VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2016 Markéta Nykrýnová

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ
DEPARTMENT OF BIOMEDICAL ENGINEERING

METODY PREDIKCE GENŮ V PROKARYOTICKÝCH GENOMECH
METHODS FOR GENE PREDICTION IN PROKARYOTIC GENOMES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Markéta Nykrýnová**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Denisa Maděránková**
SUPERVISOR

BRNO 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Biomedicínská technika a bioinformatika**
Ústav biomedicínského inženýrství

Studentka: Markéta Nykrýnová ***ID:*** 164215
Ročník: 3 ***Akademický rok:*** 2015/16

NÁZEV TÉMATU:

Metody predikce genů v prokaryotických genomech

POKYNY PRO VYPRACOVÁNÍ:

1) Vypracujte literární rešerši na téma metody vyhledávání genů v prokaryotických genomech včetně popisu alespoň tří volně dostupných softwarů. 2) Vybrané softwary otestujte na zvoleném prokaryotickém genomu a výsledky vyhodnoťte. 3) Vybranou metodu vyhledávání genů implementujte v libovolném programovém prostředí. 4) Implementovanou metodu otestujte na zvoleném prokaryotickém genomu a výsledky porovnejte s výsledky volně dostupných softwarů.

DOPORUČENÁ LITERATURA:

[1] HYATT, D., et al. Prodigal: prokaryotic gene recognition and translation initiation site identification. BMC Bioinformatics. 2010, 11(1):119.

[2] PAREJA-TOBES, P., et al. BG7: A New Approach for Bacterial Genome Annotation Designed for Next Generation Sequencing Data. PLoS ONE. 2012, 7(11):e49239.

Termín zadání: 8.2.2016 ***Termín odevzdání:*** 27.5.2016

Vedoucí práce: Ing. Denisa Maděránková
Konzultant bakalářské práce:

prof. Ing. Ivo Provazník, Ph.D., *předseda oborové rady*

ABSTRAKT

Tato bakalářská práce se zabývá metodami predikce genů v prokaryotických organismech. V první části je popsána prokaryotická buňka včetně genomu, exprese genetické informace a rozdělení metod pro predikci genů. Dále je zde uveden popis tří vybraných softwarů, které predikci provádějí. V praktické části je rozebráno testování softwarů a vyhodnocení jejich účinnosti na daném genomu. Nakonec je zde popsán vytvořený program pro hledání genů a jsou zde uvedeny výsledky jeho účinnosti.

KLÍČOVÁ SLOVA

predikce genů, prokaryota, gen

ABSTRACT

This bachelor thesis deals with methods of gene prediction in prokaryotic genomes. First part of the thesis introduces prokaryotic cell, its genome, expression of genetic information and methods for gene prediction. Following part describes three software products for gene prediction. Chosen software was tested against specific genome and next chapter presents obtained results. The last part describes program called Gene_finder and its results.

KEYWORDS

gene prediction, prokaryote, gene

NYKRÝNOVÁ, Markéta *Metody predikce genů v prokaryotických genomech*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství, 2016. 43 s. Vedoucí práce byla Ing. Denisa Maděránková

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Metody predikce genů v prokaryotických genomech“ jsem vypracovala samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autorka uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědoma následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora(-ky)

PODĚKOVÁNÍ

Ráda bych poděkovala vedoucí bakalářské práce paní Ing. Denise Maděránkové za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno ....

podpis autora(-ky)

OBSAH

Úvod **10**

1 Teoretická část **11**
1.1 Prokaryotická buňka 11
1.2 Prokaryotický genom 11
1.3 Exprese genetické informace 12
1.3.1 Transkripce 12
1.3.2 Translace 12
1.4 Metody predikce genů 13
1.4.1 Vyhledávání genů ab initio 13
1.4.2 Metody založené na homologii 14
1.5 Programy pro predikci genů 14
1.5.1 GeneMark.hmm 14
1.5.2 Glimmer 16
1.5.3 Prodigal 21

2 Praktická část **24**
2.1 Testování softwarů 24
2.1.1 GeneMark.hmm 24
2.1.2 Glimmer 25
2.1.3 Prodigal 25
2.1.4 Shrnutí výsledků 26
2.2 Program Gene_finder 27
2.2.1 Načtení sekvence 28
2.2.2 Hledání stop kodonů 28
2.2.3 Hledání start kodonů 28
2.2.4 Frekvence kodonů 29
2.2.5 Skórování potencionálních genů 30
2.2.6 Výběr finálních genů 30
2.2.7 Uložení finálních genů 30
2.3 Testování program Gene_finder 31

3 Závěr **35**

Literatura **37**

Seznam symbolů, veličin a zkratek **39**

A Vývojový diagram programu **40**

B Výsledky testování programu **41**

C Obsah přiloženého CD **43**

SEZNAM OBRÁZKŮ

1.1 Prokaryotická buňka, z [3]. 11
1.2 Exprese genetické informace u prokaryot, z [7]. 13
1.3 Skrytý Markovův model prokaryotické nukleotidové sekvence používaný v GeneMark.hmm, z [9]. 15
1.4 Případ překryvu č. 1, z [12]. 19
1.5 Případ překryvu č. 2, z [12]. 19
1.6 Případ překryvu č. 3, z [12]. 20
1.7 Případ překryvu č. 4, z [12]. 20
2.1 Výstupní obrazovka programu GeneMark.hmm. 24
2.2 Výstupní obrazovka programu Glimmer. 25
2.3 Výstupní obrazovka programu Prodigal. 26
2.4 Výstup programu v Excelu. 28
2.5 Úspěšnost predikce pro ±3 báze. 32
2.6 Úspěšnost predikce pro ±30 bází. 32
2.7 Úspěšnost predikce pro ±60 bází. 33
2.8 Úspěšnost predikce pro ±90 bází. 33

SEZNAM TABULEK

2.1 Výsledky predikce u analyzovaných softwarů. 27
2.2 Úspěšnost analyzovaných softwarů. 27
2.3 Přehled testovaných genomů. 31
2.4 Srovnání analyzovaných a vytvořeného softwaru. 34
B.1 Výsledky predikce programu pro jednotlivé genomy. 41
B.2 Úspěšnost programu pro jednotlivé genomy. 42

ÚVOD

Tato bakalářská práce se zabývá metodami pro predikci genů. Pokud máme DNA sekvenci, chceme u ní určit oblasti, které jsou zodpovědné za vznik proteinů, tedy geny. K tomuto účelu slouží programy pro jejich vyhledávání.

Jelikož v dnešní době neustálé narůstá počet osekvenovaných genomů, je potřeba mít vhodné nástroje pro jejich zpracování. Z tohoto důvodu dochází k tvorbě nových programů a ke zlepšení programů již vytvořených, které jsou schopny v genomu najít jednotlivé kódující úseky a odlišit je od úseků nekódujících. Po správné identifikaci a následné anotaci můžeme určit, k čemu jednotlivý gen slouží, což nám pomáhá v pochopení funkce celého organismu.

Predikce u prokaryotických organismů je o něco snazší než u organismů eukaryotických, jelikož prokaryota mají menší genomy, jednodušší genovou strukturu, neobsahují introny a navíc je u nich osekvenovaný velký počet genomů.

Cílem teoretické části této práce je seznámit čtenáře s prokaryotickými organismy, jejich genomem a expresí genové informace a poté ho poučit o metodách predikce a vybraných softwarech, kterými jsou GeneMark.hmm, Glimmer a Prodigal. První dva zmíněné programy používají pro predikci metody založené na Markovových modelech a třetí uvedený algoritmus využívá dynamické programování.

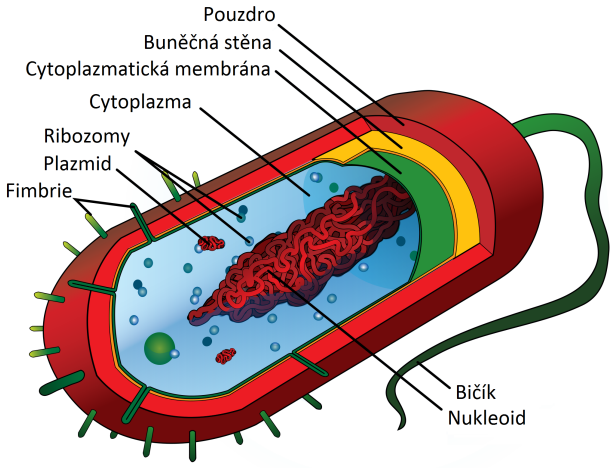
V prvním oddíle praktické části je popsáno testování výše uvedených programů. Je zde uvedeno, jak se ovládají a jaké mají výstupy. Následně jsou programy vyhodnoceny z hlediska jejich přesnosti při hledání genů v genomu *Escherichia coli*.

Ve druhém oddíle praktické části je představen vytvořený program Gene_finder, který byl realizován v prostředí MATLAB R2015a. Jsou zde uvedeny jednotlivé funkce, z kterých se algoritmus skládá, a u každé je stručně vysvětlen její princip. Poté je program otestován na pěti vybraných genomech a vyhodnocen. Nakonec je provedeno srovnání s volně dostupnými softwary.

1 TEORETICKÁ ČÁST

1.1 Prokaryotická buňka

Prokaryotické buňky jsou evolučně starší, vznikly přibližně před 3-3,5 miliardami let a později se z nich vyvinuly buňky eukaryotické. Jejich velikost se pohybuje v desítkách mikrometrů a mají rozmanitý tvar. Není u nich přítomno jádro ani jadérko, genetická informace je nesena pouze jedním chromosomem, zvaným nukleoid, který obsahuje 10^5-10^6 bp a od zbytku buňky ho neodděluje žádná membrána. Jelikož tedy neexistuje jaderný obal, nenalezneme u nich ani endomembránový systém. Organely v buňkách nacházíme ojediněle, v cytoplazmě můžeme nalézt ribozomy, které jsou menší než u eukaryot. Buňka je chráněná buněčnou stěnou, která je tvořena peptidoglykanem. Mezi prokaryotní organismy řadíme bakterie a Archea. [1], [2]



Obr. 1.1: Prokaryotická buňka, z [3].

1.2 Prokaryotický genom

Pojmem genom označujeme soubor všech molekul DNA v buňce. Genom u prokaryot je tvořen chromozomem, který se skládá z jednoho dvouřetězce DNA kruhovitého tvaru a v jednom místě je připojen na plazmatickou membránu buňky. Geny na chromozomu jsou uspořádány blízko vedle sebe s minimálním mezigenovým prostorem a nenalezneme zde introny.

Prokaryota mohou kromě jaderného chromozomu obsahovat i plazmidy, což jsou malé kruhové nebo lineární molekuly DNA v cytoplazmě. Plazmidy sice nejsou esenciální pro život buňky, ale často obsahují geny, které jsou pro buňku velmi výhodné,

například zprostředkovávají resistenci vůči antibiotikům či produkci toxinů. [4], [5]

1.3 Exprese genetické informace

Exprese genetické informace se skládá ze dvou částí: transkripce a translace. Nejprve dochází k přepisu nukleotidových sekvencí DNA podle templátového vlákna do sekvencí RNA a poté k překladu do posloupnosti aminokyselin.

1.3.1 Transkripce

Syntézu RNA podle matrice DNA katalyzuje enzym RNA polymeráza, který u prokaryot katalyzuje syntézu všech druhů RNA. Skládá se ze dvou α a dvou β podjednotek a jedné podjednotky σ, kterou označujeme jako sigma faktor a zprostředkovává vazbu enzymu na promotorové sekvence DNA. Transkripce probíhá po transkripčních jednotách, což jsou definované úseky DNA, které se skládají z promotoru, strukturních genů a terminátoru. Přepis DNA probíhá ve třech fázích: iniciace, elongace a terminace.

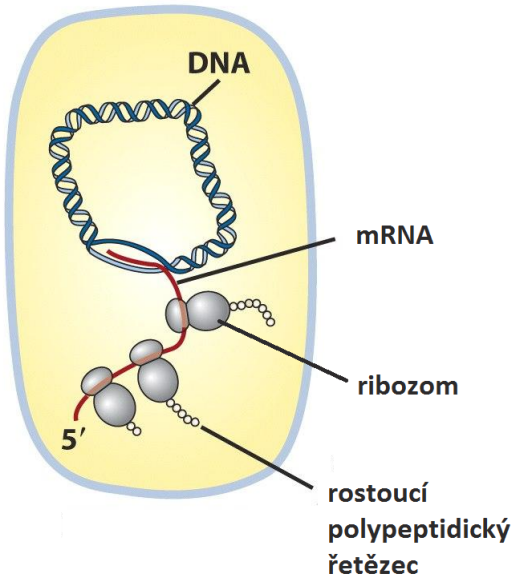
Exprese začíná navázáním RNA polymerázy na promotor, což je specifická sekvence DNA o délce asi 40 bp. Po navázání dochází k rozpletení dvoušroubovice a enzym zahajuje syntézu RNA prostřednictvím přidáváním nukleotidů. Po dosažení délky asi 9 nukleotidů se od polymerázy odštěpí sigma faktor a polymeráza se posunuje dále podél molekuly DNA a tím dochází k prodlužování vlákna RNA. Syntéza probíhá ve směru od 5‘ konce ke 3‘ konci. Jakmile se syntetizující enzym dostane k terminátoru, polymeráza uvolní jak mRNA, tak DNA, a tím dojde k ukončení transkripce. [2]

1.3.2 Translace

Druhá část genové exprese spočívá v překladu mRNA do pořadí aminokyselin, jež probíhá podle pravidel genetického kódu.

Iniciace translace začíná vytvořením translančního komplexu, který je proveden ve třech krocích. Nejprve se tři iniciační faktory (IF1-IF3) a GTP navážou na menší ribosomální podjednotku. Poté se na další vazebná místa připojí tRNA a mRNA a nakonec se připojí větší podjednotka ribozomu. Iniciační tRNA je navázána na P místo ribozomu.

Následně dochází k přikládání dalších tRNA na příslušné kodony mRNA a začíná se tvořit polypeptidový řetězec, který se prodlužuje, dokud není načten jeden ze tří stop kodonů (UAG, UAA, UGA), který interaguje s uvolňovacími faktory a tím dojde k ukončení translace a oddělení polypeptid od tRNA. [6]



Obr. 1.2: Exprese genetické informace u prokaryot, z [7].

1.4 Metody predikce genů

Pokud chceme pochopit genomy jednotlivých organismů, je nezbytné, je identifikovat a popsat, a tudíž je potřeba nalézt program, který tento objem dat zpracuje a jednotlivé geny správně identifikuje a anotuje. Metody predikce můžeme obecně rozdělit do dvou skupin: vnitřní (ab initio) a vnější (homology-based). [8]

1.4.1 Vyhledávání genů ab initio

Predikce ab initio patří mezi vnitřní metody, což znamená, že zkoumá danou sekvenci, kde hledá přítomnost signálů, mezi které patří promotory, start a stop kodony a poté určí statistiky výskytu kodonů, které jsou typické pro daný organismus.

Nejjednodušší metoda predikce ab initio zkoumá ORF (Open Reading Frame) neboli otevřené čtecí rámce, což jsou dostatečně dlouhé úseky DNA ohraničené start a stop kodonem, které kódují polypeptidický řetězec. U každé DNA můžeme nalézt těchto rámců šest, tři na vedoucím vlákně a tři na vlákně komplementárním. Většinou pouze jeden ORF je použit pro translaci genů. Výhodou prokaryotických organismů je, že mají dlouhá ORF a většinou nejdelší obsahuje gen. Avšak neplatí to ve sto procentech případů, takže zde vzniká problém mezi odlišením krátkých a dlouhých genů. Mezi další komplikace patří překryv ORF, který sice není u prokaryot častý, ale může se vyskytnout. [8]

Nejrozšířenější algoritmy predikce ab initio jsou založeny na Markovových modelech a dynamickém programování, kterým se budeme věnovat v dalších částech.

1.4.2 Metody založené na homologii

U homologního vyhledávání identifikujeme geny pomocí srovnáváním s existujími proteinovými sekvencemi. Vezmeme zkoumanou sekvenci, v knihovně sekvencí jiných organismů hledáme shodu, a pokud ji najdeme, můžeme sekvenci identifikovat. Mezi metody, které jsou založené na hledání podobností, můžeme zařadit lokální a globální zarovnání.

Výhodou homologního vyhledávání je snadnost a velmi vysoká přesnost. Nicméně na druhou stranu pro hledání je potřeba velké množství externích dat, ve kterých lze geny vyhledávat. Další nevýhodou je, že spousta genů není významně homologní k již známým genům. [8]

1.5 Programy pro predikci genů

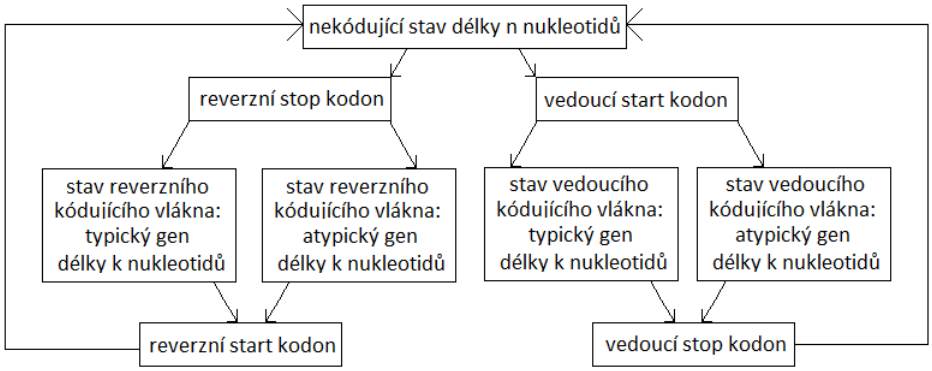
1.5.1 GeneMark.hmm

GeneMark.hmm patří do skupiny GeneMark, což je rodina programů sloužící k predikci genů již od roku 1995. Samotný GeneMark.hmm byl vyvinut v roce 1998 za účelem zlepšení predikce genů, a to díky přesnému nalezení hranice mezi nimi. Využívá soustavu skrytých Markovových modelů (HMM) a generalizovaný Viterbiho algoritmus k určení nejpravděpodobnější sekvence skrytých stavů založených na celé pozorované DNA sekvenci.

Pokud máme sekvenci $S \in \{b_1, b_2, ..., b_L\}$, kde b_i odpovídá bázi T, C, G nebo A a L je délka sekvence, můžeme pro ni vytvořit funkční sekvenci $A \in \{a_1, a_2, ..., a_L\}$, kde každé a_i může nabývat hodnoty 0, pokud nukleotid b_i je součástí nekódující oblasti nebo hodnoty 1, pokud je součástí genu na vedoucím vlákně DNA nebo hodnoty 2, pokud se gen nachází na komplementárním vlákně. Dojde ke spočítání $P(A|S)$ hodnoty a eventuálně se definuje funkční sekvence A^*, která popisuje nejpravděpodobnější anotaci sekvence S. Výpočet $P(A|S)$ je prováděn pomocí skrytých Markovových modelů. [9]

Obecně je skrytý Markovův model definován jako pravděpodobnostní stavový model s konečným počtem stavů, jenž na základě pravděpodobnosti přechází mezi jednotlivými stavy. Skrytý se nazývá, protože z venku nemůžeme určit stav, ve kterém se právě nachází, získáme pouze informaci o výstupu, který nastane s určitou pravděpodobností. [10]

Architektura skrytých Markovových modelů, kterou využívá GeneMark.hmm, je zobrazena na obr. 1.3.



Obr. 1.3: Skrytý Markovův model prokaryotické nukleotidové sekvence používaný v GeneMark.hmm, z [9].

Pokud chceme najednou řešit vedoucí i komplementární vlákno DNA, musíme definovat devět skrytých stavů, které korespondují s funkčními jednotkami bakteriálního genomu. Je potřeba definovat typické geny, atypické geny, start a stop kodony a nekódující oblasti a to jak pro vedoucí, tak pro komplementární vlákno. Nevýhodu je, že tento skrytý Markovův model nepočítá s překryvem genů.

Jednou z vlastností HMM je, že každý kódující i nekódující skrytý stav může generovat nukleotidovou sekvenci, pozorovanou sekvenci, délky skrytých stavů. Pozorovaná sekvence DNA $S \in \{b_1, b_2, ..., b_L\}$ je generována pomocí HMM, jak je zobrazeno v obr. 1.3, paralelně s HMM přechodem z jednoho skrytého stavu do dalšího. Trajektorie skrytého stavu A, jedna z mnoha variací dovolených cest, může být stručně reprezentována jako sekvence M skrytých stavů a_i mající délku d_i: $A = \{(a_1 d_1)(a_2 d_2)...(a_M d_M)\}$. Pro danou sekvenci pozorovaných stavů (nukleotidů) $S \in \{b_1, b_2, ..., b_L\}$ je optimální trajektorie skrytých stavů A^* definována jako trajektorie (funkční sekvence) A s maximální hodnotou podmíněné pravděpodobnosti $P(A|S)$.

Pomocí Viterbiho algoritmu najdeme pro danou nukleotidovou sekvenci S maximálně pravděpodobnou trajektorii A^* mezi skrytými stavy. Trajektorii A^* můžeme popsat následovně: $A^* = \{(a_1^* d_1^*)(a_2^* d_2^*)...(a_M^* d_M^*)\}$. Tato trajektorie má nejvyšší pravděpodobnost výskytu simultánně se sekvencí S ve srovnání se všemi dalšími možnými trajektoriemi.

V posledním kroku programu je hledáno místo, kde se navazuje ribozom (RBS). Toto hledání bylo zavedeno, jelikož se ukázalo, že Viterbiho algoritmus predikuje

geny, které jsou součástí překryvu, kratší než ve skutečnosti jsou. Pro predikovaný gen je hledáno RBS v intervalu -19 až -4 nukleotidů proti směru každého alternativního start kodonu, ležícího mezi pozicí start kodonu navrhovaného algoritmem a pozicí start kodonu neseným nejdelším ORF pro predikovaný gen a dle skóre je vybrán správný začátek translace. [9]

1.5.2 Glimmer

Glimmer neboli Gene Locator and Interpolated Markov ModelER je systém pro hledání genů v mikrobiální DNA. Funguje na principu interpolovaných Markovových modelů, které využívá pro hledání kódujících oblastí a jejich oddělení od oblastí nekódujících.

Glimmer byl vyvinut v institutu pro genomických výzkum v roce 1998. O rok později byla vydána druhé verze označená jako Glimmer 2.0 a v roce 2007 byla vydána třetí verze. V současné době je nejnovější Glimmer 3.02, který je přístupný na webových stránkách NCBI.

Glimmer 1.0

První verze programu používá pro hledání kódujících oblastí interpolované Markovovy modely (IMM). Nejprve dojde k vytvoří IMM pro všech šest čtecích rámců, které jsou poté využity pro ohodnocení celých ORF. Pokud se dva čtecí rámce s vysokým skórem překrývají, překryv je skórován samostatně a dojde k určení, kde pravděpodobně leží gen.

Abychom mohli definovat IMM, musíme nejdříve objasnit, co je Markovův řetězec. 1. řád Markovova řetězce můžeme popsat následovně: pokud máme sekvenci X_1, X_2, ... X_i, kde i je pozice v sekvenci a $X \in \{A, C, G, T\}$, přičemž pravděpodobnost, že X_i bude obsahovat nějakou bázi, závisí pouze na bázi X_{i-1}. U k-tého řádu pravděpodobnost, že X_i bude obsahovat nějakou bázi, závisí na k předchozích bází. Pro hledání genů pomocí řetězce musíme vytvořit 7 submodelů, 6 pro každý čtecí rámec a 1 pro nekódující sekvence, kde každý provádí predikce pro báze ve všech třech pozicích v kodonu a poté pomocí modelů ohodnotí každé ORF a vybere model s nejvyšším skórem. Pokud model korespondující se správnou kódovací oblastí ve správném rámci má nejvyšší skóre, může být označen jako gen.

IMM využívají kombinaci pravděpodobností z $0, 1, 2...k$ předchozích bází, kde k má hodnotu 8. Tudíž pro oligomery vyskytující se často, může být použit 8. řád, zatímco pro oligomery méně časté se využije nižší řád. Dojde ke spočítání pravděpodobnosti pro 0 až 8 přechozích bází a poté jsou spočítány váhy pro použití v kombinaci s predikcí jiných řádu modelu. Glimmer ohodnotí novou sekvenci spočítáním pravděpodobnosti, že model M vytvořil sekvenci S. Pravděpodobnost je

spočítaná jako:

$$P(S|M) = \sum_{x=1}^{n} IMM_8(S_x), \tag{1.1}$$

kde S_x je oligomer končící na pozici x a n je délka sekvence. Skóre pro $IMM_8(S_x)$, což značí 8.řád interpolovaného Markovova modelu spočítáme jako:

$$IMM_k(S_k) = \lambda_k(S_{x-1}) \cdot P_k(S_x) + [1 - \lambda_k(S_{x-1})] \cdot IMM_{k-1}(S_x), \tag{1.2}$$

kde $\lambda_k(S_{x-1})$ je číselná váha k-meru, který končí na pozici *x-1* v sekvenci S a $P_k(S_x)$ je odhad získaný z trénovacích dat pravděpodobnosti báze lokalizované na x v k-tém řádu modelu. Z toho plyne, že 8.řád IMM skóre oligomeru je lineární kombinací predikcí vytvořených všemi 8 řády.

Z trénovací množiny genů jsou stanoveny frekvence výskytu pro všechny podřetězcové vzory délky 1 až k+1 pro každý čtecí rámec. Pro jeden čtecí rámec *f(S)* označuje frekvence výskytu v sekvenci $S = s_1 s_2 \ldots s_n$. Z těchto frekvencí dostaneme počáteční odhady pravděpobnosti báze s_x, která je dána kontextovým řetězcem $s_{x-i}, s_{x-i+1}, \ldots, s_{x-1}$, označeným jako $S_{x,i}$ (i bází předešlých pozici x). Pravděpodobnost báze s_x danou i předchozími bázemi spočítáme jako:

$$P(s_x|S_{x,i}) = \frac{f(S_{x,i})}{\sum_{b \in acgt} f(S_{x,i}, b)}, \tag{1.3}$$

Pokud počet výskytů v řetězci $S_{x,i}$ v trénovacích datech přesáhne specifický práh, potom $\lambda_i(S_x)$ je nastaveno na 1 a použijeme tyto vzorky pravděpodobností. Pokud ovšem v datech není dostatek výskytu k dostatečnému odhadu pravděpodobnosti další báze, k λ hodnotě je přidáno dodatečné kritérium. Pro daný kontextový řetězec $S_{x,i}$ délky i, srovnáme pozorované frekvence: f($S_{x,i}$, a), f($S_{x,i}$, c), f($S_{x,i}$, g) a f($S_{x,i}$, t) s vypočítanými IMM pravděpodobnostmi využívající další kratší kontext, IMM_{i-1} ($S_{x,i}$, a), IMM_{i-1} ($S_{x,i}$, c), IMM_{i-1} ($S_{x,i}$, g) a IMM_{i-1} ($S_{x,i}$, t) a pomocí testu dobré shody určíme, jak se 4 pozorované frekvence liší od IMM hodnot. Pokud se frekvence velmi liší, je lepší pro odhad báze použít právě tyto frekvence, tzn. dát jim vyšší λ hodnotu. Naopak, když jsou frekvence shodné s IMM hodnotami, nabízejí malou prediktivní hodnotu, a proto jim dáme menší λ hodnotu. Tato λ hodnota nyní definuje pravděpodobnosti IMM_i ($S_{x,i}$,b) kde $b \in \{a, c, g, t\}$.

Samotný Glimmer systém se skládá ze dvou programů. První nazývaný „build-imm“ vytvoří pro vstupní množinu sekvencí IMM, jak je popsáno výše a druhý program, zvaný glimmer, potom použije IMM k identifikaci domnělých genů v celém genomu a to tak, že nejprve identifikuje všechny ORF delší než daný práh a ohodnotí každé ve všech čtecích rámcích. Vybraná ORF potom zkoumáme pro překryv. Každá překrývající se oblast je hodnocena zvlášť a tato skóre porovnáváme, abychom zjistili, který rámec dosahuje vyššího skóre. Obecně pokud se delší ORF překrývá s kratším

a překrývající oblast má vyšší skóre ve čtecím rámci delšího ORF, tak kratší je zamítnuto. [11]

Glimmer 2.0

Druhá verze Glimmeru byla vydána v roce 1999 a oproti Glimmeru 1.0 má lehce vyšší sensitivitu a je mnohem lepší v řešení překrývajích se genů. Jako základ používá interpolovaný kontextový model.

Interpolovaný kontextový model je rozšířením pro interpolované Markovovy modely. Pro daný kontext $C = b_1 b_2 \ldots b_k$ délky k, může ICM vybrat jakoukoliv bází v kontextu C (ne jenom ty, které sousedí s b_{k+1}) k určení pravděpodobnosti b_{k+1}. Jako kritérium k určení, kterou bázi z kontextu C vybrat, je použita tzv. společná informace, která je mezi daným párem diskrétních náhodných proměnných X a Y definována jako:

$$I(X;Y) = \sum_i \sum_j P(x_i, y_j) \log(\frac{P(x_i)P(y_j)}{P(x_i, y_j)}), \tag{1.4}$$

kde X_i a Y_j jsou hodnoty vzaté náhodnou proměnnou X, respektive Y a P(x_i, y_j) je společná pravděpodobnost x_i a y_j.

Abychom mohli z trénovací množiny T DNA sekvencí vytvořit ICM délky k, musíme nejprve začít zvážením všech oken (tj. oligomerů) délky *k+1* , které se objevují v T. Náhodná proměnná X_1 udává distribuci bází v první pozici tohoto okna, X_2 ve druhé pozici a tak dále až k X_{k+1}. Následně jsou spočítány vzájemné informační hodnoty I(X_1;X_{k+1}), I(X_2;X_{k+1}), ... I(X_k;X_{k+1}) a poté je z nich vybráno maximum. Pokud je maximum I(X_j;X_{k+1}), rozdělíme naši množinu oken do 4 podmnožin založených na nukleotidu, který se v okně vyskytuje na pozici *j*.

Stejný postup je proveden znovu pro každou ze 4 množin oken. V každé množině je vybrána pozice, která má nejvyšší společnou informaci s bází na pozici *k+1* a je vyvoláno další dělení současné množiny oken ve 4 podmnožiny, protože existují 4 typy nukleotidů.

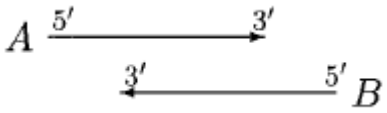
Tímto postupem je vytvořen strom pozic pro každý kontextový řetězec. Tvoření stromu skončí, když hloubka stromu dosáhne předurčeného limitu nebo když je velikost množiny oken moc malá pro efektivní stanovení pravděpodobnosti poslední pozice báze.

Každý uzel ve stromu reprezentuje množinu oken, které poskytují pravděpodobností rozložení pro každou bázi na výsledné pozici. Kořen stromu, jež reprezentuje 0. řád Markovova modelu, obsahuje všechna možná okna a další uzly udávají pravděpodobnost distribuce pro výslednou pozici báze, která je podmíněná množinou bází, jež se vyskytuje na pozicích u cesty z uzlu ke kořenu.

Interpolovací mechanismus využitý v druhé verzi Glimmeru je stejný jako ten v první. Jediný rozdíl je, že ICM interpolace probíhá jako interpolování mezi distribucemi v rodičovském a dceřinném uzlu ve stromě, zatímco IMM interpolace je vždy mezi distribucemi získaných využitím různých čísel bazí na konci kontextového okna.

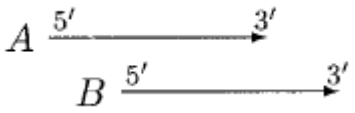
Pokud se dva geny překrývají, oblast překryvu je nejprve ohodnocena stejně jako v předcházející verzi programu, ale poté se systém pokusí posunout pozice start kodonů, jak je uvedeno níže. Posun startovacího kodonu funguje následovně: systém zkrátí predikovaný gen posunutím startu na další dostupný start kodon. Pokud nedojde k vyřešení překryvu, posune se start kodon znovu a tento postup pokračuje tak dlouho, dokud je výsledný gen delší než minimální délka genu.

Pokud domnělý gen A má vyšší skóre než gen B, jsou zváženy 4 různé orientace:



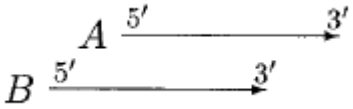
Obr. 1.4: Případ překryvu č. 1, z [12].

Na obr. 1.4 je ilustrován první případ, kdy posun startu A nebo startu B překryv nevyřeší. Takže pokud je A delší než B, potom B zamítneme jako gen. Jestliže A není delší než B, budou oba dva geny označeny jako geny s poznámkou, že zde byl nalezen nejasný překryv.



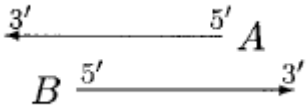
Obr. 1.5: Případ překryvu č. 2, z [12].

Druhý případ je uveden na obr. 1.5. V tomto případě lze překryv vyřešit jedině posunutím genu B, tudíž pokud s ním může pohnout, tak to uděláme. Pokud ne a B je znatelně kratší než A, potom je B zamítnuto. Jinak jsou oba geny označeny jako geny s poznámkou značící překryv.



Obr. 1.6: Případ překryvu č. 3, z [12].

Ve třetím případě, jak vidíme na obr. 1.6, může být překryv vyřešen pouze posunem startu A. Jelikož A má vyšší skóre, zkusíme ho posunout. Posun provádíme do té doby, dokud je překryv relativně malou frakcí délky A. Pokud přizpůsobení A není úspěšné, B je zamítnuto.



Obr. 1.7: Případ překryvu č. 4, z [12].

V posledním případě (obr. 1.7) lze posunout oba starty. Nejprve posouváme start B, dokud oblast překryvu nemá větší skóre pro gen B. Následně posouváme start A, dokud nebude mít větší skóre a poté opakujeme posun genu B, atd., dokud můžeme posun provádět nebo dokud není odstraněn překryv. [12]

Glimmer 3.0

Třetí a zatím poslední dostupná verze byla vypuštěna v roce 2007. Dle autorů je snížena míra falešně pozitivních predikcí, je správně predikován větší počet startovacích míst a stále je zachována vysoká míra správně pozitivních predikcí.

Autoři se v této verzi zabývali problematikou separace a identifikace DNA hostitele a jeho symbionta. Proto vyvinuli nový algoritmus, ve kterém se IMM uvnitř Glimmeru učí zvlášť na hostiteli a endosymbiontovi a poté se promění ve třídící program, který je schopen tyto dvě sekvence oddělit. [13]

1.5.3 Prodigal

Prodigal neboli Prokaryotic Dynamic Programming Genefinding Algorithm je program pro vyhledávání genů, který byl vyvinut v roce 2007, avšak publikován byl až o tři roky později.

Program začíná projitím celé vstupní sekvence a v každém otevřeném čtecím rámci spočítá počet bází guaninu a cytosinu pro každou kodonovou pozici, z kterého určí bias skóre, pomocí kterého je vytvořeno předběžné kódovací skóre pro každý gen v genomu. Skóre S pro daný gen začínající na pozici n1 a končící na pozici n2 je dáno vzorcem:

$$S(n1..n2) = \sum_{i=3}^{3} B(i) * l(i), \tag{1.5}$$

kde *B(i)* je bias skóre pro pozici v kodonu i a l je počet bází v genu. Tímto skórem je ohodnocený každý start-stop pár, který je delší než 90 bp. Při dynamickém programování je každý uzel v matici buď start nebo stop kodon. Spojení startu k jeho odpovídajícímu stopu reprezentuje gen, s ohledem na to, že spojení 3' konce k novému 5' konci reprezentuje mezigenový prostor. Jelikož dynamické programování nemůže jít zpětně, protože částečné řešení daného bodu, musí být součástí výsledného řešení, je potřeba určit pravidla pro překrývající se geny. Prodigal ohodnotí překrývající se geny pro všechny tři rámce pro každý 3' konec v genomu a vytvoří nový typ spojení z 3' konce jednoho genu k 3' konci druhého genu na stejném vlákně. Mezi dvěmi geny na stejném vlákně je povolen maximální překryv 60 bp a pro opačné vlákno 200 bp mezi 3' koncemi genu a pro 5' konce genů překryvy povoleny nejsou.

Po skončení předběžného algoritmu dynamického programování, dojde k vyhodnocení domnělých genů a vytvoření více důkladného kódovacího skóre. Na sekvenci je pohlíženo jako na řadu slov délky 6, tedy hexamery. Frekvence hexameru uvnitř rámce mohou sloužit k rozdělení kódujících regionů od těch nekódujících. Prodigal počítá skóre pro hexamer, tedy slovo w jako:

$$C(w) = \log(G(w)/B(w)), \tag{1.6}$$

kde C je kódovací skóre, G je procentuální výskyt daného slova v naší trénovací množině a B je procentuální výskyt daného slova v celé sekvenci. Finální kódovací skóre pro gen začínající na pozici n1 a končící na n2 může být zapsáno jako:

$$S(n1..n2) = \sum_{i=n1}^{n2} C(w(i)), \tag{1.7}$$

kde S je suma kódovacích skór (C) pro hexamery uvnitř rámce (množina slov w). Toto kódovací skóre je poté poupravené podle toho, co leží proti směru vybraného

startu, aby se zamezilo výběru zkrácené verze genu, když může být vybrána delší verze toho samého genu s vyšším skórem.

Dalším krokem je vytvoření skórovacího systému pro translační iniciační místo. Ze všech startovacích uzlů je získáno pozadí ATG, GTG a TTG frekvencí a též je vytvořeno pozadí RBS motivů založených na Shine-Dalgarno (SD) sekvenci, což je krátký úsek ležící před iniciačním kodonem.

Pro RBS motivy je používán koncept 'košů'. Každý koš koresponduje určité sadě RBS motivů a vzdálenosti mezi motivem a iniciačním kodonem translace. Tyto koše byly určené z sady dat z GenBanku.

Na začátku, pokud jsou dva motivy nalezeny proti směru od startovacího místa, má ten, který spadá do více početného koše přednost než ten, spadající do koše méně početného. Prodigal zkoumá výchozí kódovací vrcholy, což jsou nejvíce skórující start-stop páry pro daný stop kodon v každém ORF, které mají skóre vyšší než daný práh. Z těchto kódovacích vrcholů postaví log-pravděpodobnostní model podobný kódujícímu skóre, popsaný následovně:

$$S(n) = \log(R(n)/B(n)), \tag{1.8}$$

kde S je skóre, R je pozorované procento daného typu v naší trénovací množině a B je procento výskytu v pozadí. Tato metoda je používána pro použití start kodonu stejně jako pro koš SD motivu. Skóre jsou sečtená dohromady a vynásobena konstantou a poté přidána do kódovacího skóre. Prodigal projde skrz každý start-stop uzel a provede tento výpočet a modifikaci, což vede k novému nastavení vrcholů pro množinu trénovacích ORFs.

Jakmile je určena nová množina vrcholů, Prodigal rekonstruuje pozadí pro oba SD motivy a pro použití start kodonu. V této iteraci a v té následující už dále nepředpokládá, že vyšší očíslovaný koš je lepší pro RBS motivy a místo toho spoléhá na log-pravděpodobnost vypočítanou v předchozí iteraci k nalezení nejlepšího motivu pro daný start. Prodigal provádí několik iterací tohoto procesu a dokud se vrcholy pohybují významně, tak s nimi pohybuje. Když se vrcholy dále nepohybují, určí finální množinu vah založených na statistice získané z finální množiny domnělých pravých start kodonů.

Výsledný výsledek je množina log-pravděpodobnostních vah pro ATG/GTG/TTG a pro každý RBS koš.

Pokud se ukáže, že organismus nepoužívá SD motiv, Prodigal prohledá všechny alternativní motivy a potom provede iterativní algoritmus stejný jako je uveden výše. Koše v tomto případě korespondují každému slovu délky 3-6 bp.

Nakonec vytvoříme další skórovací systém, abychom získali informace o regionech, které již nejsou hodnoceny pomocí RBS skóre, tzn. pro několik bází proti směru

iniciačního místa translace. Tento systém postaví matici váhovaných pozic pro celou oblast.

Jakmile je vytvořeno skóre vah pro všechny typy startovacích kodonů, pro RBS motivy a pro vzdálenosti mezi motivy a iniciačním místem translace, potom je ohodnocen každý start a to v celé sekvenci. Finální skóre pro startovací uzel je vyjádřeno jako:

$$S(n) = 4,25 * (R(n) + T(n) + 0,4 * U(n)) + C(n), \tag{1.9}$$

kde S je výsledné skóre, R je RBS motiv skóre, T je skóre pro daný typ startu, U je skóre pro oblasti, které nebyly hodnoceny pomocí RBS skóre a C je kódovací skóre.

Jakmile je vypočítané výsledné skóre, DP proběhne podruhé a pro spojení genů používá již přesnější skóre. Po proběhnutí dynamického programování, program projde výsledné geny a odstraní ty, které mají záporné skóre a výsledkem je seznam genových koordinátů. [14]

2 PRAKTICKÁ ČÁST

2.1 Testování softwarů

V této části je popsáno testování softwarů, jež jsou uvedeny v kapitole 1.5, na prokaryotickém genomu *Escherichia coli* O111:H- str. 11128 DNA. Tento genom byl vybrán, jelikož na rozdíl od mnoha ostatních genomů v NCBI databázi, nebyl popsán pomocí automatických anotačních programů. V následujících kapitolách je nejprve popsáno ovládání jednotlivých programů s náhledem výstupních dat a poté je již vyhodnocena jejich účinnost.

2.1.1 GeneMark.hmm

Program GeneMark.hmm je dostupný na webových stránkách `http://exon.gatech.edu/gmhmmp.cgi`. Zde můžeme sekvenci, kterou chceme analyzovat, nahrát přímo pomocí textového pole nebo nahrát ze souboru. V obou případech musí být ve FASTA formátu. Dále volíme ze seznamu organismů ten, který chceme analyzovat, což považuji za nevýhodu programu, protože pokud se náš organismus v seznamu nenalézá, nelze ho analyzovat. V dalším nastavení lze zvolit výstupní formát, dále, co vše chceme, aby bylo součástí výstupu a zda výsledky chceme poslat na e-mail.

Výstupem je webová stránka, která je zobrazena na obr. 2.1. Zde najdeme FASTA hlavičku a dále predikované geny, jejich začátek, konec, délku, na jakém vlákně se nacházejí a třídu, která udává, zda se jedná o typický nebo atypický gen.

```
GeneMark.hmm PROKARYOTIC (Version 3.26)
Date: Sat May 14 07:06:11 2016
Sequence file name: seq.fna
Model file name:
/home/genemark/parameters/prokaryotic/Escherichia_coli_O111_H__11128/GeneMark
_hmm_combined.mod
RBS: true
Model information: Escherichia_coli_O111_H__11128

FASTA definition line: gi|257762509|dbj|AP010960.1| Escherichia coli O111:H-
str. 11128 DNA, complete genome
Predicted genes
   Gene    Strand    LeftEnd    RightEnd       Gene     Class
    #                                        Length
    1        +          <3          98          96         1
    2        +         294        2798        2505         1
    3        +        2800        3732         933         1
    4        +        3733        5019        1287         1
    5        +        5087        5236         150         2
    6        +        5233        5529         297         1
    7        -        5683        6459         777         1
```

Obr. 2.1: Výstupní obrazovka programu GeneMark.hmm.

2.1.2 Glimmer

Glimmer ve verzi 3.02 je přístupný na webových stránkách `http://www.ncbi.nlm.nih.gov/genomes/MICROBES/glimmer_3.cgi`. Zde opět můžeme použít k nahrání sekvence textové pole nebo lze sekvenci nahrát ze souboru. Sekvence stejně jako u předcházejícího softwaru musí být ve formátu FASTA. Po nahrání vybereme, o jaký genetický kód se jedná (Bacteria, Archea nebo Mycoplasma/Spiroplasma) a nakonec zvolíme topologii genomu (lineární nebo kruhovou).

Výstup nalezneme na webové stránce, kde se nachází FASTA hlavička sekvence, dále ID ORF, pozice start a stop kodonu, rámec a skóre, jak vidíme na obr. 2.2. U indexů pro začátky a konce genů si musíme dát pozor, jelikož na rozdíl od ostatních programů, jsou u komplementárního vlákna prohozené.

```
>gi|257762509|dbj|AP010960.1| Escherichia coli
 O111:H- str. 11128 DNA, complete genome
orf00002      336     2798  +3    11.00
orf00003     2800     3732  +1    10.94
orf00005     3733     5019  +1    14.29
orf00007     5536     5309  -2     3.54
orf00009     6459     5683  -1    13.05
orf00010     7959     6529  -1     9.24
orf00011     8307     9191  +3    14.37
orf00012     9306     9893  +3    10.68
```

Obr. 2.2: Výstupní obrazovka programu Glimmer.

2.1.3 Prodigal

Program je dostupný na webových stránkách `http://compbio.ornl.gov/prodigal/server.html`. Po otevření stránky musíme buď načíst sekvenci ve FASTA formátu ze souboru nebo zadat znaky ručně. Dále volíme translační tabulka a to buď pro standardní bakterie/Archea nebo pro Mycoplasma/Spiroplasma. Z možností vybereme, co chceme, aby bylo ve výstupu programu: pouze pozice genu, pozice genu s translací proteinu, pozice genu s detailními informacemi o startu nebo kombinaci všech uvedených možností.

Výstupem programu je jako v předchozích programech webová stránka, kterou vidíme na obr. 2.3. Nalezneme zde hlavičku sekvence ve FASTA formátu, pozice jednotlivých genů a zda se nacházejí na vedoucím či komplementárním vlákně.

Sequence: >gi|257762509|dbj|AP010960.1| Escherichia coli O111:H- str. 11128 DNA, complete genome
Length: 5447895 characters

Your analysis has been dispatched to the server and should be done momentarily. Do not press the back button or leave this page until the job is complete. The request ID for your job is **1463225314-15644**.

```
Gene Coordinates:

DEFINITION  >gi|257762509|dbj|AP010960.1| Escherichia coli O111:H- str. 11128
            DNA,complete genome
     CDS             336..2798
     CDS             2800..3732
     CDS             3733..5019
     CDS             5233..5529
     CDS             complement(5683..6459)
     CDS             complement(6529..7959)
     CDS             8238..9191
     CDS             9306..9893
     CDS             complement(9928..10494)
     CDS             complement(10643..11356)
     CDS             complement(11382..11786)
     CDS             12163..14079
     CDS             14168..15283
     CDS             complement(15387..15596)
     CDS             16125..17291
     CDS             17357..18256
```

Obr. 2.3: Výstupní obrazovka programu Prodigal.

2.1.4 Shrnutí výsledků

Uvedené programy byly testovány na kompletním genomu *Escherichia coli* O111:H- str. 11128 DNA, který byl získán z databáze NCBI. Organismus *E.coli* obsahuje 5264 anotovaných genů, přičemž v databázi jsou pro každý gen uvedeny indexy začátku a konce. Získané indexy jsme následně vyhledávali ve výsledcích programů pro predikci. Pro hledání pozic byla nastavena tolerance ± 3 báze, což odpovídá jednomu kodonu. Výsledky predikce genů pro jednotlivé softwary jsou uvedeny v tab. 2.1 a v tab. 2.2.

Senzitivita programů byla spočítána jako počet správně identifikovaných genů dělený počtem genů uvedených v databázi NCBI a specificita jako počet správně identifikovaných genů dělený celkovým počtem nalezených genů pro daný genom. Nakonec byla určena přesnost jednotlivých programů jako součet senzitivity a specificity dělený dvěma. Senzitivita, specificita i přesnost byly nakonec vynásobeny stem, aby výsledné hodnoty vyšly v procentech. Vzorce pro výpočet byly převzaty z [15].

Tab. 2.1: Výsledky predikce u analyzovaných softwarů.

analyzovaný software	GeneMark.hmm	Glimmer	Prodigal
celkový počet nalezených genů	5327	5504	5217
počet správně identifikovaných genů	4542	4386	4727
počet správně identifikovaných začátků	4841	4727	4904
počet správně identifikovaných konců	4820	4696	4919

Tab. 2.2: Úspěšnost analyzovaných softwarů.

analyzovaný software	GeneMark.hmm	Glimmer	Prodigal
senzitivita [%]	86,28	83,32	89,80
specificita [%]	85,26	79,69	90,61
přesnost [%]	85,77	81,51	90,21

Z celkového počtu 5264 genů, nalezly první dva zmíněné programy více genů, než ve skutečnosti genom obsahuje. Tato skutečnost může být dána tím, že ne každý start kodon musí automaticky znamenat start, ale může být i uprostřed genu a zde kódovat aminokyselinu.

Z hlediska přesnosti dosáhl nejvyšší úspěšnosti Prodigal, jehož přesnost vyšla 90,21 % a poté GeneMark.hmm s 85,77 %. Přesnost Glimmeru dosáhla nejmenší hodnoty a to 81,51 %.

2.2 Program Gene_finder

Program *Gene_finder* byl vytvořen v prostředí MATLAB R2015a. Při jeho tvorbě byl využit i Bioinformatics toolbox. Úkolem programu je načíst FASTA soubor, jež obsahuje sekvenci prokaryotického organismu a v dané sekvenci najít jednotlivé geny. Výsledkem je tabulka nalezených začátků, konců a délky predikovaných genů a dále, ve kterém čtecím rámci se nacházejí. Pokud geny leží na vedoucím vlákně, je hodnota čtecího rámce kladná, pokud leží na komplementárním vlákně, je hodnota záporná. Výstup programu je ukládán ve formátu 'results_of_gene_finder.xls'. Náhled na výstupní tabulku je zobrazen na obr. 2.4.

Samotný program se skládá z několika funkcí, jejichž princip je teoreticky popsán v následujících sekcích. Vývojový diagram navrženého algoritmu je uveden v příloze A.

	A	B	C	D
1	gene_start	gene_end	reading_frame	gene_length
2	337	2799	1	2463
3	2801	3733	2	933
4	3734	5020	2	1287
5	4501	4701	-1	201
6	6139	6378	1	240
7	7366	7773	1	408
8	8238	9191	3	954
9	8451	8807	-3	357
10	9306	9893	3	588

Obr. 2.4: Výstup programu v Excelu.

2.2.1 Načtení sekvence

V prvním kroku dochází k načtení sekvence pomocí funkce *readfasta*, do které vstupuje název souboru s analyzovanou sekvencí ve formátu 'nazev.fasta'. Výstupem funkce je samotná sekvence a její hlavička.

Následně je pomocí funkce *leading_and_complement* vytvořen reverzní komplement sekvence, který je uložen do jedné proměnné společně s hlavní sekvencí. Program *Gene_finder* probíhá nejprve pro vedoucí vlákno a následně pro reverzní komplement.

2.2.2 Hledání stop kodonů

Do funkce *find_ends_in_frame*, která vyhledává stop kodony, vstupuje sekvence a čtecí rámec, ve kterém chceme hledat. Nejprve provedeme kontrolu, zda je zadán správný čtecí rámec, tj. zda rámec nabývá hodnot 1, 2 nebo 3. Následně provedeme samotné hledání stop kodonů. Začínáme od začátku sekvence. Vždy vezmeme triplet bází a porovnáme jej se zadanými stop kodony (TAA, TAG, TGA). Pokud nalezneme shodu, dojde k uložení pozice první báze stop kodonu. Hledání kodonů končí, když projdeme celou sekvenci. Zároveň je zde ošetřen případ, kdy bychom brali poslední triplet, který by již nebyl úplný (chyběla by jedna nebo dvě báze).

Výstupem funkce je seznam pozic, které náleží první bázi v nalezených stop kodonech.

2.2.3 Hledání start kodonů

V dalším kroku programu provádíme hledání start kodonů pomocí dvou funkcí a to *find_gene_with_start* a *find_gene_in_frame*. Vstupem první funkce je sekvence, start kodon a pozice dvou stop kodonů, které jsou vedle sebe. Nejprve se provede

kontrola, zda zadané start a stop kodony jsou opravdu start a stop kodony. Následně se provádí již samotné vyhledávání start kodonu, které probíhá od konce sekvence. Vezmeme pozici posledního a jemu předcházejícího stop kodonu a mezi nimi provedeme vyhledávání start kodonu. Zde může nastat několik možností, které jsou rozebrány níže.

Pokud žádný start nenalezneme, přesouváme se po vlákně dopředu, tudíž z předcházejícího stop kodonu se stane koncový a poté hledáme mezi ním a jemu předcházejícím stopem.

Pokud nalezneme jeden start kodon, provedeme kontrolu, zda leží ve stejném rámci jako stop kodon. Pokud ne, přesouváme se mezi další dvojici stop kodonů, ale pokud ano, uložíme si pozici první báze start a poslední báze stop kodonu potencionálního genu.

Jestliže se mezi dvěma stop kodony nachází více možných startů, provádíme kontrolu rámce od nejvzdálenějšího a pokud ji nesplňuje, pokračujeme dále směrem blíže ke stop kodonu.

Tímto způsobem projdeme celé vlákno, přičemž když se dostaneme k pozici prvního stop kodonu, start hledáme mezi ním a první bází sekvence. Na výstupu funkce tedy získáme seznam pozic začátků a konců potencionálních genů.

Funkce *find_gene_in_frame* má za vstupy sekvenci, pozice stop kodonů a čtecí rámec. Tato funkce volá funkci předchozí a zároveň provádí hledání startů pro všechny možné start kodony. Nejprve se mezi dvěmi stopy hledá start kodon ATG. Pokud není nalezen, proběhne hledání GTG a pokud se ani ten v daném úseku nenachází, vyhledá se kodon TTG. Tato posloupnost není náhodná, je zvolená tak, aby odpovídala reálné situaci, kde nejčastěji vyskytující se start kodon je právě ATG, poté GTG a nakonec TTG. Start kodonem mohou být i triplety ATT a CTG, které se ovšem vyskytují velmi vzácně, a proto je ve vyhledávání neuvažujeme.

Na konci funkce dochází k volání funkce *gene_length_ok*, která odstraňuje ze seznam potencionálních genů ty, které jsou kratší 200 bází, jelikož existuje pravděpodobnost, že delší otevřené čtecí rámce kódují skutečné geny.

2.2.4 Frekvence kodonů

Následně provedeme výpočet frekvence kodonu skrz funkci *codon_frequency_map*, kam vstupuje sekvence a čtecí rámec. V každém čtecím rámci sekvence spočítáme, kolikrát se vyskytuje který kodon. Získaný počet vydělíme počtem všech kodonů a vynásobíme tisícem, čím dosáhneme hodnoty vztažené na tisíc kodonů. Vypočtené frekvence pro jednotlivé kodony použijeme v následujícím kroku.

2.2.5 Skórování potencionálních genů

Abychom mohli z potencionálních genů vybrat finální geny, ohodnotíme je pomocí skórovacího systému, k čemuž využijeme funkci *get_scoring_windows* a *gene_score*. Potencionální geny převedeme do textových řetězců pomocí funkce *genes_to_string*. Abychom zabránili tomu, že delší geny budou mít nižší skóre než kratší geny, provádíme hodnocení v okně. Okno počítáme pomocí první zmíněné funkce, kam vstupují jednotlivé textové řetězce. Délku okna jsme zvolili 48. Tato hodnota byla vybrána tak, aby byla dělitelná třema a dvěma. Také z experimentálních pokusů vycházeli hodnoty nejlépe pro výše zvolenou velikost. Okno se posouvá o půlku své délky. Pomocí funkce vypočítáme, kde okno začíná a jak bude dlouhé. Pokud délka poklesne pod 48, je jasné, že jsme narazili na konec genu a dál již okno posouvat nebudeme. Výstupem funkce je matice, kde v prvním sloupci jsou hodnoty, kde okno začíná a ve druhém jak bude dlouhé.

Poté již přistupujeme k samotnému skórování pomocí druhé zmíněné funkce. Sem nám vstupuje textový řetězec, matice s okny a frekvenční mapa. Postupně bereme jednotlivé kodony v rámci okna. Vezmeme první kodon, vyhledáme ho pomocí frekvenční mapy a přiřadíme k němu danou frekvenci. Posuneme se na další kodon, opět najdeme jeho frekvenci a vynásobíme ji s předcházející hodnotou. Takto pokračujeme do konce okna. Okno posuneme o polovinu jeho délky a znovu provádíme jednotlivé výpočty skór. Po projití celého genu určíme celkové skóre a to tak, že vezmeme hodnoty pro jednotlivá okna, která nejprve sečteme a následně je vydělíme celkovým počtem oken pro daný gen.

2.2.6 Výběr finálních genů

Pomocí funkce *apply_scoring* provedeme výběr genů, které označíme jako finální. Vstupují nám sem potencionální geny a jejich skóre. Pokud je skóre potencionálního genu vyšší, než je průměrná hodnota všech skór, označíme gen jako finální. Při tomto způsobu výběru genů se geny mohou překrývat v rámci čtecích rámců, což odpovídá jejich skutečnému umístění.

2.2.7 Uložení finálních genů

Poslední funkcí *save_as_table* spojíme výsledky pro vedoucí a reverzní komplementární vlákno dohromady a výsledek převedeme do tabulky, kterou poté uložíme jako 'results_of_gene_finder.xls'. Zároveň zde probíhá převedení pozic genů na reverzním komplementu, a to do takové podoby, aby měly výsledky stejnou strukturu, jakou mají zmiňované programy.

2.3 Testování program Gene_finder

Program pro vyhledávání genů v prokaryotických organismech byl testován celkem na 5 genomech: *Escherichia coli* str. K-12 substr. MC4100, *Escherichia coli* strain JEONG-1266, *Escherichia coli* strain MRE600, *Finegoldia magna* ATCC 29328 DNA a *Escherichia coli* O111H-str.11128 DNA, na kterém byly testovány i programy uvedené v předchozích kapitolách. První čtyři zmíněné genomy byly anotovány automaticky, tudíž popis jednotlivých genů nemusí být stoprocentně správný. Všechny uvedené genomy jsou volně dostupné v databázi GenBank pod následujícími přístupovými čísly: HG738867.1, CP014314.1, CP014197.1, NC_010376.1, AP010960.1.

Jednotlivé genomy s počtem genů, které obsahují a počtem genů, které našel testovaný program, jsou uvedeny v tab. 2.3.

Tab. 2.3: Přehled testovaných genomů.

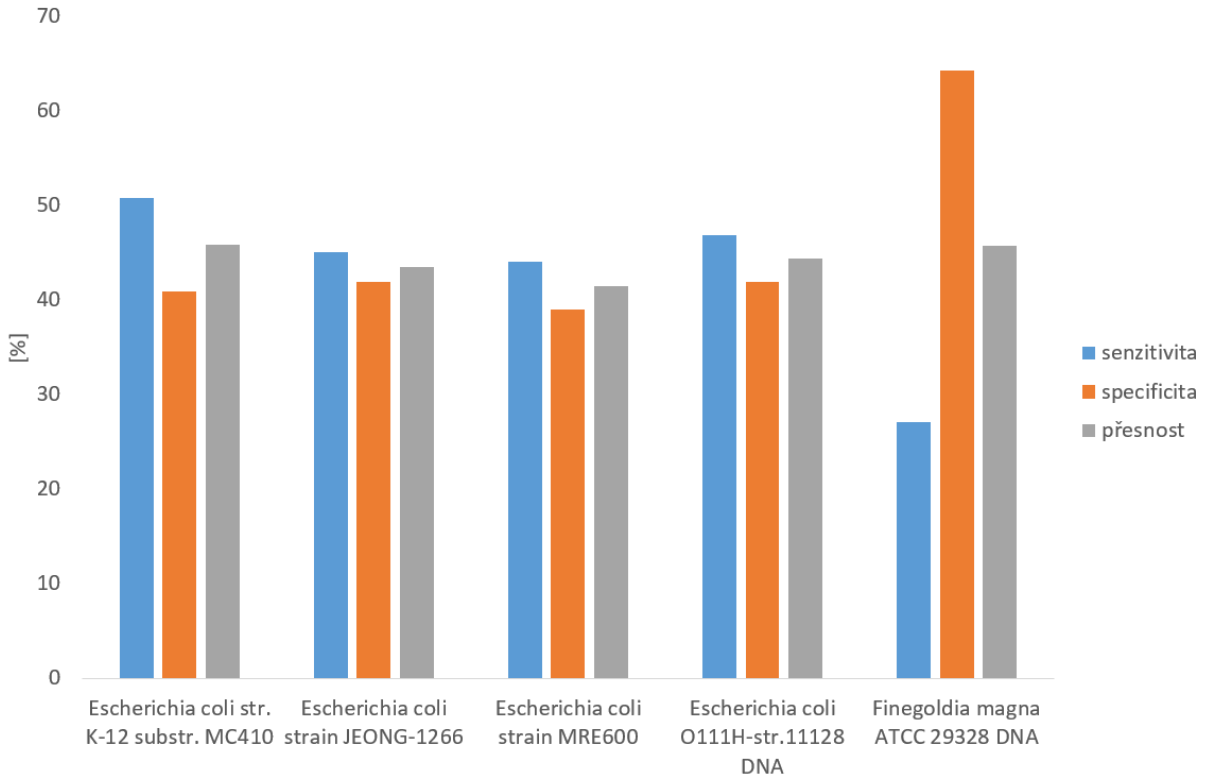
analyzovaný genom	počet genů	počet nalezených genů
Escherichia coli str. K-12 substr. MC4100	4026	4999
Escherichia coli strain JEONG-1266	5481	5896
Escherichia coli strain MRE600	4814	5429
Escherichia coli O111H-str.11128 DNA	5264	5886
Finegoldia magna ATCC 29328 DNA	1645	694

Vyhledávání bylo prováděno s tolerancí ±3, ±30, ±60, ±90 bází. Pro každý genom byly určeny následující hodnoty: počet celkem nalezených genů, počet správně nalezených genů, počet správně nalezených začátků a konců genů, senzitivita, specificita a přesnost. Na obr. 2.5, obr. 2.6, obr. 2.7 a obr. 2.8 jsou uvedeny grafy pro specificitu, senzitivitu a přesnost predikce pro jednotlivých genomy a to pro všechny tolerance. Kompletní získané výsledky jsou uvedeny v tab. B.1 a tab. B.2.

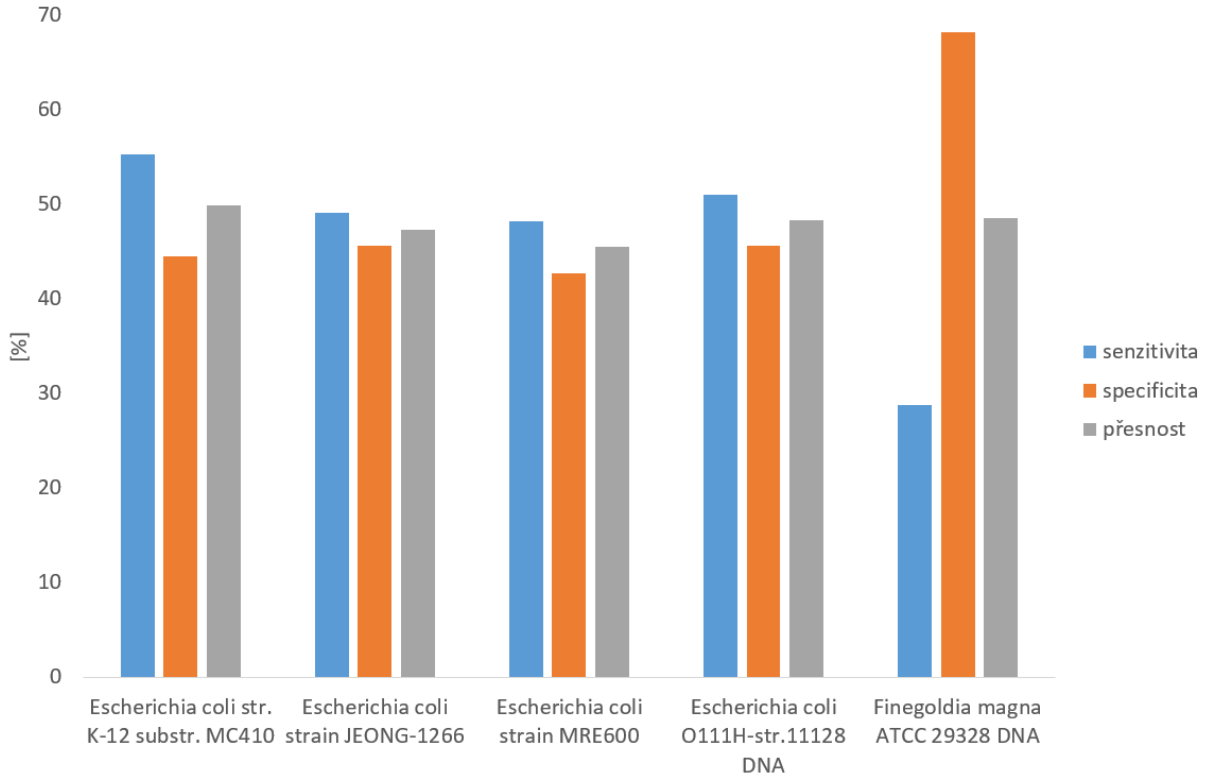
Nejlépe si program Gene_finder vedl u genomu *Escherichia coli* str. K-12 substr. MC410, kde dosáhl nejvyšší přesnosti. Z celkového počtu 4026 genů nalezl celkem 4999, přičemž správně identifikovaných jich bylo při nejvyšší toleranci 2438 a při nejnižší 2042. Přesnost predikce se tedy v tomto případě pohybuje od 46 % do 55 %.

U genomu *Escherichia coli* O111H-str.11128 DNA, na kterém byly testovány předešlé zmíněné programy, bylo identifikováno 5886 genů. Ve skutečnosti genom obsahuje 5264 genů. Správně určeno bylo při toleranci ±3 báze 2468 genů a při ±90 bází 2938 genů. Výsledná přesnost se v tomto případě pohybuje o 2 % níže než v předcházejícím případě.

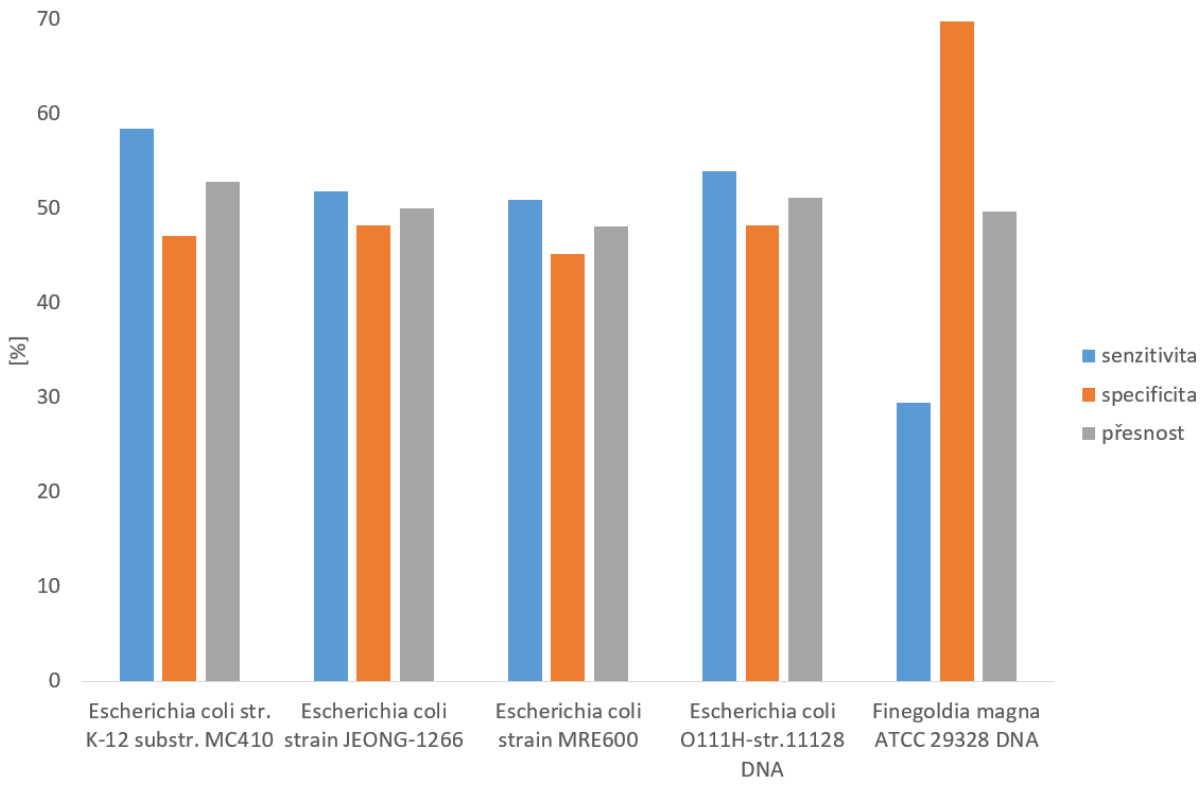
Nejhůře dopadlo testování *Escherichia coli* strain MRE600. Program nalezl celkem 5429 genů, z toho správně identifikovaných při nejnižší toleranci bylo pouhých 2117 genů. Přesnost zde dosáhla pouhých 41,5 %.



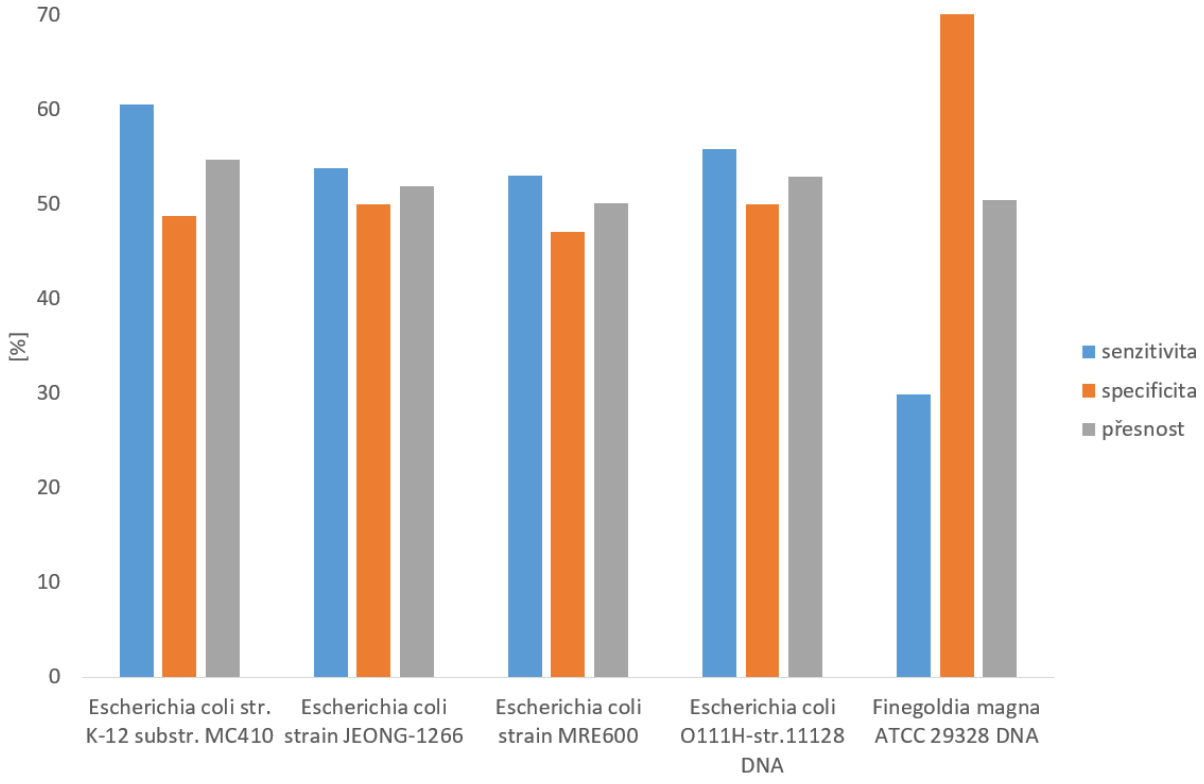
Obr. 2.5: Úspěšnost predikce pro ±3 báze.



Obr. 2.6: Úspěšnost predikce pro ±30 bází.



Obr. 2.7: Úspěšnost predikce pro ±60 bází.



Obr. 2.8: Úspěšnost predikce pro ±90 bází.

Z uvedených výsledků je patrné, že program najde větší množství genů než se ve skutečnosti v genomu objevuje. To může být zapříčeněno tím, že pokud jsme nenašli start kodon ATG, automaticky jsme začali hledat alternativní start kodony a neuvažovali jsme, že by se mohlo jednat o nekódující oblast. Dále z výsledků vidíme, že nachází správně skoro stejný počet začátků a konců genů. Ovšem tyto hodnoty jsou nižší než je celkový počet genů v organismu, a proto by bylo zapotřebí dosáhnout jejich zvýšení.

Tab. 2.4: Srovnání analyzovaných a vytvořeného softwaru.

analyzovaný software	GeneMark.hmm	Glimmer	Prodigal	Gene_finder
přesnost [%]	85,77	81,51	90,21	44,41

Jak vidíme v tab. 2.4, za pomoci základního námi vytvořeného programu lze dosáhnout 40 - 50 % úspěšnosti, z čehož ve srovnání s ostatními plyne, že pro lepší výsledky je potřeba využít mnohem pokročilejších algoritmů a metod, které přesahují obsah této práce.

3 ZÁVĚR

Náplní této bakalářské práce jsou metody pro vyhledávání genů v prokaryotický organismech s popisem tří volně dostupných softwarů, které predikci provádějí a dále vytvoření vlastního programu pro hledání genů v genomech.

V první polovině teoretické části práce jsou popsány prokaryotické organismy se zaměřením na jejich genom a průběh exprese genetické informace.

Ve druhé polovině jsme se již zaměřili na popis predikce genů v prokaryotických organismech s následným podrobným popisem tří volně dostupných softwarů a to těchto: GeneMark.hmm, Glimmer a Prodigal. Tyto softwary byly vybrány proto, že jejich kombinaci používá NCBI a společně dosahují dobrých výsledků. Z tohoto důvodu jsme chtěli otestovat, jak si povedou jednotlivě.

V další části jsou dostupné softwary popsány z hlediska praktického použití. Je zde uvedeno jejich ovládání, následně jsou zde popsány jejich výstupy, které jsou ve všech případech zobrazeny na obrázku. Po seznámení s programy je poté popsáno jejich testování na genomu *Escherichia coli* 0111:H- str 11128 DNA.

U každého programu byl určen celkový počet nalezených genů, dále počet správně identifikovaných genů a počet správně najitých začátků a konců. Pro uvedené hodnoty byla spočítána senzitivita, specificita a celková přesnost.

Všechny softwary pro predikci slibují nalezení až 95 % všech genů v genomu, avšak naše testování takovýchto hodnot nedosáhlo. Nejúspěšnější se ukázal software Prodigal, který správně identifikoval 90,21 % genů, program GeneMark.hmm nalezl 85,77 % genů a nejhůře dopadl software Glimmer, který byl schopný správně popsat 81,51 %.

Další část práce je zaměřena na program pro predikci genů Gene_finder, který byl vytvořen v prostředí MATLAB R2015a. Jsou zde popsány jednotlivé funkce programu, co mají za vstupy a výstupy a konstanty, se kterými pracují.

Dále je navržený program otestován na pěti genomech, z nichž jeden je stejný jako ten, co jsme používali u testování programů, takže lze následně porovnat výsledky navrženého algoritmu s dostupnými softwary.

Gene_finder dosahuje přesnosti predikce okolo 50 %, hodnoty senzitivity a specificity kolísají kromě jednoho případu okolo stejných hodnot. Pro přesnější predikci genů by bylo vhodné zpřesnit pozici start kodonů, například vyhledáváním motivů, které jim předcházejí, nebo uvažovat překryv genů i v rámci jednoho čtecího rámce. Oproti zmíněným volně dostupným softwarům má Gene_finder výhodou v tom, že jeho výstup se ukládá do Excel tabulky a tudíž nemusíme získaná data pracně ukládát a extrahovat jednotlivé hodnoty, jak je tomu u webových softwarů, ale můžeme s nimi okamžitě pracovat.

Na závěr lze říci, že vyhledávání genů je stále nevyřešený problém, jelikož zatím žádný software nedosahuje stoprocentní úspěšnosti, a proto si predikce genů do budoucna zaslouží naši pozornost.

LITERATURA

[1] NEČAS, Oldřich. *Obecná biologie pro lékařské fakulty.* 3. přeprac. vyd., V nakl. H. Jinočany: H, 2000, 554 s. ISBN 80-860-2246-3.

[2] CAMPBELL, Neil A a Jane B REECE. *Biologie.* Vyd. 1. Brno: Computer Press, c2006, xxxiv, 1332 s. ISBN 80-251-1178-4.

[3] RUIZ, Mariana a Michal MAŇAS. *Average prokaryote cell.* 2008. Dostupné z URL: <https://commons.wikimedia.org/wiki/File:Average_prokaryote_cell_cs.svg>.

[4] GRIFFITHS, Anthony J. F., William M. GELBART, Jeffrey H. MILLER a Richard C. LEWONTIN. *Modern genetic analysis* 2. print. New York: Freeman, 1999. ISBN 07-167-3118-5.

[5] BROWN, Terence A. Genomes. *Genomes.>* 2nd ed. New York: Wiley-Liss, c2002, xxvii, 572 p. ISBN 04-712-5046-5.

[6] SLÁDEK, Zbyšek. *Buněčná biologie.* Vyd. 1. V Brně: Mendelova zemědělská a lesnická univerzita, 2007, 106 s. ISBN 978-80-7375-086-2.

[7] Modern Genetics. *Wikispaces Search (APBio-Werle Wiki).* [online]. [cit. 2015-12-04]. Dostupné z URL: <https://apbio-werle.wikispaces.com/file/view/figure-08-01.JPG/68532827/560x303/figure-08-01.JPG>.

[8] ANGELOVA, Mihaela, Slobodan KALAJDZISKI a Ljupco KOCAREV. *Computational Methods for Gene Finding in Prokaryotes.* ICT Innovations [online]. 2010 [cit. 2015-12-03]. Dostupné z URL: <http://fac.ksu.edu.sa/sites/default/files/004635244f8d1c7267000000.pdf>.

[9] LUKASHIN, Alexander V. a Mark BORODOVSKY. *GeneMark.hmm: new solutions for gene finding.* Nucleic Acids Research [online]. 1998, 26(4): 1107-1115 [cit. 2015-11-06]. DOI: 10.1093/nar/26.4.1107. ISSN 13624962. Dostupné z URL: <http://nar.oxfordjournals.org/content/26/4/1107.full.pdf+html>.

[10] RŮŽEK, Václav. *Algoritmy pro rozpoznání ručně psaných znaků.* Zlín, 2010. Bakalářská práce. Univerzita Tomáše Bati ve Zlíně. Fakulta aplikované informatiky. Ústav řízení procesů. Vedoucí práce Ing. Petr Chalupa, Ph.D.

[11] SALZBERG, Steven L., Arthur L. DELCHER, Simon KASIF a Owen WHITE. *Microbial gene identification using interpolated Markov models.* Nucleic Acids

Research [online]. 1998, 26(2) [cit. 2015-12-03]. Dostupné z URL: <https://ccb.jhu.edu/papers/glimmer-nar.pdf>.

[12] DELCHER, Arthur L., Douglas HARMON, Simon KASIF, Owen WHITE a Steven L. SALZBERG. *Improved microbial gene identification with GLIMMER.* Nucleic Acids Research [online]. 1999, 27(23) [cit. 2015-12-03]. Dostupné z URL: <https://ccb.jhu.edu/papers/glimmer2.pdf>.

[13] DELCHER, Arthur L., Kirsten A. BRATKE, Edwin C. POWERS a Steven L. SALZBERG. *Identifying bacterial genes and endosymbiont DNA with Glimmer.* Bioinformatics [online]. 2007, 23(6) [cit. 2015-12-03]. Dostupné z URL: <https://ccb.jhu.edu/papers/glimmer3.pdf>.

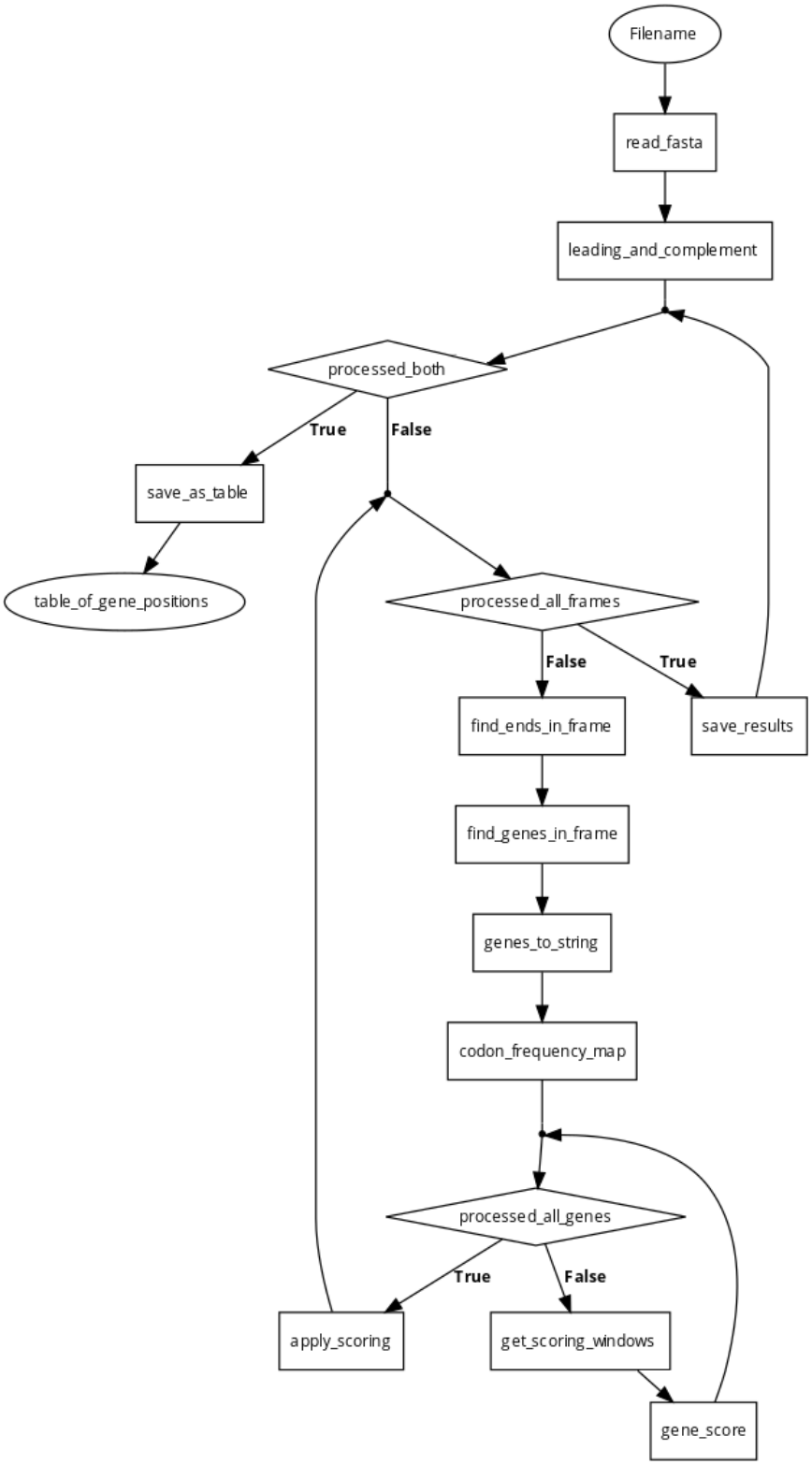
[14] HYATT, Doug, et al. *Prodigal: prokaryotic gene recognition and translation initiation site identification.* BMC Bioinformatics. 2010, 11(1):119. Dostupné z URL: <http://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-11-119>.

[15] Prodigal Gene Prediction Results and Comparison with Genbank. *Prodigal: prokaryotic gene recognition and translation initiation site identification.* Prodigal: Microbial Gene Prediction Software [online]. [cit. 2016-05-18] Dostupné z URL: <http://compbio.ornl.gov/prodigal/results.php>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

A	adenin
bp	páru bazí
C	cytosin
DNA	deoxyribonukleová kyselina
DP	dynamické programování
G	guanin
GTP	guanosintrifosfát
HMM	skrytý Markovův model
ICM	interpolovaný kontext model
IF	iniciační faktor
IMM	interpolovaný Markovův model
mRNA	mediátorová ribonukleová kyselina
NCBI	National Center for Biotechnology Information
ORF	otevřený čtecí rámec
RNA	ribonukleová kyselina
tRNA	transferová ribonukleová kyselina
T	thymin
U	uracyl

A VÝVOJOVÝ DIAGRAM PROGRAMU



B VÝSLEDKY TESTOVÁNÍ PROGRAMU

Tab. B.1: Výsledky predikce programu pro jednotlivé genomy.

Escherichia coli str. K-12 substr. MC410	tolerance	±3	±30	±60	±90
	počet správně nalezených genů	2042	2223	2352	2438
	počet správně nalezených začátků	2300	2391	2465	2525
	počet správně nalezených konců	2325	2425	2494	2537
Escherichia coli strain JEONG-1266	tolerance	±3	±30	±60	±90
	počet správně nalezených genů	2471	2687	2839	2949
	počet správně nalezených začátků	2814	2939	3030	3099
	počet správně nalezených konců	2843	2953	3037	3100
Escherichia coli strain MRE600	tolerance	±3	±30	±60	±90
	počet správně nalezených genů	2117	2318	2450	2554
	počet správně nalezených začátků	2477	2596	2680	2745
	počet správně nalezených konců	2502	2605	2672	2740
Escherichia coli O111H-str.11128 DNA	tolerance	±3	±30	±60	±90
	počet správně nalezených genů	2468	2684	2838	2938
	počet správně nalezených začátků	2813	2922	3005	3070
	počet správně nalezených konců	2789	2912	3004	3073
Finegoldia magna ATCC 29328 DNA	tolerance	±3	±30	±60	±90
	počet správně nalezených genů	446	473	484	492
	počet správně nalezených začátků	484	498	500	504
	počet správně nalezených konců	469	483	494	498

Tab. B.2: Úspěšnost programu pro jednotlivé genomy.

Escherichia coli str. K-12 substr. MC410	tolerance	±3	±30	±60	±90
	senzitivita [%]	50,72	55,22	58,42	60,56
	specificita [%]	40,85	44,47	47,05	48,77
	přesnost [%]	45,79	49,85	52,74	54,67
Escherichia coli strain JEONG-1266	tolerance	±3	±30	±60	±90
	senzitivita [%]	45,08	49,02	51,80	53,80
	specificita [%]	41,91	45,57	48,15	50,02
	přesnost [%]	43,50	47,30	49,98	51,91
Escherichia coli strain MRE600	tolerance	±3	±30	±60	±90
	senzitivita [%]	43,98	48,15	50,89	53,05
	specificita [%]	38,99	42,70	45,13	47,04
	přesnost [%]	41,49	45,43	48,01	50,05
Escherichia coli O111H-str.11128 DNA	tolerance	±3	±30	±60	±90
	senzitivita [%]	46,88	50,99	53,91	55,81
	specificita [%]	41,93	45,60	48,22	49,92
	přesnost [%]	44,41	48,30	51,07	52,87
Finegoldia magna ATCC 29328 DNA	tolerance	±3	±30	±60	±90
	senzitivita [%]	27,11	28,75	29,42	29,91
	specificita [%]	64,27	68,16	69,74	70,89
	přesnost [%]	45,69	48,46	49,58	50,40

C OBSAH PŘILOŽENÉHO CD

- **tex/** - složka se zdrojovým kódem práce pro systém LaTeX
- **matlab/** - zdrojový kód programu `Gene_finder` a skript pro testování účinnosti
- **tests/** - složka s testovanými genomy ve FASTA formátu