, Feb 2006.

[34] Vardges Ter-Hovhannisyan, Alexandre Lomsadze, Yury O Chernoff, and Mark Borodovsky. Gene prediction in novel fungal genomes using an ab initio algorithm with unsupervised training. *Genome research*, pages gr–081612, 2008.

[35] A. Lomsadze, P. D. Burns, and M. Borodovsky. Integration of mapped rna-seq reads into automatic training of eukaryotic gene finding algorithm. *Nucleic Acids Res*, 42, 2014.

[36] J. E. Allen, M. Pertea, and S. L. Salzberg. Computational gene prediction using multiple sources of evidence. *Genome Research*, 14, 2004.

[37] D. DeCaprio, J. P. Vinson, M. D. Pearson, P. Montgomery, M. Doherty, and J. E. Galagan. Conrad: gene prediction using conditional random fields. *Genome Res*, 17, 2007.

[38] G. Schweikert, A. Zien, G. Zeller, J. Behr, C. Dieterich, C. S. Ong, P. Philips, F. De Bona, L. Hartmann, A. Bohlen, N. Kruger, S. Sonnenburg, and G. Ratsch. mgene: accurate svm-based gene finding with an application to nematode genomes. *Genome Res*, 19, 2009.

[39] D. M. Kupfer, S. D. Drabenstot, K. L. Buchanan, H. Lai, H. Zhu, and D. W. Dyer. *Introns and splicing elements of five diverse fungi introns and splicing elements of five diverse fungi* †. 2004.

[40] Ian Reid, Nicholas O'Toole, Omar Zabaneh, Reza Nourzadeh, Mahmoud Dahdouli, Mostafa Abdellateef, Paul MK Gordon, Jung Soh, Gregory Butler, Christoph W. Sensen, and Adrian Tsang. Snowyowl: accurate prediction of fungal genes by using rna-seq and homology information to select among ab initio models. *BMC Bioinformatics*, 15(1):229, Jul 2014.

[41] A. M. McGuire, M. D. Pearson, D. E. Neafsey, and J. E. Galagan. Cross-kingdom patterns of alternative splicing and splice recognition. *Genome Biol*, 9, 2008.

[42] S. Nakagawa, Y. Niimura, T. Gojobori, H. Tanaka, and K. Miura. Diversity of preferred nucleotide sequences around the translation initiation codon in eukaryote genomes. *Nucleic Acids Res*, 36, 2008.

[43] 1000 fungal genomes project. http://1000.fungalgenomes.org,.

[44] Fungal genome initiative. http://www.broadinstitute.org/scientific-community/science/projects/fungal-genome-initiative/fungal-genome-initiative,.

[45] T. Steijger, J. F. Abril, P. G. Engström, F. Kokocinski, J. F. Abril, and M. Akerman. Assessment of transcript reconstruction methods for rna-seq. *Nat Methods*, 10, 2013.

[46] Autor neznámy. Markov chain monte carlo, 2019. `https://en.wikipedia.org/wiki/Markov_chain_Monte_Carlo`.

[47] AD Neverov, MS Gelfand, and AA Mironov. Gipsygene: a statistics-based gene recognizer for fungal genomes. *Biophysics*, 48(1):71–75, 2003.

[48] C. Holt and M. Yandell. Maker2: an annotation pipeline and genome-database management tool for second-generation genome projects. *BMC Bioinformatics*, 12, 2011.

[49] K. J. Hoff, S. Lange, A. Lomsadze, M. Borodovsky, and M. Stanke. Braker1: Unsupervised rna-seq-based genome annotation with genemarket and augustus. *Bioinformatics*, 32, 2016.

[50] Chuhu Yang, Eugene Bolotin, Tao Jiang, Frances M Sladek, and Ernest Martinez. Prevalence of the initiator over the tata box in human and yeast genes and identification of dna motifs enriched in human tata-less core promoters. *Gene*, 389(1):52–65, 2007.

[51] Temple F Smith, Michael S Waterman, et al. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981.

[52] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990.

[53] Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

[54] W James Kent. Blat—the blast-like alignment tool. *Genome research*, 12(4):656–664, 2002.

[55] Yuzhen Ye. Hidden markov models, 2013. `http://homes.sice.indiana.edu/yye/lab/teaching/get.php?course=spring2013-I529&name=HMM.pdf`.

[56] Richard Durbin, Sean R Eddy, Anders Krogh, and Graeme Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

[57] Tamara Steijger, Josep F Abril, Pär G Engström, Felix Kokocinski, Martin Akerman, Tyler Alioto, Giovanna Ambrosini, Stylianos E Antonarakis, Jonas Behr, Paul Bertone, et al. Assessment of transcript reconstruction methods for rna-seq. *Nature methods*, 10(12):1177, 2013.

[58] John Henderson, Steven Salzberg, and Kenneth H Fasman. Finding genes in dna with a hidden markov model. *Journal of Computational Biology*, 4(2):127–141, 1997.

[59] NCBI. A. query input and database selection, 2019. `https://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastDocs&DOC_TYPE=BlastHelp`.

[60] Michael D. Barton and Hazel A. Barton. Scaffolder - software for manual genome scaffolding. *Source Code Biol Med*, 7(1):4–4, May 2012. 22640820[pmid].

[61] NCBI. Annotating genomes with gff3 or gtf files, 2018. `https://www.ncbi.nlm.nih.gov/genbank/genomes_gff/`.

[62] Hong Wang, Hen-ming Wu, and Alice Y. Cheung. Pollination induces mrna poly(a) tail-shortening and cell deterioration in flower transmitting tissue. *The Plant Journal*, 9(5):715–727, 1996.

[63] Karl Sigman. Discrete-time markov chains, 2009. http://www.columbia.edu/~ks20/stochastic-I/stochastic-I-MCI.pdf.

[64] Mark Craven. Markov chain models (part ii), 2011. https://cw.fel.cvut.cz/b172/_media/courses/bin/markov-chains-2.pdf.

[65] Michael Zhang and T G Marr. A weight array method for splicing signal analysis. *Computer applications in the biosciences : CABIOS*, 9:499–509, 11 1993.

[66] G David Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[67] Jiajie Wei, Ying Zhang, Ivaylo P. Ivanov, and Matthew S. Sachs. The stringency of start codon selection in the filamentous fungus neurospora crassa. *J Biol Chem*, 288(13):9549–9562, Mar 2013. 23396971[pmid].

[68] Ian Korf, Paul Flicek, Daniel Duan, and Michael R Brent. Integrating genomic homology into gene structure prediction. *Bioinformatics*, 17(suppl_1):S140–S148, 2001.

[69] Jing Wu and David Haussler. Coding exon detection using comparative sequences. *Journal of Computational Biology*, 13(6):1148–1164, 2006.

[70] Irwin Jungreis, Michael F. Lin, Rebecca Spokony, Clara S. Chan, Nicolas Negre, Alec Victorsen, Kevin P. White, and Manolis Kellis. Evidence of abundant stop codon readthrough in drosophila and other metazoa. *Genome Res*, 21(12):2096–2113, Dec 2011. 21994247[pmid].

[71] Ed. T. Bray. The javascript object notation (json) data interchange format, 2017. https://tools.ietf.org/html/rfc8259.

[72] EMBL-EBI. Gff/gtf file format - definition and supported options, 2019. https://www.ensembl.org/info/website/upload/gff.html.

[73] NCBI. Genbank flat file format, 2006. https://www.ncbi.nlm.nih.gov/Sitemap/samplerecord.html.

[74] David Martin Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. 2011.

[75] Claude Sammut and Geoffrey I. Webb, editors. *Zero-One Loss*, pages 1031–1031. Springer US, Boston, MA, 2010.

[76] Vladimir Vapnik. Principles of risk minimization for learning theory. In *Advances in neural information processing systems*, pages 831–838, 1992.

[77] Autor neznámy. Cross-validation (statistics), 2019. https://en.wikipedia.org/wiki/Cross-validation_(statistics).

[78] Yann Guédon. Estimating hidden semi-markov chains from discrete sequences. *Journal of Computational and Graphical Statistics*, 12(3):604–639, 2003.

Dodatok B

Typické chyby predikcie

B.1 Krátke intróny

scaffold_26-

Info

Gene Name: fgenesh1_kg.26_#_52_#_NP_009887.1

Protein ID: 38718

Sequence

25853	AAACATACATCTATCCCGTTATGAAGTTTTCTGCTGGTGCCGTCCTGTCATGGTCCTCCCTGCTGCTCGCCTCCTCTGTT	25933
25933	TTCGCCCAACAAGAGGCTGTGGCCCCTGAAGACTCCGCTGTCGTTAAGTTGGCCACCGACTCCTTCAATGAGTACATTCA	26013
26013	GTCGCACGACTTGGTGCTTGCGGAGTTTTTTGCTCCATGGTGTGGCCACTGTAAGAACATGGCTCCTGAATACGTTAAAG	26093
26093	CCGCCGAGACTTTAGTTGAGAAAAACATTACCTTGGCCCAGATCGACTGTACTGAAAACCAGGATCTGTGTATGGAACAC	26173
26173	AACATTCCAGGGTTCCCAAGCTTGAAGATTTTCAAAAACAGCGATGTTAACAACTCGATCGATTACGAGGGACCTAGAAC	26253
26253	TGCCGAGGCCATTGTCCAATTCATGATCAAGCAAAGCCAACCTGCTGTCGCCGTTGTTGCTGATCTACCAGCTTACCTTG	26333
26333	CTAACGAGACTTTTGTCACTCCAGTTATCGTCCAATCCGGTAAGATTGACGCCGACTTCAACGCCACCTTTTACTCCATG	26413
26413	GCCAACAAACACTTCAACGACTACGACTTTGTCTCCGCTGAAAACGCAGACGATGATTTCAAGCTTTCTATTTACTTGCC	26493
26493	CTCCGCCATGGACGAGCCTGTAGTATACAACGGTAAGAAAGCCGATATCGCTGACGCTGATGTTTTTGAAAAATGGTTGC	26573
26573	AAGTGGAAGCCTTGCCCTACTTTGGTGAAATCGACGGTTCCGTTTTCGCCCAATACGTCGAAAGCGGTTTGCCTTTGGGT	26653
26653	TACTTGTTCTACAATGACGAGGAAGAATTGGAAGAATACAAGCCTCTCTTTACCGAGTTGGCCAAAAAGAACAGAGGTCT	26733
26733	AATGAACTTTGTTAGCATCGATGCCAGAAAATTCGGCAGACACGCCGGCAACTTGAACATGAAGGAACAATTCCCTCTAT	26813
26813	TTGCCATCCACGACATGACTGAAGACTTGAAGTACGGTTTGCCTCAACTCTCTGAAGAGGCGTTTGACGAATTGAGCGAC	26893
26893	AAGATCGTGTTGGAGTCTAAGGCTATTGAATCTTTGGTTAAGGACTTCTTGAAAGGTGATGCCTCCCCAATCGTGAAGTC	26973
26973	CCAAGAGATCTTCGAGAACCAAGATTCCTCTGTCTTCCAATTGGTCGGTAAGAACCATGACGAAATCGTCAACGACCCAA	27053
27053	AGAAGGACGTTCTTGTTTTGTACTATGCCCCATGGTGTGGTCACTGTAAGAGATTGGCCCCAACTTACCAAGAACTAGCT	27133
27133	GATACCTACGCCAACGCCACATCCGACGTTTTGATTGCTAAACTAGACCACACTGAAAACGATGTCAGAGGCGTCGTAAT	27213
27213	TGAAGGTTACCCAACAATCGTCTTATACCCAGGTGGTAAGAAGTCCGAATCTGTTGTGTACCAAGGTTCAAGATCCTTGG	27293
27293	ACTCTTTATTCGACTTCATCAAGGAAAACGGTCACTTCGACGTCGACGGTAAGGCCTTGTACGAAGAAGCCCAGGAAAAA	27373
27373	GCTGCTGAGGAAGCCGAAGCTGACGCCGAAGCCGAAGCTGACGCTGACGCTGAATTGGCTGACGAAGAAGATGCCATTCA	27453
27453	CGATGAATTGTAATTCTGATCACTTTGGTTTTT	27533

Legend:

AACGT - Non coding

AACGT - Exon

AACGT - Intron

Obrázok B.2: Predikované štruktúra génovej oblasti

Info

Gene Name: fgenesh1_kg.26_#_52_#_NP_009887.1

Protein ID: 38718

Sequence

25853 AAACATACATCTATCCCGTTATGAAGTTTTCTGCTGGTGCCGTCCTGTCATGGTCCTCCCTGCTGCTCGCCTCCTCTGTT 25933

25933 TTCGCCCAACAAGAGGCTGTGGCCCCTGAAGACTCCGCTGTCGTTAAGTTGGCCACCGACTCCTTCAATGAGTACATTCA 26013

26013 GTCGCACGACTTGGTGCTTGCGGAGTTTTTTGCTCCATGGTGTGGCCACTGTAAGAACATGGCTCCTGAATACGTTAAAG 26093

26093 CCGCCGAGACTTTAGTTGAGAAAAACATTACCTTGGCCCAGATCGACTGTACTGAAAACCAGGATCTGTGTATGGAACAC 26173

26173 AACATTCCAGGGTTCCCAAGCTTGAAGATTTTCAAAAACAGCGATGTTAACAACTCGATCGATTACGAGGGACCTAGAAC 26253

26253 TGCCGAGGCCATTGTCCAATTCATGATCAAGCAAAGCCAACCTGCTGTCGCCGTTGTTGCTGATCTACCAGCTTACCTTG 26333

26333 CTAACGAGACTTTTGTCACTCCAGTTATCGTCCAATCCGGTAAGATTGACGCCGACTTCAACGCCACCTTTTACTCCATG 26413

26413 GCCAACAAACACTTCAACGACTACGACTTTGTCTCCGCTGAAAACGCAGACGATGATTTCAAGCTTTCTATTTACTTGCC 26493

26493 CTCCGCCATGGACGAGCCTGTAGTATACAACGGTAAGAAAGCCGATATCGCTGACGCTGATGTTTTTGAAAAATGGTTGC 26573

26573 AAGTGGAAGCCTTGCCCTACTTTGGTGAAATCGACGGTTCCGTTTTCGCCCAATACGTCGAAAGCGGTTTGCCTTTGGGT 26653

26653 TACTTGTTCTACAATGACGAGGAAGAATTGGAAGAATACAAGCCTCTCTTTACCGAGTTGGCCAAAAAGAACAGAGGTCT 26733

26733 AATGAACTTTGTTAGCATCGATGCCAGAAAATTCGGCAGACACGCCGGCAACTTGAACATGAAGGAACAATTCCCTCTAT 26813

26813 TTGCCATCCACGACATGACTGAAGACTTGAAGTACGGTTTGCCTCAACTCTCTGAAGAGGCGTTTGACGAATTGAGCGAC 26893

26893 AAGATCGTGTTGGAGTCTAAGGCTATTGAATCTTTGGTTAAGGACTTCTTGAAAGGTGATGCCTCCCCAATCGTGAAGTC 26973

26973 CCAAGAGATCTTCGAGAACCAAGATTCCTCTGTCTTCCAATTGGTCGGTAAGAACCATGACGAAATCGTCAACGACCCAA 27053

27053 AGAAGGACGTTCTTGTTTTGTACTATGCCCCATGGTGTGGTCACTGTAAGAGATTGGCCCCAACTTACCAAGAACTAGCT 27133

27133 GATACCTACGCCAACGCCACATCCGACGTTTTGATTGCTAAACTAGACCACACTGAAAACGATGTCAGAGGCGTCGTAAT 27213

27213 TGAAGGTTACCCAACAATCGTCTTATACCCAGGTGGTAAGAAGTCCGAATCTGTTGTGTACCAAGGTTCAAGATCCTTGG 27293

27293 ACTCTTTATTCGACTTCATCAAGGAAAACGGTCACTTCGACGTCGACGGTAAGGCCTTGTACGAAGAAGCCCAGGAAAAA 27373

27373 GCTGCTGAGGAAGCCGAAGCTGACGCCGAAGCCGAAGCTGACGCTGACGCTGAATTGGCTGACGAAGAAGATGCCATTCA 27453

27453 CGATGAATTGTAATTCTGATCACTTTGGTTTTT 27533

Obrázok B.1: Skutočná štruktúra génovej oblasti

Info

Gene Name: fgenesh1_kg.27_#_20_#_NP_013018.1

Protein ID: 38762

Sequence

```
33402  TACACTAAGAATACATAGCAATGGCTTCCAAGAAAATTAAAGTTGACGAAGTGCCAAAATTAAGTGTTAAGGAGAAGGAG  33482
33482  ATTGAAGAGAAATCTTCTTCTTCTTCTTCCTCTTCTTCCTCTTCCTCCTCGTCCTCCTCCTCCTCTTCTTCTTCATC  33562
33562  TTCCTCGTCCAGTTCTAGCGGCGAATCTTCAAGCAGCAGCAGCAGCAGTGACAGCAGTGACAGCAGTGACTCTGAATCTA  33642
33642  GCTCATCATCTTCTTCTTCTAGCTCAAGCTCAAGTTCTAGTGACTCTGAATCTTCTAGTGAATCCGACTCTAGTTCCAGT  33722
33722  GGATCTTCTTCTTCCTCTAGCTCTAGTTCTGACGAATCTTCTTCGGAATCCGAATCAGAAGATGAAACCAAGAAAAGAGC  33802
33802  AAGAGAATCAGACAACGAAGATGCTAAGGAGACTAAAAAGGCTAAAACTGAACCAGAAAGCTCTTCCTCATCGGAGTCAA  33882
33882  GCTCTAGCGGGTCATCGTCATCTTCAGAATCAGAATCAGGTTCGGAATCTGATTCTGACTCAAGTTCTAGCTCAAGTAGT  33962
33962  TCTTCTGATTCCGAATCAGATTCTGAATCAGATTCTCAATCAAGTTCTAGCTCCAGCTCTAGCGACTCTTCGTCTGACTC  34042
34042  CGACTCCAGCTCCAGCGACTCTTCATCTGACTCCGACTCTAGCTCCAGCTCTAGCAGCTCTTCCTCGGATTCAGATTCGG  34122
34122  ATTCCGACTCAAGCTCGGACTCAGACTCTTCAGGCTCGTCCGATTCTTCAAGCTCGTCTGATTCAAGTTCCGACGAATCT  34202
34202  ACATCTTCTTCTGATTCCTCCGATTCGGATTCCGACTCAGACTCTGGTTCTAGCTCTGAATTGGAAACAAAAGAAGCAAC  34282
34282  TGCAGACGAATCCAAGGCCGAAGAAACGCCTGCCTCATCTAACGAATCAACTCCATCTGCTTCCAGTTCATCAAGCGCCA  34362
34362  ATAAATTGAACATTCCTGCTGGAACCGACGAGATCAAAGAGGGCCAAAGAAAGCATTTTTCCAGAGTAGATAGATCGAAG  34442
34442  ATTAATTTCGAAGCTTGGGAGTTGACAGACAACACCTACAAAGGCGCGGCTGGAACATGGGGTGAAAAGGCAAATGAAAA  34522
34522  ATTAGGTAGAGTAAGAGGTAAAGATTTCACGAAGAACAAAAATAAAATGAAGAGAGGTTCTTACAGAGGTGGATCCATTA  34602
34602  CATTAGAAAGTGGTTCATACAAATTCCAAGATTAAAAGAATAGGAAACATAAAAC                            34682
```

Obrázok B.3: Skutočná štruktúra génovej oblasti

B.2 Posun intrónu

scaffold_27+

Info

Gene Name: fgenesh1_kg.27_#_20_#_NP_013018.1

Protein ID: 38762

Sequence

```
33402  TACACTAAGAATACATAGCAATGGCTTCCAAGAAAATTAAAGTTGACGAAGTGCCAAAATTAAGTGTTAAGGAGAAGGAG  33482
33482  ATTGAAGAGAAATCTTCTTCTTCTTCTTCCTCTTCTTCCTCTTCCTCCTCGTCCTCCTCCTCCTCTTCTTCTTCATC  33562
33562  TTCCTCGTCCAGTTCTAGCGGCGAATCTTCAAGCAGCAGCAGCAGCAGTGACAGCAGTGACAGCAGTGACTCTGAATCTA  33642
33642  GCTCATCATCTTCTTCTTCTAGCTCAAGCTCAAGTTCTAGTGACTCTGAATCTTCTAGTGAATCCGACTCTAGTTCCAGT  33722
33722  GGATCTTCTTCTTCCTCTAGCTCTAGTTCTGACGAATCTTCTTCGGAATCCGAATCAGAAGATGAAACCAAGAAAAGAGC  33802
33802  AAGAGAATCAGACAACGAAGATGCTAAGGAGACTAAAAAGGCTAAAACTGAACCAGAAAGCTCTTCCTCATCGGAGTCAA  33882
33882  GCTCTAGCGGGTCATCGTCATCTTCAGAATCAGAATCAGGTTCGGAATCTGATTCTGACTCAAGTTCTAGCTCAAGTAGT  33962
33962  TCTTCTGATTCCGAATCAGATTCTGAATCAGATTCTCAATCAAGTTCTAGCTCCAGCTCTAGCGACTCTTCGTCTGACTC  34042
34042  CGACTCCAGCTCCAGCGACTCTTCATCTGACTCCGACTCTAGCTCCAGCTCTAGCAGCTCTTCCTCGGATTCAGATTCGG  34122
34122  ATTCCGACTCAAGCTCGGACTCAGACTCTTCAGGCTCGTCCGATTCTTCAAGCTCGTCTGATTCAAGTTCCGACGAATCT  34202
34202  ACATCTTCTTCTGATTCCTCCGATTCGGATTCCGACTCAGACTCTGGTTCTAGCTCTGAATTGGAAACAAAAGAAGCAAC  34282
34282  TGCAGACGAATCCAAGGCCGAAGAAACGCCTGCCTCATCTAACGAATCAACTCCATCTGCTTCCAGTTCATCAAGCGCCA  34362
34362  ATAAATTGAACATTCCTGCTGGAACCGACGAGATCAAAGAGGGCCAAAGAAAGCATTTTTCCAGAGTAGATAGATCGAAG  34442
34442  ATTAATTTCGAAGCTTGGGAGTTGACAGACAACACCTACAAAGGCGCGGCTGGAACATGGGGTGAAAAGGCAAATGAAAA  34522
34522  ATTAGGTAGAGTAAGAGGTAAAGATTTCACGAAGAACAAAAATAAAATGAAGAGAGGTTCTTACAGAGGTGGATCCATTA  34602
34602  CATTAGAAAGTGGTTCATACAAATTCCAAGATTAAAAGAATAGGAAACATAAAAC                            34682
```

Legend:

AACGT - Non coding

AACGT - Exon

AACGT - Intron

Obrázok B.4: Predikované štruktúra génovej oblasti

B.3 Readthroug jav

scaffold_88+

Info

Gene Name: estExt_fgenesh1_pg.C_880010

Protein ID: 772493

Sequence

18900	CAAAAACAAACACCAAAAGAATGAAGGCCTTCATCATCAAGCAATACGGTGCTCCCCACGAAGCCCTCTCCCTAGTCGAC	18980
18980	CTCCCTCTACCAACACCCAAAGAAAACGAAGTAACCGTCCGTGTGCACGCGGTCAGCCTCAACGCCCTTGACTGGCACCT	19060
19060	CACCCGCGGCGATCCCTACATCGTGCGCTTCCTCACCGGCTTCACCCGCCCCAAACGCGACAACTGGGTCGCAGGCTCAG	19140
19140	GGTTCGCAGGAACCGTCGAATCCATCGGCGCTTCCGTGACAACATTCAAAGTCGGTGATGAAGTCATGGCCGATCCGGGC	19220
19220	GTCGATTTGGGCGGGCTGGCTGAAGTCGCCGTTGTTTCGCACAAGGATCTTTGCTTGAAGCCGGCGAATGTTCGTTTGCG	19300
19300	GAGGCGGCGGCGGTGGGTGTGTCAGCCACGACGGCGTTCCAAGCGCTGCATGATGTGGCGAAGGTACAAGCGGGGCAGAA	19380
19380	GGTGTTGGTGAATGGAGCGTCTGGGGTGTTGGTACTGCTGCGATTCAGATTGCTAAGGCGGCAGGGGCTCAGGTTACGGC	19460
19460	TGTGTGCAGTGCAAGGAATATTGATCTCGTGAAGTCTCTTGGCGCTGACTATGCGATTGACTATGCTCAGGAAAACTTTA	19540
19540	CTCAGCTTGGAACGTGTATGATATTATTATTGATAATGTGGGCACACAATCTGAGACGGATTGTTGTAAAGTGTTGTCAC	19620
19620	CAGAGGGTGTTATTGTCCAAGTCGGATCTCTGTCACCAAAAGGTACTTTGACAAATGGACTTCTGAGTGGTGCTGTAGCT	19700
19700	GGTTTATTCACCCAGAAGCAGAAGGGGCAACGAATTGAGGGAATCATGGCTAAGGCTGGAGGAAAATGTTTGGAACAAAT	19780
19780	CAAGCCACTATTGGAGCAAGGTGCTTTAAAACCTCCAATCTTCAAGACATTTACCTTTGAGCAAGCTCCGGATGCATTAG	19860
19860	TCTTGCAAGAAGAAGGTCACGTTGCTGGGAAGATTGTTATCACTGTGCTTTGAATGCAGGGAGGGAAACTACA	19940

Legend:

AACGT - Non coding

AACGT - Exon

AACGT - Intron

Obrázok B.5: Skutočná štruktúra génovej oblasti

scaffold_88+

Info

Gene Name: estExt_fgenesh1_pg.C_880010

Protein ID: 772493

Sequence

18900	CAAAAACAAACACCAAAAGAATGAAGGCCTTCATCATCAAGCAATACGGTGCTCCCCACGAAGCCCTCTCCCTAGTCGAC	18980
18980	CTCCCTCTACCAACACCCAAAGAAAACGAAGTAACCGTCCGTGTGCACGCGGTCAGCCTCAACGCCCTTGACTGGCACCT	19060
19060	CACCCGCGGCGATCCCTACATCGTGCGCTTCCTCACCGGCTTCACCCGCCCCAAACGCGACAACTGGGTCGCAGGCTCAG	19140
19140	GGTTCGCAGGAACCGTCGAATCCATCGGCGCTTCCGTGACAACATTCAAAGTCGGTGATGAAGTCATGGCCGATCCGGGC	19220
19220	GTCGATTTGGGCGGGCTGGCTGAAGTCGCCGTTGTTTCGCACAAGGATCTTTGCTTGAAGCCGGCGAATGTTCGTTTGCG	19300
19300	GAGGCGGCGGCGGTGGGTGTGTCAGCCACGACGGCGTTCCAAGCGCTGCATGATGTGGCGAAGGTACAAGCGGGGCAGAA	19380
19380	GGTGTTGGTGAATGGAGCGTCTGGGGTGTTGGTACTGCTGCGATTCAGATTGCTAAGGCGGCAGGGGCTCAGGTTACGGC	19460
19460	TGTGTGCAGTGCAAGGAATATTGATCTCGTGAAGTCTCTTGGCGCTGACTATGCGATTGACTATGCTCAGGAAAACTTTA	19540
19540	CTCAGCTTGGAACGTGTATGATATTATTATTGATAATGTGG	19620

Obrázok B.6: Predikované štruktúra génovej oblasti

B.4 Nenájdenie stop kodónu

scaffold_88+

Info

Gene Name: estExt_fgenesh1_pm.C_880006

Protein ID: 759481

Model name: Rhizoclosmatium globosum

Logodd: 259.0715137741934

Sequence

63487 TTCCGAAACCAAACCTTAAAATGTCCGCCTCCCCAGCCATTCTTGCCCCAACCGAGCAGGACATCACGTCCCTCCTCGCC 63567
63567 GCCCAGTCCCACATCGGAACAAAGAACTTGAACGTGCACATGCAGCCATACGTGTGGAAGCGCCGTGCCGACGGTGTGCA 63647
63647 CATCATCAACATCGGCAAGACCTACGAGAAGATGGTCCTTGCCGCCGTATCATTGCCGCCGTTGAGAACCCTGCCGACAT 63727
63727 CTGTGTCGTCTCTGCCCGTCCATACGGTCAACGTGCTGCCCTCAAGTTTGCCAACTACACCGGTGCCCAGGCTATTGCCG 63807
63807 GCCGCTTCACCCCCGGTACCTTCACCAACTACATCACCCGTACCTTCAGAGAGCCCCGCTTGATCATCGTCACTGACCCA 63887
63887 CGTACCGACCACCAGGCCATCAAGGAGGCCTCCTACGTCAACATCCCTGTCATTGCCTTTGCTGACTGTGACGCCCCACT 63967
63967 CAAGTTTGTTGACTGTGTTATCCCAACCAACAACAAGGGTAAGCACGCTATTGGTCTTGCCTACTGGCTCCTTGCCCGTG 64047
64047 AGGTTCTCCGTCTCCGTGGAACCATCTCCCGCTCGAGCCATGGTCTGTTATGACCGATATGTTCTTCTACCGTGACCCAG 64127
64127 AGGAGGCTGAGAAGGAGGCTGAGGCTCTTGCTGCCGCTGCCGCCGCTGCTGCTGCTCCAGTTGCTGAGGAGGAGGCTGCC 64207
64207 AACCCTGAGTGGGAGGTCTCTGCCTCTGGCGCTGCTGGCCTTGCCGGAACTGTTGCTGAGGGAGAGTGGGGTGCCAACGC 64287
64287 CAACGTTGAGTGGGGAACTGACGCTTAAGTCAATTGTTTGATGCTTTA 64367

Obrázok B.7: Skutočná štruktúra génovej oblasti

scaffold_88+

Info

Gene Name: estExt_fgenesh1_pm.C_880006

Protein ID: 759481

Model name: Rhizoclosmatium globosum

Logodd: 259.0715137741934

Sequence

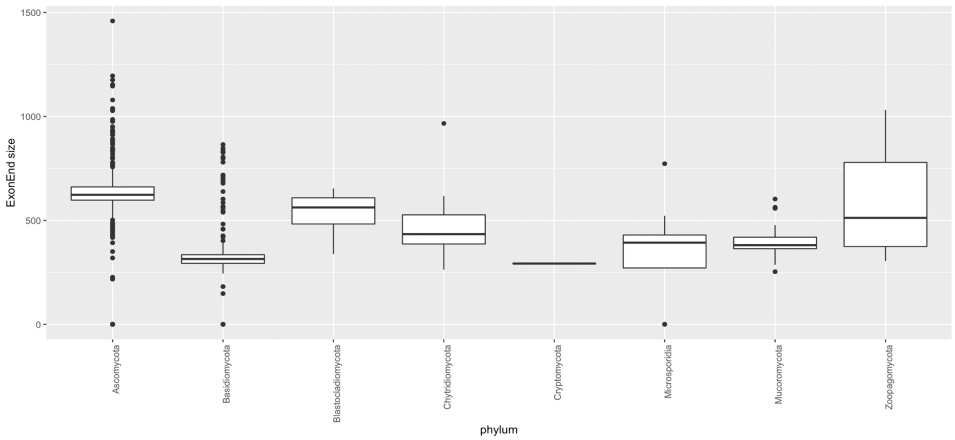
63487 TTCCGAAACCAAACCTTAAAATGTCCGCCTCCCCAGCCATTCTTGCCCCAACCGAGCAGGACATCACGTCCCTCCTCGCC 63567
63567 GCCCAGTCCCACATCGGAACAAAGAACTTGAACGTGCACATGCAGCCATACGTGTGGAAGCGCCGTGCCGACGGTGTGCA 63647
63647 CATCATCAACATCGGCAAGACCTACGAGAAGATGGTCCTTGCCGCCGTATCATTGCCGCCGTTGAGAACCCTGCCGACAT 63727
63727 CTGTGTCGTCTCTGCCCGTCCATACGGTCAACGTGCTGCCCTCAAGTTTGCCAACTACACCGGTGCCCAGGCTATTGCCG 63807
63807 GCCGCTTCACCCCCGGTACCTTCACCAACTACATCACCCGTACCTTCAGAGAGCCCCGCTTGATCATCGTCACTGACCCA 63887
63887 CGTACCGACCACCAGGCCATCAAGGAGGCCTCCTACGTCAACATCCCTGTCATTGCCTTTGCTGACTGTGACGCCCCACT 63967
63967 CAAGTTTGTTGACTGTGTTATCCCAACCAACAACAAGGGTAAGCACGCTATTGGTCTTGCCTACTGGCTCCTTGCCCGTG 64047
64047 AGGTTCTCCGTCTCCGTGGAACCATCTCCCGCTCGAGCCATGGTCTGTTATGACCGATATGTTCTTCTACCGTGACCCAG 64127
64127 AGGAGGCTGAGAAGGAGGCTGAGGCTCTTGCTGCCGCTGCCGCCGCTGCTGCTGCTCCAGTTGCTGAGGAGGAGGCTGCC 64207
64207 AACCCTGAGTGGGAGGTCTCTGCCTCTGGCGCTGCTGGCCTTGCCGGAACTGTTGCTGAGGGAGAGTGGGGTGCCAACGC 64287
64287 CAACGTTGAGTGGGGAACTGACGCTTAAGTCAATTGTTTGATGCTTTAAAAGTCATGCAATCGATTTTTGTTTGTTGGAT 64367
64367 TCTTGTTCTCAATTCTCTCTCTTATTTGTGCGTTTCTTTCTTTGTTCCTTTGCTGGATCGAGTTGGACCAAATAAATCAG 64447
64447 ACTTCAATGTTATTTATGATTTGACAGTATGACAATTCTTGTTTGGATCAAGTTGACTCACGCTTGGTTTGCTATTGCCG 64527
64527 GCCGCTTCACCCCCGGTACCTTCACCAACTACATCACCCGTACCTTCAGAGAGCCCCGCTTGATCATCGTCACTGACCCA 64607
64607 CGTACCGACCACCAGGCCATCAAGGAGGCCTCCTACGTCAACATCCCTGTCATTGCCTTTGCTGACTGTGACGCCCCACT 64687
64687 CAAGTTTGTTGACTGTGTTATCCCAACCAACAACAAGGGTAAGCACGCTATTGGTCTTGCCTACTGGCTCCTTGCCCGTG 64767
64767 AGGTTCTCCGTCTCCGTGGAACCATCTCCCGCTCGAGCCATGGTCTGTTATGACCGATATGTTCTTCTACCGTGACCCAG 64847
64847 AGGAGGCTGAGAAGGAGGCTGAGGCTCTTGCTGCCGCTGCCGCCGCTGCTGCTGCTCCAGTTGCTGAGGAGGAGGCTGCC 64927
64927 AACCCTGAGTGGGAGGTCTCTGCCTCTGGCGCTGCTGGCCTTGCCGGAACTGTTGCTGAGGGAGAGTGGGGTGCCAACGC 65007
65007 CAACGTTGAGTGGGGAACTGACGCTTAAGTCAATTGTTTGATGCTTTAAAAGTCA 65087

Obrázok B.8: Predikované štruktúra génovej oblasti

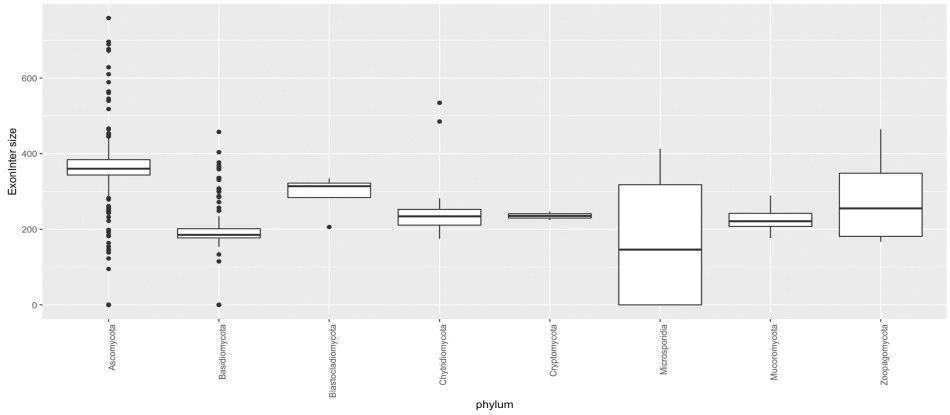
Dodatok C

Štatistiky genómov húb

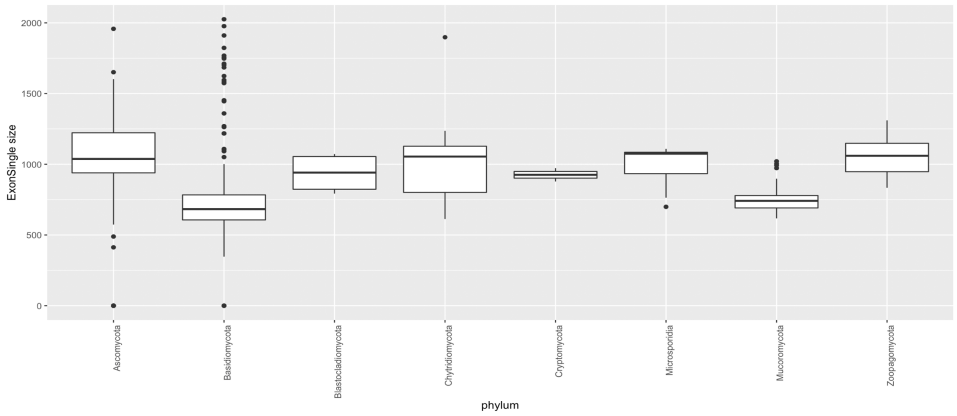
C.1 Box ploty



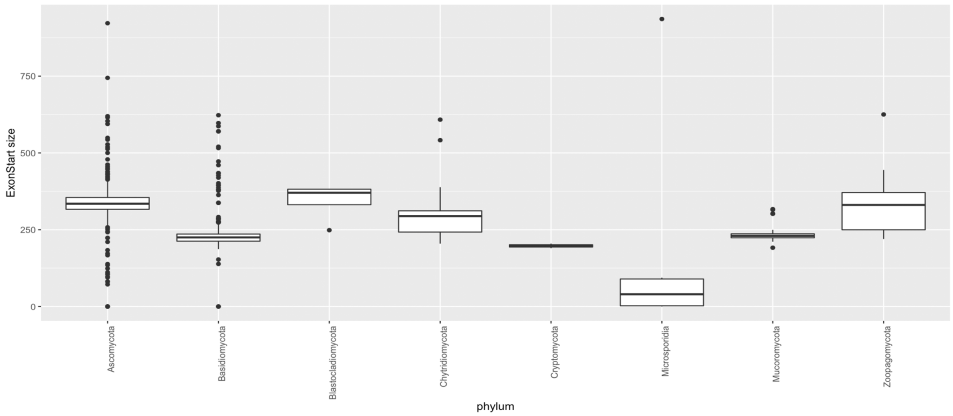
Obrázok C.1: Rozdelenie dĺžok sekvencií oblastí *exonEnd* pre dané phylum



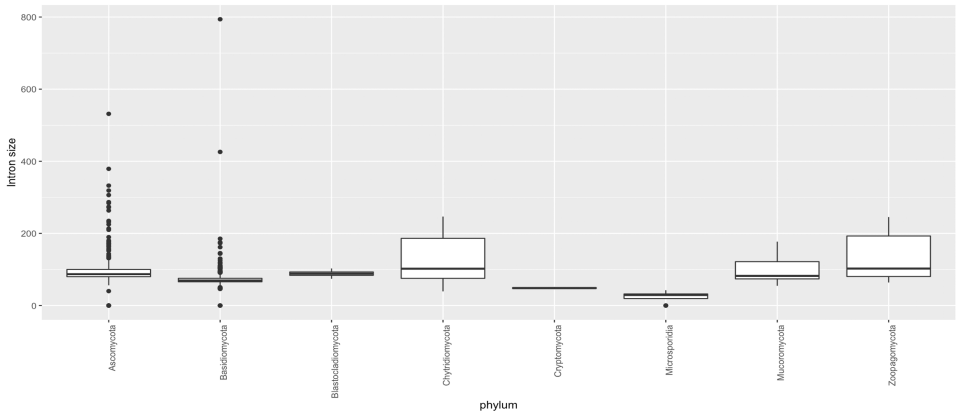
Obrázok C.2: Rozdelenie dĺžok sekvencií oblastí *exonInter* pre dané phylum



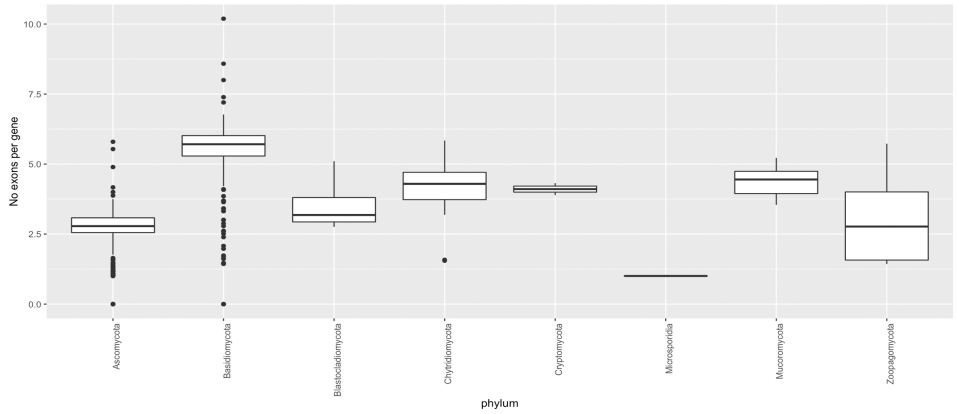
Obrázok C.3: Rozdelenie dĺžok sekvencií oblastí *exonSingle* pre dané phylum



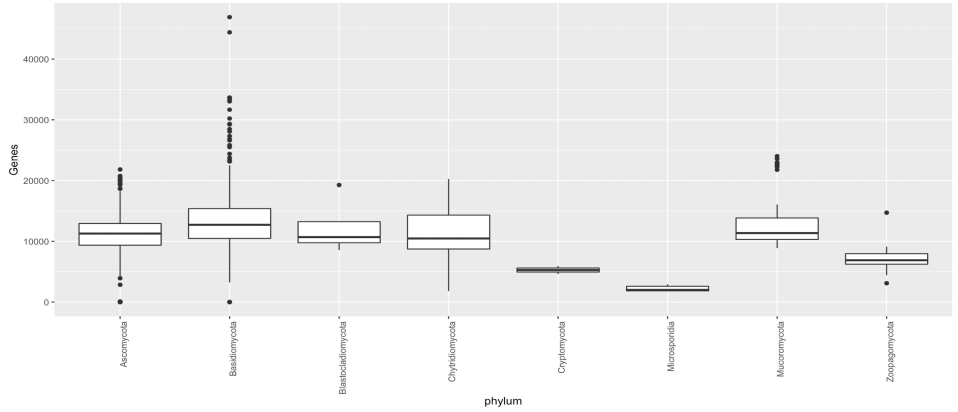
Obrázok C.4: Rozdelenie dĺžok sekvencií oblastí *exonStart* pre dané phylum



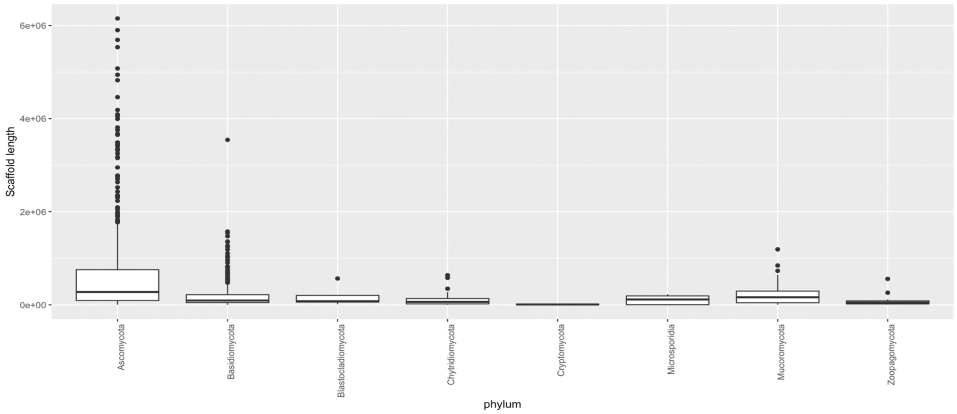
Obrázok C.5: Rozdelenie dĺžok sekvencií oblastí *intron* pre dané phylum



Obrázok C.6: Rozdelenie počtu exónov na gén pre dané phylum

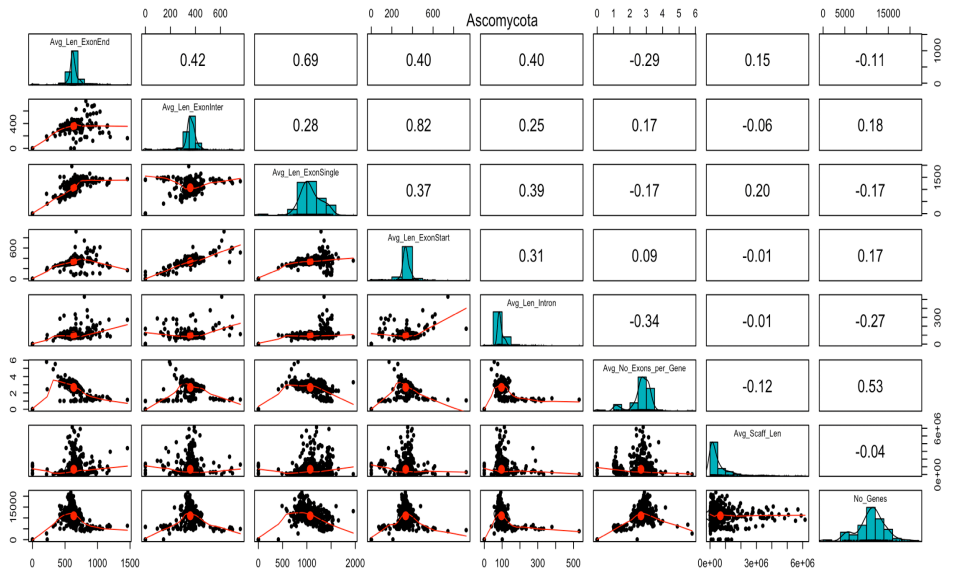


Obrázok C.7: Rozdelenie počtu génov pre dané phylum

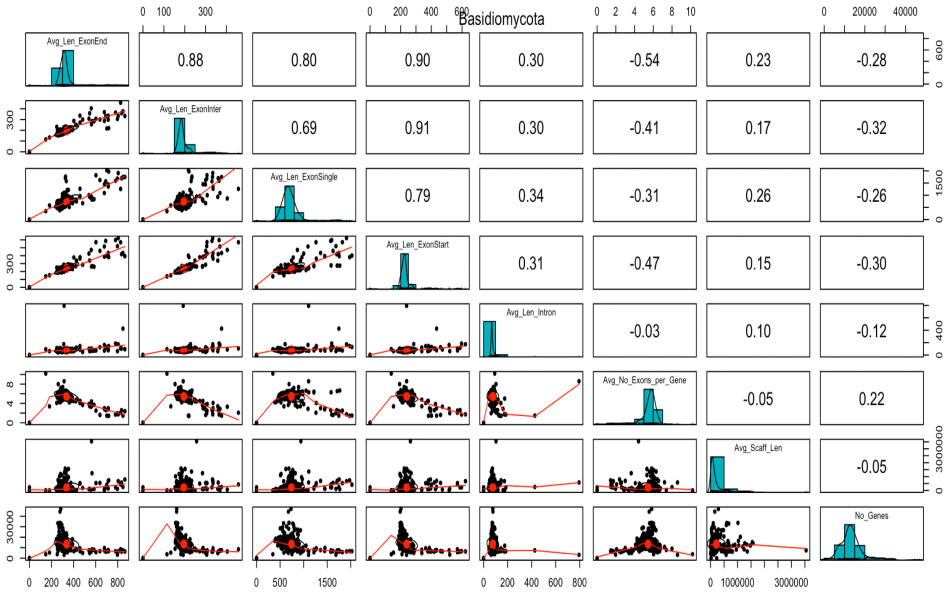


Obrázok C.8: Rozdelenie dĺžok scaffoldov pre dané phylum

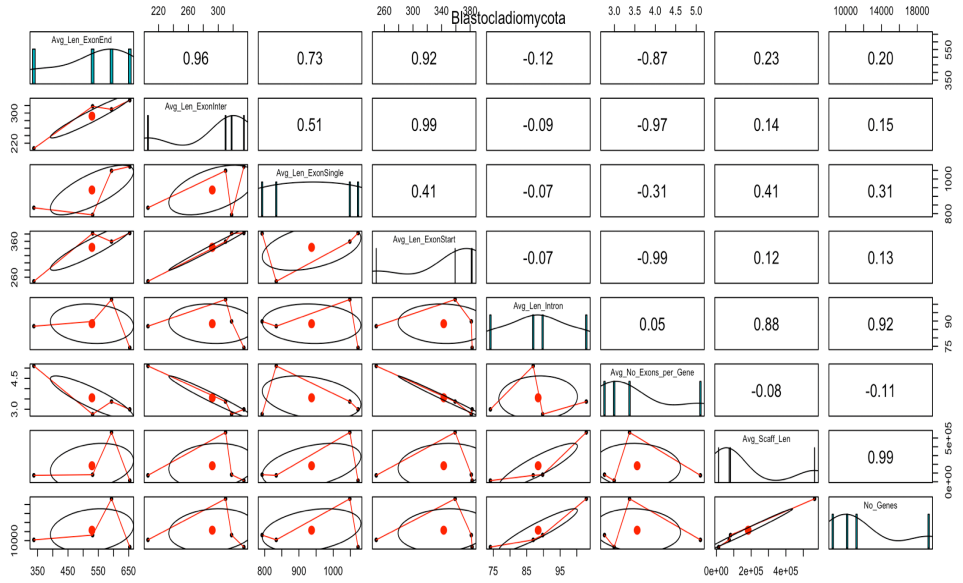
C.2 Korelácie sekvencií vstupných stavov genómového modelu



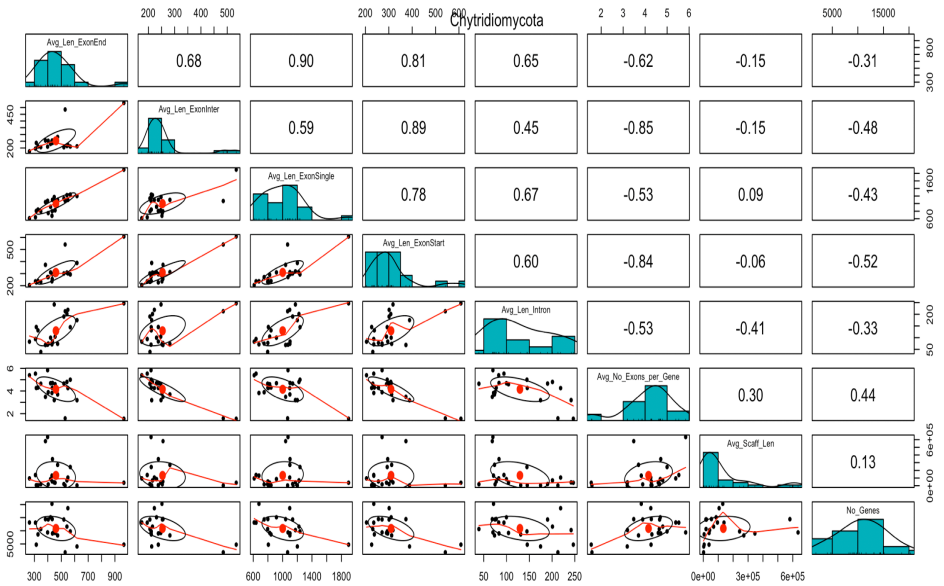
Obrázok C.9: Korelácie dĺžok jednotlivých oblastí genómu pre phylum Ascomycota



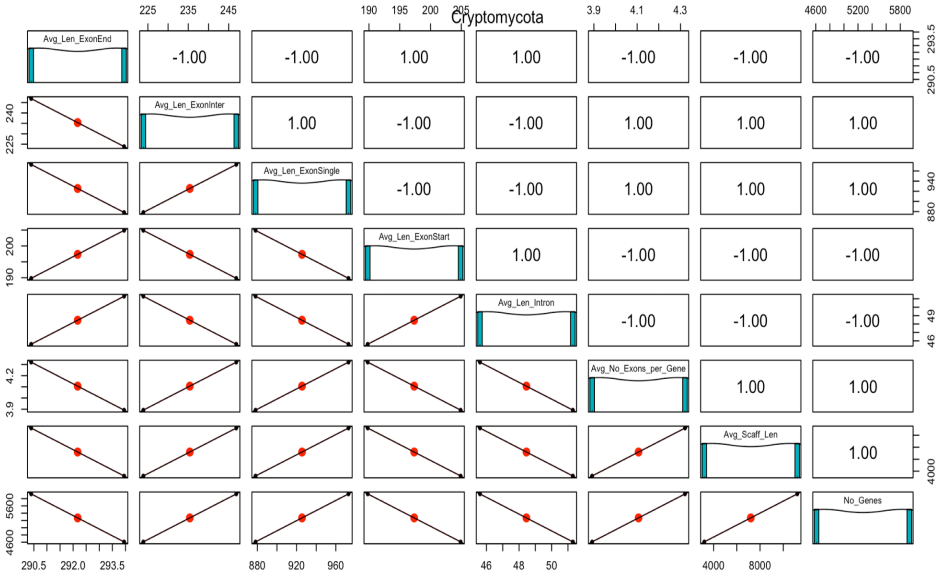
Obrázok C.10: Korelácie dĺžok jednotlivých oblastí genómu pre phylum Basidiomycota



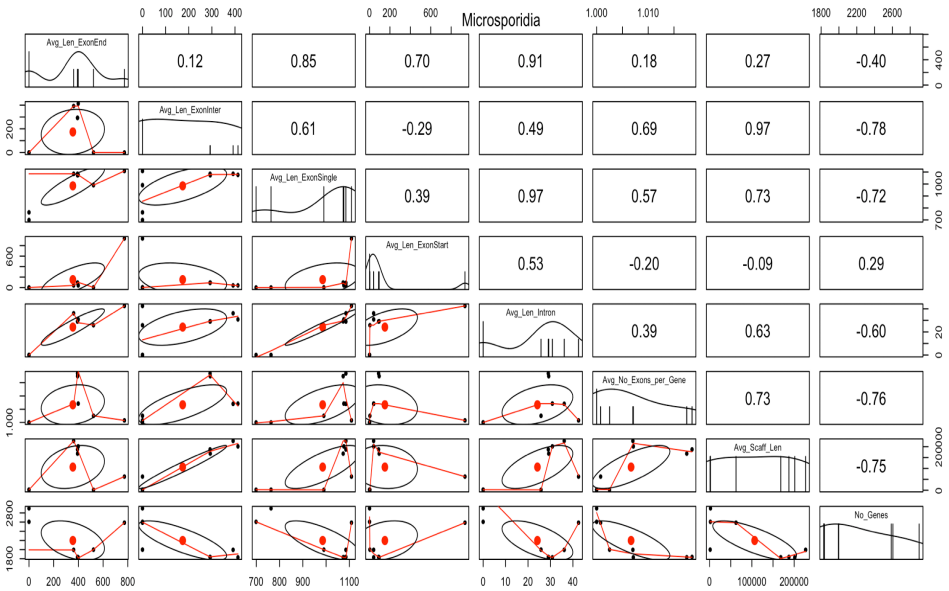
Obrázok C.11: Korelácie dĺžok jednotlivých oblastí genómu pre phylum Blastocladiomycota



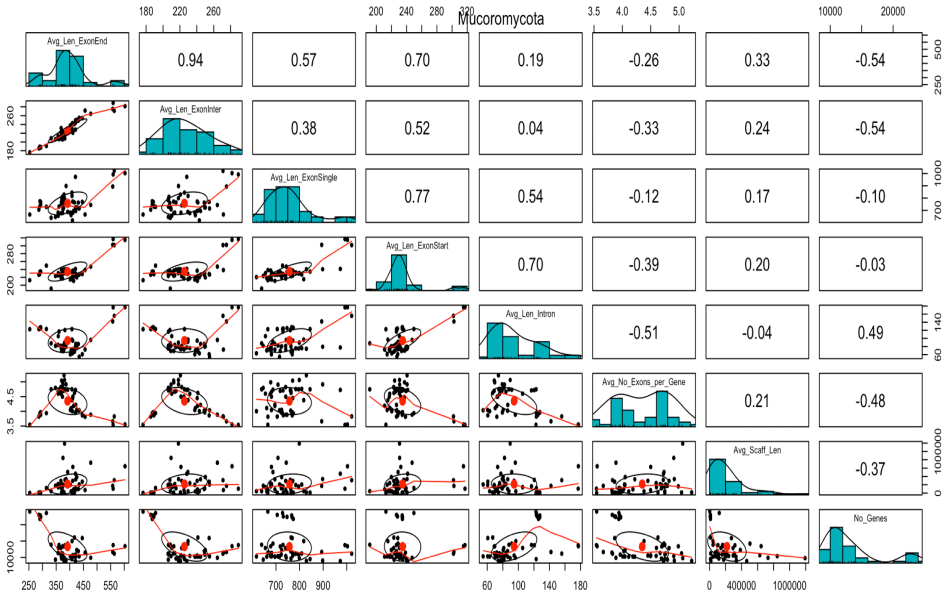
Obrázok C.12: Korelácie dĺžok jednotlivých oblastí genómu pre phylum Chytridiomycota



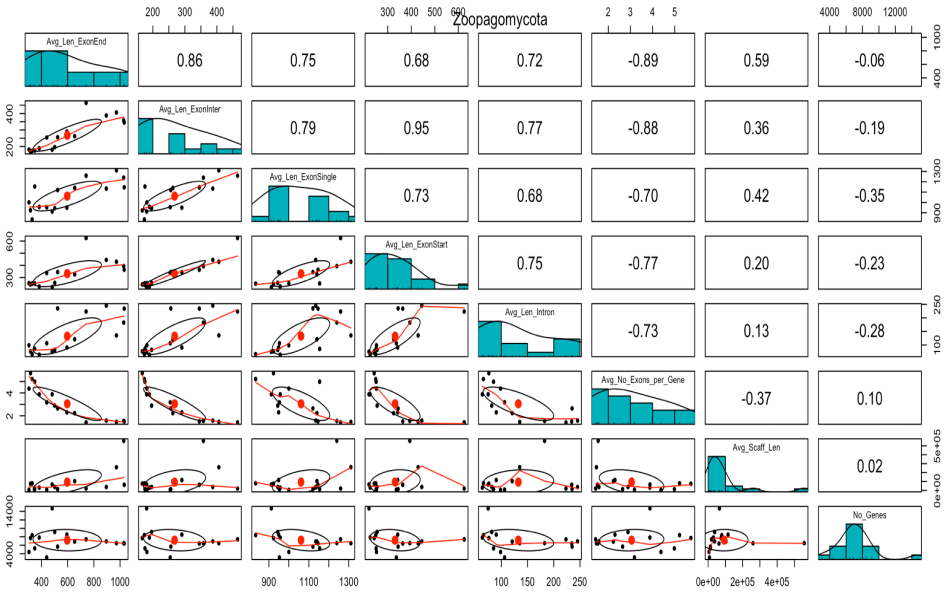
Obrázok C.13: Korelácie dĺžok jednotlivých oblastí genómu pre phylum Cryptomycota



Obrázok C.14: Korelácie dĺžok jednotlivých oblastí genómu pre phylum Microsporidia



Obrázok C.15: Korelácie dĺžok jednotlivých oblastí genómu pre phylum Mucoromycota



Obrázok C.16: Korelácie dĺžok jednotlivých oblastí genómu pre phylum Zoopagomycota

Dodatok D

Implementačné detaily

D.1 Ukážka konfiguračného súboru

```
{
    "ascomycota" : {
        "genome" : "./genomeConfig.txt",
        "transcript" : "./transcriptConfig.txt"
    }
}
```

Obrázok D.1: Konfiguračný súbor pre Fungi model. Špecifikuje cesty ku genómovému a transkriptovému konfiguračnému súboru.

```
{
    "name" : "ascomyConf",
     "utrConf" : {
        "min" : 0,
        "max" : 300
     },
     "intronConf : {
        "min" : 10,
        "max" : 980
     },
     "exonSingleConf : {
        "min" : 10,
        "max" : 2800
     },
     "exonInterConf" : {
        "min" : 10,
        "max" : 980
     },
     "exonStartConf" : {
        "min" : 10,
        "max" : 1222
     },
     "exonEndConf" : {
        "min" : 10,
        "max" : 1950
     },
     "spliceSiteConf" : {
        "exonSide" : 15,
        "intronSide" : 3
     },
     "promoterConf" : {
        "leftLen" : 11,
        "rightLen" : 9
     }
     "stopCodonConf" : {
        "leftLen" : 18,
        "rightLen" : 0
     }
}
```

(a) : Konfiguračný súbor pre genóm model

```
{
    "name" : "ascomyConf",
     "utrConf" : {
        "min" : 0,
        "max" : 300
     },
     "cdsConf : {
        "min" : 10,
        "max" : 980
     },
     "ncdsConf : {
        "min" : 10,
        "max" : 2800
     },
     "promoterConf" : {
        "leftLen" : 11,
        "rightLen" : 9
     }
     "stopCodonConf" : {
        "leftLen" : 18,
        "rightLen" : 0
     }
}
```

(b) : Konfiguračný súbor pre transkript model

Dodatok E

Obsah priloženého CD

K práci je priložené CD, ktoré obsahuje nasledujúce zložky:

- CODE
  - obsahuje všetky Python moduly implementovaného nástroja popísané v kapitole 6
- DOCUMENTATION
  - obsahuje text samotnej práce v podobe tohto pdf
  - obsahuje zdrojové kódy použité pri tvorbe dokumentácie v jazyku LaTeX