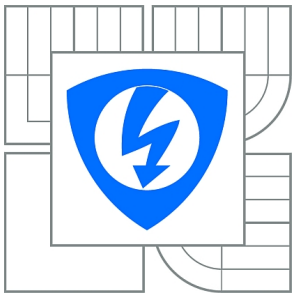
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF BIOMEDICAL ENGINEERING

VYHLEDÁVÁNÍ EXONŮ POMOCÍ FOURIEROVY TRANSFORMACE

FOURIER TRANSFORMATION FOR EXON PREDICTION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

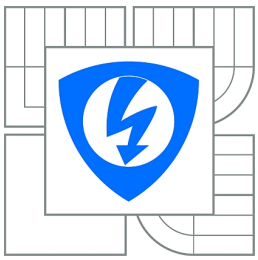
AUTOR PRÁCE — MICHAL RUSINA

AUTHOR

VEDOUCÍ PRÁCE — Ing. DENISA MADĚRÁNKOVÁ

SUPERVISOR

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav biomedicínského inženýrství

Bakalářská práce

bakalářský studijní obor
Biomedicínská technika a bioinformatika

Student:	Michal Rusina	***ID:***	133980
Ročník:	3	***Akademický rok:***	2012/2013

NÁZEV TÉMATU:

Vyhledávání exonů pomocí Fourierovy transformace

POKYNY PRO VYPRACOVÁNÍ:

1) Zpracujte literární rešerši na téma vyhledávání kódujících úseků v genomech prokaryotických i eukaryotických organismů. 2) Vytvořte přehledný seznam dostupných metod s jejich výhodami, nevýhodami a možnostmi použití. 3) Popište a vyzkoušejte alespoň dva volně dostupné nástroje pro vyhledávání exonů na souboru sekvencí. 4) Navrhněte pseudokód a vývojový diagram funkce pro vyhledávání exonů pomocí diskrétní Fourierovy transformace s volitelnou délkou okna a překryvem okna. 5) Navrženou funkci implementujte v libovolném programovém prostředí. 6) Funkci vyzkoušejte na souboru sekvencí a výsledky porovnejte s výsledky z volně dostupných nástrojů.

DOPORUČENÁ LITERATURA:

[1] WANG, Zhuo, CHEN, Yazhu a LI, Yixue. A Brief Review of Computational Gene Prediction Methods. Geno. Prot. Bioinfo., 2004, roč. 2(4), s. 216-221.
[2] MATHÉ, Catherine; SAGOT, Marie-France; SCHIEX, Thomas a ROUZÉ, Pierre. Current methods of gene prediction, their strengths and weaknesses. Nucleic Acids Research, 2002, roč. 30(19), s. 4103-4117.

Termín zadání:	11.2.2013	***Termín odevzdání:***	31.5.2013

Vedoucí práce: Ing. Denisa Maděránková
Konzultanti bakalářské práce:

prof. Ing. Ivo Provazník, Ph.D.
Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt v českém jazyce

V této bakalářské práci jsou popsány metody predikce exonů. První část práce je zaměřena na rozdíl mezi prokaryotickými a eukaryotickými organismy, popis struktury DNA a vysvětlení pojmů exon a intron. Druhý úsek teoretické části obsahuje čtyři metody predikce exonů a to dynamické programování, neuronové sítě, skryté Markovovy modely a diskrétní Fourierovu transformaci. V praktické části byl vytvořen program Predikce_exonu, který vyhledává exony v nukleotidových sekvencích a pracuje na principu Fourierovy transformace. Tento algoritmus spolu s třemi volně dostupnými programy byl testován na 25 sekvencích a úspěšnost jejich predikce byla popsána senzitivitou a specificitou.

Klíčová slova v českém jazyce

DNA, exon, intron, transkripce, ab initio predikce, hledání podobností, Fourierova transformace

Abstract in English language

In this bachelor thesis there are described the methods of the prediction of exon. The first part is aimed at the difference between the prokaryotic and eukaryotic organisms, the description of the DNA structure and the explanation of the terms exon and intron. The second section of the theoretic part includes four methods of the prediction of exons namely dynamic programming, neural networks, hidden Markov models and discrete Fourier transform. In the practical part there was created the program called Predikce_exonu that searches exons in nucleotide sequences and works on the principle of the Fourier transform. This algorithm together with 3 freely available programs was tested on 25 sequences and the success of their prediction was described by sensitivity and specificity.

Key words in English langugage

DNA, exon, intron, transcription, ab initio prediction, similarity searches, Fourier transformation

BIBLIOGRAFICKÁ CITACE

RUSINA, M. *Vyhledávání exonů pomocí Fourierovy transformace.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 59 s. Vedoucí bakalářské práce Ing. Denisa Maděránková.

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma „Vyhledávání exonů pomocí Fourierovy transformace“ jsem vypracoval 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 autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009Sb.

V Brně dne ……..

podpis autora

Poděkování

Děkuji vedoucí bakalářské práce Ing. Denise Maděránkové za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady pří zpracování mé bakalářské práce. Dále bych chtěl poděkovat celé mé rodině za podporu během studia a psaní bakalářské práce.

V Brně dne .....

podpis autora

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

Tato bakalářská práce se zabývá metodami predikce exonů. Exony, jakožto kódující úseky DNA jsou zachovány ve zralých molekulách mRNA a mají hlavní význam při syntéze jednotlivých proteinů. Právě tyto části jsou později exprimovány translací do aminokyselinové struktury. Tedy určením pozic a z nich nukleotidové složení daných exonů, nalezneme geny a můžeme určovat jejich funkce. Pojmem gen budeme označovat jen takovou část sekvence, která kóduje bílkoviny. [1]

S významným pokrokem v oblasti sekvenování DNA narůstá množství dat (sekvencí) v databázích. V dnešní době existuje mnoho programů zabývající se sekvenováním DNA různých organismů, avšak biologická interpretace nedrží krok s rychlostí sekvenování. To vede ke vzniku velkého množství surových dat. Z těchto důvodů se výpočetní predikce stává zásadní pro automatickou analýzu a popis velkého množství necharakterizovaných genomických sekvencí. [2], [3]

První oddíl teoretické části je věnován rozboru problematiky související se základními poznatky v této oblasti. Jsou zde vysvětleny hlavní rozdíly mezi eukaryotickými a prokaryotickými organismy. Další kapitola obsahuje popis struktury DNA, RNA a rozdíly v jejich struktuře vztažené k jednotlivým organismům. V následujících kapitolách je vysvětlen proces transkripce a translace a rozdílný průběh těchto procesů u eukaryotických a prokaryotických organismů, včetně vysvětlení pojmů exonů a intronů, které patří mezi zásadní odlišnosti v DNA těchto dvou organismů.

Druhý oddíl teoretické části obsahuje stručné vysvětlení dvou přístupů k predikci, predikce založena na vyhledávání podobnosti sekvence a ab initio přístup. Dále se zde nachází popis čtyř metod používaných pro predikci exonů a to dynamické programování, neuronové sítě, skryté Markovovy modely a diskrétní Fourierova transformace.

V praktické části byl vytvořen program Predikce_exonu, který využívá k detekci exonu metodu Fourierovy transformace. Byla provedena analýza, za účelem zjistit optimální hodnoty délky okna a prahu a s takto zjištěnými vstupními hodnotami byl program testován na 25 sekvencích.

Druhý oddíl praktické části je zaměřen na vyzkoušení tří volně dostupných programů pro predikci kódujících úseků, z nichž jeden, GeneID, využívá metodu dynamického programování a dva, GeneMark.hmm a FGENESH, metodu skrytých Markovových modelů. Tyto programy byly testovány na stejné sadě sekvencí jako program Predikce_exonu, byla u nich vypočtena senzitivita a specificita a následně byla porovnána úspěšnost jednotlivých programů.

2 Teoretická část

2.1 Eukaryotická a prokaryotická buňka

Buňka je základní života schopná jednotka. V buňkách, nebo jejich prostřednictvím probíhají všechny základní životní procesy. Tyto procesy jsou prováděny přímo v nich nebo jsou uskutečňovány interakcemi mezi nimi. Buňku lze považovat za individuum, základní strukturní jednotku. Neznamená to, že by se dále nedala dělit na menší podstruktury, avšak ty nejsou schopny samostatného života. Hlavní úkoly buňky se rozdělují na dva:

1. Zachování její existence.
2. Reprodukce.

Všechny procesy v buňce sledují tyto dva cíle. A porušení těchto úkolů by vedlo k jejímu zániku. Veškeré výzkumy vedly k zjištění, že existují pouze dva typy buněk prokaryotické a eukaryotické. [4]

2.1.1 Prokaryotická buňka

Jedná se o buňku fylogeneticky starší a rozměrově menší v průměru kolem 1 až 5 μm, avšak jsou známy i výjimky, které dosahovaly mnohem větší velikosti. Kolem většiny prokaryot se nachází buněčná stěna, která poskytuje ochranu. Tato stěna však není tvořena celulózou, jak tomu je u buněčné stěny rostlin, ale tvoří jí speciální látka peptydoglykan. Její struktura je jednodušší. [1]

Jádro, nazývané nukleoid, není odděleno jadernou membránou, ale tvoří jej pouze volně uložený chromozom, který se nachází přímo v cytoplazmě. Tyto buňky také neobsahují žádné organely, které by měly specializované funkce. Většina zástupců tohoto druhu je jednobuněčných, ale objevují se i druhy shlukující se do takzvaných kolonií, ve kterých můžeme pozorovat rozdělení pracovních rolí. [1], [4]

2.1.2 Eukaryotická buňka

Jedná se o fylogeneticky mladší buňku, dosahující větší velikosti a to 10-100 μm. Funkcí plazmatické membrány u eukaryot je jak oddělení celé buňky od vnějšího prostředí, tak také rozděluje vnitřní část buňky na různé oblasti tím, že ohraničuje jednotlivé organely. Z toho plyne, že obsahuje specializované organely. Například mitochondrie, endoplazmatické retikulum, chloroplasty a další. [1], [4]

Jádro eukaryotické buňky je odděleno od cytoplasmy jaderným obalem, což je dvojitá membrána. DNA je uspořádaná do vláknitého materiálu, takzvaného chromatinu. Ten je

tvořen oddělenými strukturami chromozomy, kterých je v jádře většinou více. Dalším popis DNA se nachází v následujících kapitolách. Přehlednější rozdíl mezi eukaryotickou a prokaryotickou buňkou je uveden v Tabulce 1. [1], [4]

Tabulka 1 Rozdíly mezi eukaryotickou a prokaryotickou buňkou. [4]

	Prokaryotická buňka	**Eukaryotická buňka**
Vnitřní prostor	Nedělený	Rozdělený membránou
Organely	Nejsou (ojedinělé)	Jsou (jádro, mitochondrie atd.)
Buněčné jádro	Neoddělené	Oddělené dvojitou membránou
Velikost	Menší	Větší
Genetická informace	10^6-10^7 bp2	10^8-10^{10} bp2
Počet chromozomů	Většinou jeden	Více

2.2 Struktura DNA a RNA

Oba typy buněk mají svou genetickou informaci uloženou v DNA. Jedná se o kyselinu deoxyribonukleovou. Jejími monomery jsou nukleotidy, které jsou tvořeny spojením pentózy, organické dusíkaté báze a kyseliny fosforečné. Nukleotidy se mohou kovalentně spojovat, takzvanými fosfodiesterovými vazbami, mezi cukrem jednoho nukleotidu a fosfátem druhého. Tímto spojením vzniká z nukleotidů polynukleotid. [4]

U nukleotidů se rozlišují dva typy pentóz a to: ribóza v ribonukleových kyselinách a důležitější deoxyribóza, která je součástí kyseliny deoxyribonukleové tedy DNA. Existují také dvě skupiny dusíkatých bází, kterými jsou pyrimidinové a purinové. Jako pyrimidinové se označují cytosin (C), tymin (T) a uracil (U). Purinové jsou adenin (A) a guanin (G). Pravidlem je, že tymin se nachází jen v DNA a uracil jen v RNA. [1], [4]

Zásadním rozdílem mezi DNA a RNA je v tom, že DNA je tvořena dvěma polynukleotidovými vlákny, která probíhají vedle sebe a navzájem jsou spojeny vodíkovými můstky mezi jednotlivými bázemi a vzniká vždy pár cytosin-guanin, nebo adenin-tymin. Tedy je-li na jednom vlákně přítomen guanin, na druhém vlákně naproti němu bude cytosin. Toto pravidlo se nazývá párování bází. Z tohoto řádu vyplývá, že obě vlákna DNA jsou odhadnutelným doplňkem druhého. Proto každé ze dvou vláken může být využito jako vzor pro uspořádání a vytvoření nové dvoušroubovice DNA. Důležité je také poznamenat, že mezi adeninem a tyminem vznikají 2 vodíkové můstky, zatímco mezi guaninem a cytosinem vznikají 3 vodíkové můstky. [1]

U DNA se rozlišují dva typy struktur. Primární struktura je dána zastoupením různých nukleotidů v řetězci, tudíž jejich sekvencí. Sekundární struktura vychází z objevu Watsona a Cricka. Ti zjistili, že oba řetězce DNA, které jsou propojeny vodíkovými můstky, jsou stočeny do šroubovice. Nejčastěji se vyskytuje dvoušroubovice pravotočivá. Je známa i levotočivá. Ta se dá pozorovat na úsecích tvořených opakujícími se puriny a pyrimidiny. V tomto případě se jedná o Z-formu. Sekundární strukturu se dá jednoduše rozrušit, takzvanou denaturaci, kdy dojde k oddělení řetězců a to buď částečně, nebo úplně. Denaturace může být způsobena například působením močoviny, zvýšením teploty atd. [1], [5]

2.2.1 DNA typická pro jednotlivé druhy

V předešlé kapitole byla ve zkratce popsána DNA, dále by bylo vhodné uvést rozdíly uložení genetické informace u prokaryotických a eukaryotický buněk. V obou typech organismů jsou geny uloženy převážně v takzvaných chromozomech. Důležité je definovat co se myslí pod pojmem genom. Slovem genom se označuje soubor všech molekul DNA v buňce tedy jak DNA genová, tak i negenová. [4]

U prokaryot je genom tvořen jedním prokaryontním chromozomem a menší množství DNA se také nachází v plazmidech. Prokaryontní chromozom je nejdůležitější část prokaryotického genomu, jedná se o kružnicovou molekulu DNA. Jeho součástí můžou být i proteiny podobné histonů a bývá daným místem připojen k plazmatické membráně. U některých prokaryotických buněk může mít velikost až 4,6 milionu nukleotidových páru. To by odpovídalo rozměrům až 1 mm. Avšak DNA je v uvnitř buňky sbalená a zabírá jen malý úsek. Tento chromozom slouží k binárnímu dělení. Jedná se o metodu, kterou se některé prokaryotické buňky rozmnožují. Tomuto ději předchází replikace DNA. Syntéza probíhá v obou směrech cirkulární DNA od jednoho místa zvaného počátek replikace. Například některé bakterie se ve vhodném prostředí můžou velice rychle množit, E. colli každých 20 minut. [1], [4]

Téměř všechny geny eukaryotních buněk jsou uloženy v chromosomech. Každý jeden chromozom je tvořen jednou molekulou DNA. Určitá část genů je uložena i v mimo jaderných oblastech a to v mitochondriích nebo chloroplastech. Eukaryotický chromozom je značně složitější než chromozom prokaryotický, obsahuje totiž velké množství proteinů a je také mnohem delší. Každý z těchto chromozomů může dosahovat délky až 100 miliónů nukleotidových páru, rozvinutý by tak měřil až 6 cm. Například u člověka se do jádra musí vejít všech 46 chromozomů. Tato nutnost je zabezpečena díky propracovanému, víceúrovňovému systému sbalování DNA. [1], [6], [7]

2.3 Transkripce a translace

Aby byla genetická informace, která je uložena v DNA, uplatněna, musí dojít ke dvěma dějům, transkripci a translaci. Transkripce neboli přepis je první z nich. Jedná se o syntézu RNA podle řetězce DNA. Jelikož i RNA je tvořena sekvencí nukleotidů, dochází pouze k sestavení pořadí nukleotidů podle DNA. DNA tedy tvoří takzvaný templát, podle kterého se vytvoří RNA. Avšak tyto dva řetězce nebudou úplně shodné. RNA se od DNA liší ve dvou bodech:

1. Cukernatou složkou v RNA je ribóza (Z toho plyne název ribonukleová kyselina).
2. Změna v jedné bázi, kdy místo tyminu je přítomen uracil.

Ale zásadním rozdílem je, že RNA netvoří dvojšroubovici. [1], [5]

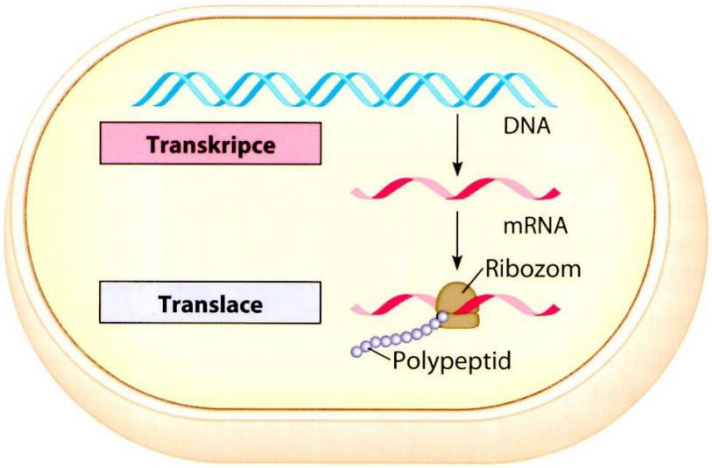
Pomocí transkripčního procesu vzniká všechna RNA v buňce. Transkripce začíná rozvolněním úseku DNA. Tento děj vykonává enzym RNA polymeráza, a následně připojuje RNA nukleotidy podle komplementarity bází k DNA templátu. Výsledkem je vlákno RNA, které má stejnou sekvenci jako komplementární vlákno k danému templátu DNA. Aby tento enzym poznal, kde má začít, respektive skončit s transkripcí, nacházejí se v sekvenci DNA úseky promotor a terminátor. Další důležitou poznámkou je, že výsledná RNA nezůstává připojena na templátovou DNA, ale po přidání jednoho ribonukleotidu dochází k okamžitému obnovení spojení dvojšroubovice. [1], [5]

V buňce se rozlišuje několik typu RNA. Největší množství vznikající RNA je takzvaná mediátorová RNA (mRNA), jejímž úkolem je řídit následný vznik proteinů. Dále existují RNA, které mají strukturní a enzymatickou funkci. Ribosomální RNA (rRNA), která tvoří základ ribozomů. Transferová RNA (tRNA), která vybírá správné aminokyseliny, podle sekvence mRNA a začleňuje je do aminokyselinového řetězce. [5]

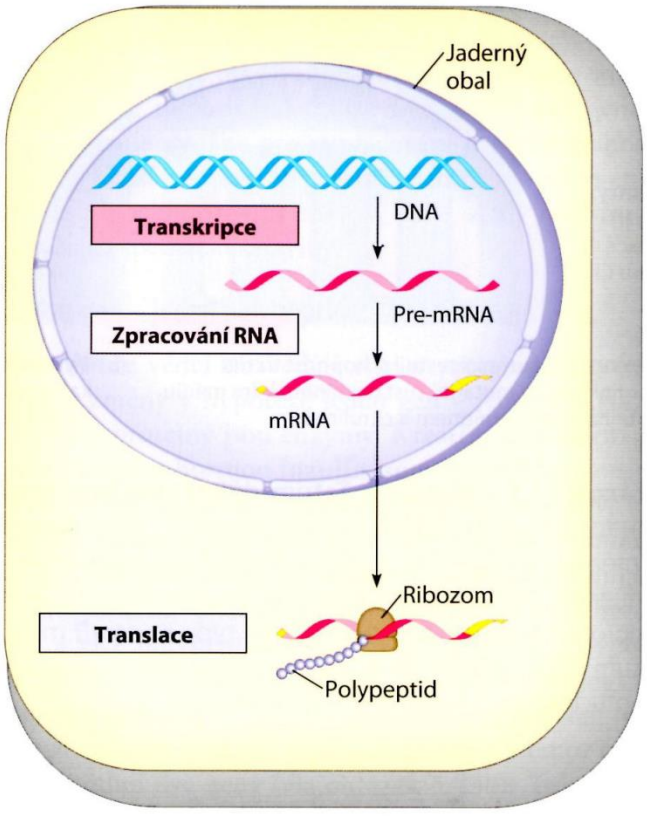
Když se vytvoří mRNA, může se přejít k samotné syntéze proteinu. Tento proces se nazývá translace tedy překlad. Během tohoto procesu dojde ke změně „jazyku“, kdy je potřeba přeložit sekvenci bází v mRNA do řetězce aminokyselin v proteinu. V genetickém kódu je vždy daná aminokyselina v řetězci kódována tripletem nukleotidů. K tomuto ději dochází na ribozomech. [1]

Ačkoli by se podle tohoto popisu mohlo zdát, že nebude existovat rozdíl mezi transkripcí prokaryotické a eukaryotické buňky není tomu tak. Jak již bylo uvedeno výše, prokaryotické buňky postrádají jádro, to znamená, že jejich DNA není nijak oddělena od ribozomů, na kterých bude probíhat syntéze proteinů. V případě objevení volného 5´ konce, nasedá na něj ribozom a dochází k proteosyntéze. U eukaryotické buňky tomu je jinak. DNA je u nich uzavřená v jádře, proto musí k transkripci dojít tam. A až poté je transportována malými póry

v jaderné membráně do cytoplasmy. Vědci však zjistili, že dochází k rozsáhlému odstranění úseků RNA, která byla původně syntetizována. Jedná se o takzvaný sestřih RNA. [1], [5]



Obrázek 1 Translace a transkripce prokaryotické buňky. [1]



Obrázek 2 Translace a transkripce eukaryotické buňky. [1]

2.4 Exony a introny

Pro další popis rozdílu si je nutné definovat, co znamenají pojmy exony a introny. Introny jsou dlouhé úseky DNA eukaryotických buněk, které nenesou genetickou informaci, nazývají se nekódujícími oblastmi. Tyto oblasti přerušují kódující úseky exony. Důležitým poznatkem bylo zjištění, že introny jsou oblasti, nepřekládající se do proteinu, ale v průběhu tvorby mRNA se vystříhávají. Tento proces se nazývá splicing. V dnešní době je zjištěno jen minimum informací o funkci intronů. Jednou z úloh intronů může být regulace exprese genů a to z hlediska času, díky tomu, že velikost intronů je velice rozmanitá, ovlivňují nám dobu transkripce. [6]

V této oblasti se nachází zásadní rozdíl mezi eukaryotickými a prokaryotickými organismy, jelikož prokaryota neobsahují introny. Jejich geny nejsou složené, ale jsou tvořeny jen exony. Výhodou je, že genom je kratší a dochází tak k urychlení tvorby bílkovin. [4]

2.5 Metody vyhledávání exonů

Díky vzniku programu Human Genome Program (v roce 1990) bylo mnoho organismů sekvenováno a to jak v říši rostlin, tak v říši zvířat. V současnosti je spuštěno více než 50 projektů pro sekvenování eukaryotických genomů to je důvodem, že se databáze sekvenované DNA rychle zvětšuje. Avšak biologická interpretace nedrží krok s rychlostí získávání surových sekvencí. Z tohoto důvodu existuje velké množství nepopsaných dat. Výpočetní genová predikce se stává zásadním nástrojem pro analýzu DNA a slouží k popsání velkého množství necharakterizovaných genomických sekvencí. V posledních 20 letech vzniklo mnoho programů zabývajících se předpovědí genů. [2], [3]

Z rozdílu mezi obsahem eukaryotického a prokaryotického genomu, uvedeného výše, je zřejmé, že bude přítomná odlišnost také při hledání jednotlivých genů.

2.5.1 V genomech prokaryotických buněk

Vyhledávání genů v prokaryotickém genomu je mnohem jednodušší, a to především kvůli vyšší hustotě genů a absencí intronů v oblastech kódujících proteiny. DNA sekvence je transkribovaná do mRNA a ta se překládá do bílkoviny bez významných změn. Výsledkem tedy je, že nejdelší ORF (open reading frames) začínající od prvního START kodonu až k následujícímu STOP kodonu funguje dobře, ale není zajištěno vyhledávání pouze kódujících oblastí pro bílkoviny. Několik metod používá různé typy Markovových modelů, aby bylo možno zachytit rozdíly mezi kódujícími oblastmi, „stínovými“ kódujícími oblastmi a nekódujícími oblastmi. Markovovy modely budou popsány níže. Ve výsledku, jsou programy využívající tyto metody schopny identifikovat většinu oblastí kódujících bílkoviny, s vysokou úspěšností. [2]

2.5.2 V genomech eukaryotických buněk

Vyhledávání kódujících oblastí v eukaryotických genomech se potýká s docela jiným problémem. Transkripce je sice zahájena specifickou sekvencí promotorů, ale poté musí následovat oddělení nekódujících úseků, intronů, z pre-mRNA, tento proces se nazývá sestřih (splicing). Po tomto kroku nám zůstane jen oblast exonů, tedy oblast kódující proteiny. Když jsou introny odstraněny, výsledná mRNA muže být přeložena do proteinu. Z těchto poznatků tedy vyplývá, že ORF musí být přerušeny přítomností intronů. [2]

Existují dvě hlavní skupiny metod výpočetní predikce genů. První z nich je metoda podobnosti, která se někdy označuje jako „vnější přístup". Druhá skupina je pojmenována jako „vnitřní přístup" a jedná se o takzvané „ab initio" metody predikce. [2], [3]

2.6 Hledání podobností sekvence

Jedná se o poměrně jednoduchý přístup založený na vyhledávání dostatečné podobnosti mezi oblastmi analyzované genomové sekvence a bílkovinnými nebo DNA sekvencemi přítomnými v databázi. Důvodem tohoto porovnání je zjistit, zda se jedná o oblast kódující, nebo zda se daná oblast překládá do bílkoviny. Z těchto metod mohou podobnosti s třemi různými typy sekvencí, poskytnout informaci o exonech (intronech). [2], [3]

První a nejpoužívanější jsou bílkovinné sekvence. Odhaduje se, že na základě dostatečné podobnosti s danou proteinovou sekvencí může být identifikováno až 50 % genů. Avšak tento postup má i svou nevýhodu. I když se nalezne dostatečné shoda, sestavení přesné struktury genu může být stále složité. [3]

Druhým typem je porovnávání a hledání podobností s EST (expressed sequence tags). EST jsou krátké sub-sekvence DNA, které jsou generovány sekvenováním exprimovaného genu. ESTy podávají informaci umožňující identifikovat části exonů. Nevýhodou tedy je, že jsou získány pouze omezené informace o genové struktuře, protože EST odrážejí jen části DNA. [3], [8]

Ve všech případech hlavní nevýhoda metod hledání podobností spočívá v tom, že shoda nebude nalezena, pokud databáze neobsahuje dostatečně podobnou sekvenci. Také může být problém v dané databázi, kdy nemusí obsahovat dostatečně kvalitní informace. [3]

Důležitá výhoda všech těchto strategií je, že předpovědi jsou založeny na tom, co je již známe a tedy v případě dobré kvality databází se získává spolehlivá predikce, i když jen částečná. [3]

2.7 Ab initio

Jedná se o souhrnné označení další velké skupiny metod predikce. Slovní spojení ab initio se překládá jako od začátku. Jedná se tedy o metody predikce, které pracují od začátku, to znamená s primární strukturou sekvenované DNA. [2]

2.7.1 Metoda dynamického programování (DP)

Dynamické programování řeší problém přesného identifikování vnitřních exonů a intronů. Tato metoda vychází se znalostí, které charakterizují danou sekvenci. Například použití kodónů v sekvenci, délkovou distribucí, pravidelnou asymetrii a frekvencí šestic. [9]

Kterékoli sekvence genomové DNA délky *N*, může produkovat dvě poloviční matice, které se označí jako L_E a L_I, ve kterých každá subsekvence začíná na pozici *i* a končí na pozici *j*, získá se tedy odpovídající matice L_E *(i,j)* a L_I*(i,j),* které představují logaritmickou pravděpodobnost, že je subsekvence exon nebo intron. [9]

Když je získána taková to matici, lze použít dynamické programování, k nalezení optimálních vzorů pro sestřih na základě důkazů z těchto matic. Pomocí dynamického programování se prosazuje omezení, že introny a exony se střídají v premRNA a jsou vedle sebe. Výsledkem DP jsou dva vektory, označené jako D_E a D_I, jejichž každý prvek obsahuje skóre pro nejlepší možnou kombinaci intronů a exonů končících na pozici *j*. Tyto skoré se dají vypočítat pomocí rovnice:

$$D_E(j) = max\left\{max_{k:2\rightarrow j-m}^{L_E(1,j)}[L_E(k,j) + D_I(k-1)]\right.. \quad (1)$$ [9]

D_I lze vypočítat obdobným způsobem. Existují jen tři možné výsledky této rovnice:

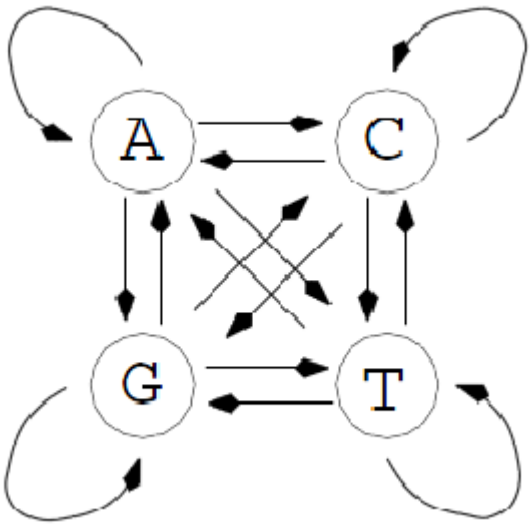
1. $D_E(j) = L_E(1,j)$
   Tento výsledek říká, že segment od 1 do *j* přesně definuje exon.
2. $\mathrm{D_E(j) = max_{k:2\rightarrow j-m}[L_E(k,j) + D_I(k-1)]}$
   Segment od 1 do *j* končí v exonu. D_E *(j)* pak udává skóre pro nejlepší kombinaci exonů, L_E *(k,j),* a nejlepší kombinaci konců intronů na pozici *k-1*, což udává D_I *(k-1)*, které se určuje dříve. D_I *(k-1)* může nabývat i nulových hodnot, v případě, že žádné introny nepředcházejí exonům na pozici *k.*
3. $D_E(j) = 0$
   Jedná se o poslední možnou hodnotu, jaká pomocí DP může vyjít. V tomto případě jsou obě předchozí možnosti negativní. Neexistuje tedy kombinace sekvencí s exonem končícím na pozici *j*. [9]

Shrnutím výše uvedených pravidel vede k zjištění, že D_E *(j)* udává skóre pro nejlepší vnitřní uspořádání intronů a exonů s exonem končícím na pozici *j*. [9]

2.7.2 Metoda skrytých Markovových modelů (HMM)

Tato metoda patří do skupiny metod souhrnně označovaných jako vnitřní obsahové čidla. Původně byly tyto metody definovány především pro prokaryotické genomy, ve kterých jsou přítomny pouze dva typy regionů a to regiony kódující protein a mezigenetické regiony. U prokaryot geny definují nepřerušený kódující úsek, který nesmí obsahovat STOP kodón. Zde se tedy nabízí jednoduchý přístup, který by vedl k nalezení potenciálních kódujícího úseku a to použití dostatečně dlouhého úseku otevřeného čtecího rámce, který by neobsahoval STOP kodóny. To by tedy znamenalo objevení sekvence mezi START a STOP kodóny. To však neleze použít pro eukaryotický genom, kde překládající se úseky, exony, jsou krátké a absence STOP kodónů ztrácí smysl. Z tohoto důvodu je nutné zavést další opatření, které se snaží blíže charakterizovat fakt, že sekvence je kódující pro protein: použití kodónu, frekvenci šestic (to znamená použít slovo délky šesti nukleotidů). Bylo zjištěno, že právě použití frekvence šestic je nejvíce diskriminující mezi kódujícími a nekódujícími úseky. [3]

Před definováním HMM je nutné definovat několik dalších pojmů a to především Markovův řetězec. Úkolem tohoto řetězce je vytvořit pravděpodobností model dané sekvence. Tento model vychází z předpokladu, že pravděpodobnost výskytu jednoho nukleotidu závisí na pravděpodobnosti nukleotidu předchozího. [16]



Obrázek 3 Markovův řetězec. [16]

Na Obrázku 3 lze vidět grafický model Markovova řetězce, u kterého kruhy charakterizují čtyři možné stavy: v tomto případě *A*, *C*, *G* a *T*. Šipky popisují možné přechody mezi stavy. Každý tento přechod je charakterizován přechodovou pravděpodobností $a_{st}=P(x_i=t|x_{i-1}=s)$.

Markovův řetězec prvního řádu je systém *(S, A)*, který se skládá z konečného množství stavů $S= \{s_1, s_2..., s_n\}$ a přechodové matice $A= \{a_{st}\}$, kde platí $\sum_{t \in S} a_{st} = 1$ pro všechna *s* patřící do *S*. Tímto je určena pravděpodobnost přechodu $s \rightarrow t$, jako: $P(x_{i+1}=t|x_i=s)=a_{st}$. Tedy v každém okamžiku *i* se nachází řetězec ve stavu x_i a řetězec se mění na stav x_{i+1} podle dané pravděpodobnosti. [16]

Avšak je nutné pamatovat, že tento řetězec začíná ve stavu x_1 s počáteční pravděpodobností $P(x_1)$. Z tohoto důvodu je potřeba do modelu přidat počáteční stav, který se označuje jako „*b*" a platí $x_0=b$, $P(x_1=s)=abs=P(s)$, kde $P(s)$ je takzvaná pravděpodobnost pozadí symbolu *s*. Obdobně je popsán i konec sekvence pomocí koncového stavu "*e*", který popisuje pravděpodobnost, že skončí právě v tomto stavu. $P(x_L=t)=a_{xLe}$. [16]

V případě exonu lze vypočítat pravděpodobnost, zda sekvence pochází z exonu, proti pravděpodobnosti, že nepochází. Nejprve je důležité určit příslušné přechodové matice. Tyto matice můžou být vypočítány z trénovací sady dat. Matici přechodu pro sekvenci DNA, která pochází z exonu, je označena A^+ a stanovena jako:

$$a_{st}^{+} = \frac{c_{st}^{+}}{\sum_{t'} c_{st'}^{+}}, \tag{2}$$

kde c_{st} je počet pozic v trénovací sadě, ve kterých je stav *s*, následován stavem *t*. Podobně je vyjádřena přechodová matice A^-, která obsahuje přechodové pravděpodobnosti pro případ, že sekvence nepochází z exonu. [16]

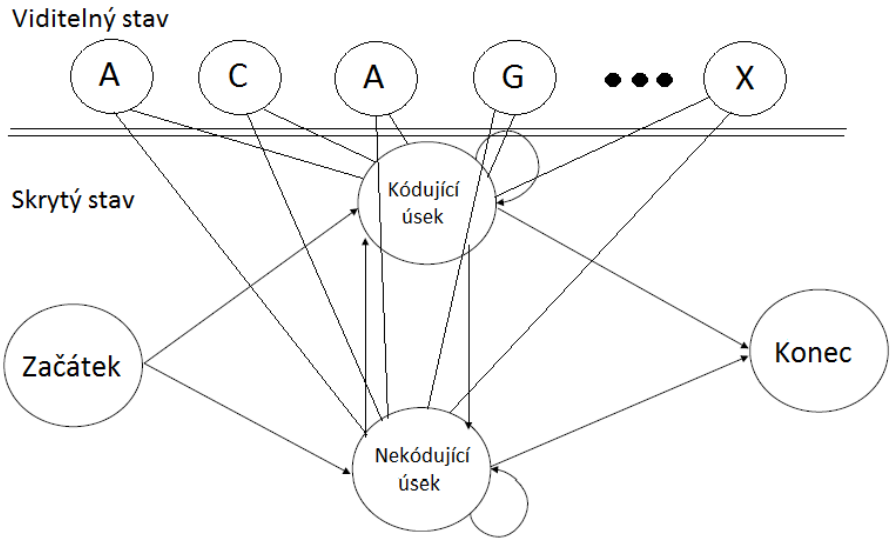
Následně je určené skóre, které udává, zda daný úsek sekvence pochází z oblasti exonů či nikoli. V případě, že ano: $P(x|model^+) = \prod_{i=2}^{L} a_{x_i x_{i+1}}^{+}$ a pro druhý případ platí: $P(x|model^-) = \prod_{i=2}^{L} a_{x_i x_{i+1}}^{-}$. Celkové skóre pak je vypočítáno podle vzorce:

$$S(x) = log \frac{P(x|model^+)}{P(x|model^-)} = \sum_{i=0}^{L} log \frac{a_{x_i x_{i+1}}^{+}}{a_{x_i x_{i+1}}^{-}}. \tag{3}$$

Čím vyšší bude toto skóre, tím vyšší je pravděpodobnost, že *x* pochází z exonu.[16]

Skrytý Markovův model lze definovat následujícím způsobem. Jedná se o pravděpodobnostní stavový model, který se skládá z konečného počtu stavů, z nichž každý může emitovat symbol z konečné abecedy s pevně stanovenou pravděpodobností mezi těmito symboly a sadou přechodů mezi těmito stavy, které umožňují model měnit po každém emitovaném symbolu. Skrytým se nazývá proto, že zvenku se nedá přesně zjistit stav, ve kterém se nachází, ale dostáváme informace o výstupu, který nastává s určitou pravděpodobností. Každý krok má výstup v podobě daného symbolu. A pro každý stav je dána pravděpodobnost pro výskyt konkrétního symbolu na výstupu. [15]

Takovýto model je získán spojením dvou Markovových řetězců, které jsou výše pojmenovány jako model^{-}(nekódující úsek) a model^{+}(kódující). [16]



Obrázek 4 Skrytý Markovův model.

Obecně je skrytý Markovův model popsán jako systém $M=(\Sigma, Q, A, e)$, kde

Σ je abeceda znaků (v tomto případě A, G, C, T)

Q je sada stavů (zde kódující (A+, G+, C+, T+), nekódující (A-, G-, C-, T-))

A matice přechodových pravděpodobností ($A=\{a_{kl}\}$)

e matice emisní pravděpodobnosti. [16]

Z Obrázku 4 je tedy patrné, že pomocí HMM lze získat pravděpodobnostní model nukleotidů, ale už se nedá zjistit, zda daný nukleotid pochází z kódujícího úseku nebo úseku nekódujícího. Pro zjištění, z jakého stavu pochází daný nukleotid, musí být použit Viterbiho algoritmus. [16]

2.7.2.1 Viterbiho algoritmus

Pokud je určena sekvence symbolů Σ (v tomto případě A, C, G a T), existuje několik cest přes skryté stavy, které vedou k dané sekvenci, avšak tyto cesty nemají stejnou pravděpodobnost. Viterbiho algoritmus je algoritmus dynamického programování, který umožňuje vybrat nejpravděpodobnější cestu a to podle rovnic 4 a 5. [16]

$$\pi^* = arg\max_{\pi} P(x, \pi). \quad (4)$$

Pokud jsou zvoleny úvodní symboly *(x_1, x_2, ..., x_i)*, poté $v_k(i)$ označuje pravděpodobnost, že nejpravděpodobnější cesta je ve stavu *k*, při generování symbolu *xi* na pozici *i*. Pak:

$$v_l(i+1) = e_l(x_{i+1}) \max_{k \in Q}(v_k(i) a_{kl}), \quad (5)$$

kde $v_k(i)$ se nazývá viterbiho proměnná. [16]

Vstup: Vstupem do viterbiho algoritmu je právě HMM *M = (Σ, Q, A, e)* a symbol sekvence *x*.

Výstup: Výstupem je nejpravděpodobnější cesta označovaná π^*

Začátek (i=0): $v(0)=1$, $v_k(0)=0$ pro $k \neq 0$

Pro všechny i=1…L, l ∈ Q: $v_l(i) = e_l(x_i) \max_{k \in Q}(v_k(i-1) a_{kl})$

$$ptr_i(l) = arg\max_{k \in Q}(v_k(i-1) a_{kl})$$

Ukončení: $P(x, \pi^*) = \max_{k \in Q}(v_k(L) a_{k0})$

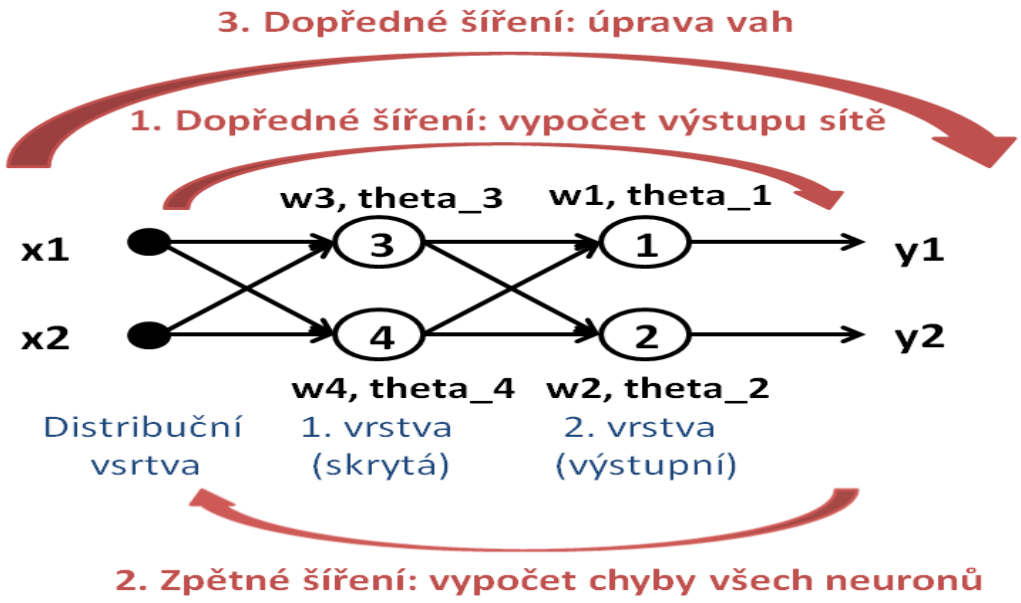
$$\pi_L^* = \text{argmax}_{k \in Q}(v_k(L) a_{k0})$$

Je získána nejpravděpodobnější cesta skrytými stavy, a tedy zjištěno, který nukleotid v dané sekvenci pochází z oblasti kódující respektive z oblasti nekódující. [16]

2.7.3 Neuronová síť (NN)

V případě, že se použije DP se souborem libovolných počátečních vah, jejichž velikosti v průběhu predikce se nemění, nemusí se podařit zpracovat sekvence správně. Pro přesnější predikci tedy může být použita metoda neuronových sítí, jejichž jedna z vlastností je i možnost predikce. Zde se využívají takzvané aproximátory, jedná se o sítě, které si vytvoří vlastní funkční model podle vstupních informací, který aproximuje funkci skutečného systému. Právě tato vlastnost umožní síti predikovat. Hlavním typem této sítě je takzvaná vícevrstvá perceptronová síť, která může být známá pod zkratkou síť BP (název pochází z anglického spojení back propagation, síť se zpětným šířením), příklad takovéto sítě se 4 neurony je na Obrázku 5. Síla této metody spočívá právě v chybách, které se vrací zpět na učební algoritmus, na kterém dochází k jemnému doladění velikostí vah, toto ladění vede k zlepšení predikce. Chyba této sítě se počítá od výsledku. Síť se tedy učí s učitelem. Úkolem tohoto postupu je nalézt optimální hodnoty pro váhové vektory *w*. Aby síť v budoucnu správně pracovala, musí být natrénována, k tomuto kroku je potřeba mít takové vstupy, o

kterých je známo, do které třídy patří. Bude potřeba trénovací dvojice a to vstupní vektor a požadovanou odezvu. [9], [11]



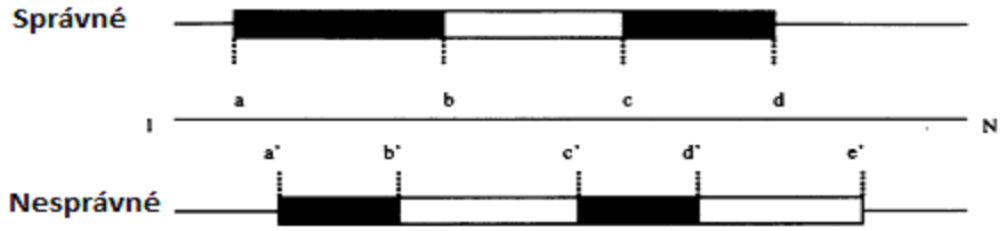
Obrázek 5 Algoritmus zpětného šíření chyby. Příklad sítě BP s 4 neurony. [14]

Váhy v druhé vrstvě se vypočítají jednoduše, pomocí delta pravidla, jelikož je znám požadovaný výstup a také vstup sítě. Váhy neuronů 3,4 se určí pomocí algoritmu zpětného šíření chyby. Učení neuronové sítě probíhá v epochách, jedná se o ukázání všech dvojic neuronové síti a upravování vah. [9], [11], [14]

Je dán vektor T_{xi} *(a, b)*, které reprezentuje skoré pro danou statistiku *i* (například použité kodónu, délkovou distribuci nebo použití šestic) exonu začínajícího na pozici *a* a končícího na pozici *b*. Toto číslo je uloženo ve vrstvě *i* matice T_E a musí být počítáno pouze jednou během trénovacího procesu. Obdobně je získán T_{xi} *(a, b)*, reprezentující skóre pro statistiku *i* danou intronem, uložené ve vrstvě *i* matice T_I. Dále je zavedena matici L_E, která bude obsahovat hodnoty pro exony začínající na pozici *a* a končící na pozici *b*. Tuto hodnotu lze vypočítat jako váhovaný součet všech statistických příspěvků podle následujícího vzorce:

$$L_E = \frac{1}{1+e^{-\sum_i w_{xi} \cdot (T_{xi}(a,b)+c_{xi})}}, \tag{6}$$

kde w_{xi} je hodnota váhy a c_{xi} je předsudek pro statistiku *i*. [9]



Obrázek 6 Příklad sekvence. V horní části obrázku je správné řešení, pod ním je nesprávné. Plné obdélníky odpovídají exonům, prázdné intronům. [9]

Pomocí takto získané informace z L_E a L_I metodou dynamického programování je zjištěno nejvyšší možné skóre kombinací sousedních nepřekrývajících se intron a exonů. Jako příklad lze použít správné a nesprávné řešení v Obrázku 6. [9]

Výsledek pomocí dynamického programování pro jednotlivé řešení je roven:

$$D(aktuální) = L_E(a,b) + L_I(b+1, c-1) + L_E(c,d). \quad (7)$$

$$D(nesprávné) = L_E(a',b') + L_I(b,+1,c'-1) + L_E(c',d') + L_E(d'+1,e'-1)\,. \quad (8)$$

Úkolem tedy je najít vhodné váhy a předsudky statistik aby platilo:

D(aktuální) > D(nesprávné)

Cílem je dosažení stavu, kdy aktuální struktura genu bude mít větší skóre než případné možné nesprávné řešení. Toho lze dosáhnout pomocí NN, které hledá optimální váhy tak, aby byla splněna podmínka *D(aktuální) > D(nesprávné)*. Vstupem do neuronových sítí je rozdíl mezi sumou skóre pro každou statistiku pro správní a nesprávné řešení. Výsledkem jsou hodnoty pro exony a pro introny. [9]

Exony: $\Delta T_{xi} = T_{xi}(a,b) + T_{xi}(c,d) - T_{xi}(a',b') - T_{xi}(c',d').$ (9)

Introny: $\Delta T_{\mu i} = T_{\mu i}(b+1, c-1) - T_{\mu i}(b'+1, c'-1) - T_{\mu i}(d'+1, e'-1).$ (10)

Trénování neuronových sítí:

1. Nastavení náhodných váh.
2. Sestavení matice *T* pro každou sekvenci. (jedná se o řadu čísel, která statisticky klasifikuje dané subregiony)
3. Vytvoření matice *L* pro každou sekvenci. (obsahuje váhované složky skóre z odpovídajících části matic *T*)
4. Spočítání hodnoty *D* pomocí dynamického programování ze souboru *L* matici.
5. Zjištění, zda je dosažená dostatečná přesnost.

   Pokud ano: Učení sítě je ukončeno.

Pokud ne: Učení pokračuje dále.

6. Trénování neuronové sítě.
7. Aktualizace vah podle výsledků z předchozího trénování.
8. Pokračování bodem 3. [9]

2.7.4 Diskrétní Fourierova Transformace (DTF)

Je dobře známo, že sekvence DNA kódující protein vykazují periodicitu rovnou 3, kvůli kodónové struktuře zapojené do translace.[1] Důležité zjištění bylo, že u eukaryotických buněk se tato periodicita vyskytuje v exonech, zatímco v intronech se nevyskytuje vůbec. Problém však nastává u prokaryot, ve kterých se periodicita projevuje i mimo kódující oblasti. V případě eukaryot, periodicita charakterizuje oblasti kódující proteiny. Z tohoto důvodu se dá DFT využít jako identifikátor genů u eukaryotických organismů. [1], [2], [10]

Diskrétní Fourierova transformace se využívá pro zpracování periodicity. V případě, že je dána DNA sekvence délky *N*, dá se předpokládat $u_A\,(n)$, $u_T\,(n)$, $u_C\,(n)$ a $u_G\,(n)$, jedná se o 4 indikační vektory, které reprezentují přítomnost nebo nepřítomnost daného nukleotidu na pozici *n*. Aplikováním DFT na tyto indikační sekvence získáme 4 spektrální reprezentace a to $U_A\,(k)$, $U_T\,(k)$, $U_C\,(k)$ a $U_G\,(k)$. [2]

Rovnice pro diskrétní Fourierovu transformaci:

$$U_X[k] = \sum_{n=0}^{N-1} u_X \cdot (n) \cdot e^{-j \cdot 2 \cdot \pi \cdot k \cdot n/N}\,, \tag{11}$$

kde *N* je délka okna (tedy část sekvence pro, kterou se počítá DFT),

k je koeficient spektra z intervalu 1 až *N/2*,

j je imaginární číslo,

$u_X\,(n)$ je *n*-tá hodnota v indikační sekvenci vymezené oknem *N*. [10]

Celkové (výkonnostní) spektrum dané DNA sekvence je definováno takto:

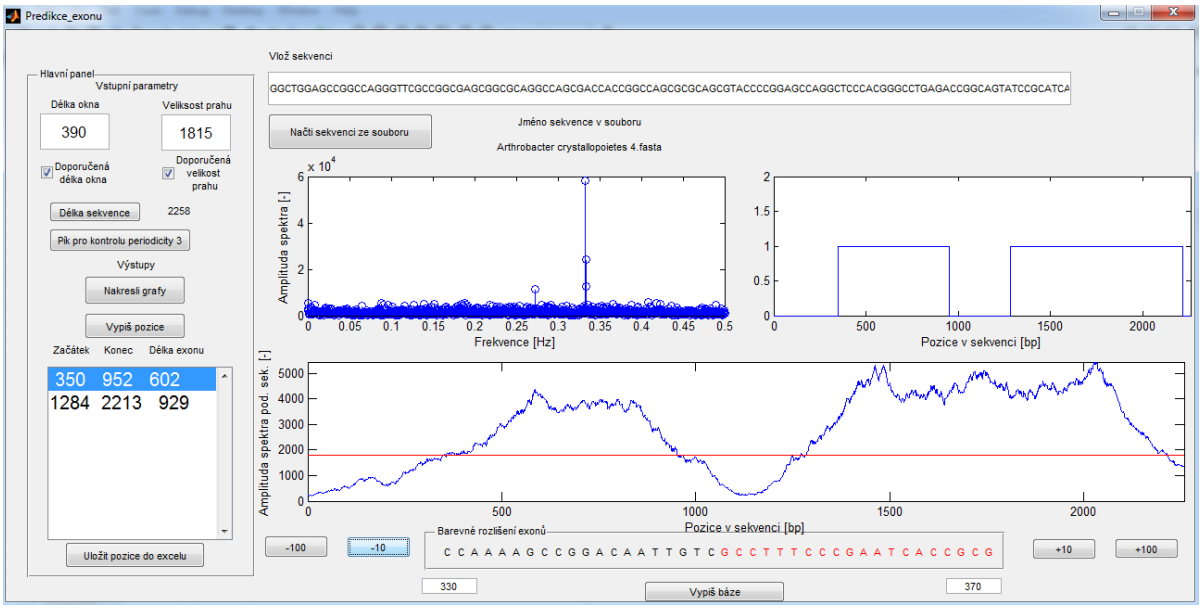
$$S[k] = |U_A(k)|^2 + |U_C(k)|^2 + |U_G(k)|^2 + |U_T(k)|^2. \tag{12}$$

V kódujících regionech DNA, má $S\,(k)$ pík na frekvenci $k = N/3$, zatímco v nekódujících oblastech nejsou přítomny žádné významné píky. Velikost píku je pak závislá na typu genu. Je důležité dodržet pravidlo, že *k* může nabývat pouze hodnot *N/3*. Posouváním okna podél sekvence je obdržen obraz změny $S\,[N/3]$ podél sekvence. Velikost okna musí být vhodně zvolena. Příliš velké okno způsobuje prodloužení výpočetní doby a snižuje rozlišení pro predikci exonu. Většinou se *N* volí ve stovkách až tisících. [1], [2], [10]

3 Praktická část

3.1 Program Predikce_exonu

Program *Predikce_exonu* byl vytvořen v prostředí MATLAB R2010b. Jeho hlavní část byla realizována v grafickém prostředí GUIDE. Úkolem programu je načíst a zpracovat sekvenci DNA. Sekvenci načte z FASTA souboru, který si uživatel zvolí, nebo lze sekvenci zadat ručně do připraveného okna. Jako výsledek program vypíše pozice začátků a konců detekovaných exonů, spolu s jejich délkou. Tyto výstupy lze uložit do Excelu ve formátu *.xls*. Program dále vykreslí výkonnostní spektrum sekvence, jehož důležitost je popsaná níže. Dalším grafickým výstupem programu je změna výkonnostního spektra podél sekvence pro frekvenční koeficient 3, podle něhož se určují pozice exonu. Posledním grafem, který program vykreslí, je zobrazení pozic detekovaných exonů. Závěrečná část programu se věnuje zobrazení bází sekvence, která se dá prohlížet v rozsahu 40 nukleotidů. Nalezené úseky exonů jsou barevně odlišeny a to červenou barvou. Pomocí tlačítek se uživatel může přesouvat po sekvenci o větší počet nukleotidů na požadované pozice. V následujících kapitolách je uveden podrobný popis programu a uživatelského prostředí. V příloze B se pak nachází manuál k programu.



Obrázek 7 Čelní panel programu Predikce_exonu s vyplněnými vstupy a výstupy.

3.1.1 Načtení sekvence

V programu *Predikce_exonu* byly vytvořeny dvě možnosti jak zadat analyzovanou sekvenci. První z nich je sekvenci ručně vypsat do edit okna. Tento způsob je však časově náročný a uživatel se zde může dopustit mnoha chyb.

Druhou, lepší možností je načíst sekvenci ze souboru a to ve formátu FASTA. K tomuto úkolu slouží tlačítko *Nacti_sekvenci*. Stisknutím se spustí funkce *NactiSoubor*. Tato funkce má dva výstupy a to název sekvence a samotnou sekvenci.

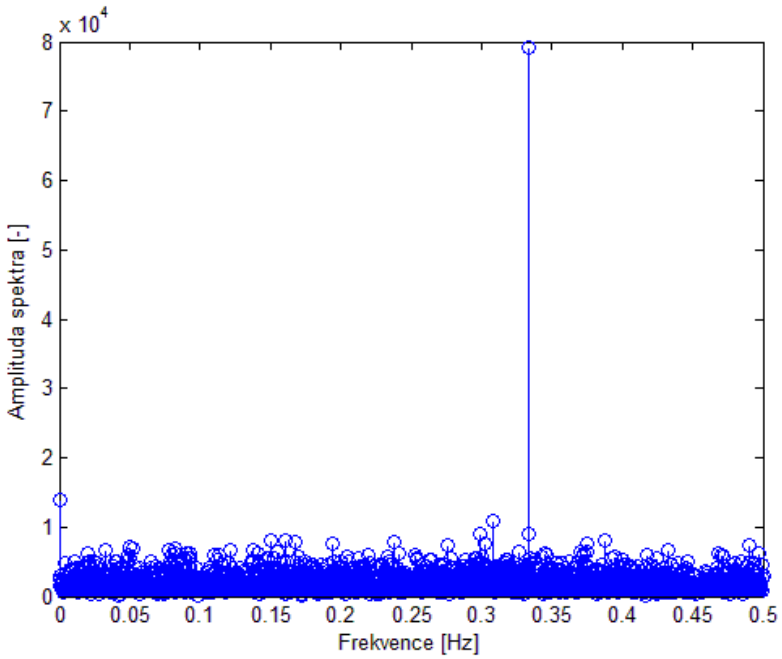
3.1.2 Ověření periodicity 3 v sekvenci

Jak již bylo popsáno v teoretické části, v odstavci 2.7.4 o Diskrétní Fourierově Transformaci, základní podmínkou pro možnost vyhledávání exonů pomocí této metody je předpoklad, že exony obsahují periodicitu rovnou 3. Program *Predikce_exonu* vykresluje výkonnostní spektrum testované sekvence, ze kterého je možno zjistit, zda se v této sekvenci vyskytuje potřebná periodicita 3. Výkonnostní spektrum se vykreslí pomocí tlačítka *Nakresli_piky*, kterým se spustí funkce *Nakresli_piky_Callback*. V této funkci je vnořená další funkce *vypocet_piku*. Jejím úkolem je výpočet výkonnostního spektra.

```
vypocet_piku(sek)
1   L ← délka sekvence
2   for k ←1 to L/2
3           n ← vektor délky L s lineárně rozloženými body mezi 0 a L-1, včetně těchto
    bodů
4           sumaAp_k ← Σ(ua.*exp(-2*i*pi*k*n/L))
5           sumaCp_k ← Σ(uc.*exp(-2*i*pi*k*n/L))
6           sumaGp_k ← Σ(ug.*exp(-2*i*pi*k*n/L))
7           sumaTp_k ← Σ(ut.*exp(-2*i*pi*k*n/L))
8   sumapik ← |(sumaAp)|.^2 + |(sumaCp)|.^2+ |(sumaGp)|.^2 + |(sumaTp)|.^2
9   return sumapik
```

Kde *sumaAp*, *sumaCp*, *sumaGp* a *sumaTp* jsou matice frekvenčních koeficientu
ua, *uc*, *ug* a *ut* jsou jednotlivé binární reprezentace
sumapik je vektor hodnot odpovídající výkonnostnímu spektru sekvence.

Po vykreslení vektoru hodnot *sumapik*, je možno pozorovat, zda se v sekvenci vyskytuje perioda rovna třem. Velikost píku také udává, jak je tato periodicita výrazná. V případě, že se zde periodicita 3 nevyskytuje, případně by její velikost byla nízká, musí se uvážit, zda má smysl pokračovat ve vyhledávání exonů touto metodou. Existuje však případ, že by exony byly krátké a mezi nimi hodně dlouhé introny, v tomto případě by pík 3 nemusel být výrazný, avšak metoda by fungovala. Na Obrázku 8 je příklad, kdy se ve výkonnostním spektru vyskytuje výrazný pík odpovídající periodicitě 3.



Obrázek 8 Příklad spektra sekvence.

3.1.3 Vyhledávání exonů

Stisknutím tlačítka *nakresli_grafy* se spustí vnořená funkce *fourier_hledani*, tato funkce tvoří základ celého programu. Aby funkce správně fungovala, potřebuje vstupní data, kterými jsou sekvence a parametry, které jsou vloženy uživatelem v *edit* políčcích *okno* a *Prah*. Tato políčka je nutné vyplnit. V případě, že by takto nebylo učiněno a políčka by nebyla vyplněna, nebo by byla vyplněna špatně, vyskočí upozornění, které bude odkazovat na chybu. Parametry jsou zadávané v datovém typu *string*, proto je potřeba je ihned převést na formát *double*. Poté je možné je načíst do funkce, spolu se sekvencí, kterou jsme získali načtením nebo vepsáním, podle kapitoly 3.1.1.

Úkolem funkce *fourier_hledani* je nalézt pozice exonů v zadané sekvenci. Hned na začátku je nutné převést sekvenci na numerickou reprezentaci, v tomto případě 4D binární reprezentaci, tato metoda vytvoří čtyři indikační vektory, vždy jeden pro daný nukleotid, které indikují přítomnost či nepřítomnost daného nukleotidu na pozici *n*:

$$u_x[k] = 1 \; jestliže \; s[k] = X, \tag{13}$$

kde *s[k]* pro $k = 0,1,\ldots,N-1$ je symbolická sekvence o délce *N*.

Z takto získaných binárních reprezentací program spočítá změnu výkonnostního spektra podél sekvence pro koeficient $k = N/3$.

```
1   for q ←1 to K
2           for n ← 1 to N
3                   sumaA_q ← sumaA_q+ua_pom+n*exp(-2*i*pi*k*(n-1)/N)
```

4 $sumaC_q \leftarrow sumaC_q+uc_{pom+n}$*exp(-2*i*pi*k*(n-1)/N)

5 $sumaG_q \leftarrow sumaG_q+ug_{pom+n}$*exp(-2*i*pi*k*(n-1)/N)

6 $sumaT_q \leftarrow sumaT_q+ut_{pom+n}$*exp(-2*i*pi*k*(n-1)/N)

7 pom←pom+1

8 suma←|(sumaA)|.^2+|(sumaC)|.^2+|(sumaG)|.^2+|(sumaT)|.^2;

9 **return** suma

Kde *sumaA, sumaC, sumaG* a *sumaT* jsou matice frekvenčních koeficientu

ua, uc, ug a *ut* jsou jednotlivé binární reprezentace

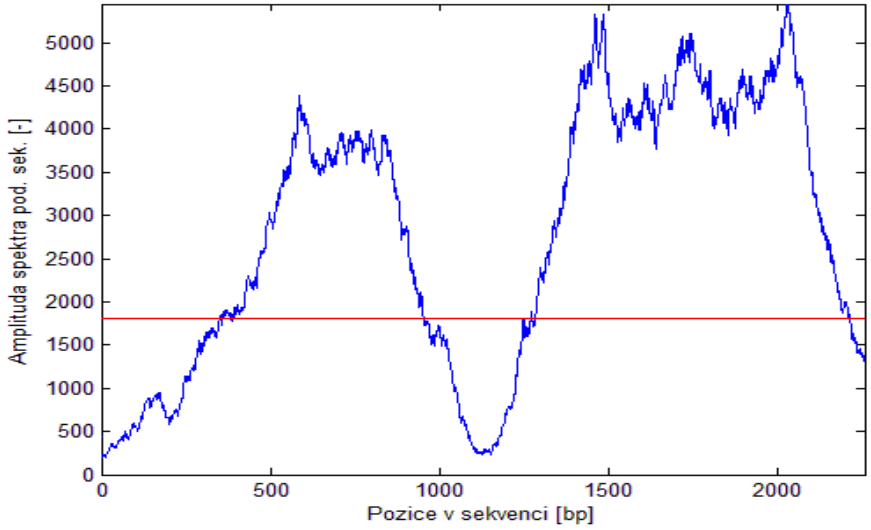
pom je pomocná proměnná

suma je vektor hodnot odpovídající průběh výkonnostního spektra podél sekvence.

A právě z takto získaného vývoje výkonnostního spektra podél sekvence je algoritmus schopen získat pozice exonů. V tomto kroku, výpočtu výkonnostního spektra podél sekvence, je program velice ovlivněn uživatelem, jelikož je potřeba vhodně zvolit délku použitého okna a velikost prahu. Existují dvě možnosti. Je ručně zadána hodnota délky okna a program pracuje s touto hodnotou. Nebo je využita doporučená hodnota okna, která byla zvolena na základě prováděné analýzy, která je popsána v kapitole 3.2. V této části je podrobněji uvedeno, jaké hodnoty okna jsou vhodné, musí být však splněna podmínka, že délka okna je dělitelná třemi. Doporučená velikost okna je vybrána pomocí *checkbutton Dop_delka_okna*.

Obdobný princip je použit u volby velikosti prahu. Uživatel má možnost zadat vlastní velikost prahu, ale také, pomocí *checkbutton Dop_hodnota_prahu*, může nastavit hodnotu doporučenou, která vychází z analýzy programu a je zvolena na hodnotu *max (suma)/3*. Při těchto hodnotách dopadla analýza programu, provedena na 4 sekvencích, nejlépe. Avšak program ponechává možnost volby i na uživateli, pro případ, že by vyhledávání nedopadlo uspokojivě.

Po zvolení okna a prahu program spočítá vývoj výkonnostního spektra podél sekvence pro koeficient $k = N/3$. Takto získané hodnoty program vykreslí. Následně má uživatel možnost upravit hodnoty okna a prahu a znovu nechat spočítat a vykreslit výkonnostní spektrum podél sekvence. Příklad vykresleného výkonnostního spektra je zobrazen na Obrázku 9.



Obrázek 9 Průběh výkonnostního spektra podél sekvence pro k=N/3 a prahu.

Další část programu *fourier_hledani* je zaměřena na detekci pozic začátků a konců exonů. Jedná se o prosté podmínky, kdy program, jako začátek označí pozici, kdy hodnota *suma* v daném bodě je menší než hodnota zvoleného prahu a zároveň hodnota *suma* na pozici o jedna větší bude větší, než daný práh. Nalezení konce exonu je řešeno obdobným způsobem. Programové řešení detekcí obou pozic je popsáno pomocí pseudokódu uvedeného níže.

1. **for** i← 1 **to** L-1
2. **if** $suma_i$<prah **and** $suma_{i+1}$>prah
3. $pozicezac_i$←i
4. **return** $pozicezac_i$
5. **elseif** $suma_i$>prah **and** $suma_{i+1}$<prah
6. $poziceend_i$←i
7. **return** $poziceend_i$

Kde *suma* je vektor odpovídající průběh výkonnostního spektra podél sekvence
pozicezac je pozice začátků exonů
poziceend je pozice konců exonů
prah je hodnota prahu.

3.1.4 Vylepšení detekce pozic

Ve funkci *fourier_hledani* byla realizována dvě vylepšení, umožňující přesnější vyhledání pozic a odstranění nevhodných exonů. První z nich spočívá v sečtení pozic exonů v případě, že by dva detekované exony byly od sebe vzdáleny méně než 9 nukleotidů. Tento krok zamezí vzniku velkého množství rozdělených exonů, které se však dají považovat za exon jeden.

```
1.  P←0
2.  A←1
3.  i←0
4.  if length(zacatek)<4
5.          zacatek1←zacatek_1
6.          konec1←konec_1
7.          return zacatek1
8.          return konec1
9.  else
10.         while P<délka(zacatek)-1
11.                 c←1+1
12.                 i←i+1
13.                 if |(zacatek_{i+1}- konec_i)|≤9 and |(zacatek_{i+1}- konec_i)|≠0
14.                         while |(zacatek_{i+1}- konec_i)|≤9
15.                                 zacatek1_A←zacatek_c
16.                                 konec1_A←konec_{i+1}
17.                                 i←i+1;
18.                                 P←i+1;
19.                 A=A+1
20.                 else
21.                         zacatek1_A←zacatek_i
22.                         konec1_A←konec_i
23.                         P←P+1;
24.                         A←A+1;
25. return zacatek1
26. return konec1
```

Kde *A, c* a *i* jsou pomocné proměnné pro indexovaní

zacatek a *konec* jsou původní nalezené začátky a konce exonů

zacatek1 a *konec1* jsou začátky a konce exonů po sečtení blízkých exonů.

Druhé vylepšení se týká odstranění krátkých exonů. Jedná se o prosté smazání exonů, které byly detekovány, ale jejich délka je menší než 9 nukleotidů. Jak leze vidět na pseudokódu uvedeném níže, tento problém byl vyřešen tím, že v případě, že exon bude kratší než 9 nukleotidů, bude nahrazen exonem následujícím.

```
1.  a←1
2.  LZ← délka zacatek1
3.  for l←1 to LZ-1
4.      if konec1_l-zacatek1_l≤9
```

5. zacatek$_a$ ←zacatek1$_{l+1}$
6. konec$_a$ ←konec1$_{l+1}$
7. **else**
8. zacatek$_a$ ←zacatek1$_l$
9. konec$_a$ ←konec1$_l$
10. a←a+1;
11. **return** zacatek
12. **return** konec

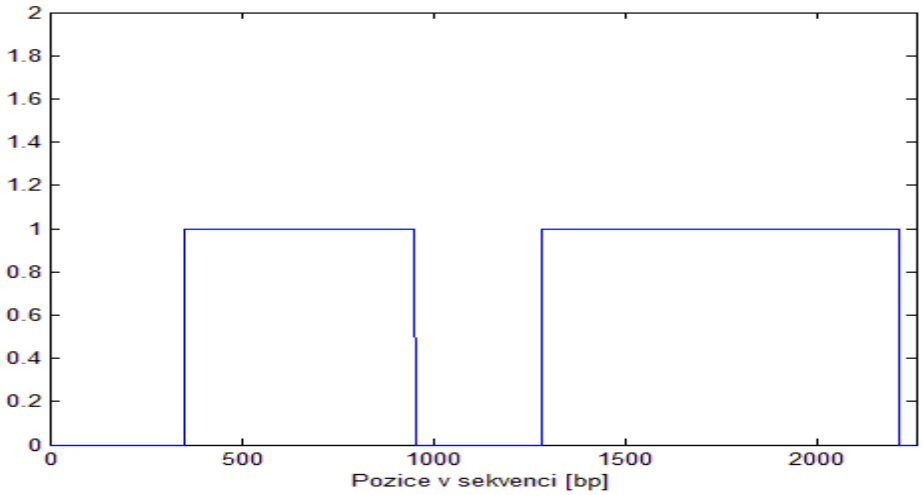
Kde *zacatek1* a *konec1* jsou původní začátky a konce detekovaných exonů
zacatek a *konec* jsou začátky a konce exonů po odstranění krátkých exonů.

3.1.5 Grafická reprezentace nalezených úseků a uložení

Poslední úsek funkce *fourier_hledani* slouží ke grafickému znázornění nalezených exonů. Jedná se o vytvoření vektoru, kde místa v sekvenci, která program označil za exony, budou obsahovat jedničky ostatní pozice sekvence nuly. Takto zobrazený výsledek můžete vidět na Obrázku 10.

1. R← počet detekovaných exonů
2. tip← matice 0 o délce sekvence
3. **if** R<2
4. tip(zacatek_konec$_{1,1}$:zacatek_konec$_{1,2}$) ←1
5. **else**
6. **for** i←1 **to** R
7. tip(zacatek_konec$_{i,1}$:zacatek_konec$_{i,2}$) ←1
8. **return** tip

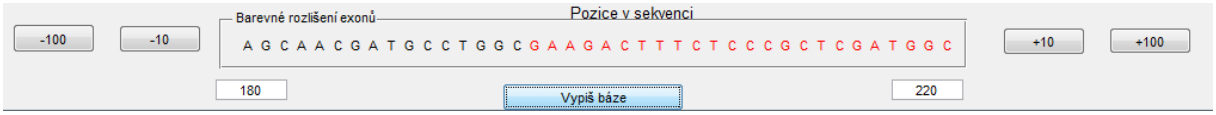
Kde *tip* je výsledný vektor hodnot reprezentující přítomnost, nepřítomnost exonu
zacatek_konec je matice obsahující pozice začátků a konců detekovaných exonů.



Obrázek 10 Grafická reprezentace nalezených exonů.

Dalším nástrojem ve funkci *Predikce_exonu* je vykreslení nukleotidů, ve kterých budou jednotlivé báze přebarveny na červeno v případě, že se na dané části nachází exon. Báze se vypisují do panelu s rozsahem 40 nukleotidů. V tomto panelu se dá pohybovat pomocí tlačítek po stranách nebo přímým přepisováním pozic. Existuje zde však omezení, kdy se nelze dostat na pozice delší než je sekvence, nebo pozice záporné v tomto případě budou vypsány pozice posledních 40 nukleotidů respektive prvních.

V případě zvolení vykreslení tedy probíhá analýza zvolených pozic a zjišťuje se, zda se na vybraných pozicích nachází exon. V případě, že se na dané pozici exon nacházet bude, barva tohoto písmene se obarví na červeno a zařadí se místo předchozí hodnoty do proměnné *znak*. Po skončení cyklu obsahuje tato proměnná červené nebo černé nukleotidy a vypíše se do panelu. Na Obrázku 11 je příklad takovéhoto vypsání sekvence.



Obrázek 11 Příklad barevného vyznačení nukleotidů v sekvenci.

3.2 Analýza programu Predikce_exonu

Analýza programu byla provedena na 4 sekvencích a to Magnaporthe oryzae 1, Homo sapiens 3, Listeria monocytogenes 1 a Drosophila melanogaster 1. Více informací o těchto sekvencích je uvedeno v Tabulce 5. Hlavním důvodem a cílem analýzy bylo zjistit, jaká volba délky okna a prahu je optimální, případně zda by se dala volit délka okna v závislosti na délce sekvence.

Provedeno bylo několik analýz, při kterých se volila okna od 300-630 vždy po 30. Některé sekvence byly testovány do nižší maximální délky okna, například do 420, jelikož při vyšších hodnotách vycházely detekované pozice hůř.

Průběh analýzy byl prostý, pro tři velikosti prahu (max(suma/3), max(suma/2) a max(suma)/2,5) byly měněny okna a detekovány pozice, následně byla spočítána senzitivita a specificita pro každý práh a okno, poté byly sestrojeny ROC křivky popisující vzájemnou závislost senzitivity a specificity na velikosti prahu pro různé volby délky okna. Jako optimální velikost okna a prahu byla zvolena hodnota, pro kterou byla specificita i senzitivita největší. Tedy tak, aby byl program co nejvíce specifický a zároveň co nejvíce senzitivní.

Senzitivita je míra pravdivé pozitivity. V tomto případě je to pravděpodobnost, že v případě, že se jedná o exon, vyjde pozitivní test. Senzitivita se označuje jako TPR a lze ji vyjádřit pomocí kontingenční tabulky jako:

$$TPR = \frac{TP}{TP+FN}. \tag{14}$$

Zatímco specificitou je myšlena míra pravdivé negativity a tedy pravděpodobnost, že v případě, že se nejedná o exon, vyjde negativní test. Pomocí kontingenční tabulky se vyjadřuje jako:

$$TNR = \frac{TN}{TN+FP}, \tag{15}$$

kde hodnoty *TP*, *TN*, *FN* a *FP* jsou hodnoty z kontingenční tabulky. Příklad tabulky je v Tabulce 2. TP pochází z anglického spojení true positive a znamená pravdivě pozitivní. TN z anglického true negative tedy pravdivě negativní. FP vychází z anglického false posivie, falešně pozitivní. FN tedy false negative, znamená falešně negativní. [17]

Tabulka 2 Kontingenční tabulka. [17]

	EXON	NENÍ EXON
POZITIVNÍ TEST	TP	FP
NEGATIVNÍ TEST	FN	TN

Tabulka 3 obsahuje příklad prováděné analýzy. Vybraná byla tabulka, ve které vyšly maximální hodnoty specificity i senzitivity, proto z analýzy sekvence Magnaporthe oryzae 1, byla odhadnuta doporučena hodnota prahu max(suma)/3 a délka okna 390. Tyto hodnoty byly potvrzené i vykreslenými ROC křivkami. V Tabulce 4 je příklad vypočtených senzitivit a specificit pro sekvenci Magnaporthe oryzae 1 pro všechny testované velikosti prahů a oken. Takto byly spočteny hodnoty i pro ostatní analyzované sekvence. Z těchto hodnot pak byly sestrojeny ROC křivky pro všechny 4 sekvence. Příklad ROC křivek, pro sekvence Magnaporthe oryzae 1 a Homo sapiens 3 je v Grafech 1 a 2. Výsledky kompletní analýzy jsou k dispozici na přiloženém CD.

Tabulka 3 Příklad prováděné analýzy.

Délka okna(N)=390	Magnaporthe oryzae 1			
Práh=max(suma)/3	Detekované pozice			
Délka sekvence(L)=1418	Začátek	Odchylka od správné pozice začátků	Konec	Odchylka od správných pozic konců
L/N=3.6359	225	0	506	7
Senzitivita: 98,77 %	610	-18	1235	11
Specificita: 95,25 %				

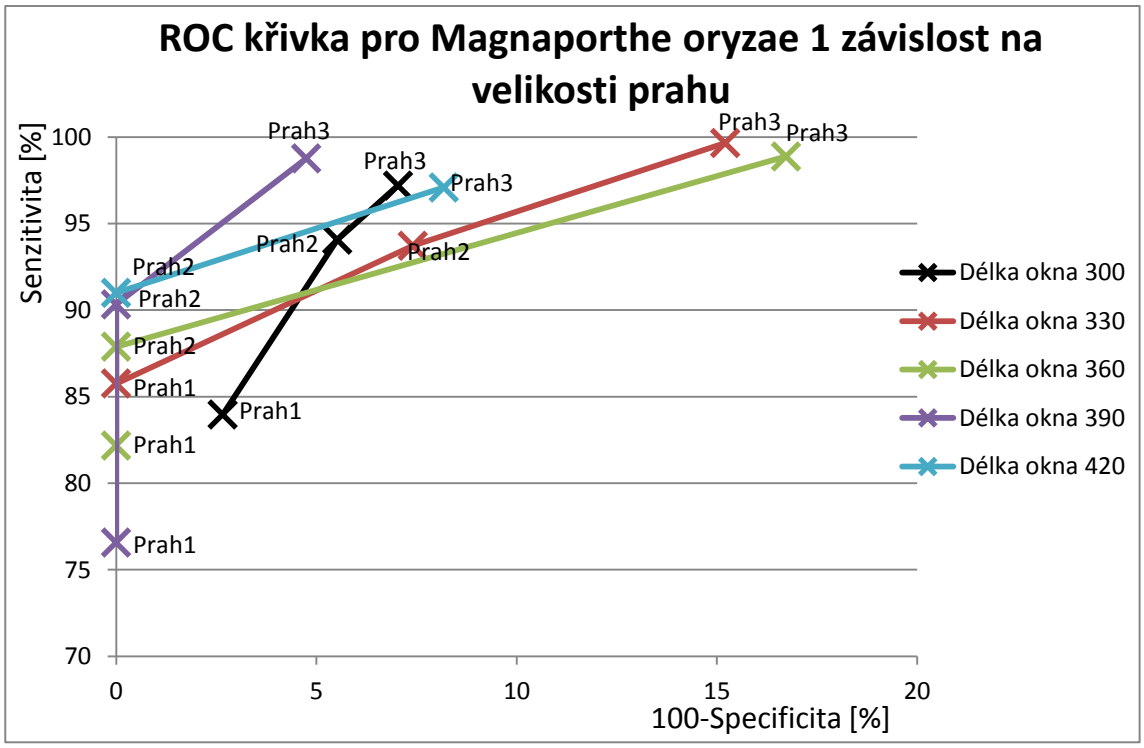
Jak je patné z grafů uvedených níže, nejvyšších a vyrovnaných hodnot senzitivity a specificity bylo dosaženo pro práh max(suma)/3. Tato hodnota byla potvrzena i v testech pro další 3 sekvence. Z těchto důvodů byla doporučená hodnota prahu nastavena na max(suma)/3.

Větší problém nastal při hledání vhodné délky okna. Během testování nebyla zjištěna závislost délky okna na délce sekvence. Jelikož poměr délky sekvence k délce okna, který by byl optimální pro kratší sekvence, nebylo možné aplikovat na sekvence delší a obráceně. Z tohoto důvodu bylo zjištěno, jakých optimálních hodnot nabývá délka okna v analýze a zjistil se rozsah hodnot od 300 do 570 pro různé délky sekvencí. Přesněji byly u 4 testovaných sekvencí zjištěny tyto optimální hodnoty oken: 390, 570, 300 a 300. Na základě analýzy tedy bylo rozhodnuto jako doporučenou hodnotu délky okna volit hodnotu 390.

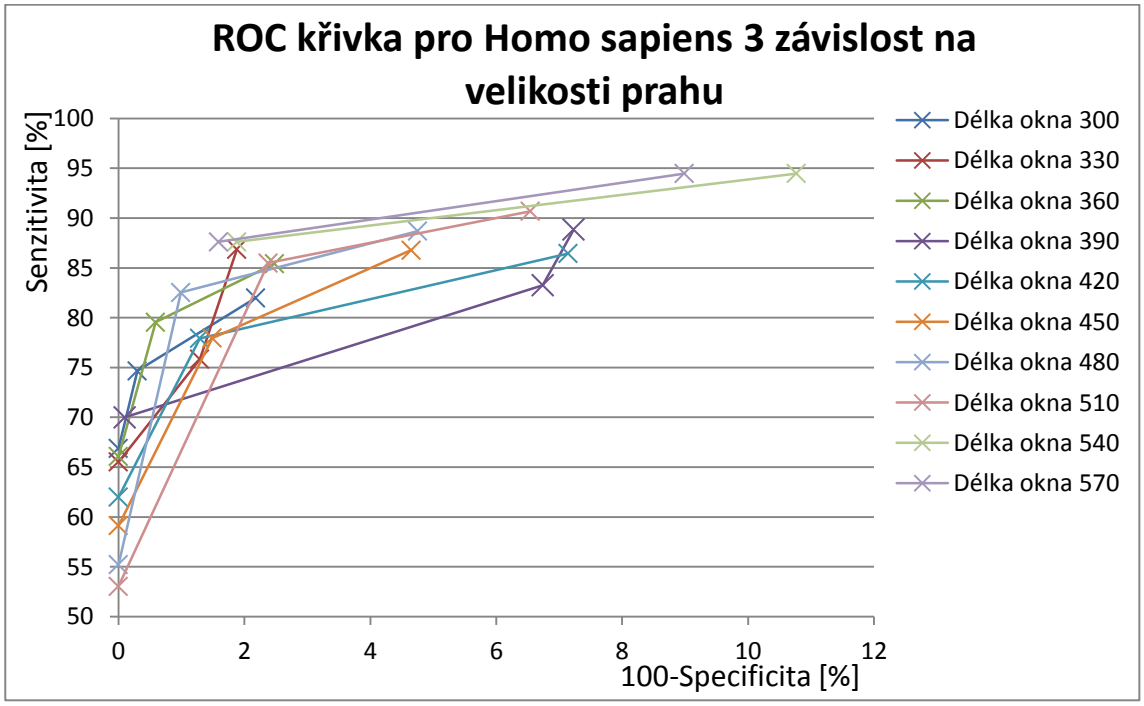
Hodnoty, které byly určeny na základě testování, jsou hodnoty doporučené, uživatel má možnost si zvolit vlastní hodnoty podle svého uvážení v případě, že by nebyl spokojen s hodnotami nastavenými. Avšak tyto hodnoty musí splňovat základní požadavky na délku okna a to především, že musí být dělitelná 3.

Tabulka 4 Příklad senzitivit a specificit pro sekvenci Magnaporthe oryzae 1.

Práh1=max(suma)/2		
Délka okna	senzitivita	specificita
300	83,97	97,34
330	85,76	100
360	82,17	100
390	76,57	100
Práh2=max(suma)/2.5		
Délka okna	senzitivita	specificita
300	94,06	94,48
330	93,72	92,59
360	87,89	100
390	90,34	100
420	91	100
450	90,68	100
Práh3=max(suma)/3		
Délka okna	senzitivita	specificita
300	97,2	92,96
330	99,66	84,79
360	98,88	83,27
390	98,77	95,25
420	97,09	91,82



Graf 1 ROC křivka pro Magnaporthe oryzae 1 závislost na velikosti prahu.



Graf 2 ROC křivka pro Homo sapiens 3 závislost na velikosti prahu.

3.3 Testování programu Predikce_exonu a tří volně dostupných programů

V této kapitole jsou popsány tři volně dostupné internetové vyhledávače pro predikci exonů, z nich některé využívají metody popsány v kapitole 2.7. Programy byly testovány na 25 sekvencích a pro každý program byla spočítána specificita a senzitivita a na základě těchto hodnot bylo zjištěno jak je daný program, respektive metoda, kterou používá, úspěšný. Testování bylo provedeno na 17 sekvencích eukaryotických organismů a na 8 sekvencích prokaryotických organismů. Sekvence byly získány z databáze NCBI a zde uvedené oblasti exonů jsou považovány za správné, avšak nemusí tomu tak vždy být, jelikož velké množství sekvencí, zde uvedených bylo analyzováno pomocí automatických analyzátorů s určitou genovou predikční metodou. Další informace o použitých sekvencích jsou v Tabulce 5.

Tabulka 5 Seznam testovaných sekvencí. [13]

Označení sekvence v dokumentu	Symbol	Organismus	Typ organismu	Délka sekvence (bp)	Počet exonu	Exony podle NCBI	Dostupné
Homo sapiens 1	PABPC3	Homo sapiens	Eukaryotický	2430	1	62..1957	1
Homo sapiens 2	SLC7A4	Homo sapiens	Eukaryotický	3841	4	747..1728,2208..2848,3115..3223,3329..3504	2
Homo sapiens 3	GALR3	Homo sapiens	Eukaryotický	2114	2	26..384, 1342..2089	3
Pan troglodytes 1	TAAR5	Pan troglodytes	Eukaryotický	1014	1	1..1014	4
Pan troglodytes 2	SLC32A1	Pan troglodytes	Eukaryotický	5167	2	501..890, 3248..4435	5
Pongo abelii 1	ADAM21	Pongo abelii	Eukaryotický	2049	1	652..2049	6
Pongo abelii 2	CHST12	Pongo abelii	Eukaryotický	1990	1	76..1320	7
Equus caballus 1	SOX2	Equus caballus	Eukaryotický	2138	2	1..161, 268..1052	8
Equus caballus 2	BGN	Equus caballus	Eukaryotický	5354	7	8..257, 618..730, 1259..1472200 2..2112 2367..2460 2541..2679412 1..4318	9
Mus musculus 1	Sry	Mus musculus	Eukaryotický	1188	1	1..1188	10
Aspergillus niger 1	ANI_1_866124	Aspergillus niger	Eukaryotický	967	2	11..94, 182..967	11

Magnaporthe oryzae 1	MGG_05802	Magnaporthe oryzae	Eukaryotický	1418	2	225..499, 628..1246	12
Magnaporthe oryzae 2	MGG_07258	Magnaporthe oryzae	Eukaryotický	2947	1	183..2450	13
Oryza sativa 1		Oryza sativa Indica Group	Eukaryotický	537	1	1..537	14
Drosophila melanogaster 1	VINC	Drosophila melanogaster	Eukaryotický	8174	5	2031..2087 4365..6555 6617..6747 6807..7205 7338..7445	15
Caenorhabditis elegans 1	vit-5	Caenorhabditis elegans	Eukaryotický	5075	6	1..864, 935..1219, 1267..4304 4354..4582 4628..4763 4816..5075	16
Caenorhabditis elegans 2	R09H3.1	Caenorhabditis elegans	Eukaryotický	4205	6	1..1524, 1579..1792 1948..2035 2140..3054 3675..3881 3932..4205	17
Escherichia coli 1		Escherichia coli W26	Prokaryotický	3472	3	154..306, 376..1278, 1355..3079	18
Escherichia coli 2		Escherichia coli O91:H21	Prokaryotický	6230	4	1..2616, 2914..3198 4655..4954 5356..>6230	19
Escherichia coli 3		Escherichia coli O91:H21	Prokaryotický	3252	1	78..>3252	20
Arthrobacter crystallopoietes 1		Arthrobacter crystallopoietes BAB-32	Prokaryotický	4165	2	1..363, 2382..3920	21
Arthrobacter crystallopoietes 2		Arthrobacter crystallopoietes BAB-32	Prokaryotický	2610	2	1..875, 1363..2610	22
Arthrobacter crystallopoietes 3		Arthrobacter crystallopoietes BAB-32	Prokaryotický	3530	3	1..1327, 1581..2342, 2619..3356	23
Arthrobacter crystallopoietes 4		Arthrobacter crystallopoietes BAB-32	Prokaryotický	2258	2	1..1057, 1269..2258	24
Listeria monocytogenes 1		Listeria monocytogenes	Prokaryotický	2363	2	17..1057, 1456..2361	25

3.3.1 GeneID

Tento program je volně dostupný z http://genome.crg.es/geneid.html. Pro predikci využívá metodu dynamického programování popsanou v kapitole 2.7.1. Udávaná přesnost je srovnatelná s ostatními „ab initio“ programy pro predikci exonů. Mezi velkou výhodu tohoto programu patří rychlost predikce a velikost paměti potřebné k predikci. Na webových stánkách tohoto programu uvádí autor, že tento program je schopný zpracovat sekvenci délky 1Gbp za hodinu (na procesoru Intel (R) Xeon CPU 2,80 Ghz). Nesporná přednost spočívá v možnosti sloučit program na predikci s vnějšími důkazy (anotacemi, poznámkami), například už s popsanými geny. Tento vnější důkaz je potřeba načíst z dalšího souboru ve formátu GFF. [12]

Po načtení webové stránky bylo potřeba zadat danou sekvenci a to buď ručně vypsat znaky v sekvenci, nebo načíst ze souboru dat ve formátu FASTA. Dále je zde možnost zadat GFF důkazy, avšak toto není povinný krok. Program byl testován pouze na predikci exonů, proto tato možnost nebyla vyzkoušená. V GeneID existovaly další možnosti nastavení a to především výběr organismu s velké škály možnosti, například člověk, octomilka čtverzubec zelený (druh ryby), ale také pšenice nebo rýže. Program však neumožňoval vybrat prokaryotické organismy a proto na nich nebyl testován. V další části byl nastaven režim predikce a směr vlákna DNA. V posledním úseku byl vybrán výstup, jako nejpřehlednější byl zvolen výstup v podobě GFF. [12]

Samotným výstupem programu byla obrazovka obsahující informace o verzi použitého programu, časovém údaji spuštění programu, ale také důležitější informace jako počet párů bazí sekvence a nejpodstatnější informaci o poloze předpovídaných exonů. V prvním sloupečku šlo vidět, zda se jedná o první exon, vnitřní exon, nebo poslední. Další sloupec obsahoval čísla pozic začátku daného exonu a třetí sloupec čísla pozic, na kterých exon končí. Jak byl program úspěšný, udává Tabulka 6.

Tabulka 6 Výsledky detekce pomocí programu GeneID.

	Pozice exonů předpovězena programem.	**Pozice exonů podle NCBI.**
Homo sapiens 1	2..1957	62..1957
Homo sapiens 2	747..1728, 2208..2848, 3115..3504	747..1728, 2208..2848, 3115..3223, 3329..3504
Homo sapiens 3	26..384, 1342..2089	26..384,1342..2089
Pan troglodytes 1(testováno podle člověka)	58…458, 651..969	1..1014
Pan troglodytes 2 (testováno podle člověka)	501..890, 3248..4435	501..890, 3248..4435

Pongo abelii 1 (testováno podle člověka)	652..1842	652..2049
Pongo abelii 2 (testováno podle člověka)	76..1320	76..1320
Equus caballus 1	Není na výběr.	1..161,268..1052
Equus caballus 2	Není na výběr.	8..257, 618..730, 1259..1472, 2002..2112, 2367..2460, 2541..2679, 4121..4318
Mus musculus 1	367..1035	1..1188
Aspergillus niger 1	11..94, 182..802	11..94, 182..967
Magnaporthe oryzae 1	225..499, 628..1246	225..499, 628..1246
Magnaporthe oryzae 2	183..2450	183..2450
Oryza sativa 1	1..315, 385..525	1..537
Drosophila melanogaster 1	4365..6555, 6617..6747, 6807..7205, 7338..7445	2031..2087, 4365..6555, 6617..6747, 6807..7205, 7338..7445
Caenorhabditis elegans 1	1..864, 935..1219, 1267..4304, 4354..4582, 4628..4763, 4816..5008	1..864, 935..1219, 1267..4304, 4354..4582, 4628..4763, 4816..5075
Caenorhabditis elegans 2	1..1935, 2342..3133, 3995..4126	1..1524, 1579..1792, 1948..2035, 2140..3054, 3675..3881, 3932..4205
Escherichia coli 1	x	154..306, 376..1278, 1355..3079
Escherichia coli 2	x	1..2616, 2914..3198, 4655..4954, 5356..>6230
Escherichia coli 3	x	78..>3252
Arthrobacter crystallopoietes 1	x	1..363, 2382..3920
Arthrobacter crystallopoietes 2	x	1..875, 1363..2610
Arthrobacter crystallopoietes 3	x	1..1327, 1581..2342, 2619..3356
Arthrobacter crystallopoietes 4	x	1..1057, 1269..2258
Listeria monocytogenes 1	x	17..1057, 1456..2361

Z tohoto testování byla spočítána celková senzitivita a specificita programu. Senzitivita: **92,52 %** a specificita: **98,27 %**. Výsledná čísla byla vysoká a to především hodnota specificity. Porovnání programu GeneID vzhledem k ostatním testovaným programům je uvedeno v kapitole 3.4.

3.3.2 GeneMark.hmm

Program dostupný z http://exon.gatech.edu/hmmchoice.html. Predikce je založena na principu skrytých Markovových modelech pospaných v kapitole 2.7.2. Po otevření odkazu se zobrazila hlavní stránka, zde byla možnost zvolit typ organismu, tedy zda je testován prokaryotický, nebo eukaryotický organismus. Právě tato možnost dala značnou výhodu tomuto programu

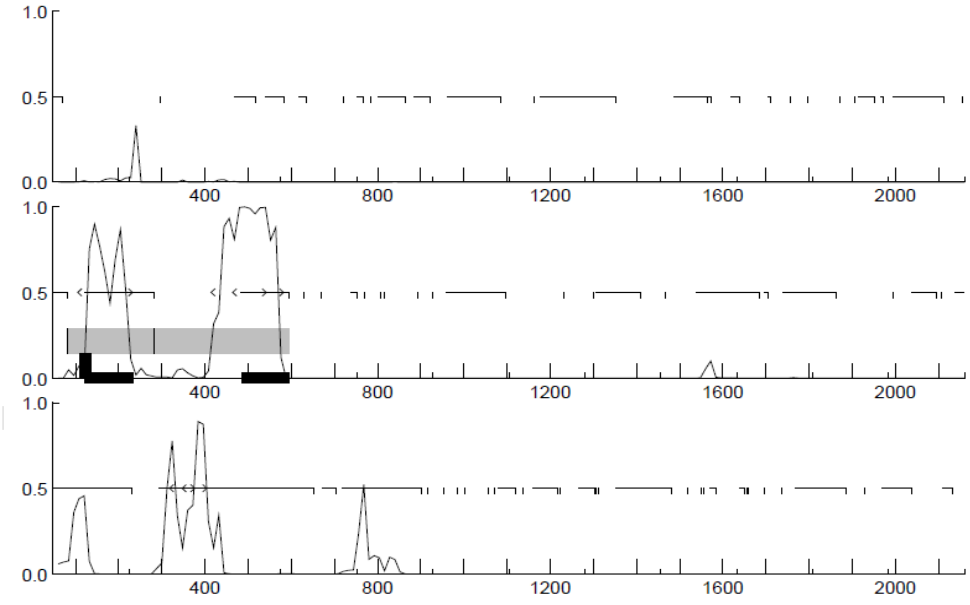
oproti předchozímu. Po vybrání typu organismu bylo nutné zadat testovanou sekvenci, opět zde byla volba mezi ručním zadáním, anebo výběrem ze souboru ve formátu FASTA. Dále bylo nutné zvolit, o jaký organismus se jedná, na výběr bylo asi z 20 druhů eukaryotických organismů. Nakonec se zvolil výstup programu a to ze 4 možností. Textový výstup se zobrazil vždy, navíc zde byla možnost grafického znázornění případně odeslání výstupu na zadanou emailovou adresu.

Výstupem programu bylo okno, ve kterém se nacházela informace o verzi použitého programu, určen byl i chromozom, ze kterého sekvence pocházela, délka sekvence, ale navíc i informace o procentuálním zastoupení C+G nukleotidů v sekvenci. Hlavním výstupem byla tabulka, která obsahovala údaje o nalezených exonech. O jaký typ exonu se jedná, pozice začátků a konců exonů a jejich délku. Pozice exonů, které detekoval program GeneMark.hmm jsou uvedeny v Tabulce 7, grafický výstup je na Obrázku 12.

Tabulka 7 Pozice exonů podle programu GeneMarak.hmm.

	Pozice exonů předpovězena programem.	**Pozice exonů podle NCBI.**
Homo sapiens 1	152…262, 275..1957	62..1957
Homo sapiens 2	747..1728, 2208..2848, 3115..3223, 3329..3504	747..1728,2208..2848,3115..3223,3329..3504
Homo sapiens 3	7..384, 1342..2018	26..384,1342..2089
Pan troglodytes 1 (testováno podle člověka)	78..458, 651..969	1..1014
Pan troglodytes 2 (testováno podle člověka)	501..890, 3248..4435	501..890, 3248..4435
Pongo abelii 1 (testováno podle člověka)	263..447, 652..2011	652..2049
Pongo abelii 2 (testováno podle člověka)	76..1320	76..1320
Equus caballus 1	Není na výběr.	1..161,268..1052
Equus caballus 2	Není na výběr.	8..257, 618..730, 1259..1472, 2002..2112, 2367..2460, 2541..2679, 4121..4318
Mus musculus 1	367..1103	1..1188
Aspergillus niger 1	Není na výběr.	11..94, 182..967
Magnaporthe oryzae 1	Není na výběr.	225..499, 628..1246
Magnaporthe oryzae 2	Není na výběr.	183..2450
Oryza sativa 1	142..525	1..537
Drosophila melanogaster 1	2031..2087, 4365..6555, 6617..6747, 6807..7205, 7338..7445	2031..2087, 4365..6555, 6617..6747, 6807..7205, 7338..7445

Caenorhabditis elegans 1	82..864, 935..1219, 1267..4304, 4354..4582, 4628..4763, 4816..5063	1..864, 935..1219, 1267..4304, 4354..4582, 4628..4763, 4816..5075
Caenorhabditis elegans 2	220..1935, 2342..3133	1..1524, 1579..1792, 1948..2035, 2140..3054, 3675..3881, 3932..4205
Escherichia coli 1	1..306, 376..1278, 1355, 3079	154..306, 376..1278, 1355..3079
Escherichia coli 2	1..3354, 3999..4235, 4270..4407, 4655..4954, 5356..6228	1..2616, 2914..3198, 4655..4954, 5356..>6230
Escherichia coli 3	3..83, 78..3251	78..>3252
Arthrobacter crystallopoietes 1 (Testováno jako Agrobacterium tumefaciens)	1..363, 2418..3920	1..363, 2382..3920
Arthrobacter crystallopoietes 2	3..1145, 1363..2610	1..875, 1363..2610
Arthrobacter crystallopoietes 3	2..1327, 1581..2315, 2619..3356	1..1327, 1581..2342, 2619..3356
Arthrobacter crystallopoietes 4	2..1057, 1290..2258	1..1057, 1269..2258
Listeria monocytogenes 1	17..1057, 1456..2361	17..1057, 1456..2361



Obrázek 12 Část grafického výstupu z programu GeneMark pro sekvenci 1.

Hlavní výhoda tohoto programu spočívala v možnosti testovat prokaryotické i eukaryotické organismy. Dosažená senzitivita: **93,12 %**, specificita: **91,78 %**. Program dosahoval nižších hodnot specificity oproti senzitivitě.

3.3.3 FGENESH

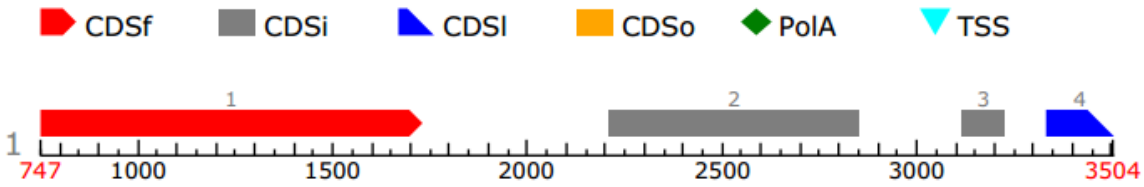
Jedná se o další z programů využívající skrytých Markovových modelů. Dostupný z: http://linux1.softberry.com/berry.phtml?topic=fgenesh&group=programs&subgroup=gfind. Ovládání programu bylo založeno na velice podobném principu. Nejdříve bylo nutné vložit sekvenci, opět se zde dalo volit mezi vložením FASTA souboru nebo ručním vypsáním sekvence nukleotidů. Následně se zvolil organismu, jehož sekvence byla testována. V programu FGENESH byl velký výběr organismů od člověka přes žábu, až k různým druhům ryb. V základním nastavení nebyla možnost jiných změn, ovšem existovala možnost otevření vyššího nastavení, kde se naskytly různé eventuality, které umožňovaly ovlivnit, jak samotnou predikci, tak i výstup programu. Měnit se daly parametry jako minimální délka prvního exonu a vnitřních exonů, hodnota váhy, kdy v případě, že váha daného exonu byla menší než stanovené číslo, nebyl exon brán v úvahu pro predikci a mnoho dalších parametrů ovlivňující predikci.

Výstup byl rozšířen o zobrazení mRNA sekvence predikovaných genů, nebo zobrazení sekvence nukleotidů jednotlivých exonů. Výsledkem programu byly informace o verzi použitého programu, času predikce, o druhu organismu dále se zde nalézaly údaje o délce sekvence počtu predikovaných exonů. Následoval výpis jednotlivých exonů ve sloupečku, kde se opět dalo nalézt pozice začátku a konce exonů a délku jednotlivých exonů. Navíc program uváděl i vypočtené skóre pro dané exony. V případě, že se zadal požadavek, výstup obsahoval i mRNA sekvenci, kterou byl program schopen přeložit do proteinů, případně existovala i možnost nechat vypsat sekvence jednotlivých exonů. Další výhodou programu byla schopnost vytvořit PDF soubor, který obsahoval grafické zobrazení predikovaných exonů. V Tabulce 8 jsou vypsány pozice exonu detekované tímto programem. A na Obrázku 13 je jeho grafický výstup.

Tabulka 8 Výsledné pozice exonů podle programu FGENESH.

	Pozice exonů předpovězena programem.	**Pozice exonů podle NCBI.**
Homo sapiens 1	62..1957	62..1957
Homo sapiens 2	747..1728, 2208..2848, 3115..3223, 3329..3504	747..1728,2208..2848, 3115..3223,3329..3504
Homo sapiens 3	26..384, 1342..2089	26..384,1342..2089
Pan troglodytes 1	1..1014	1..1014
Pan troglodytes 2	501..890, 3248..4435	501..890, 3248..4435
Pongo abelii 1		652..2049
Pongo abelii 2	79..1320	76..1320
Equus caballus 1	Není na výběr.	1..161,268..1052

Equus caballus 2	Není na výběr.	8..257, 618..730, 1259..1472, 2002..2112, 2367..2460, 2541..2679, 4121..4318
Mus musculus 1	Není na výběr.	1..1188
Aspergillus niger 1	11..94, 182..967	11..94, 182..967
Magnaporthe oryzae 1	225..499, 628..1246	225..499, 628..1246
Magnaporthe oryzae 2	183..2450	183..2450
Oryza sativa 1	Není na výběr.	1..537
Drosophila melanogaster 1	2031..2087, 4365..6555, 6617..6747, 6807..7205, 7338..7445	2031..2087, 4365..6555, 6617..6747, 6807..7205, 7338..7445
Caenorhabditis elegans 1	1..864, 935..1219, 1267..4304, 4354..4582, 4628..4763, 4816..5075	1..864, 935..1219, 1267..4304, 4354..4582, 4628..4763, 4816..5075
Caenorhabditis elegans 2	1..1935	1..1524, 1579..1792, 1948..2035, 2140..3054, 3675..3881, 3932..4205
Escherichia coli 1	x	154..306, 376..1278, 1355..3079
Escherichia coli 2	x	1..2616, 2914..3198, 4655..4954, 5356..>6230
Escherichia coli 3	x	78..>3252
Arthrobacter crystallopoietes 1	x	1..363, 2382..3920
Arthrobacter crystallopoietes 2	x	1..875, 1363..2610
Arthrobacter crystallopoietes 3	x	1..1327, 1581..2342, 2619..3356
Arthrobacter crystallopoietes 4	x	1..1057, 1269..2258
Listeria monocytogenes 1	x	17..1057, 1456..2361



Obrázek 13 Grafický výstup z PDF souboru z programu FGENESH pro sekvenci Homo sapiens 2.

Tento program dosahoval senzitivity: **93,74 %**, a specificity: **98,73 %**.

3.3.4 Implementovaná funkce Predikce_exonu

Jedná se o program, který byl navržen jako hlavní úkol této bakalářské práce. Tato kapitola obsahuje jeho testování na 25 testovacích sekvencích. V této zkoušce byly použity doporučené hodnoty délky okna (390) a prahu (max(suma)/3), které byly zjištěny na základě analýzy popsané v kapitole 3.2. Byla vypočtena senzitivita a specificita a v následující kapitole jsou porovnány všechny 4 programy. Výsledné pozice, které program označil jako exony, se nachází v Tabulce 9.

Tabulka 9 Výsledky predikce pomocí programu Predikce_exonu.

	Pozice exonů předpovězena programem.	**Pozice exonů podle NCBI.**
Homo sapiens 1	104..136, 190..322, 411..884, 902..997, 1123..1140, 1274..1846	62..1957
Homo sapiens 2	933..1650, 2209..2648, 2691..2707, 3188..3360	747..1728,2208..2848,3115..3223,3329..3504
Homo sapiens 3	68..426, 1291..1322, 1342..2008	26..384,1342..2089
Pan troglodytes 1	76..893	1..1014
Pan troglodytes 2	543..575, 602..775, 3195..4426	501..890, 3248..4435
Pongo abelii 1	276..333, 516..806, 823..1065, 1082..1104, 1189..1215, 1284..1297, 1472..1959	652..2049
Pongo abelii 2	176..201, 231..1266	76..1320
Equus caballus 1	411..959	1..161,268..1052
Equus caballus 2	1146..1521, 4108..4317	8..257, 618..730, 1259..1472, 2002..2112, 2367..2460, 2541..2679, 4121..4318
Mus musculus 1	436..1059	1..1188
Aspergillus niger 1	196..351, 398..967	11..94, 182..967
Magnaporthe oryzae 1	225..506, 610..1235	225..499, 628..1246
Magnaporthe oryzae 2	465..1440, 1465..1485, 1557..1588, 1726..1918, 1933..2072	183..2450
Oryza sativa 1	1..537	1..537
Drosophila melanogaster 1	4461..6407, 6898..7102	2031..2087, 4365..6555, 6617..6747, 6807..7205, 7338..7445

Caenorhabditis elegans 1	193..232, 266..783, 1017..1387, 1502..1535, 1546..3452, 3619..3631, 3656..3668, 3679..4267, 4457..4815	1..864, 935..1219, 1267..4304, 4354..4582, 4628..4763, 4816..5075
Caenorhabditis elegans 2	75..756, 823..1100, 1123..1228, 1907..2018, 2045..2287, 2704..2755, 2784..2835, 4001..4012	1..1524, 1579..1792, 1948..2035, 2140..3054, 3675..3881, 3932..4205
Escherichia coli 1	401..1219, 1465..2927	154..306, 376..1278, 1355..3079
Escherichia coli 2	111…3252, 5545..6145	1..2616, 2914..3198, 4655..4954, 5356..>6230
Escherichia coli 3	377..3126	78..>3252
Arthrobacter crystallopoietes 1	2502..3863	1..363, 2382..3920
Arthrobacter crystallopoietes 2	82..854, 871..884, 1507..2583	1..875, 1363..2610
Arthrobacter crystallopoietes 3	17..1212, 1659..2247, 2764..3380	1..1327, 1581..2342, 2619..3356
Arthrobacter crystallopoietes 4	350..952, 1270..2213	1..1057, 1269..2258
Listeria monocytogenes 1	44..997, 1485..1500, 1523..2192	17..1057, 1456..2361

Dosažené hodnoty senzitivity: **80,36 %** a specificity: **96,28 %**. Program s doporučeným nastavením nedosahoval vysokých hodnot senzitivity, těch by se dalo, dosáhnou snížením velikosti prahu, avšak bylo by zde riziko, vysokého poklesu specificity.

Jelikož program pracuje na základě periodicity tří, je důležité, aby testované sekvence tuto periodicitu obsahovaly. V případě, že ji neměly, nebo nebyla výrazná, nedocházelo k přesným detekcím a nalezené pozice byly zkreslené. Jedním z dalších problémů bylo testování sekvencí, které neobsahovaly výrazné exony, nebo jeden výrazný exon a další menší exony. Tato komplikace byla dána především navázáním velikosti prahu na maximum *sumy*. V případě, že by sekvence měla jeden výrazný exon, výsledkem by byl vysoký práh a kratší exony by nebyly detekovány. Chyba také mohla nastat pro případ, kdy mezi exony v sekvenci byly jen malé vzdálenosti. Tímto mohlo docházek ke spojení blízkých exonů.

Obecně pokud by měl program raději zachytit větší množství správných exonů a nevadilo by, že by označil více pozic za exony, které exony nejsou, bylo by vhodné zvolit nižší hodnotu prahu. Program by tedy dosahoval vyšších hodnot senzitivity, ale nižších hodnot specificity.

3.4 Srovnání programů

Každý z testovaných programů měl určité výhody a přednosti, ale také své zápory. Jednotlivé programy se svými klady a vadami byly popsány v kapitolách týkajících se daného programu.

Avšak pro větší přehlednost byla vytvořena Tabulka 10 obsahující výhody jednotlivých programů a Tabulka 11, ve které jsou jejich negativa.

Tabulka 10 Výhody jednotlivých programů.

	GeneID	**GeneMark.hmm**	**FGENESH**	**Predikce_exonu**
Výhody	Rychlost (tuto výhodu bychom ocenili až při predikci sekvencí větší délky)	Možnost predikce prokaryotických organismů.	Nastavení vah (využije především zkušenější uživatel, kdy si může přizpůsobit predikci svým potřebám)	Možnost predikce prokaryotických organismů.
	Možnost kombinace s vnějšími důkazy. Srovnání už s popsanými geny.	Grafický výstup.	Grafický výstup. Možnost zadání minimální délky exonů.	Grafický výstup.
			Vypsání sekvence mRNA, jednotlivých exonů, případně jejich translace do proteinů.	
	Senzitivita: 92,52 % Specificita: 98,27 %	Senzitivita: 93,12%	Senzitivita: 93,74 % Specificita: 98,73 %	Specificita: 96,28 %

Tabulka 11 Nevýhody jednotlivých programů.

	GeneID	GeneMark.hmm	FGENESH	Predikce_exonu
Nevýhody	Nemožno vybrat pro predikci prokaryotický organismus.	Nižší hodnota specificity: 91,78%	Nemožno vybrat pro predikci prokaryotický organismus.	Dlouhá výpočetní doba. Testované sekvence nebyly dlouhé, ale pro sekvence delší by se jednalo o problém
	Není grafický výstup.			Není vhodné pro všechny sekvence. Testovat můžeme všechny, ale je určeno pro sekvence, které obsahují periodicitu 3.
				Nižší hodnota senzitivity: 80,36%

4 Závěr

Hlavní úkolem této bakalářské práce bylo vytvořit program pro vyhledávání exonů v prokaryotických a eukaryotických organismech. Tento program byl otestován a jeho přesnost byla porovnána se třemi volně dostupnými programy, které jsou schopny vyhledávání pozic exonů.

Práce se skládá z teoretické části, kde je popsán rozdíl mezi eukaryotickými a prokaryotickými organismy, stručný popis RNA a DNA z hlediska jejich struktury, funkce a rozdílu mezi organismy. Následně jsou zde popsány průběhy transkripce translace a vysvětlen rozdíl mezi exony a introny. Druhý úsek teoretické části je věnován popisu metod, které se využívají pro predikci exonů a to metoda dynamického programování, metoda skrytých Markovových modelů, neuronové sítě a Fourierova transformace.

V praktické části této práce je popsán realizace programu *Predikce_exonu*. Tento program pracuje na základě diskrétní Fourierovy transformace. Umožňuje načíst testovanou sekvenci a v této sekvenci vyhledat pozice exonů. Pro správnou funkci programu je nutné zadat dvě vstupní proměnné. Hodnotu délky okna a velikosti prahu. Správnost predikce je velice ovlivněna právě těmito parametry. Z tohoto důvodu byla provedena analýza programu na čtyřech sekvencích s cílem zjistit optimální hodnoty. Tato analýza je popsána v praktické části práce.

Poslední oddíl praktické části se zabývá testováním vytvořeného programu *Predikce_exonu* a testováním a popisem třech volně dostupných programů, z nichž dva využívají metodu skrytých Markovových modelů a jeden metodu dynamického programování. Pro všechny čtyři programy byly spočteny hodnoty senzitivity a specificity a na základě těchto hodnoty byly programy porovnány a hodnoceny.

Program GeneID, který využívá metodu dynamického programování, byl ze všech programů nejrychlejší. Tato výhoda však nebyla plně doceněna, jelikož byly testovány jen krátké sekvence s malým počtem exonů. Jeho nevýhodou byla nemožnost predikce prokaryotických organismů, také chyběl grafický výstup. Při testování dosáhl hodnot Senzitivity: **92,52 %** a specificity: **98,27 %.**

Program GeneMark.hmm využíval skryté Markovovy modely, měl výhodu v možnosti predikce, jak eukaryotických, tak prokaryotických organismů, jeho přednost spočívala také v podrobném grafickém výstupu. Z hlediska přesnosti predikce dosahoval hodnot senzitivity: **93,12 %**, specificity: **91,78 %.**

Posledním, z volně dostupných programů, byl program FGENESH. Využíval také skryté Markovovy modely. Mezi jeho přednosti patřila především největší databáze druhů.

Poskytoval méně podrobný, ale přehledný grafický výstup. Jeho hlavní nevýhody spočívaly v nemožnosti predikce prokaryotických organismů. Avšak hlavní výhodou, kterou ocení především zkušenější uživatelé, byla možnost nastavení mnoha parametrů predikce. Program dosáhl hodnot senzitivity: **93,74 %** a specificity: **98,73 %**.

Jako poslední byl testován program *Predikce_exonu*. Jednalo se o program vytvořený jako hlavní úkol bakalářské práce. Program dosahoval hodnot senzitivity: **80,36 %** a specificity: **96,28 %**.

Navržený program dosahoval nižších hodnot senzitivity, než zbývající testované programy, avšak hodnota specificity je se zbývajícími programy srovnatelná v jednom případě dokonce vyšší. Pro přesnější predikci by bylo vhodné provést rozsáhlejší analýzu programu a zjistit další vhodné velikosti vstupních hodnot, délky okna a prahu.

5 Zdroje

[1] CAMPBELL, Neil A a Jane B REECE. *Biologie*. Vyd. 1. Brno: Computer Press, 2006, xxxiv, 1332 s. ISBN 80-251-1178-4.

[2] WANG, Zhuo; CHEN, Yazhu; LI, Yixue. *A Brief Review of Computational Gene Prediction Methods.* Geno. Prot. Bioinfo., 2004, vol. 2(4), pp. 216-221.

[3] MATHÉ, Catherine; SAGOT, Marie-France; SCHIEX, Thomas; ROUZÉ, Pierre. *Current methods of gene prediction, their strengths and weaknesses.* Nucleic Acids Research, 2002, vol. 30(19), pp. 4103-4117.

[4] NEČAS, Oldřich et al. *Obecná biologie: pro lékařské fakulty*. Jinočany: H+H, 2000, 554 s. ISBN 80-86022-46-3.

[5] ALBERTS, Bruce. *Základy buněčné biologie: úvod do molekulární biologie buňky*. 2. vyd. Překlad Arnošt Kotyk, Bohumil Bouzek, Pavel Hozák. Ústí nad Labem: Espero, c1998, xxvi, 630, G-18, A-62, I-30 s. ISBN 80-902-9062-0.

[6] SNUSTAD, D a Michael J SIMMONS. *Genetika*. 5th ed. Překlad Jiřina Relichová. Brno: Masarykova univerzita, 2009, xxi, 871 s. ISBN 978-802-1048-522.

[7] ROSYPAL, Stanislav. *Úvod do molekulární biologie: úvod do molekulární biologie buňky*. 4. inovované vyd. Překlad Arnošt Kotyk, Bohumil Bouzek, Pavel Hozák. Brno : Espero, 2006, 289 s. ISBN 80-902-5625-2.

[8] Bioinformatika - *02 Sekvence a databáze*, Brno: VUT, FEKT, ÚBMI, 2012.

[9] SYNDER, Eric a Gary STORMO. *Identification of coding regions in genomic DNA sequences: an application of dynamic programming and neural networks. Nucleic Acids Research*. 1992, Vol. 21, No. 3, s. 607-613

[10] Bioinformatika - *Návod do 12. počítačového cvičení.* Brno: VUT, FEKT, ÚBMI, 2012, 3 s.

[11] KOZUMPLÍK, J.; PROVAZNÍK, I. *Umělá inteligence v medicíně.* Brno: FEKT VUT v Brně, 2007.

[12] CAMARA, Francisco. *Geneid homepage*. [online]. [cit. 2012-11-18]. Dostupné z:http://genome.crg.es/software/geneid/index.html#tophttp://genome.crg.es/software/geneid/index.html#top

[13] *National Center for Biotechnology Information* [online]. [cit. 2012-11-18]. Dostupné z: http://www.ncbi.nlm.nih.gov/

[14] Umělá inteligence v medicíně – *Neuronové sítě 4*, Brno: VUT, FEKT, ÚBMI, 2012.

[15] RŮŽEK, Václav. *Algoritmy pro rozpoznání ručně psaných znaků.* Zlín: Univerzita Tomáše Bati ve Zlíně. Fakulta aplikované informatiky. Ústav řízení procesů, 2010, 66 s. Vedoucí bakalářské práce Ing. Petr Chalupa, Ph.D.

[16] HUSON, Daniel. *Algorithms in Bioinformatics II.* Universiät Tübingen, 2004.

[17] Úvod do medicínské informatiky – *Pravděpodobnostní usuzování v medicíně*. Brno: VUT, FEKT, ÚBMI, 2012.

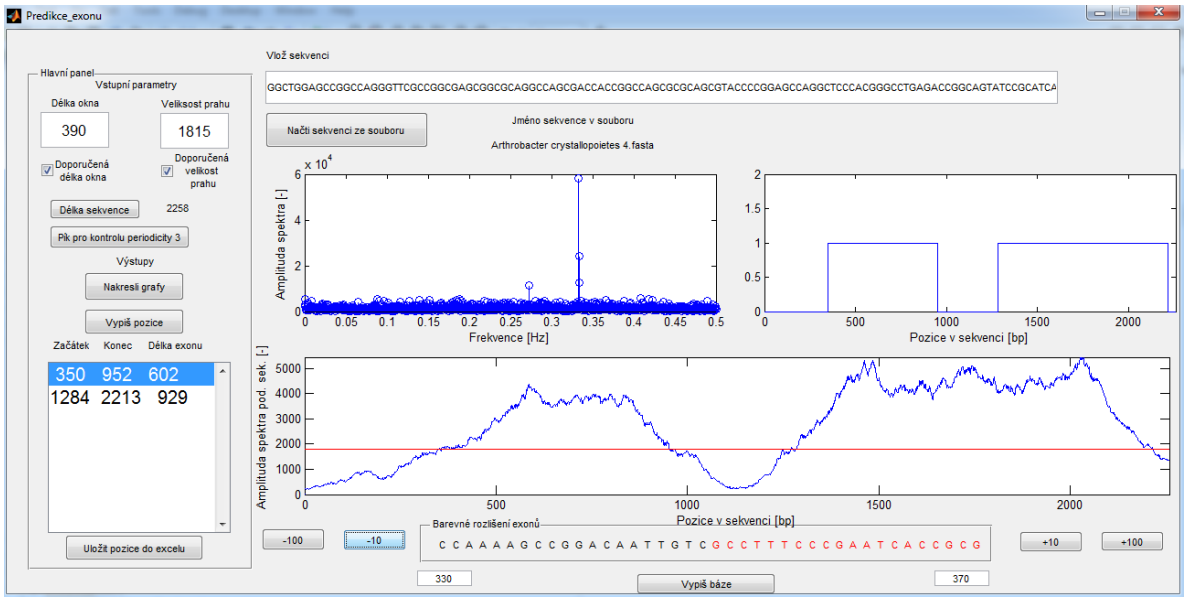
6 Přílohy

A Seznam zkratek

DNA	Deoxyribonukleová kyselina
RNA	Ribonukleová kyselina
mRNA	Mediátorová ribonukleová kyselina
G	Guanin
T	Tymin
C	Cytosin
A	Adenin
bp	Páry bazí
ORF	Open reading frame (otevřené čtecí rámce)
EST	Expressed sequence tags
DP	Dynamické programování
HMM	Skryté Markovovy modely
NN	Neuronové sítě
DFT	Diskrétní Fourierova transformace

B Uživatelský manuál

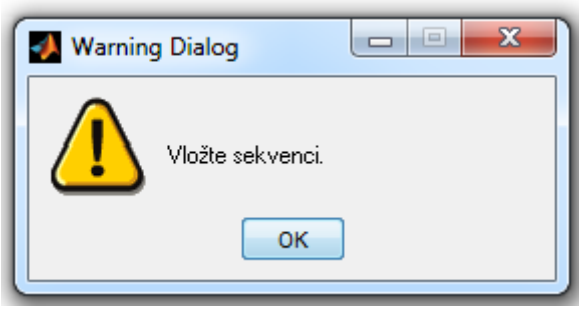
Program *Predikce_exonu* byl navržen pro vyhledávání exonů v sekvencích prokaryotických a eukaryotických organismů. Sekvenci můžeme načíst ze souboru a to ve formátu FASTA, nebo ji vypsat do připraveného okna. Program zobrazí tři grafy, kde ze dvou vychází následná predikce a třetí graf zobrazuje pozice predikovaných exonů. Vypíše pozice začátků a konců nalezených exonů spolu s jejich délkou.



Obrázek 14 Příkald grafického prostření v programu Predikce_exonu.

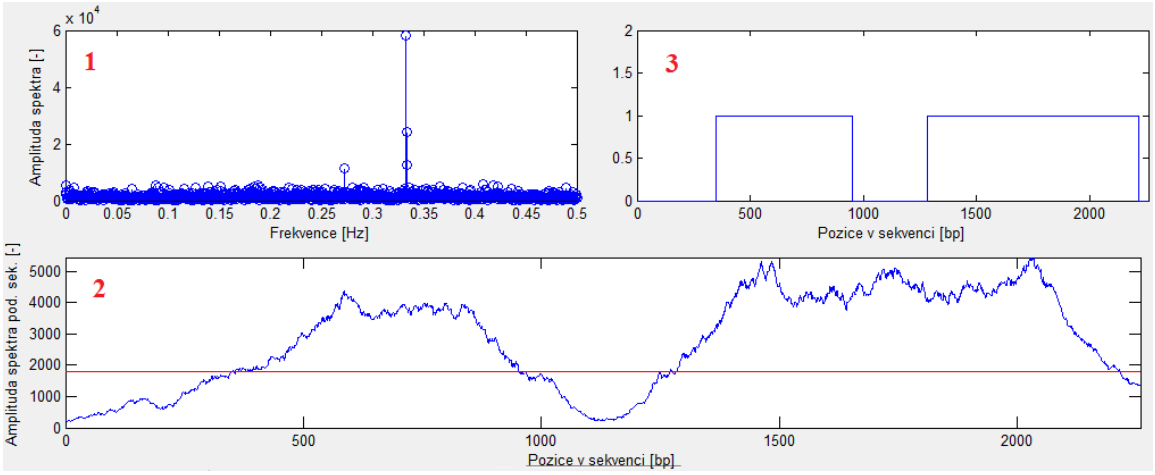
1. Program spusťte stisknutím tlačítka F5 v souboru *Predikce_exonu*.

2. Ze všeho nejdřív musíte získat sekvenci, kterou chcete testovat. Máte dvě možnosti jak sekvenci zadat. První možností je sekvenci zadat ručně do připraveného *edit okna* označeného **Vlož sekvenci**. V tomto případě, musíte zmáčknout tlačítko ***Délka sekvence***, aby program zjistil počet nukleotidů v testované sekvenci. Druhým způsobem můžete sekvenci načíst ze souboru, v případě, že ji máte uloženou ve formátu FASTA. Toto načtení provedete tlačítkem **Načti sekvenci ze souboru.** Poté se objeví sled nukleotidů sekvence v edit okně a do připraveného místa pod text **Jméno sekvence v souboru** se vypíše název sekvence, shodný s názvem uloženého souboru FASTA. V případě, že by sekvence nebyla načtena, vypíše se chybová hláška a nebude možno pokračovat dále. Příklad chybové hlášky je uveden na Obrázku 15.



Obrázek 15 Příklad chybové hlášky v případě nezadání sekvence.

3. Po úspěšném načtení sekvence, je nutné nechat si vykreslit spektrum sekvence, abyste zjistili, zda se v sekvenci nalézá periodicita rovna třem. Pomocí tlačítka **Spektrogram sekvence**, se do připraveného grafického pole **pik** vykreslí spektrum sekvence. V případě, že zde bude přítomný pík na pozici 1/3, můžete pokračovat v dalším postupu. Pokud by se zde pík nevyskytoval, je nutné zvážit správnost dalšího postupu. Na Obrázku 16 odpovídá grafickému poli pro vykreslení spektra sekvence pole číslo 1.



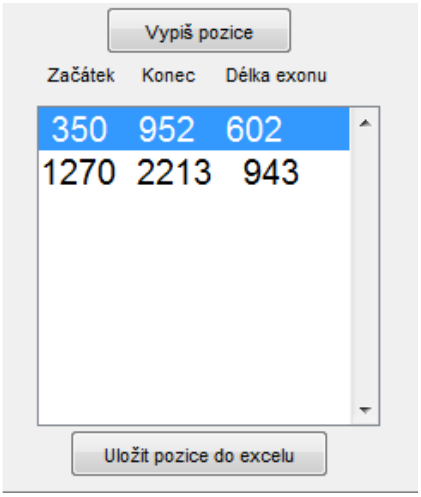
Obrázek 16 Pozice grafických výstupu v programu.

4. Pokračujte, zadáním délky okna. Délku okna zadávejte v edit políčku **Délka okna**. Můžete zadat jak vlastní zvolenou hodnotu délky okna, tak pracovat s hodnotu doporučenou. V případě špatně zadaných vlastních hodnot se řiďte podle chybových hlášek. Doporučenou hodnotu vyberete pomocí checkboxu **Doporučená délka okna**.

5. Postupujte k zadání velikosti prahu. Zde opět existují dvě možnosti, buď můžete zadat velikost prahu ručně do připraveného edit pole **Veliksost prahu**, nebo je zde možnost nechat si spočítat doporučenou hodnotu velikosti prahu. Volbu doporučených hodnot potvrdíte pomocí checkboxu **Doporučená velikost prahu**.

6. Po zdání vstupních hodnot stiskněte tlačítko **Nakresli grafy**. Tímto tlačítkem se realizuje hlavní úsek programu, od vypočtení spektra podél sekvence a jeho vykreslení až po detekci pozic začátků a konců exonů a jejich grafické zobrazení do příslušných grafických polí. Vykreslení spektra podél sekvence odpovídá grafické pole číslo 2 v Obrázku 16 a ve stejném obrázku odpovídá oblast číslo 3 místu pro vykreslení pozic nalezených exonů.

7. Pro vypsaní detekovaných pozic stiskněte tlačítko **Vypiš pozice** a do připraveného listboxu se vám vypíšou nalezené začátky a konce exonů spolu s délkou jednotlivých exonů. Tento výstup si můžete uložit do excelu ve formátu .xls pomocí políčka **Uložit pozice do excelu**. Tuto oblast můžete vidět na Obrázku 17.



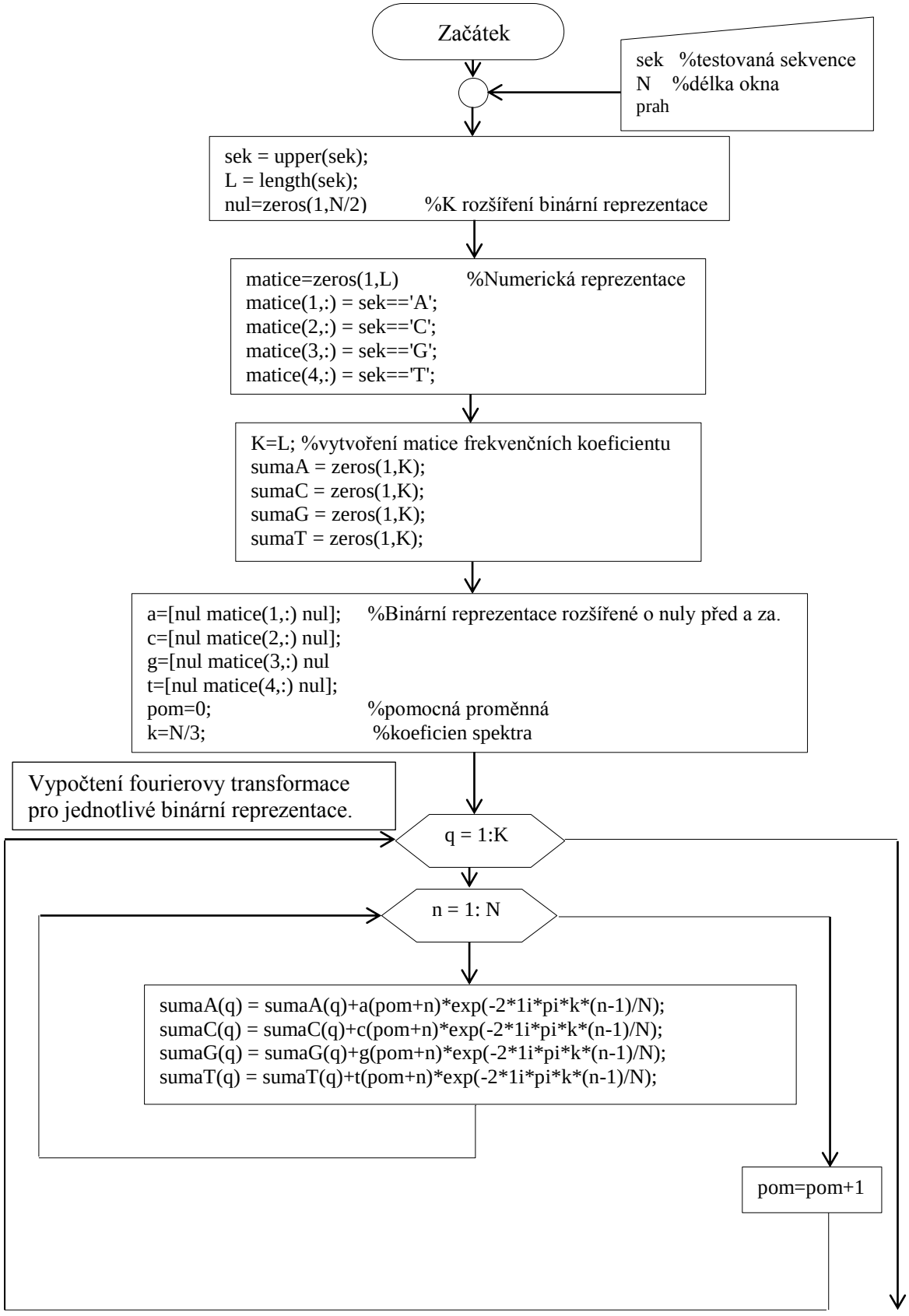
Obrázek 17 Vypsání a uložení detekovaných pozic.

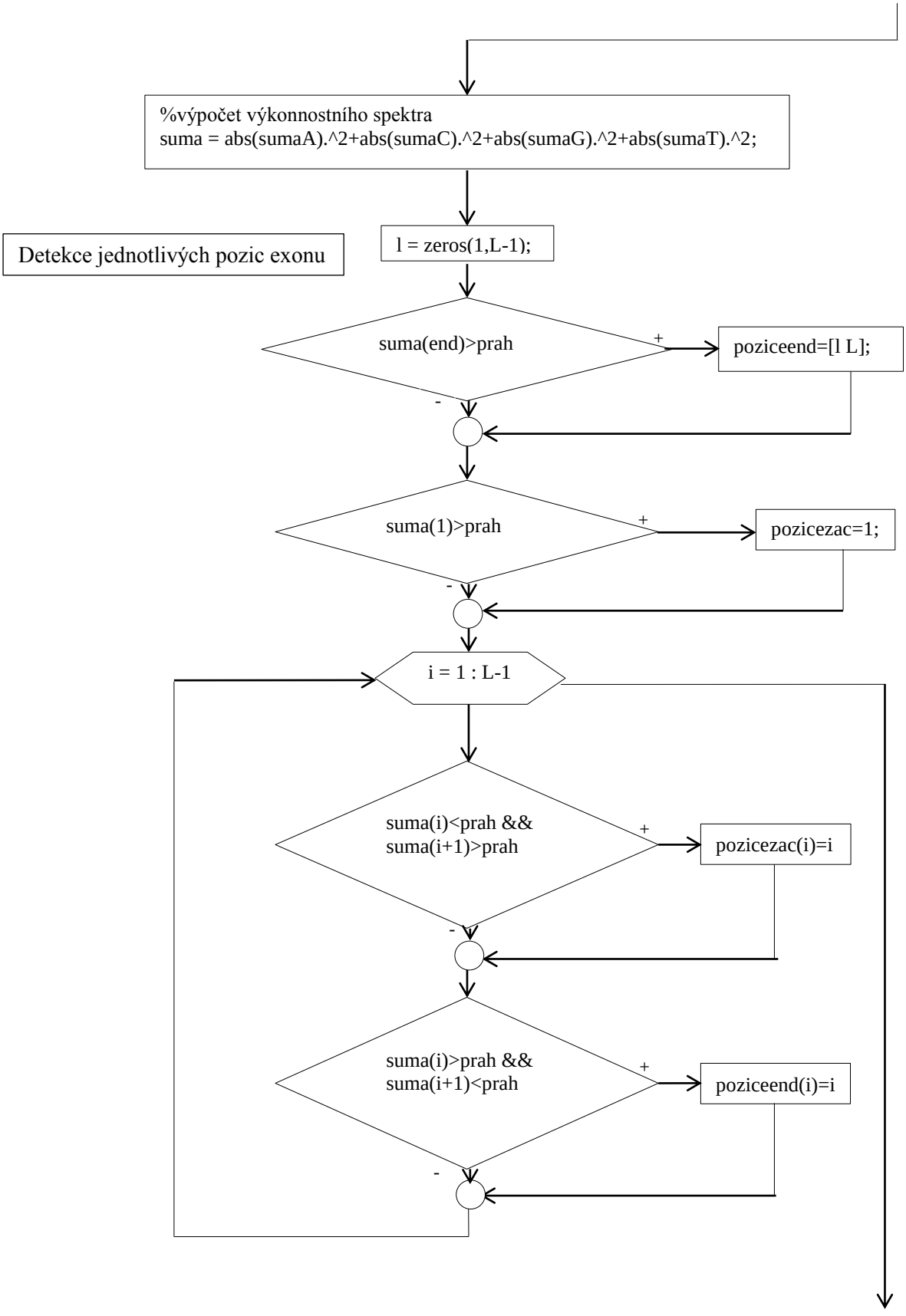
8. Poslední části programu je možnost vypsání a barevného znázornění exonů v sekvenci. Tento krok realizujete pomocí tlačítka **Vypiš báze** a do políčka **Barevné rozlišení exonů** se vypíše prvních 40 nukleotidů sekvence. Exony jsou zde zvýrazněny červenou barvou, nukleotidy, které program neoznačil, jako exony jsou barvou černou. Pohybovat se po sekvenci můžete pomocí tlačítek vpravo a vlevo a to vždy o +10, +100, -10 a -100, nebo pomocí přepisování čísel v edit oknech pod políčkem ve kterém jsou vypsány nukleotidy. Tuto oblast čelního panelu můžete vidět na Obrázku 18.

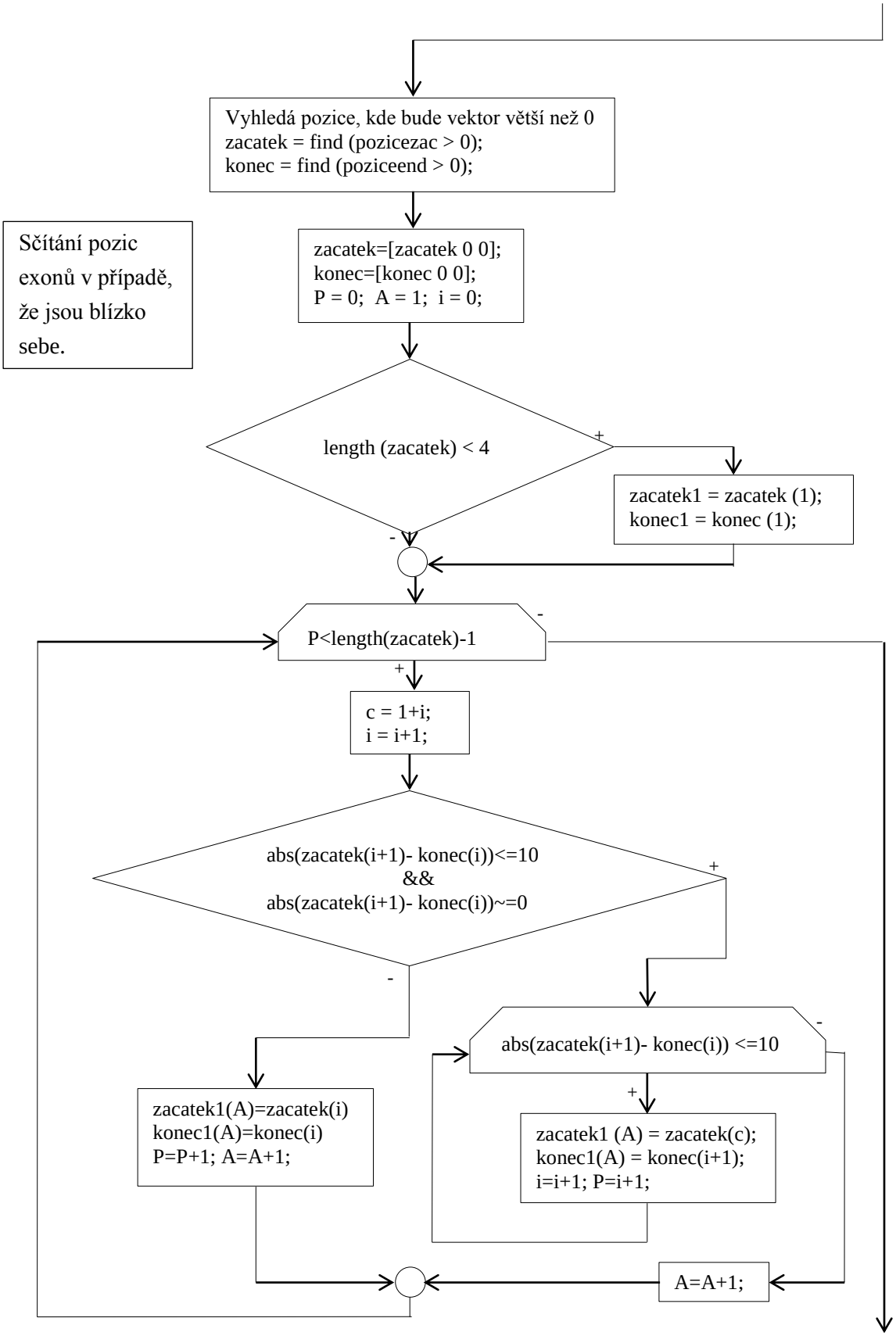
Obrázek 18 Barevné rozlišení exonů.

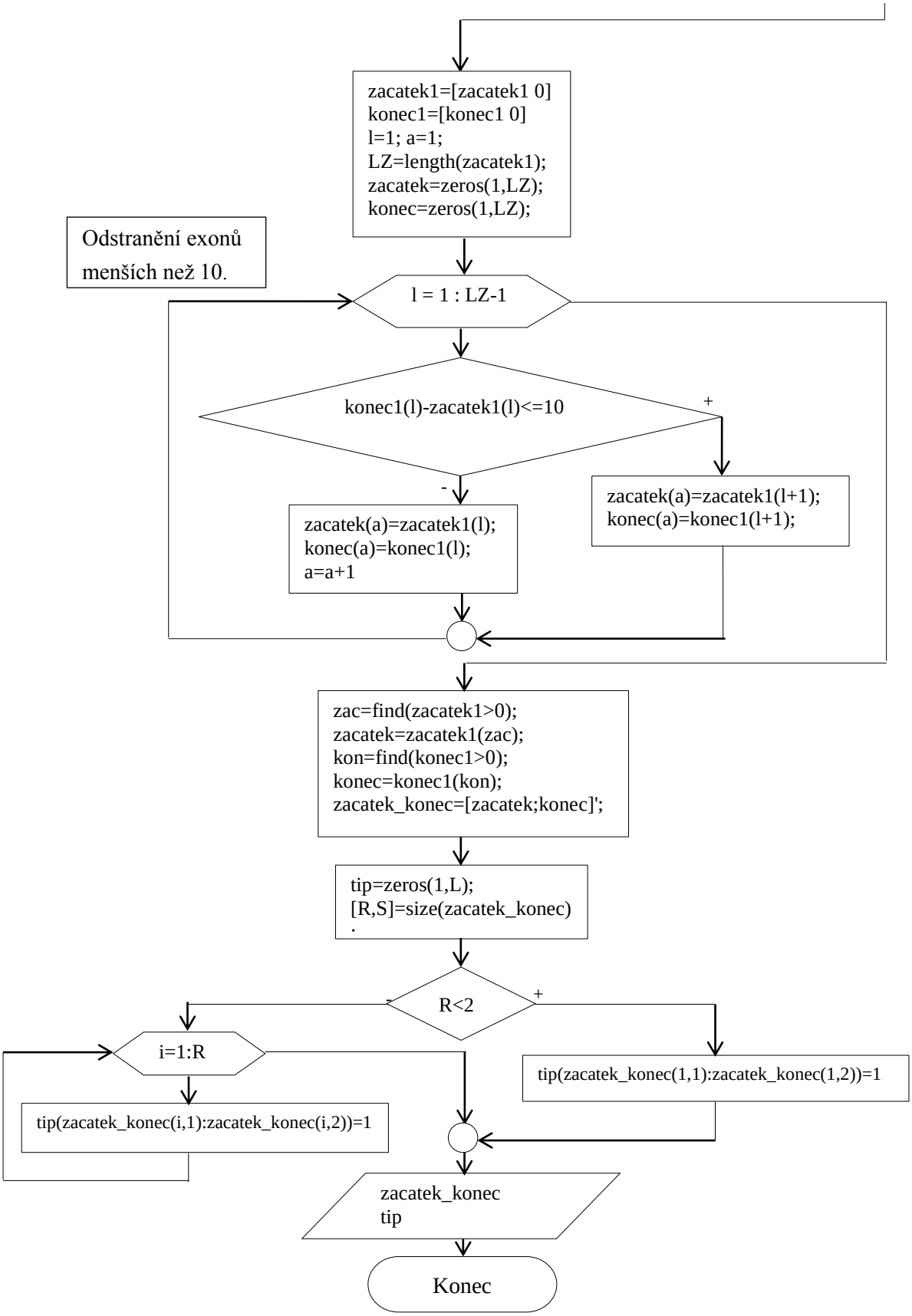
9. Program zavřete kliknutím na křížek v horní části obrazovky.

C Blokový diagram




%výpočet výkonnostního spektra
suma = abs(sumaA).^2+abs(sumaC).^2+abs(sumaG).^2+abs(sumaT).^2;
Detekce jednotlivých pozic exonu
l = zeros(1,L-1);
suma(end)>prah
+
poziceend=[l L];
-
suma(1)>prah
+
pozicezac=1;
-
i = 1 : L-1
suma(i)<prah &&
suma(i+1)>prah
+
pozicezac(i)=i
-
suma(i)>prah &&
suma(i+1)<prah
+
poziceend(i)=i
-


Vyhledá pozice, kde bude vektor větší než 0
zacatek = find (pozicezac > 0);
konec = find (poziceend > 0);
Sčítání pozic exonů v případě, že jsou blízko sebe.
zacatek=[zacatek 0 0];
konec=[konec 0 0];
P = 0; A = 1; i = 0;
length (zacatek) < 4
+
-
zacatek1 = zacatek (1);
konec1 = konec (1);
P<length(zacatek)-1
-
+
c = 1+i;
i = i+1;
abs(zacatek(i+1)- konec(i))<=10
&&
abs(zacatek(i+1)- konec(i))~=0
+
-
abs(zacatek(i+1)- konec(i)) <=10
-
+
zacatek1(A)=zacatek(i)
konec1(A)=konec(i)
P=P+1; A=A+1;
zacatek1 (A) = zacatek(c);
konec1(A) = konec(i+1);
i=i+1; P=i+1;
A=A+1;


Odstranění exonů menších než 10.
zacatek1=[zacatek1 0]
konec1=[konec1 0]
l=1; a=1;
LZ=length(zacatek1);
zacatek=zeros(1,LZ);
konec=zeros(1,LZ);
l = 1 : LZ-1
konec1(l)-zacatek1(l)<=10
+
-
zacatek(a)=zacatek1(l+1);
konec(a)=konec1(l+1);
zacatek(a)=zacatek1(l);
konec(a)=konec1(l);
a=a+1
zac=find(zacatek1>0);
zacatek=zacatek1(zac);
kon=find(konec1>0);
konec=konec1(kon);
zacatek_konec=[zacatek;konec]';
tip=zeros(1,L);
[R,S]=size(zacatek_konec)
.
R<2
-
+
i=1:R
tip(zacatek_konec(i,1):zacatek_konec(i,2))=1
tip(zacatek_konec(1,1):zacatek_konec(1,2))=1
zacatek_konec
tip
Konec