VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

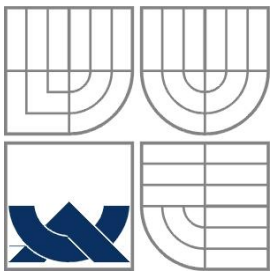
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

# DETEKCIA GÉNOV V DNA SEKVENCIÁCH

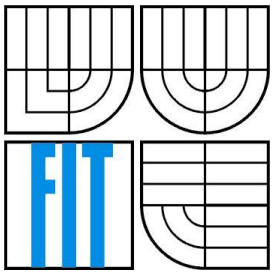
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE Marek Višňovský
AUTHOR

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

# DETEKCE GENŮ V DNA SEKVENCÍCH

GENE DETECTION IN DNA SEQUENCES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — Marek Višňovský
AUTHOR

VEDOUCÍ PRÁCE — Ing. Tomáš Martínek PhD.
SUPERVISOR

BRNO 2011

# Abstrakt

Práce se zaobírá návrhem metody na predikci prokaryotických genů, která pak bude odzkoušená a prezentována na genome *Escherichie coli*. Jednotlivé kapitoly postupně pojednávají o problematice spojené s tímto návrhem, stručným úvodem do molekulární biologie začínajíc, představením používaných metod detekce respektive predikce genů pokračujíc, až samotným návrhem metody založené především na pozičně specifických maticích končíc.



# Abstract

The main goal of this work is to create a method for gene prediction in prokaryotic genomes, which will be later demonstrated and tested on *Escherichia coli*. The first chapter contains a short introduction into molecular biology. In the second one, we take a closer look on current methods of gene detection and prediction and the work ends with an application of acquired knowledge on gene prediction mostly using position weight matrices.



# Klíčová slova

Skryté Markovovy modely, pozičně specifická matice, predikce prokaryotických genů

# Keywords

Hidden Markov Models, Position Weight Matrix, prokaryotic gene prediction


# Citace

Višňovský Marek: Detekcia génov v DNA sekvenciách, bakalářská práce, Brno, FIT VUT v Brně, 2011

# Detekcia génov v DNA sekvenciách

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Tomáše Martínka PhD.
Další informace mi poskytla Katarína Bibzová.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Marek Višňovský
5.5.2011

## Poděkování

Mé poděkování patří především vedoucímu Ing. Tomášovi Martínkovi PhD., který mne vedl k prezentovanému řešení a poskytl nespočet odborných rad, ale také Kataríne Bibzovej, která mi poskytla doplňující informace a materiály týkající se struktury a způsobu uložení genů v buňce.

# Obsah

1 Úvod ..... 2
2 Základy molekulárnej biológie ..... 3
2.1 DNA ..... 3
2.2 Transkripcia ..... 4
2.2.1 Transkripcia v prokaryotických bunkách ..... 5
2.2.2 Transkripcia v eukaryotických bunkách ..... 6
2.3 Translácia ..... 7
3 Detekcia génov ..... 9
3.1 Detekcia génov na základe podobností ..... 9
3.1.1 Smith-Watermanov algoritmus ..... 9
3.2 Predikcia génov ..... 11
3.2.1 Markovovské reťazce ..... 11
3.2.2 Skryté Markovovské modely (HMM) ..... 12
3.2.3 Pozične špecifické matice ..... 14
4 Experimentálna časť ..... 16
4.1 Poznámky k implementácii ..... 16
4.2 Návrh metódy predikcie génov ..... 16
4.2.1 Inicializácia transkripcie, detekcia promótorov ..... 17
4.2.2 Začiatok prekladu, hľadanie Shine-Dalgarna ..... 22
4.2.3 Koniec prekladu, odhad správneho ORF ..... 24
4.2.4 Terminácia transkripcie, detekcia terminátorov ..... 26
4.3 Zlúčenie ..... 28
5 Záver ..... 31
Literatúra ..... 32
Prílohy ..... 33

# 1 Úvod

Od prvého úspešného sekvenovania DNA roku 1977, mnohé projekty prinášajú nové reťazce DNA rýchlejšie ako ich dokážeme analyzovať. Preto vzniká potreba automatizácie čo najväčšej časti analýzy týchto sekvencií, aby sme pred samotným výskumom v laboratórnych podmienkach mali čo najviac informácií o týchto reťazcoch. Obľúbeným testovancím subjektom sa v tomto smere stala baktéria *Escherichia coli*, o ktorej genóme tak máme najviac znalostí, čo využijeme pri prezentácii a testovaní navrhnutej metódy predikcie génov.

Detekcia alebo predikcia génov slúži predovšetkým k identifikácii miest, pozícií, v DNA, na ktorých sa nachádzajú gény kódujúce proteíny. Z toho sme schopní vyčítať ich zloženie, čo robí z predikcie génov prvý krok k určeniu ich 3D štruktúr a funkcií, ktoré v bunke zastávajú. Napríklad spomínaná *E. coli* je tiež využívaná ako pomocná zložka pri výrobe inzulínu, za čo sú zodpovedné určité gény, u ktorých nám znalosť ich štruktúry môže napomôcť k syntéze tejto zložky bez potreby pestovania samotnej baktérie.

Nasledujúce strany popisujú cestu vedúcu k návrhu metódy predikcie génov v prokaryotických organizmoch, demonštrovanej a testovanej na *Escherichie coli*, stručným predstavením ústrednej dogmy molekulárnej biológie začnúc a samotným návrhom spolu s implementáciou navrhnutej metódy končiac.

Kapitola 2 je stručným úvodom do molekulárnej biológie. Keďže je však problematika molekulárnej biológie pomerne široká, tak sa v tomto smere obmedzíme výhradne na znalosti potrebné k detekcii génov. Nasleduje kapitola 3 približujúca metódy a algoritmy používané pri detekcii génov. Ďalšou v poradí je kapitola 4 venovaná rozšíreniu znalostí o štruktúre prokaryotických génov, návrhu a následnej implementácii metódy predikcie prokaryotických génov. V záverečnej kapitole 5 zhodnotíme výsledky práce a načrtneme jej ďalšie možné pokračovanie.

# 2 Základy molekulárnej biológie

V úvodnej kapitole sa pozrieme na ústrednú dogmu molekulárnej biológie, teda proces, pri ktorom z informácie uloženej v DNA vznikajú proteíny. Tento proces, nazývaný aj ako proteosyntéza, je vo všeobecnosti rovnaký ako pri prokaryotických, tak i pri eukaryotických organizmoch a pozostáva z dvoch krokov, prepisu genetického kódu uloženého v DNA do RNA a následnom preklade, využití tejto informácie pri syntéze proteínov. Inšpiráciou a zdrojom informácií prezentovaných v tejto kapitole bola predovšetkým publikácia [1].

$$DNA \xrightarrow{transkripcia} RNA \xrightarrow{translácia} protein$$

Proteíny sú jednou zo základných jednotiek buniek. Určujú štruktúru a funkciu buniek. Funkcia proteínov je daná lineárnym poradím ich základných stavebných kameňov, reťazcom aminokyselín. Táto sekvencia určuje, ako bude daný proteín zabalený do funkčnej informácie.

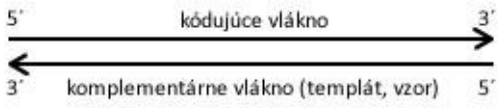
Pochopenie procesu syntézy proteínov je nevyhnutným krokom na ceste k detekcii génov, keďže práve v genetickej informácii je zakódovaná sekvencia aminokyselín tvoriacich proteíny. V procese proteosyntézy sa pre účely detekcie génov zameriame predovšetkým na transkripciu DNA do RNA, keďže práve tu dochádza k extrakcii genetickej informácie z DNA. Než sa však k transkripcii dostaneme, je treba popísať štruktúru samotnej DNA, v ktorej sú gény, genetická informácia, uložené.

## 2.1 DNA

DNA alebo kyselina deoxyribonukleová pozostáva z dvoch dlhých polynukleotidových vlákien zložených zo štyroch typov nukleotidových podjednotiek. Vlákna sú vzájomne spojené vodíkovými mostíkmi medzi bázami nukleotidov.

Nukleotid je základnou stavebnou jednotkou DNA. Tvorí ho päťuhlíkový cukor, v prípade DNA 2-deoxyribóza, na ktorý je naviazaný fosfátový zvyšok a jedna zo štyroch dusíkatých báz - adenín (A), guanín (G), cytozín (C) alebo tymín (T), v RNA nahradený uracilom (U). Nukleotidy sú spojené v reťazec väzbami medzi sacharidmi (cukrom) a fosfátovými zvyškami, ktoré tak tvoria kostru cukor – fosfát – cukor – fosfát... Keďže sa nukleotidy líšia len dusíkatými bázami, tak je DNA zaznamenávaná ako reťazec symbolov označujúcich jednotlivé bázy A, C, G a T.

Nukleotidy sa môžu spájať len jedným orientovaným spôsobom, čo dáva reťazcu DNA polaritu. Na jednom konci, označovanom ako 3´, končí reťazec hydroxilovou –OH skupinou, na druhom, označovanom ako 5´, končí fosfátovým zvyškom.




**Obrázok 2.1** *Antiparalelné vlákna DNA, v ktorých proti sebe stoja opačné konce*

Oba polynukleotidové vlákna sú v dvojzávitnici DNA spojené vodíkovými mostíkmi medzi bázami, ktoré sa nachádzajú vnútri dvojzávitnice s cukor-fosfátovou kostrou von. Dusíkaté bázy sa párujú na základe komplementarity báz, adenín s tymínom a guanín s cytozínom, vo všeobecnosti bicyklická báza purín (adenín, guanín) s monocyklickou bázou pyrimidínom (cytozín, tymín). Toto komplementárne párovanie umožňuje párom báz zaujať energeticky najvýhodnejšie postavenie v rámci dvojzávitnice. Takto majú oba páry báz podobnú šírku, ktorá umožňuje udržanie stability cu-

kor-fosfátovej kostry molekuly DNA. Obe kostry sa zároveň obtáčajú okolo seba a tvoria tak dvojzávitnicu. Bázy sa môžu spolu párovať v dvojzávitnici len vtedy, keď sú oba reťazce voči sebe antiparalelné, teda majú navzájom opačnú polaritu. Z párovania báz ďalej vyplýva, že sekvencia nukleotidov prvého reťazca DNA molekuly je presne komplementárna k sekvencii nukleotidov druhého reťazca. To znamená, že oba reťazce nesú rovnakú informáciu, čo má význam pri kopírovaní DNA (ako pri replikácii, tak i pri transkripcii do RNA).

Gény sú nositeľmi genetickej informácie, ktorá je ďalej predávaná z generácie na generáciu pri delení buniek. Sú kódované v štruktúre DNA poradím nukleotidov v reťazci, kedy môžeme každú bázu (A, C, T a G) považovať za jedno písmeno štvorpísmenovej abecedy používanej k zápisu genetickej informácie do reťazca DNA. Väčšina génov je tvorená krátkymi úsekmi DNA, ktoré kódujú jednotlivé proteíny. Pre samotný preklad je však využívaná len menšia časť génu, zvyšné tzv. regulačné oblasti určujú množstvo a čas syntézy daného proteínu. Úplná genetická informácia organizmu sa nazýva genóm, avšak toto pomenovanie sa často používa je pre označenie DNA, ktorá túto informáciu obsahuje.

# 2.2 Transkripcia

K syntéze, tvorbe, proteínov dochádza na polyribozómoch, kdežto DNA ako nositeľka informácie pre ich tvorbu sa v eukaryotických bunkách nachádza v chromozómoch, v prípade prokaryotických buniek sa úlohy chromozómov zhosťuje jadro bunky, ktoré je tvorené jednou kruhovou makromolekulou DNA. Vzniká tak potreba akéhosi prostredníka pre prenos tejto informácie. Týmto prostredníkom je kyselina ribonukleová, RNA. Táto RNA je neskôr priamo využívaná ako vzor, templát, pre tvorbu proteínov. Transkripcia alebo prepis je proces, pri ktorom vzniká RNA komplementárna k jednému z reťazcov DNA, templátovému reťazcu, a je prvým krokom pre uplatnenie genetickej informácie v bunke.

RNA je, podobne ako DNA, lineárny polymér zložený zo štyroch typov podjednotiek poprepájaných navzájom fosfodiesterovými väzbami. Rozdiel medzi RNA a DNA je v tom, že päťuhlíkvým cukrom v ribonukleotidoch, obdoba nukleotidov v DNA, je ribóza namiesto deoxyribózy a tymín je nahradený uracilom (U). Uracil sa rovnako ako tymín páruje s adenínom. Taktiež treba poznamenať, že RNA je tvorená len jedným vláknom. To jej umožňuje zbaliť sa do rôznych tvarov, čo môže byť dôležité pri prenose informácie z DNA do proteínu. Okrem prenosu informácie môže mať RNA aj iné funkcie, napríklad štruktúrnu alebo katalytickú. Všetka RNA v bunke vzniká transkripciou.


**Obrázok 2.2** *RNA počas transkripcie uvoľní dvojzávitnicu DNA. Následne po pridaní nového nukleotidu dochádza k obnoveniu dvojzávitnice DNA a vytesneniu vlákna RNA*

Transkripcia začína pouvoľnením úseku dvojzávitnice DNA, jeden z reťazcov následne slúži ako vzor pre syntézu RNA. Nukleotidová sekvencia RNA je ďalej určená komplementárnym párovaním báz. Pri párovaní uvoľneného ribonukleotidu s nukleotidom v DNA, ktorý slúži ako vzor, je ri-

bonukleotid pripojený k rastúcemu reťazcu RNA v enzýmovo katalyzovanej reakcii. Takto vznikajúci reťazec RNA sa postupne predlžuje a stáva sa kópiou kódujúceho vlákna DNA, vlákna komplementárneho k templátovému reťazcu DNA. Hneď za miestom, kde bol pridaný nový ribonukleotid, dochádza k obnoveniu dvojzávitnicovej štruktúry DNA a vytesneniu vlákna RNA. Enzýmy prepisujúce DNA do RNA sa nazývajú RNA-polymerázy a ich význam spočíva v katalyzácii vzniku väzieb medzi nukleotidmi. RNA je rovnako ako DNA syntetizovaná v smere 5′ → 3′. Vďaka skorému uvoľneniu RNA z templátového reťazca DNA počas syntézy môže podľa DNA jedného génu vzniknúť mnoho kópií RNA.

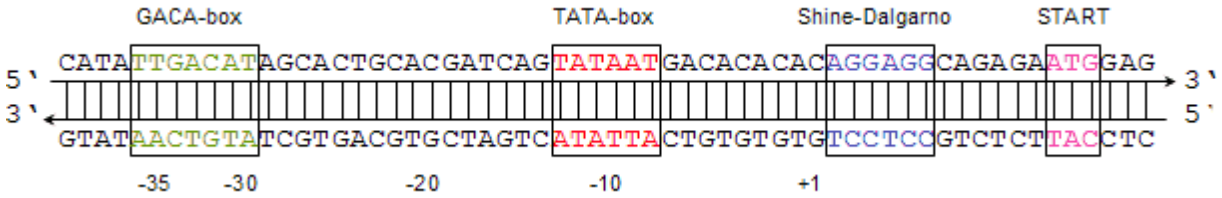
RNA vytvorená transkripciou génov, ktoré v bunke kódujú aminokyselinovú sekvenciu proteínov, sa nazýva mediátorová RNA, mRNA. Konečným produktom iných génov môže byť aj samotná RNA, ktorá má však nakoniec v bunke podobnú úlohu ako proteíny a navyše zohráva dôležitú úlohu pri preklade RNA na proteíny. Takýmito typmi RNA sú rRNA, ktorá je súčasťou ribozómov, kde prebieha preklad, alebo tRNA slúžiaca ako adaptor medzi mRNA a aminokyselinou.

Priebeh transkripcie je náramne podobný tak v eukaryotických, ako aj v prokaryotických bunkách. Pri oboch druhoch organizmov, však má transkripcia svoje špecifiká.

## 2.2.1 Transkripcia v prokaryotických bunkách

Začneme prokaryotickými organizmami, u ktorých sa nakoniec budeme pokúšať aj o samotnú detekciu génov. Tu sa RNA-polymeráza pri náhodnom stretnutí s molekulou DNA pokúša slabo naviazať, čo vedie k rýchlemu pohybu sklzom po naviazanej DNA, pri ktorom pátra po sekvencii promótora obsahujúceho informácie o začiatku transkripcie. Ako náhle RNA-polymeráza rozpozná promótor, pevne sa naviaže na tento úsek DNA a uvoľní dvojzávitnicovú štruktúru DNA pred sebou. Jedno vlákno sa takto stáva templátom pre komplementárne párovanie báz, takže je sekvencia RNA zhodná so sekvenciou kódujúcou gén. Začína sa prepis z DNA do RNA, ktorý pokračuje až dovtedy, kým RNA-polymeráza nenarazí na tzv. terminátor, signál v DNA, ktorý ma na svedomí zastavenie RNA-polymerázy a uvoľnenie od templátového reťazca DNA ako aj reťazca RNA.

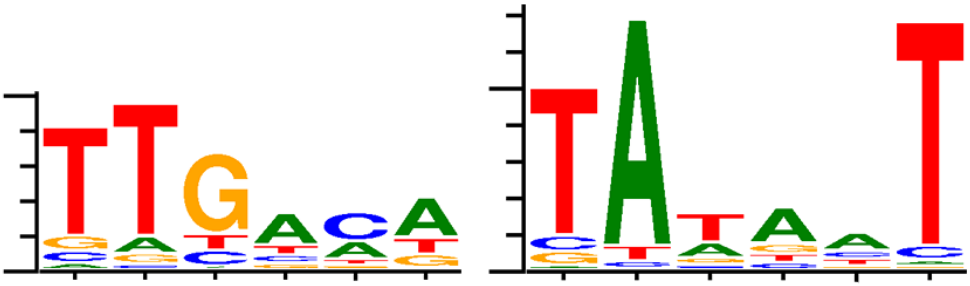
Rozpoznávanie promótorov má na starosti tzv. sigma-faktor, súčasť RNA-polymerázy. Každá molekula RNA-polymerázy má práve jeden takýto sigma-faktor, pričom napríklad v baktérii *Escherichia coli* nájdeme dokopy sedem druhov sigma-faktorov. Rozlišujú sa číslom reprezentujúcim charakteristickú molekulovú hmotnosť, napríklad $\sigma^{70}$ sa v spomínanej *E. coli* nazýva aj ako primárny sigma-faktor, keďže má na starosti inicializáciu transkripcie vo väčšine prípadov. Po naviazaní sa polymerázy na promótor a vytvorení reťazca dlhého približne 7 nukleotidov dochádza k uvoľneniu sigma-faktoru a RNA-polymeráza pokračuje ďalej v syntéze RNA v smere transkripcie až do uvoľnenia z DNA na sekvencii terminátora, kedy sa na ňu znovu naviaže sigma-faktor a hľadanie promótora začína odznova.




**Obrázok 2.3** *Štruktúra prokaryotického promótora, okolo pozície -35 sa nachádza GACA box, na -10 nájdeme Pribnowov box, niekedy označovaný aj ako TATA box. Signál Shine-Dalgarno nesúvisí priamo s transkripciou, ale zohráva svoju úlohu pri preklade a podrobnejšie bude opísaný v kapitole 2.3*

Promótor je asymetrickým signálom a preto viaže molekulu RNA-polymerázy len jedným smerom, čo má za následok jej schopnosť pohybu len týmto smerom. Takto je zaistené, že dochádza

vždy k prepisu správneho vlákna DNA v smere 5′ → 3′. Prítomnosť promótorov umožňuje presnú kontrolu sekvencií, ktoré majú byť prepisované, keďže k započatiu prepisu je potreba pevnej väzby RNA-polymerázy na DNA. Prokaryotický promótor pozostáva z dvoch častí. GACA box, s konvenčnou sekvenciou 5′-TTGACAT-3′, sa nachádza zhruba na pozícii -35 a Pribnowov box, s konvenčnou sekvenciou 5′-TATAAT-3′, okolo pozície -10. Začiatok transkripcie je označovaný pozíciou +1, nula sa pri značení pozície nepoužíva. Znalosť štruktúry promótorov považujem za kľúčovú pri predikcii génov, a preto sa im ešte budeme venovať podrobnejšie v kapitole 4 venovanej návrhu „prediktora“, konkrétne pri predikcii začiatku transkripcie.



**Obrázok 2.4** *Logá definujúce množstvo informácie obsiahnuté na jednotlivých pozíciách signálov prokaryotických promótorov, GACA box vľavo, Pribnowov box vpravo*

Ku koncu transkripcie dochádza, keď RNA-polymeráza narazí na terminátor. Existujú dva druhy terminátorov. Rho-závislý terminátor vyžaduje k ukončeniu transkripcie tzv. rho-faktor. Takýto terminátor nie je možné detekovať, pretože je závislý čisto na rho-faktore a nie je nijako závislý na štruktúre reťazca DNA. Na druhej strane, pre rho-nezávislé terminátori je charakteristická palindromatická štruktúra nasledovaná na tymín bohatou sekvenciou. Palindromatická štruktúra vytvorí niečo ako slučku, na ktorej sa RNA-polymeráza pozastaví, čo oslabí väzby medzi tymínmi a potencionálnymi adenínmi a transkripcia sa ukončí.

Medzi špecifiká transkripcie pri prokaryotických bunkách taktiež patrí skutočnosť, že nie všetky gény majú pred sebou promótor. Gény sa totiž zvyknú zhlukovať do akýchsi transkripčných jednotiek, tzv. operónov. Takto môže dôjsť k prepisu viacerých génov naraz, umiestnených medzi jedným promótorom a terminátorom.

## 2.2.2 Transkripcia v eukaryotických bunkách

Pri eukaryotách prebieha transkripcia obdobne ako v prokaryotách. Rozdiely však nájdeme napríklad v podobe promótorov, kedy eukaryotický promótor má zložitejšiu štruktúru. Na pozícii -80 sa CAAT box s konvenčnou sekvenciou 5′-GGCCAATCT-3′, ktorý zohráva dôležitú úlohu pri uplatňovaní promótora zvyšovaním jeho sily. U rastlín je CAAT box nahradený AGGA boxom. Nasleduje TATA box s konvenčnou sekvenciou 5′-TATAA-3′ okolo pozície -25, podobný Pribnowovmu boxu v prokaryotických promótoroch. TATA box je zväčša obkolesený sekvenciou bohatou na GC, tzv. GC box. GC box sa nevyskytuje len v okolí TATA boxu, ale môže byť prítomný na viacerých miestach promótoru.

Ďalším rozdielom je fakt, že zatiaľ čo v prokaryotických bunkách nasadajú ribozómy na voľný 5′ koniec RNA ešte počas priebehu transkripcie, v eukaryotických bunkách je DNA uzavretá v jadre bunky, teda k transkripcii dochádza v jadre podobne ako v prokaryotách, avšak ribozómy, na ktorých prebieha následný preklad, sa nachádzajú v cytoplazme. RNA sa do cytoplazmy dostáva skrz malé póry v membráne jadra a pred samotným odchodom z jadra ešte podlieha niekoľkým úpravám. Pre stabilizáciu molekuly RNA dochádza k úprave koncov mRNA, na 5′koniec je pridaná čiapočka v podobe atypického nukleotidu, jedná sa o guanínový nukleotid s naviazanou methylovou skupinou,

a na 3´konci dochádza k polyadenizácii, kedy je na koniec naviazaná sekvencia zložená zo samých adenínov. Okrem toho sú z DNA odstránené nekódujúce oblasti, intróny, pretože na rozdiel od prokaryotických génov, ktoré sú tvorené neprerušenou sekvenciou nukleotidov kódujúcich aminokyseliny pre tvorbu proteínov, sú gény eukaryotických buniek tvorené kódujúcimi sekvenciami, exónmi a nekódujúcimi oblasťami, intrónmi. Tento proces sa nazýva zostrih RNA.

# 2.3 Translácia

Proteíny sú tvorené 21 rôznymi aminokyselinami, avšak na ich zakódovanie v RNA sú použité iba štyri striedajúce sa nukleotidy. Je teda zrejmé, že nemôže dochádzať k prekladu jedného nukleotidu na jednu aminokyselinu. Pravidlá popisujúce preklad nukleotidovej sekvencie do aminokyselinovej sú definované ako genetický kód. Sekvencia RNA je pri preklade čítaná nie ako sekvencia jednotlivých nukleotidov, ale ako trojíc. Keďže máme 4 druhy nukleotidov, tak dostávame 64 rôznych trojíc, ktoré kódujú 21 rôznych aminokyselín. Znamená to, že niektoré aminokyseliny sú kódované viacerými trojicami nukleotidov nazývaných kodóny. Genetický kód je tak redundantný. Niektoré trojice kódujú vybrané aminokyseliny častejšie, iné menej, čo môže napomôcť pri analýze reťazca DNA za účelom predikcie génov.

| | | 2.báza | | | |
|---|---|---|---|---|---|
| | | U | C | A | G |
| 1.báza | U | UUU (Phe/F)<br>UUC (Phe/F)<br>UUA (Leu/L)<br>UUG (Leu/L) | UCU (Ser/S)<br>UCC (Ser/S)<br>UCA (Ser/S)<br>UCG (Ser/S) | UAU (Tyr/Y)<br>UAC (Tyr/Y)<br>UAA (stop)<br>UAG (stop) | UGU (Cys/C)<br>UGC (Cys/C)<br>UGA (stop)<br>UGG (Trp/W) |
| | C | CUU (Leu/L)<br>CUC (Leu/L)<br>CUA (Leu/L)<br>CUG (Leu/L) | CCU (Pro/P)<br>CCC (Pro/P)<br>CCA (Pro/P)<br>CCG (Pro/P) | CAU (His/H)<br>CAC (His/H)<br>CAA (Gin/Q)<br>CAG (Gln/Q) | CGU (Arg/R)<br>CGC (Arg/R)<br>CGA (Arg/R)<br>CGG (Arg/R) |
| | A | AUU (Ile/I)<br>AUC (Ile/I)<br>AUA (Ile/I)<br>AUG (Met/M) | ACU (Thr/T)<br>ACC (Thr/T)<br>ACA (Thr/T)<br>ACG (Thr/T) | AAU (Asn/N)<br>AAC (Asn/N)<br>AAA (Lys/K)<br>AAG (Lys/K) | AGU (Ser/S)<br>AGC (Ser/S)<br>AGA (Arg/R)<br>AGG (Arg/R) |
| | G | GUU (Val/V)<br>GUC (Val/V)<br>GUA (Val/V)<br>GUG (Val/V) | GCU (Ala/A)<br>GCC (Ala/A)<br>GCA (Ala/A)<br>GCG (Ala/A) | GAU (Asp/D)<br>GAC (Asp/D)<br>GAA (Glu/E)<br>GAG (Glu/E) | GGU (Gly/G)<br>GGC (Gly/G)<br>GGA (Gly/G)<br>GGG (Gly/G) |

**Tabuľka 2.1** *Genetický kód. Kodóny sú prekladané na aminokyseliny, v zátvorkách skratka/písmeno označujúce aminokyselinu*

Gény v drvivej väčšine prípadov začínajú štart kodónom AUG, na ktorom začína translácia mRNA, a končia jedným zo stop kodónov UAA, UAG alebo UGA. Ako štart kodón zriedka vystupujú GUG alebo UUG, ktoré sú však pri translácii v konečnom dôsledky interpretované ako methionín, teda aminokyselina kódovaná pôvodným štart kodónom AUG. V prokaryotických bunkách sa pred štart kodónom nachádza ešte tzv. Shine-Dalgarno sekvencia, väčšinou v tvare AGGAGGU, ktorá napomáha naviazaniu ribozómu k mRNA, aby mohla začať syntéza proteínu tým, že vybraný ribozóm zarovná voči štart kodónu. Sekvencia znakov medzi štart a stop kodónom sa nazýva čítací rámec alebo ORF, z anglického *Open Reading Frame*. Principiálne môže byť RNA prekladaná v troch číta-

cích rámcoch, podľa toho, ktorým nukleotidom začneme. Požadovaný proteín však vzniká len v jednom z týchto rámcov.

Samotné kodóny v mRNA sa neviažu priamo na danú aminokyselinu. Preklad mRNA na proteín tak má na starosti transferová RNA, tRNA, ktorá sa je schopná spárovať jednou časťou s kodónom a ďalšou časťou naviazať aminokyselinu. Preklad prebieha na ribozómoch zložených s viac než 50 druhov proteínov a niekoľkých druhov RNA, rRNA.

# 3 Detekcia génov

Po predstavení základných princípov, na ktorých stojí molekulárna biológia, a spôsobe uloženia génov v DNA sa môžeme bližšie pozrieť na detekciu génov z informatického hľadiska. Jedná sa v podstate o vyhľadávanie signálov v reťazci štvorpísmenovej abecedy (A, C, G a T) pomocou štatistických alebo pravdepodobnostných modelov. V súčasnosti sú zaužívané dva rôzne prístupy k tomuto problémy, a síce detekcia génov na základe vzájomnej podobnosti sekvencií alebo tzv. predikcia génov, kedy je pre lokalizáciu génov použitá množina pravidiel, ktoré musí daná sekvencia spĺňať, alebo pravdepodobnostný model, vytrénovaný na DNA sekvenciách so známou polohou funkčných oblastí. Podrobnejšie porovnanie jednotlivých metód nájdeme napríklad v [2]. Pre viac informácií ohľadne metód samotných by som odporučil publikáciu [3], ktorá bola i mne hodnotným zdrojom pri písaní nasledujúcich riadkov.

## 3.1 Detekcia génov na základe podobností

Pri detekcii génov na základe podobností jednoducho porovnávame úseky sekvencie, v ktorej gény hľadáme, s kódovaním proteínov alebo DNA sekvencií uložených v databáze. Tento prístup je založený na predpokladoch, že funkčné oblasti DNA sú menej náchylné zmenám v dôsledku evolúcie ako nekódujúce oblasti. Základnými prostriedkami pre určenie miery príbuznosti sekvencií sú metódy zarovnávania sekvencií, ako napríklad Smith-Watermanov algoritmus pre lokálne zarovnávanie sekvencií alebo rýchlejšie, no menej presné metódy pre väčšie objemy dát ako BLAST a FASTA.

Algoritmus BLAST stojí na myšlienke, že správne zarovnané sekvencie obsahujú s veľkou pravdepodobnosťou krátke úseky zhodných symbolov. Tieto úseky môžeme na začiatku vyhľadať a použiť ich ako semienka, od ktorých sa pri hľadaní dlhšieho zarovnania odraziť.

FASTA je niekoľkokroková metóda, u ktorej vyhľadávanie začína lokalizáciou zhodných slov pevne danej dĺžky medzi dvoma porovnávanými sekvenciami a označí ich skôr než sa začne zarovnávanie časovo náročnejšími algoritmami typu Smith-Waterman.

Detekcia génov na základe podobností poskytuje biologicky relevantné dáta, keďže je založená na existujúcich informáciách, s čím však súvisí problém kvality informácií uložených v databázach. Mnohé databázy sú totiž tvorené údajmi pochádzajúcimi z podobných detektorov génov, ktoré samozrejme tiež nie sú stopercentné, čo má za následok lokalizáciu ďalších chybných sekvencií a tým pádom vnášanie čoraz väčších chýb do spomínaných databáz. Taktiež majú tieto metódy problém s krátkymi génmi, ktorú sú často preskakované. No azda najväčším problémom je asi fakt, že len zhruba polovica objavených génov má význačnú podobnosť s génmi uloženými v databázach.

### 3.1.1 Smith-Watermanov algoritmus

Ako bolo vyššie naznačené, mnohé metódy detekcie génov porovnávaním sú založené na algoritmoch pre zarovnávanie sekvencií. Tento spôsob hľadania podobností medzi sekvenciami si ukážeme na Smith-Watermanovom algoritme pre lokálne zarovnávanie sekvencií. Smith-Watermanov algoritmus je v porovnaní s vyššie spomenutými heuristickými algoritmami FASTA a BLAST presnejší, no pomalší a preto sa pre vyhľadávanie vo väčších dátových súboroch nepoužíva. Napriek tomu, väčšina algoritmov na zarovnávanie sekvencií, a v podstate aj FASTA a BLAST, vychádzajú práve zo Smith-Watermana.

Pri porovnávaní sekvencií pátrame po tom, či sekvencie majú spoločného predka, od ktorého sa líšia rôznymi mutáciami. Základnými procesmi pri mutáciách sú zámena, vloženie alebo odstráne-

nie prvkov, v našom prípade nukleotidov. Vloženie a odstránenie budeme označovať ako medzery. Pre hodnotenie podobnosti je potrebné vytvoriť akýsi bodovací systém. Celkové ohodnotenie podobnosti dvoch sekvencií bude potom súčtom bodov za každý zarovnaný pár alebo medzeru. Nebudeme sa zaoberať tvorbou bodovacieho systému, pretože existuje viacero spôsobov ohodnocovania jednotlivých podobností.

Uvažujme dva reťazce, *x* o dĺžke *m* a *y* dĺžky *n*, u ktorých chceme zistiť, ako veľmi sú si podobné. Skóre z bodovacieho systému medzi i-tým znakom reťazca *x* a j-tým znakom reťazca *y* budeme značiť ako $s(x_i, y_j)$. Vytvoríme maticu *F* s rozmermi *m*n* kde platí:

$$F(i,0) = 0; pre\ 0 \leq i \leq m \tag{3.1}$$

$$F(0,j) = 0; pre\ 0 \leq j \leq m \tag{3.2}$$

$$F(i,j) = max\begin{cases} 0 \\ F(i-1,j-1) + s(i,j) \\ F(i-1,j) - d \\ F(i,j-1) - d \end{cases}; pre\ 1 \leq i \leq m, 1 \leq j \leq n \tag{3.3}$$

kde *d* je penalizáciou za medzeru. Je potrebné si zakaždým uložiť ukazovateľ na bunku, z ktorej bola hodnota *F(i, j)* odvodená. Keď je matica *F* hotová, nájdeme bunku *F(i, j)* s najvyšším ohodnotením a podľa uložených ukazovateľov zrekonštruujeme cestu, ktorou sme sa k bunke dopracovali. Súčasne môžeme zarovnané sekvencie zapisovať odzadu. Pri pohybe po diagonále zaznačíme oba symboly $x_i$ a $y_j$, ak ukazovateľ na aktuálnu pozíciu ukazoval z bunky *(i-1, j)*, zaznačíme $x_i$ a medzeru (-), ak z bunky *(i, j-1)*, tak zapíšeme medzeru (-) a $y_j$. Spätné vyhľadávanie končí, keď narazíme na bunku s hodnotou 0, ktorá zodpovedá začiatku zarovnania.

Najlepšie bude ukázať Smith-Watermanov algoritmus na nasledujúcom príklade: *Majme dve DNA sekvencie GAATTC a GATTA, medzi ktorými chceme nájsť najlepšie možné zarovnanie. Uvažujme, že skóre pri zhode symbolov bude +2, pri nesúhlasných symboloch -1 a penalizácia za medzeru d = 2.*

Zo všetkého najskôr zostavíme matice *F*, podľa postupu uvedeného vyššie.

$$F = \begin{array}{c|ccccccc} & - & G & A & A & T & T & C \\ - & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ G & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ A & 0 & 0 & 4 & 2 & 0 & 0 & 0 \\ T & 0 & 0 & 2 & 3 & 4 & 2 & 0 \\ T & 0 & 0 & 0 & 1 & 5 & 6 & 4 \\ A & 0 & 0 & 2 & 2 & 3 & 4 & 5 \end{array}$$

**Obrázok 3.1** *Matica zostrojená Smith-Watermanovým algortimom, šípky reprezentujú ukazovatele a najlepšie ohodnotené zarovnanie je vyznačené červenou farbou*

Ak existuje viacero buniek, z ktorých mohla byť aktuálna bunka odvodená, tak je výber tej predchádzajúcej ponechaný na náhode. Keď je matica hotová, vyhľadáme najvyššiu hodnotu, v tomto prípade číslo 6, a môžeme začať so spätným vyhľadávaním cesty, počas ktorého si značíme jednotlivé symboly podľa smeru ukazovateľa.

Takto nájdeme ideálne lokálne zarovnanie sekvencií GAATTC a GATTA so skóre, v podobe GAATT a GA-TT.

Smith-Watermanov algoritmus patrí k algoritmom dynamického programovania, podobne ako väčšina algoritmov používaných na analýzu sekvencií. Je variáciou staršieho Needleman-Wunschovho algoritmu pre globálne zarovnávanie. Algoritmy sa líšia v podstate iba v dvoch bodoch

a síce, že Needleman-Wunsch pracuje s maticou, v ktorej sú aj záporné hodnoty, a spätné vyhľadávanie nezačína v bunke s najvyšším ohodnotením, ale v poslednej bunke matice, *F(m, n)*.

# 3.2 Predikcia génov

Po detekcii génov sa dostávame k, pre túto prácu podstatnejšej, predikcii génov. Pod týmto pojmom budeme chápať vyhľadávanie génov metódami založenými na poznatkoch o štruktúre génov a pravidlami z týchto poznatkov odvodenými. Pre takéto modelovanie sa používajú najmä rôzne variácie Skrytých Markovovských modelov alebo Neurónové siete.

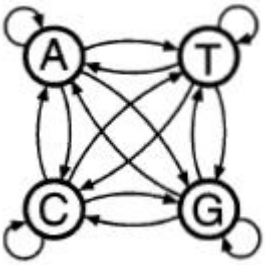
Najväčším obmedzením predikcie génov je pomerne slabá znalosť štruktúry génov, a to najmä v nových DNA sekvenciách. Taktiež množina momentálne známych génov, z ktorých by modely vychádzali, je stále malá a určite nepokrýva všetky potenciálne vlastnosti génov.

## 3.2.1 Markovovské reťazce

Než sa dostaneme k Skrytým Markovovským modelom, predstavíme si ich jednoduchšiu variantu, Markovovské reťazce. Markovovský reťazec je kolekciou stavov poprepájaných prechodmi definovanými pravdepodobnosťou, s ktorou je aktuálny stav (alebo symbol) nasledovaný iným. Tieto pravdepodobnosti nazývame pravdepodobnosti prechodu a ďalej ich budeme značiť ako $a_{st}$:

$$a_{st} = P(x_i = t | x_{i-1} = s) \tag{3.4}$$

kde $a_{st}$ je určené pravdepodobnosťou prechodu zo stavu *s* do stavu *t*.




**Obrázok 3.2** *Markovovský reťazec popisujúci DNA (zdroj: [3] strana 48)*

Pri pravdepodobnostných modeloch sekvencií môžeme pravdepodobnosť sekvencie vypočítať niekoľkonásobným použitím *P(X, Y) = P(X|Y)P(Y)* ako to je naznačené v rovnici nižšie.

$$P(x) = P(x_L, x_{L-1}, \dots, x_1) = P(x_L | x_{L-1}, \dots, x_1) P(x_{L-1} | x_{L-2}, \dots, x_1) \dots P(x_1) \tag{3.5}$$

Kľúčovou vlastnosťou Markovovských reťazcov je, že pravdepodobnosť každého symbolu $x_i$ závisí iba na predchádzajúcom symbole $x_{i-1}$, takže predchádzajúcu rovnicu môžeme zapísať ako súčin pravdepodobností prechodov postupnosti stavov definujúcich analyzovanú sekvenciu:

$$P(x) = P(x_L | x_{L-1}) P(x_{L-1} | x_{L-2}) \dots P(x_2 | x_1) P(x_1) = P(x_1) \prod_{i=2}^{L} a_{x_{i-1} x_i} \tag{3.6}$$

Okrem jednotlivých symbolov abecedy ako stavov sa do Markovovských reťazcov zvykne dopĺňať začiatočný prípadne i koncový stav pre modelovanie začiatku a konca sekvencie.

### 3.2.2 Skryté Markovovské modely (HMM)

Skrytý Markovovský model je všeobecným pravdepodobnostným modelom pre sekvencie symbolov. Užívaním tohto modelu analyzujeme iba požadovanú sekvenciu, v ktorej gény vyhľadávame, takže nedochádza k porovnávaniu s inými, napríklad uloženými v databáze, pričom samotný model je založený na znalostiach o štruktúre génov nadobudnutých učením na trénovacej množine.

Skryté Markovovské modely (ďalej len HMM z anglického *Hidden Markov Model*) majú všeobecné využitie, okrem predikcie génov sú často používané pri rozpoznávaní hlasu, kedy je zvuková nahrávka rozkúskovaná na menšie časti o veľkosti 10 – 20 milisekúnd, ktoré sú nasledovaným spracovaním roztriedené do veľkého množstva preddefinovaných kategórií procesom nazývaným vektorové kvantovanie. Nahrávka je ďalej prezentovaná ako sekvencia symbolov zodpovedajúcich jednotlivým kategóriám a z tejto sekvencie program pre rozpoznávanie hlasu určí vyslovené slová v nahrávke. Rozpoznávanie génov je podobným problémom, máme sekvenciu symbolov (rozkúskovaná nahrávka) a hľadáme jej význam (jednotlivé slová). Viac o použití HMM pri rozpoznávaní hlasu nájdeme napríklad v [4].

Hlavným rozdielom medzi vyššie spomínanými Markovovskými reťazcami a HMM tkvie v tom, že v HMM nekorešpondujú symboly so stavmi jedna k jednej. Sekvencia stavov je oddelená od sekvencie symbolov. Sekvenciu stavov budeme ďalej označovať ako cesta alebo $\pi$. Cesta samotná zodpovedá Markovovskému reťazcu, takže pravdepodobnosť nejakého stavu závisí len na stave predchádzajúcom, rovnica (3.4). Keďže sme oddelili postupnosť stavov od postupnosti symbolov, treba do modelu zaviesť nové parametre:

$$e_k(b) = P(x_i = b|\pi_i = k) \tag{3.7}$$

kde $e_k(b)$ je pravdepodobnosť, že v stave $k$ narazíme na symbol $b$. Túto pravdepodobnosť nazývame emisnou pravdepodobnosťou.

Skryté Markovovské modely voláme skrytými pretože je nám známe iba postupnosť symbolov a nie stavov, tá je skrytá. HMM sa často používajú taktiež ku generovaniu sekvencií, kedy začíname stavom $\pi_i$ vybraným na základe pravdepodobnosti $a_{0i}$, kde stav 0 reprezentuje počiatočný stav sekvencie, tak ako tomu bolo pri Markovovských reťazcoch. V tomto stave je vygenerovaný symbol podľa emisných pravdepodobností $e_{\pi 1}$. Ďalší stav generujeme podľa pravdepodobností prechodu $a_{\pi 1 i}$ a tak ďalej... Takto je vygenerovaná sekvencia symbolov, o ktorej môžeme prehlásiť, že pravdepodobnosť generovania sekvencie symbolov $x$ a sekvencie stavov $\pi$ je rovná:

$$P(x, \pi) = a_{0\pi_1} \prod_{i=1}^{L} e_{\pi_i}(x_i) a_{\pi_i \pi_{i+1}} \tag{3.8}$$

Ako bolo vyššie povedané, pri HMM nie sme schopní na prvý pohľad určiť, v ktorom stave sa pri prechode daným symbolom bude model nachádzať. Pri používaní HMM nás však práve táto informácia zaujíma najviac, preto potrebujeme spôsob, akým zistiť stav modelu zodpovedajúci danému symbolu. Na toto existuje viacero algoritmov, najčastejšími je Viterbiho algoritmus.

Viterbiho algoritmus patrí medzi algoritmy dynamického programovania, takže je príbuzný so skôr spomenutým Smith-Watermanovým algoritmom. Uvažujme napríklad jednoduchý HMM pre predikciu génov, ktorý v sekvencii nájde kódujúce a nekódujúce oblasti. Rozdelenie na tieto oblasti utvoríme vychádzajúc s najpravdepodobnejšej cesty prechodov stavmi HMM:

$$\pi^* = argmax\ P(x, \pi) \tag{3.9}$$

Túto najpravdepodobnejšiu cestu $\pi^*$ nájdeme rekurzívne. Predpokladajme, že pravdepodobnosť $v_k(i)$ je najpravdepodobnejšia cesta končiaca v stave $k$ pokrývajúca $i$ symbolov a je známa pre každý zo stavov $k$. Potom môžeme vypočítať tieto pravdepodobnosti aj pre symbol $x_{i+1}$ ako:

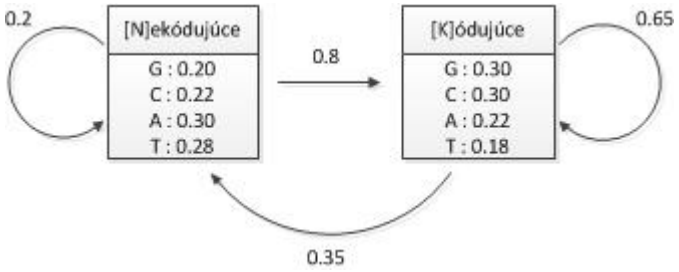
$$v_l(i+1) = e_l(x_{i+1}) max_k(v_k(i) a_{kl}) \tag{3.10}$$

Sekvencie začínajú v počiatočnom stave 0, z ktorého odvodíme počiatočnú podmienku $v_0(0)$. Uchovávaním spätných ukazovateľov môžeme ďalej nájsť sekvencie stavov backtrackingom. Viterbiho algoritmus pre počiatočnú podmienku $v_0(0) = 1$ bude vyzerať takto (prevzaté z [3] strana 56):

| | |
|---|---|
| Inicializácia premenných (i = 0): | $v_0(0) = 1, v_k(0) = 0 \; pre \; k > 0$ |
| Rekurzia (i = 1 ... L): | $v_l(i) = e_l(x_i) max_k(v_k(i-1) a_{kl})$<br>$ptr_i(l) = argmax_k(v_k(i-1) a_{kl})$ |
| Ukončenie: | $P(x, \pi^*) = max_k(v_k(L) a_{k0})$<br>$\pi_L^* = argmax_k(v_k(L) a_{k0})$ |
| Backtracking (i = L ... 1): | $\pi_{i-1}^* = ptr_i(\pi_i^*)$ |

Pri implementácii Viterbiho algoritmu, ale i ďalších algoritmov pracujúcich s pravdepodobnosťami, sa stretáme s problémom násobenia veľkého počtu malých čísel (pravdepodobností), čo môže spôsobiť chybu podtečenia. Tento problém sa rieši prevodom pravdepodobností do logaritmického priestoru, ako si ukážeme neskôr pri pozične špecifických maticiach.

Opísaný algoritmus ako i použitie HMM samotných si ukážeme na jednoduchom príklade, kedy sa pokúsime určiť kódujúce, respektíve nekódujúce úseky reťazca *ATGCA* za predpokladu, že model je zostavený podľa nasledujúceho obrázku.




**Obrázok 3.3** *Zmyslený HMM pre identifikáciu kódujúcich a nekódujúcich oblastí v nukleotidovom reťazci.*

Ďalej predpokladajme, že pravdepodobnosť začiatku sekvencie je pre oba stavy rovnaká, teda 0,5. Počiatočnú pravdepodobnosť pre začiatok v jednotlivých stavoch vypočítame jednoducho, a síce:

$$v_N(1) = e_N(A) * v_0(0) * a_{0N} = 0{,}3 * 1 * 0{,}5 = 0{,}15$$
$$v_K(1) = e_K(A) * v_0(0) * a_{0K} = 0{,}22 * 1 * 0{,}5 = 0{,}11$$

V tomto prvom kroku netreba hľadať najvyššiu pravdepodobnosť z predchádzajúceho kroku, keďže definovaným stavom v inicializácii je počiatočný stav $v_0$ s pravdepodobnosťou 1. Nasledujúce kroky už sú sprevádzané výberom najpravdepodobnejšieho prechodu a uchovaním spätného ukazovateľa na predchádzajúci stav.

$$v_N(2) = e_N(T) * \max[v_N(1) * a_{NN};\; v_K(1) * a_{KN}]$$
$$= 0{,}28 * \max[0{,}15 * 0{,}2; 0{,}11 * 0{,}35] = 0{,}01078$$
$$v_K(2) = e_K(T) * \max[v_K(1) * a_{KK};\; v_N(1) * a_{NK}]$$
$$= 0{,}18 * \max[0{,}11 * 0{,}65; 0{,}15 * 0{,}8] = 0{,}0216$$

A takto pokračujeme pre všetky znaky sekvencie. Na záver vyberieme cestu s najväčšou pravdepodobnosťou a zrekonštruujeme ju pomocou spätných ukazovateľov, ako to je naznačené v tabuľke nižšie. Takto vyčítame najpravdepodobnejšiu postupnosť stavov nášho modelu, a tak sa do-

stávame k výsledku ktorý identifikuje začiatok i koniec nekódujúcej oblasti v prvom znaku A a za kódujúcu oblasť označujeme zvyšné znaky analyzovanej sekvencie TGCA.

| | - | A | T | G | C | A |
|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| N | 0 | **0,15** | 0,01078 | 0,001512 | 0,000324324 | 0,0000862407 |
| K | 0 | 0,11 | **0,0216** | **0,004212** | **0,00082134** | **0,00011745162** |

**Tabuľka 3.1** *Výsledky prechodu sekvencie modelom. Hrubým písmom je zvýraznená najpravdepodobnejšia cesta.*

Nevyhnutným krokom k použitiu akéhokoľvek HMM je určenie jednotlivých pravdepodobností, či už emisných alebo pravdepodobností prechodov. Najbežnejším spôsobom je vytrénovanie modelu na príkladoch z trénovacej množine, u ktorých poznáme cesty prechodov stavmi HMM. Takto poznáme presný počet prechodov, $A_{kl}$, a výskytov symbolov, $E_k(b)$, v danom stave a pravdepodobnosti dopočítame jednoducho:

$$a_{kl} = \frac{A_{kl}}{\sum_{l'} A_{kl'}};\ e_k(b) = \frac{E_k(b)}{\sum_{b'} E_k(b')} \tag{3.11}$$

Ak cesty prechodov nepoznáme, tak sa pravdepodobnosti na základe počiatočných odhadov dajú dopočítať všetkými známymi algoritmami pre optimalizácie spojitých funkcií. Častejšie sa však používa iteračná metóda, Baum-Welchov algoritmus, ktorý pôvodne odhadnuté hodnoty postupne aproximuje na základe pravdepodobných ciest stavmi pre trénovaciu sekvenciu. Pre viac informácií o tomto algoritme by som odkázal na literatúru, napríklad [3].

## 3.2.3 Pozične špecifické matice

Pri hľadaní génov pátrame vo všeobecnosti po ich charakteristických znakoch, ktoré môžeme rozdeliť do dvoch skupín. Signály, čo sú krátke úseky pevnej dĺžky, ako napríklad GACA box, štart kodóny a podobne, alebo obsahové oblasti (z anglického *content regions*) rôznych dĺžok, v prokaryotách čítacie rámce, v eukaryotách intróny a exóny. A zatiaľ čo na detekciu a vyhodnocovanie obsahových oblastí sa výborne hodia napríklad Markovovské reťazce, detekciu signálov môžeme riešiť jednoduchšími pozične špecifickými maticami.

V prvom rade treba podotknúť, že pozične špecifické matice sú podobné HMM 0-tého rádu. 0-tého rádu znamená, že pravdepodobnosť výskytu aktuálneho symbolu nezávisí od toho predošlého.

Pozične špecifické matice, niekedy tiež označovaná ako profil, používame na vyjadrenie charakteristických znakov nejakého vzoru. Profil je zväčša odvodený z množiny zarovnaných funkčne príbuzných sekvencií. V jednoduchosti sa jedná o okienko pevnej dĺžky, ktoré v každej bunke zachytáva distribúciu jednotlivých nukleotidov na danej pozícii. Ako príklad si ukážeme pozične špecifickú maticu určenú pre detekciu GACA boxu, uvádzacieho signálu prokaryotických promótorov. Použité hodnoty pochádzajú z [5].

| | T | T | G | A | C | A |
|---|---|---|---|---|---|---|
| T | 0,78 | 0,82 | 0,15 | 0,20 | 0,10 | 0,24 |
| G | 0,10 | 0,05 | 0,68 | 0,10 | 0,07 | 0,17 |
| C | 0,09 | 0,03 | 0,14 | 0,13 | 0,52 | 0,05 |
| A | 0,03 | 0,10 | 0,03 | 0,58 | 0,32 | 0,54 |

**Tabuľka 3.2** *Pozične špecifická matica definujúca signál GACA box. Bunky obsahujú pravdepodobnosť výskytu daného nukleotidy (riadok) na vybranej pozícii (stĺpec).*

Ako vidíme v tabuľke vyššie, bunky matice špecifikujú percentuálne zastúpenie vybraného nukleotidu na danej pozícii. S takto definovanou maticou sme schopní určiť s akou pravdepodobnos-

ťou môžeme sekvenciu danej dĺžky nájsť v skutočnej funkčnej oblasti, ktorú matica opisuje, v našom prípade na mieste GACA boxu. Túto pravdepodobnosť vypočítame súčinom pravdepodobností výskytu jednotlivých symbolov na zodpovedajúcich pozíciách.

Takýmto spôsobom, hlavne pri väčších maticiach, dostaneme pomerne malé a často na prvý pohľad nič nehovoriace čísla s tým, že hrozí riziko podtečenia, ako sme spomenuli vyššie pri Viterbiho algoritme. Tento problém sa rieši tak, že hodnoty pravdepodobností nahradíme tzv. log-likelihood hodnotami, ktoré vyjadrujú akúsi mieru pravdepodobnosti výskytu v logaritmickom priestore. Prevod sa riadi nasledujúcim vzorcom:

$$M'_{ij} = \log\left(\frac{M_{ij}}{p_i}\right) \tag{3.12}$$

kde $M_{ij}$ je pravdepodobnosť výskytu symbolu $i$ na pozícii $j$ v matici $M$, $M'_{ij}$ je výsledná transformovaná log-likelihood hodnota a $p_i$ je všeobecná pravdepodobnosť výskytu symbolu $i$.

| | T | T | G | A | C | A |
|---|---|---|---|---|---|---|
| T | 1,13 | 1,18 | -0,51 | -0,22 | -0,91 | -0,04 |
| G | -0,91 | -1,60 | 1,00 | -0,91 | -1,27 | -0,38 |
| C | -1,02 | -2,12 | -0,57 | -0,65 | 0,73 | -1,60 |
| A | -2,12 | -0,91 | -2,12 | 0,84 | 0,24 | 0,77 |

**Tabuľka 3.3** *Pozične špecifická matica signálu GACA box po transformácii na log-likelihood hodnoty, uvažujúc všeobecnú pravdepodobnosť výskytu každého z nukleotidov za 0,25.*

Skóre $m$ sekvencie $s$ dĺžky $l$ potom takto transformovanou maticou dopočítame ako súčet hodnôt v zodpovedajúcich bunkách.

$$m_s = \sum_{i=1}^{l} M_{s_i j} \tag{3.13}$$

Toto skóre udáva mieru príbuznosti analyzovaného reťazcu s konvenčnou sekvenciou, respektíve maticou definovaným signálom. Samozrejme platí, že čím vyššie skóre, tým sú si sekvencie podobnejšie.

Práve na pozične špecifických maticiach je z väčšej časti založená metóda predikcie génov, ktorej návrh je predmetom tejto práce, a ktorej návrhu je zasvätená nasledujúca kapitola.

# 4 Experimentálna časť

Po teoretickom úvode do problematiky sa dostávame k hlavnému bodu tejto práce, a síce samotnému návrhu vhodnej metódy detekcie génov, lepšie povedané predikcie génov. Kódujúce oblasti budeme vyhľadávať na základe množiny pravidiel definovanej konečným automatom. Zameriame sa na predikciu génov v prokaryotických organizmoch, konkrétne *Escherichie coli*. V texte budeme ďalej označovať vyvíjaný program ako „prediktor".

*Escherichia coli* ako vhodný subjekt na testovanie bola zvolená kvôli rozsiahlejším znalostiam o štruktúre jej genómu. Taktiež zavážil fakt, že prokaryotické gény majú o niečo jednoduchšiu štruktúru, keďže sa kódujúca oblasť ďalej nezostriháva kvôli prítomnosti nekódujúcich exónov, ako je tomu v eukaryotách, a ich hustota v DNA sekvencii je oveľa vyššia (až 88% u *E. coli*) ako hustota výskytu génov u eukaryotov.

Pri návrhu riešenia predikcie génov kladieme dôraz aj na časovú náročnosť výsledného programu a dosiahnutie lineárnej zložitosti.

## 4.1 Poznámky k implementácii

Než sa dostaneme k popisu častí prediktora génov, zastavíme sa pri pár poznámkach k implementácii. Za implementačný jazyk pri vývoji sme zvolili skriptovací jazyk Python vo verzii 2.7, práve kvôli svojej skriptovacej povahe a z toho vyplývajúcej väčšej rýchlosti vývoja a analýzy zozbieraných dát, akej by sme dosiahli napríklad pri použití jazyka C. Okrem Pythona 2.7 je pre prístup k súborom formátu fasta, ktorý je bežný pre ukladanie DNA sekvencií, použitá nadstavba BioPython vo verzii 1.57, dostupná na http://www.biopython.org/wiki/Download.

Všetky použité skripty sú k dispozícii na priloženom CD v priečinku *src/*. Väčšina z nich je priamo spustiteľná a obsahuje stručnú nápovedu, ostatné sú súčasťou balíčka *genscan*. Dokumentácia k hlavnému programu vygenerovaná pomocou nástroja Epydoc 3.0 sa nachádza na rovnakom nosiči v priečinku *doc/*.

Reálne dáta sú umiestnené v priečinku *data/* a nájdeme tu predovšetkým zoskenovanú dna E. coli vo formáte fasta a k tomu prislúchajúce tabuľky výskytu génov a operónov. Dáta pochádzajú z UCSC Microbial Genome Browser dostupného na http://microbes.ucsc.edu/index.html. V priečinku *stats/* sú zasa uložené výsledky rôznych analýz, ktoré však z väčšej časti rozoberieme na nasledujúcich stránkach.

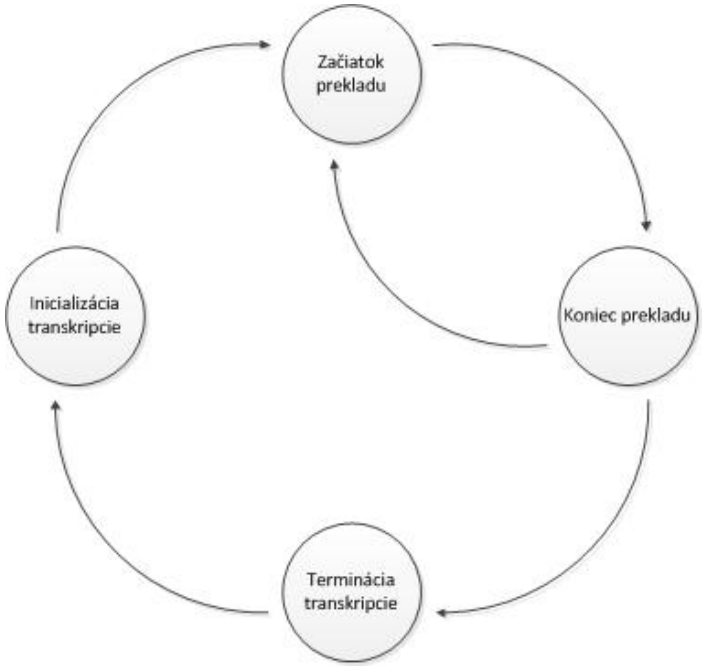
## 4.2 Návrh metódy predikcie génov

Začneme trochu obecným zadefinovaním štruktúry génov v prokaryotách. Ako bolo naznačené v kapitole 2, pri uplatňovaní genetickej informácie dochádza k dvom procesom, transkripcii a translácii. Ďalej sme sa dozvedeli, že inicializácia a ukončenie každého z týchto procesov sú nejakým spôsobom zakódované v DNA. Gén sa teda skladá z regulačnej a kódujúcej oblasti. Regulačné oblasti riadia transkripciu a pozostávajú z iniciátora transkripcie promótora a terminátora, ktorý transkripciu ukončuje. Regulačná oblasť môže byť v prokaryotách zdieľaná viacerými génmi zlúčenými v tzv. operóny. S operónmi súvisí aj prekrývanie génov, ktoré nie je bežné u eukaryotických organizmoch, i keď sa objavujú v poslednom čase objavujú informácie o výskyte takýchto génov aj u týchto organizmov. Kódujúca oblasť je ohraničená štart a stop kodónmi a pozostáva z troch čítacích rámcov, z ktorých však len jeden obsahuje informácie potrebné na syntézu požadovaného proteínu.

Špecifikom prokaryotických génov je Shine-Dalgarnova sekvencia nachádzajúca sa pred štart kodónom. Preklad i prepis prebiehajú vždy v smere 5′→3′.

Nad týmito znalosťami zostrojíme konečný automat a jeho jednotlivé časti bližšie popíšeme v nasledujúcich podkapitolách. Návrh, implementácia a testovanie v tejto práci tak povediac splývajú v jeden iteratívny proces, a tak sa každá podkapitola všetkým týmto aspektom vývoja venuje ako jednému postupne vylepšovanému celku.

Tento automat bude analyzovanú sekvenciu prechádzať posuvným okienkom premenlivej dĺžky. Tá sa mení podľa stavu, v ktorom sa nachádza.




**Obrázok 4.1** *Konečný automat popisujúci kostru navrhovanej metódy predikcie génov, vychádzajúci z referenčného návrhu a síce ústrednej dogmy molekulárnej biológie.*

## 4.2.1 Inicializácia transkripcie, detekcia promótorov

Detekcia promótorov je prvým krokom k predikcii génov, a tak sa stáva jednou z najdôležitejších súčastí navrhovaného konečného automatu. Znalosti nadobudnuté v úvodných kapitolách o transkripcii prokaryotov rozšírime o konkrétnejšie informácie získané zo štúdie [5]. Dôležitým prínosom tejto práce pre nás je predovšetkým kolekcia sekvencií zarovnaných voči energeticky najvýhodnejším pozíciám a podobám promótorov, a zhrnutie poznatkov o ich štruktúre.

Ako sme v úvodných kapitolách naznačili, prokaryotický promótor pozostáva z dvoch signálov, GACA boxu na pozícii -35 a Pribnowovho boxu na pozícii -10. Tieto signály sú okolo uvedených pozícií zarovnané približne na stred. Dá sa teda povedať, že promótor sa skladá z týchto dvoch signálov medzi ktorými je nejaká medzera. Z [5] vyplýva, že až v 92% skúmaných promótorov bola táto medzera o dĺžke 17±1 nukleotidov. U zvyšných 8% bola medzera dĺžky 15 až 21 nukleotidov. Samotná transkripcia začína najčastejšie 7±1 nukleotidov po Pribnowovom boxe, v ostatných prípadoch 4 až 12 nukleotidov po Pribnowovom boxe.

Na detekciu signálov použijeme pozične špecifické matice s log-likelihood hodnoteniami výskytu nukleotidov na jednotlivých pozíciách. Každý zo signálov je definovaný vlastnou maticou. Zostrojená matica GACA boxu bola uvedená vyššie ako Tabuľka 3.3. Maticu pre detekciu Pribnowovho boxu nadefinujeme podobne, opäť vychádzajúc z údajov práce [5], kde na základe zarovnaných sekvencií taktiež zostrojili pozične špecifickú maticu, ale s percentuálnymi ohodnoteniami. Pre

naše účely, prevedieme aj túto maticu do logaritmického priestoru tak, aby hodnoty v jednotlivých bunkách vyjadrovali log-likelihood výskytu nukleotidu na danej pozícii.

| | T | A | T | A | A | T |
|---|---|---|---|---|---|---|
| T | 0,82 | 0,07 | 0,52 | 0,14 | 0,19 | 0,89 |
| G | 0,07 | 0,01 | 0,12 | 0,15 | 0,11 | 0,02 |
| C | 0,03 | 0,03 | 0,14 | 0,13 | 0,52 | 0,05 |
| A | 0,08 | 0,89 | 0,26 | 0,59 | 0,49 | 0,03 |

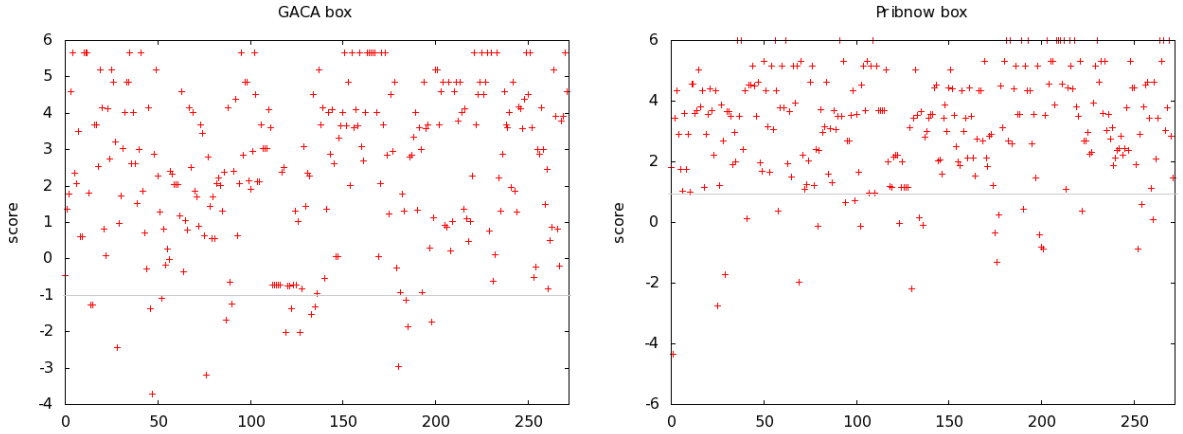
**Tabuľka 4.1** *Pribnowov box ako pozične špecifická matica s percentuálnymi ohodnoteniami v bunkách.*

| | T | A | T | A | A | T |
|---|---|---|---|---|---|---|
| T | 1,18 | -1,27 | 0,73 | -0,57 | -0,27 | 1,26 |
| G | -1,27 | -3,21 | -0,73 | -0,51 | -0,82 | -2,52 |
| C | -2,12 | -2,12 | -0,91 | -0,73 | -0,17 | -1,60 |
| A | -1,13 | 1,26 | 0,03 | 0,85 | 0,67 | -2,12 |

**Tabuľka 4.2** *Pribnowov box definovaný pozične špecifickou maticou v logaritmickom priestore s hodnotami vyjadrujúcimi log-likelihood.*

Podoba s hľadaným signálom je takto špecifikovaným maticiam definovaná ako súčet ohodnotení nukleotidov na jednotlivých pozíciách, takže analýza sekvencie nám vždy vráti nejaké skóre. Definitívne rozhodnutie, či je sekvencia dostatočne podobná s konvenčnou sekvenciou detekovaného signálu a teda môže byť označená za ten-ktorý signál, odvodíme zo spomínaného skóre. Pre maticu každého signálu teda určíme spodnú hranicu skóre, od ktorej je analyzovaná sekvencia označená za hľadaný signál.

Hľadanie vhodných spodných hraníc skóre pre oba signály, GACA box a Pribnowov box, začíname na kompilácii zarovnaných regiónov promótorov.



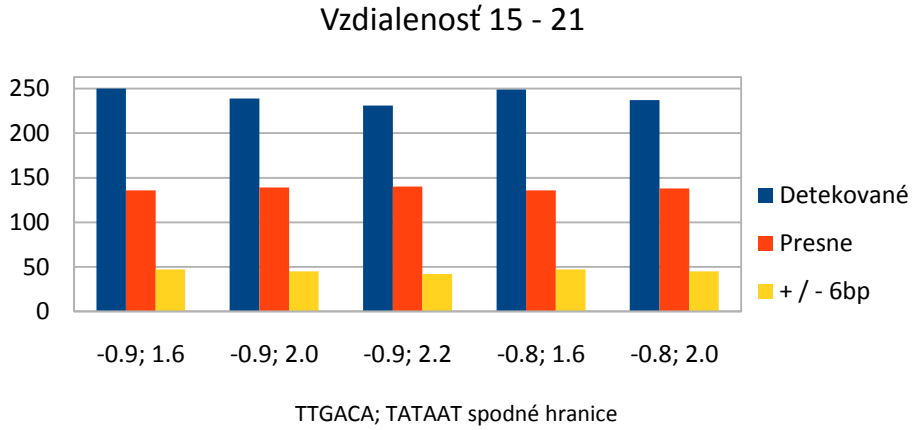

**Obrázok 4.2** *Distribúcia hodnôt skóre u GACA boxu (vľavo) a Pribnowovho boxu (vpravo). Šedé čiary reprezentujú prvý odhad vhodných spodných hraníc jednotlivých signálov.*

Z obrázku vyššie vidno, že Pribnowov box je oveľa zakonzervovanejší ako GACA box, čo nakoniec potvrdzujú aj priemerné hodnoty skóre promótorov v kompilácii, kedy priemerné ohodnotenie GACA boxu vyšlo na 2,47 a Pribnowovho box 3,17. Tieto čísla naznačujú, že pri detekcii promótorov bude mať najväčší význam podobnosť druhého signálu s Pribnowovým boxom, u ktorého budú prísnejšie kritéria klasifikácie príslušnosti k tomuto signálu.

Na základe získaných poznatkov bude detekcia promótorov prebiehať nasledovne. V sekvencii hľadáme najskôr GACA box signál. Každé prekročenie spodnej hranice skóre pre GACA box je zaznamenané, a ak sa nachádzame vo vhodnej vzdialenosti od GACA boxu, tak začína zároveň

s ním aj hľadanie Pribnowovho boxu. Za vhodnú vzdialenosť môžeme považovať 15 až 21 alebo potrebe pri prísnejších kritérií 16 až 18 nukleotidov po koncovom nukleotide GACA boxu.

Takto implementovanú detekciu promótorov sme otestovali na kompilácii zarovnaných promótorov postupne meniac hodnoty spodných hraníc pre jednotlivé signály.



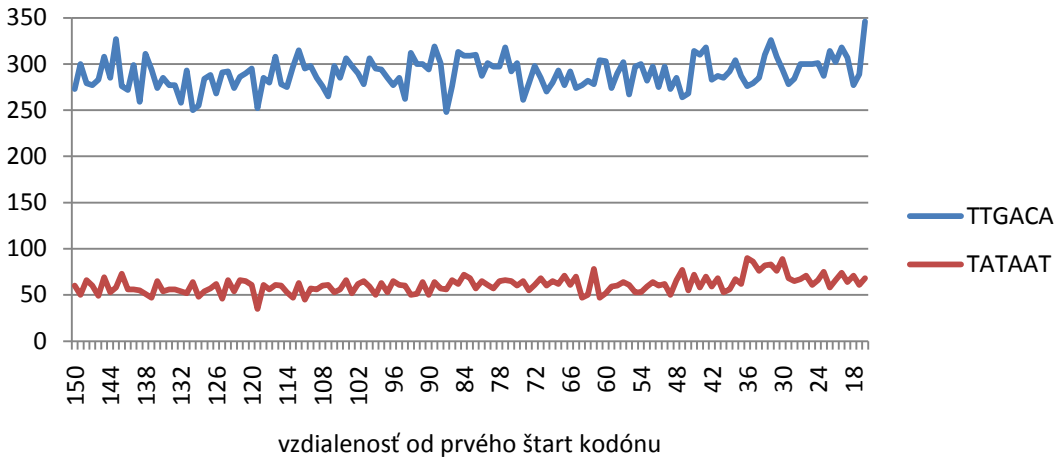

**Obrázok 4.3** *5 najúspešnejších kombinácií spodných hraníc pre jednotlivé signály so vzdialenosťou 15 – 21 nukleotidov.*

Na grafe vyššie vidieť výsledky testovania navrhnutého spôsobu detekcie promótorov. V najlepšom prípade, kedy mal GACA box spodnú hranicu nastavenú na -0,9 a Pribnowov box na 2,2, sa nám podarilo presne nájsť 140 promótorov a 42 ďalších s posunutím menším ako šesť nukleotidov. To dáva dokopy skoro 68% úspešnosť. Nesprávne sme identifikovali 49 promótorov, v zvyšných 41 sekvenciách sme promótor nenašli vôbec. Potom nasledoval rovnaký test s povolenou vzdialenosťou medzi signálmi v intervale <16; 18> nukleotidov. Tu boli výsledky mierne odlišné, kedy sa nám pri hodnotách spodných hraníc -0,9 a 1,1 podarilo presne označiť 146 promótorov a 44 približne, čo činí skoro 70% úspešnosť oproti 57 nesprávne identifikovaným promótorom.

Tieto testy ukázali, že pri prísnejšie nastavenej vzdialenosti nám postačuje benevolentnejšia spodná hranica skóre Pribnowovho boxu. V oboch prípadoch sa spodná hranica pre identifikáciu GACA boxu pohybovala okolo hodnoty -0,9. V prípade Pribnowovho boxu prichádzajú do úvahu hodnoty spodnej hranice od 1,0 v prípade menšieho rozsahu povolenej vzdialenosti medzi dvoma signálmi, pri vzdialenostiach 15 až 21 od hodnoty 1,6 a pre oba prípustné intervaly vzdialenosti sa hodnoty prijateľných spodných hraníc šplhajú až k 2,2. Tieto hodnoty budeme brať ako východzie pre všetky ďalšie testy, pretože vychádzajú z analýzy sekvencií, u ktorých vieme s istotou povedať, na ktorom mieste sa signály promótorov nachádzajú. Na druhej strane je potrebné ďalšie testovanie, keďže doterajšia testovacia množina pozostávala len z krátkych úsekov DNA, a tak takmer určite došlo k určitému skresleniu výsledkov.

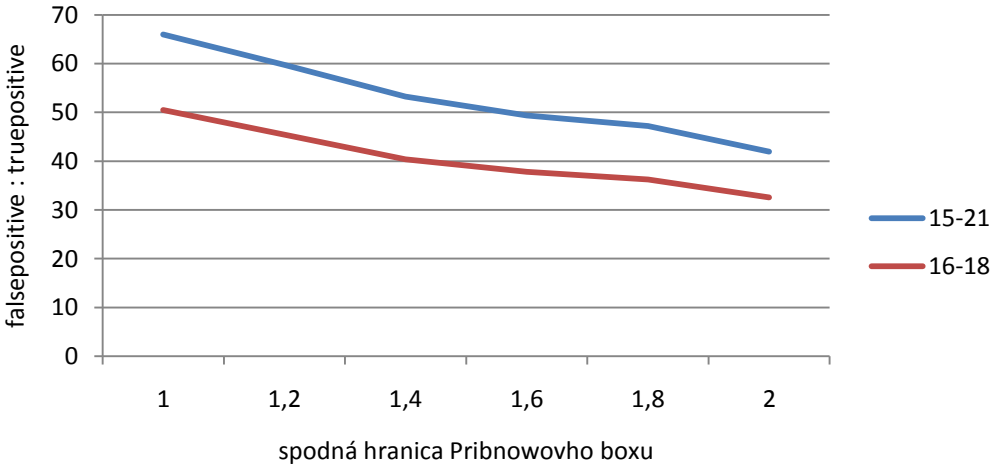
Predtým než sa pustíme do testovania na kompletnej sekvencii DNA *E. coli*, musíme určiť množinu potenciálnych transkripčných jednotiek. Za transkripčnú jednotku budeme považovať kódujúce oblasti s jedným promótorom a terminátorom, teda gény mimo operónov a samotné operóny. Testovanie zúžime na + vlákno, na ktorom sa nachádza 1204 takýchto transkripčných jednotiek. Za správne nájdený promótor budeme považovať promótor vo vzdialenosti od 16 do 100 nukleotidov od prvého štart kodónu transkripčnej jednotky. Minimálna vzdialenosť vychádza z približnej dĺžky signálu Shine-Dalgarno (12 nukleotidov), ktorý je prítomný pred začiatkom translácie v prokaryotických bunkách, a minimálnej vzdialenosti začiatku transkripcie po Pribnowovom boxe (4 nukleotidy). Maximálna vzdialenosť nie je podporená žiadnymi štatistikami, iba odhadom. Pokúsil som sa nájsť oblasť pred transkripčnými jednotkami, ktorá by vykazovala výraznejšiu frekvenciu výskytu promótorov. Jediný takýto región, o ktorom by sa dalo povedať, že sa v ňom končilo viac promótorov sa na-

chádzal 30 až 40 nukleotidov pred prvým štart kodónom (graf nižšie), čo podporuje pôvodný odhad maxima 100 nukleotidov, avšak zvýšenie frekvencie oproti ostatným oblastiam bolo minimálne.




**Obrázok 4.4** *Analýza výskytu signálov promótorov v regióne 150 nukleotidov pred transkripčnými jednotkami.*

Prechádzame teda ku komplexnejšiemu testovaniu na väčšej vzorke, a síce celom + vlákne DNA *E. coli* o dĺžke okolo 4,5 milióna nukleotidov. V týchto testoch nás bude zaujímať počet identifikovaných promótorov, všeobecné vlastnosti týchto sekvencií a počet identifikovaných potenciálnych promótorov vo vzdialenosti maximálne 100 nukleotidov od začiatku nejakej transkripčnej jednotky. Taktiež budeme rôzne kombinovať spodné hranice ohodnotenia jednotlivých signálov, zameraj úc sa predovšetkým na východzie hodnoty z prvého testovania.




**Obrázok 4.5** *Počet falsepositive nálezov prislúchajúcich k jednému truepositive nálezu. Spodná hranica GACA boxu nastavená na -0,9, hranica pre Pribnow box je vynesená na osi x.*

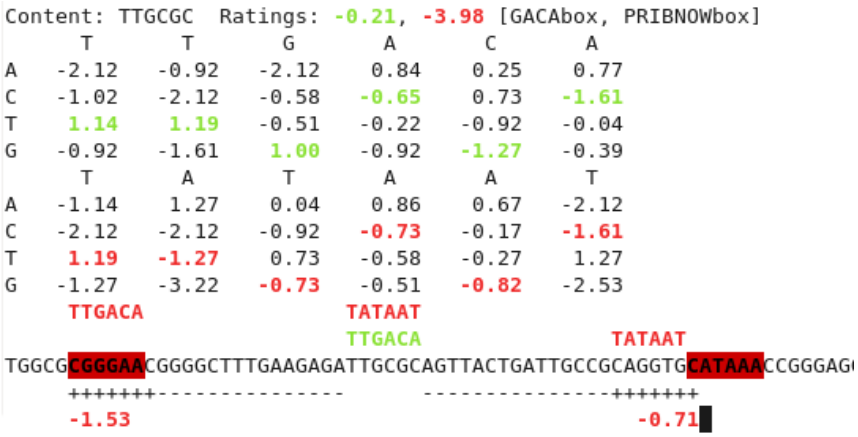
Pri prvom pohľade na výsledky zarazí počet nesprávne identifikovaných promótorov v pomere k počtu potenciálne správne detekovaných, kedy aj pri najprísnejších kritériách vychádzajúcich z predchádzajúceho testovania, teda konfigurácia s povolenou vzdialenosťou 16 až 18 a spodnými hranicami -0,9 a 2,0, nájdeme v DNA sekvencii 31419 potenciálnych promótorov vyhovujúcich našim pravidlám, z čoho pred 936 transkripčnými jednotkami sa potenciálny promótor nachádzal vo vzdialenosti do 100 nukleotidov.

Na druhej strane je pozitívne zistenie, že pri správnej kombinácii spodných hraníc sme schopní zaregistrovať promótori pred 1173 z 1204 transkripčných jednotiek, čo činí 97%. Z tohto pohľadu sme pomerne úspešní aj pri východzích hodnotách spodných hraníc signálov

z predchádzajúcich meraní. Napríklad pri povolenej vzdialenosti v rozmedzí od 15 do 21 nukleotidov medzi signálmi a spodných hraniciach -0,9 a 2,0 úspešne nájdeme promótor pred 1076 transkripčnými jednotkami, čo je stále slušných 89%.

Ďalej z meraní vyplýva, že promótori sa v priemere nachádzajú zhruba 39 nukleotidov pred prvým štart kodónom transkripčných jednotiek. Priemerné ohodnotenia signálov sa pohybujú v okolí 2,5 pre GACA box a 3,2 pre Pribnowov box, čo korešponduje s priemernými hodnoteniami nameranými v množine zarovnaných promótorov. To nasvedčuje tomu, že signály ktoré sme zachytili pred transkripčnými jednotkami budú s vysokou pravdepodobnosťou skutočnými promótormi.

Pred tým než sa pokúsime znížiť počet falsepositive nálezov promótorov, tak sa pozrieme na dôvody, prečo sme neboli ani únosným upravovaním spodných hraníc signálov schopní nájsť promótori pred 31 transkripčnými jednotkami.



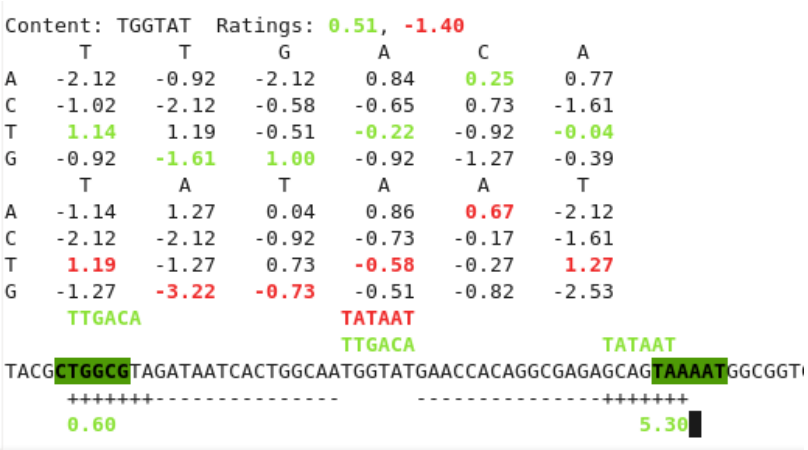

**Obrázok 4.6** *Analýza regiónu predchádzajúcemu transkripčnú jednotku v textovom rozhraní skriptu promoter_analysis.py.*

Podrobná správa o tejto analýze, spolu s výsledkami všetkých spomínaných testov, je uložená na priloženom CD s priečinku *stats/*. Najčastejším problémom bolo, že ak bol aj jeden zo signálov silný, tak k nemu neexistoval žiaden pripomínajúci signál druhý alebo ak aj existoval, tak jeho ohodnotenie bolo nižšie ako prípustná spodná hranica. Taktiež som našiel zo tri promótori, ktorých signály mali medzi sebou nevyhovujúcu vzdialenosť (13, 14 alebo 22). No a ďalšia možná príčina je demonštrovaná na obrázku vyššie, kde pred jednou z transkripčných jednotiek som našiel pomerne silný signál GACA boxu v podobe TTGCGC a k nemu vo vzdialenosti 21 nukleotidov sekvenciu s nízkym ohodnotením, no náramne pripomínajúcu Pribnowov box, CATAAA. V mnohých prípadoch som však pred transkripčnými jednotkami nenašiel ani náznak niečoho, čo by promótor pripomínalo.

Problém s tým, že v niektorých prípadoch bol jeden zo signálov promótora silný a druhý nepreliezol spodnú hranicu o niekoľko desatín, sme skúsili vyriešiť adaptívnou spodnou hranicou Pribnowovho boxu. Predstava bola taká, že ak je signál GACA boxu výrazne silnejší než jeho spodná hranica, tak sa o zlomok tohto rozdielu zníži spodná hranica Pribnowovho boxu. Toto vylepšenie však neprinieslo žiadaný efekt, kedy sa síce o niečo zväčšil počet správne detekovaných promótorov, no neúmerne k tomu narástol počet tých nesprávne detekovaných. Napríklad pri spodných hraniciach -0,9 a 1,6 s povolenou vzdialenosťou 15 – 21, sme boli bez adaptívnej spodnej hranice Pribnowovho boxu správne detekovať 1129 promótorov, po pridaní adaptívnej spodnej hranice tento počet stúpol o 1,3% na 1143, zároveň však počet nesprávne detekovaných promótorov stúpol z 57066 na 62871, čo je nárast o skoro 10%.

Ďalej bolo zaujímavé zistiť, prečo sme našli toľko promótorov v regiónoch, kde by sa podľa záznamov v databázach nachádzať nemali. Vybrali sme preto náhodne región, ktorý má čo najďalej od najbližšej transkripčnej jednotky a jeden príklad, ktorý hovorí za všetky zachytáva obrázok nižšie,

kde vidíme úspešne detekovaný promótor s takmer ideálnou vzdialenosťou 19 nukleotidov medzi jednotlivými signálmi a nadpriemerne vysokým skóre predovšetkým u signálu zodpovedajúcemu Pribnowovmu boxu. Ohodnotenie pravdepodobného GACA boxu je taktiež nad spodnou hranicou, o čom svedčí zelená farba. Tohto konkrétneho príkladu sa síce zbavíme použitím prísnejšieho intervalu pre povolené vzdialenosti medzi signálmi, ale určite nie je jediným, ktorý nájdeme v reťazci DNA.




**Obrázok 4.7** *Snímka promótoru nájdeného v oblasti, kde by sa podľa databáz žiaden nachádzať nemal.*

Nezodpovedanou otázkou zostáva, čo s veľkým množstvom nesprávne identifikovaných promótorov. Dopredu môžem povedať, že zhruba 10% ich bude odignorovaných zavedením detekcie začiatku translácie.

Počet nesprávnych nálezov som sa pokúsil zavedením akéhosi celkového skóre promótora, ktoré sa rovnalo súčtu ohodnotení oboch signálov. Ako sa však ukázalo, toto vylepšenie malo len kozmetický vplyv na výsledok, kedy sa počet nesprávne detekovaných promótorov zhruba o dve percentá bez toho, aby to malo vplyv na správne nájdené promótori.

Na záver sme do návrhu detekcie promótorov zahrnuli dve zo štyroch zovšeobecnených pravidiel o *E. coli*, ku ktorým dospel McClure v [6]. Prvé pravidlo hovorí o tom, že všetky promótori využívajúce sigma-faktor 70, čo je pre pripomenutie primárnym sigma-faktorom v *E. coli* a teda má na starosti detekciu väčšiny promótorov, majú aspoň dva z troch najkonzervovanejších nukleotidov v Pribnowovom boxe (TA...T). Druhé súvisí s GACA boxom a opäť sa týka najkonzervovanejších nukleotidov, tentoraz však stačí iba jeden z trojice TTG na začiatku signálu. Zavedenie týchto pravidiel nemá žiaden vplyv na počet detekovaných promótorov, keďže už len k dosiahnutiu požadovanej spodnej hranice u oboch signálov treba naplniť tieto pravidlá, boli skôr zavedené kvôli zvýšeniu rýchlosti predikcie promótorov kedy iba po splnení týchto pravidiel dochádza k vyhodnoteniu analyzovanej sekvencie, a teda prístupu a počítaniu s maticou.

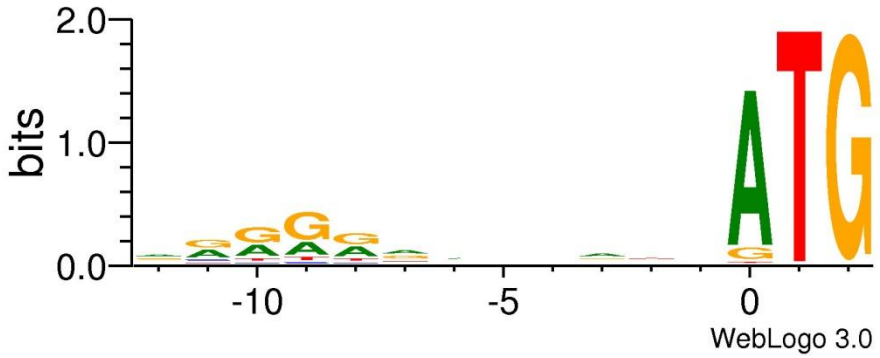
O výbere najvhodnejšej konfigurácie spodných hraníc signálov a povolenej vzdialenosti medzi nimi rozhodneme až na záver návrhu po získaní výsledkov predikcie génov na celom genóme *E.coli*, pričom sa sústredíme na východzie hodnoty prvého testovania.

### 4.2.2 Začiatok prekladu, hľadanie Shine-Dalgarna

Preklad, translácia, v prokaryotách začína zväčša na štart kodóne ATG. Preto pôvodným plánom bolo, po nájdenom promótore hľadať túto postupnosť znakov. Lenže táto myšlienka veľmi rýchlo dostáva trhliny vďaka početnosti takýchto kodónov v reťazci DNA, čo môže byť problém hlavne pri detekcii génov v operónoch, kde sa pred štart kodónom nenachádza detekciu upresňujúci promótor. A aj s detekovaným promótorom takýmto spôsobom často nenájdeme správny štart kodón. Ďalšou chybou na detekcii týmto spôsobom je prosté odignorovanie zvyšných možných štart kodónov,

avšak napríklad alternatívnym štart kodónom GTG začína až zhruba 8% génov v *E. coli* a okolo 1,3% génov začína štart kodónom TTG.

Preto identifikáciu začiatku prekladu rozširujeme o detekciu Shine-Dalgarnovej sekvencie. Trochu problémom je fakt, že táto sekvencia býva rôznej dĺžky, čo ovplyvňuje podobu štart kodónu alebo aj vzdialenosť od neho. Avšak v [7] nájdeme analýzu tejto sekvencie na 1055 génoch, kde pomerne úspešne použili na detekciu Shine-Dalgarnovej sekvencie pozične špecifickú maticu, a tak nemá zmysel vymýšľať nejaké ďalšie metódy. Taktiež dáta potrebné k zostrojeniu pozične špecifickej matice použijeme z tejto práce. Pozične špecifická matica okrem spresnenia detekcie správneho štart kodónu, zahrnie do detekcie aj alternatívne štart kodóny, čím vyriešime oba úvodné problémy. Pred vytvorením tejto matice som sa chcel presvedčiť a sile tohto signálu vo všetkých génoch *E.coli*, čo prezentuje obrázok alebo logo nižšie zobrazujúce množstvo informácie obsiahnuté nukleotidmi na jednotlivých pozíciách.




**Obrázok 4.8** *Logo zachytávajúce informačnú hodnotu Shine-Dalgarnovej sekvencie vo všetkých génoch E. coli. Výsledok zodpovedá tomu, ku ktorému dospeli aj v [7]*

Po overení výskytu tohto signálu v podobe, v akej je prezentovaný v [7] pristúpime teda k vytvoreniu matice, rovnakým spôsobom akým sme to urobili v prípade promótorov. Do matice zahrnieme len prvý nukleotid štart kodónu, keďže zvyšné dva TG sa vyskytujú vo všetkých prípadoch. Táto vlastnosť taktiež pomôže k urýchleniu detekcie Shine-Dalgarnovej sekvencie, keďže k vyhodnocovaniu matice dôjde iba v prípade, že posledné dva nukleotidy analyzovaného okienka budú T a G.

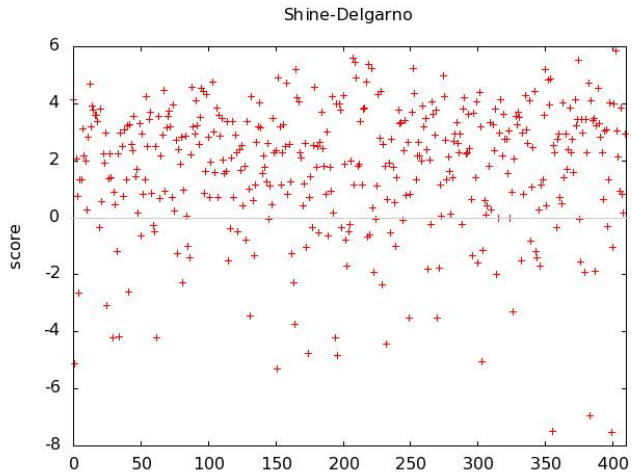
| | -12 | -11 | -10 | -9 | -8 | -7 | -6 | -5 | -4 | -3 | -2 | -1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T | -0,27 | -0,92 | -0,92 | -0,92 | -0,65 | -0,33 | 0,08 | 0,00 | 0,04 | -0,39 | 0,22 | 0,04 | -3,21 |
| G | 0,28 | 0,58 | 0,67 | 0,71 | 0,52 | 0,34 | 0,00 | -0,18 | -0,27 | -0,04 | -0,58 | -0,45 | -1,27 |
| C | -0,65 | -0,92 | -1,14 | -1,27 | -0,92 | -0,73 | -0,51 | -0,22 | -0,22 | -0,33 | 0,11 | 0,15 | -3,22 |
| A | 0,34 | 0,34 | 0,28 | 0,25 | 0,34 | 0,34 | 0,28 | 0,31 | 0,34 | 0,49 | 0,08 | 0,15 | 1,30 |

**Tabuľka 4.3** *Pozične špecifická matica pre Shine-Dalgarnov signál s log-likelihood hodnotami. Posledná pozícia, pozícia 0, patrí prvému nukleotidu štart kodónu.*

Podobný, ale nie až tak rozsiahly, postup ako pri promótoroch pokračoval aj pri hľadaní spodnej hranice pre túto maticu. Nakoniec sme za túto medz zvolili nulu, čomu vyhovuje 3760 z 4294 začiatkov génov v *E. coli*, čo činí 87,6%. Mohlo by sa zdať, že sa dalo vyťažiť väčšie percento detekovateľných génov znížením spodnej hranice o niečo, no vzhľadom k počtu nesprávne nájdených promótorov, je treba pri hodnotení Shine-Dalgarnovej sekvencie, ktorá je akýmsi rozšírením detekcie promótorov, zvoliť prísnejšie kritéria.

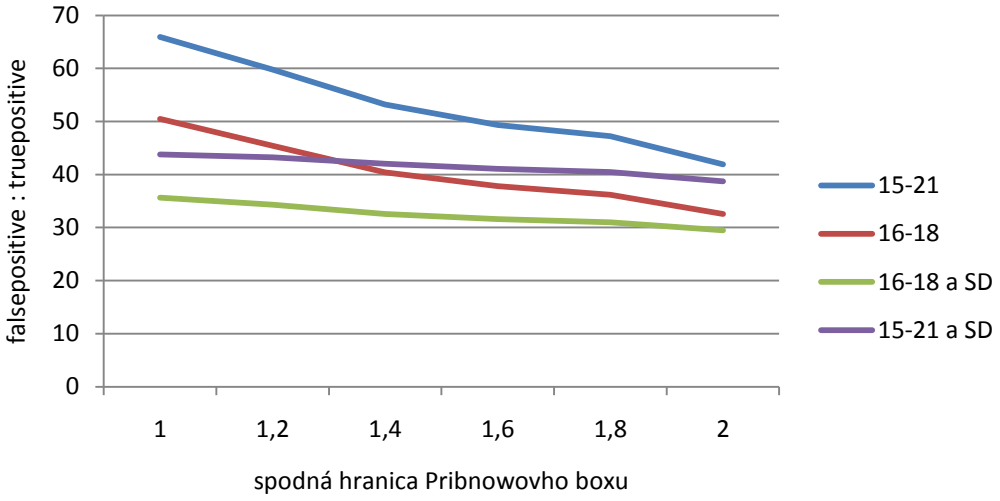
Detailnejšie sme sa pozreli na začiatky translácií, kedy ohodnotenie maticou viedlo ku skóre menšiemu ako 0. Často sa stávalo, že skóre bolo len o stotiny menšie od nuly, čo by však bolo prob-

lémom pri akejkoľvek spodnej hranici ohodnotenia. Ďalšou častou príčinou neúspechu identifikácie začiatku prekladu bolo veľké množstvo tymínu na miestach, kde sa v Shine-Dalgarnovej sekvencii očakávajú guanín alebo adenínom. Predpokladám, že tento problém súvisí s použitím pozične špecifickej matice na identifikovanie Shine-Dalgarna, keďže ako bolo na začiatku povedané, pozícia regiónu bohatého na A a G zároveň závisí i na jeho dĺžke a potom štart kodóne.




**Obrázok 4.9** *Vizualizácia ohodnotení začiatkov translácie v prípade operónov v E. coli.*

Tu sa dostávame k naplnenie ďalšieho z prísľubov spresnenia detekcie začiatku translácie, a síce s tým súvisiacou presnejšou detekciou promótorov. Teraz sme totižto schopní prehlásiť promótor za falošný, ak sa do 100 nukleotidov za ním nenájde Shine-Dalgarno. Ako som naznačil v predchádzajúcej podkapitole, počet nesprávne označených promótorov klesol v priemere o 10%, napríklad u nami preferovanej kombinácie -0,9 a 2,0 s povolenou vzdialenosťou <15; 21> z 46205 na 41686 nesprávne detekovaných promótorov. Prvý krát sa taktiež dostávame k počtu presne nájdených začiatkov translácie transkripčných jednotiek, ktorý pri vyššie spomenutej konfigurácii nadobúda hodnoty 933.




**Obrázok 4.10** *Zlepšenie predikcie promótorov začlenením Shine-Dalgarna.*

## 4.2.3 Koniec prekladu, odhad správneho ORF

Po nájdenom začiatku prekladu nasleduje hľadanie správneho stop kodónu. Ako bolo v teoretickom úvode povedané, každý gén kóduje aminokyseliny po kodónoch a podľa toho, na kto-

rom nukleotide za štart kodónom začneme čítať, má každý gén tri rôzne čítacie rámce (ORF), avšak len jeden z nich skutočne kóduje proteín.

Prvým nápadom bolo označiť za koniec prekladu prvý stop kodón, na ktorý po ceste od štart kodónu narazíme. Takto však správne identifikujeme len minimum génov. Vyvstáva teda otázka, ako identifikovať správny ORF?

Odpoveď nachádzame vo frekvencii kódovania aminokyselín kodónmi[1]. Je totiž známe, že aminokyseliny sú niektorými kodónmi kódované častejšie ako druhými. Zvyknutí na počty v logaritmickom priestore sme ohodnotenia použitia jednotlivých kodónov transformovali na loglikelihood hodnoty, ktoré vyjadrujú pravdepodobnosť použitia vybraného kodónu pre zápis danej aminokyseliny.

| | | 2.báza | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | U | | C | | A | | G | |
| 1.báza | U | UUU (Phe/F) | 0,13 | UCU (Ser/S) | 0,00 | UAU (Tyr/Y) | 0,15 | UGU (Cys/C) | -0,11 |
| | | UUC (Phe/F) | -0,15 | UCC (Ser/S) | -0,06 | UAC (Tyr/Y) | -0,17 | UGC (Cys/C) | 0,10 |
| | | UUA (Leu/L) | -0,21 | UCA (Ser/S) | -0,21 | UAA (stop) | 0,63 | UGA (stop) | -0,10 |
| | | UUG (Leu/L) | -0,21 | UCG (Ser/S) | -0,06 | UAG (stop) | -1,42 | UGG (Trp/W) | 0,00 |
| | C | CUU (Leu/L) | -0,37 | CCU (Pro/P) | -0,39 | CAU (His/H) | 0,13 | CGU (Arg/R) | 0,84 |
| | | CUC (Leu/L) | -0,47 | CCC (Pro/P) | -0,39 | CAC (His/H) | -0,15 | CGC (Arg/R) | 0,86 |
| | | CUA (Leu/L) | -1,39 | CCA (Pro/P) | -0,22 | CAA (Gin/Q) | -0,11 | CGA (Arg/R) | -0,83 |
| | | CUG (Leu/L) | 1,12 | CCG (Pro/P) | 0,71 | CAG (Gln/Q) | 0,28 | CGG (Arg/R) | -0,47 |
| | A | AUU (Ile/I) | 0,42 | ACU (Thr/T) | -0,33 | AAU (Asn/N) | -0,08 | AGU (Ser/S) | -0,06 |
| | | AUC (Ile/I) | 0,22 | ACC (Thr/T) | 0,52 | AAC (Asn/N) | 0,08 | AGC (Ser/S) | 0,49 |
| | | AUA (Ile/I) | -1,30 | ACA (Thr/T) | -0,51 | AAA (Lys/K) | 0,41 | AGA (Arg/R) | -1,16 |
| | | AUG (Met/M) | 0,00 | ACG (Thr/T) | 0,04 | AAG (Lys/K) | -0,69 | AGG (Arg/R) | -1,67 |
| | G | GUU (Val/V) | 0,08 | GCU (Ala/A) | -0,39 | GAU (Asp/D) | 0,23 | GGU (Gly/G) | 0,31 |
| | | GUC (Val/V) | -0,17 | GCC (Ala/A) | 0,08 | GAC (Asp/D) | -0,30 | GGC (Gly/G) | 0,44 |
| | | GUA (Val/V) | -0,45 | GCA (Ala/A) | -0,13 | GAA (Glu/E) | 0,31 | GGA (Gly/G) | -0,73 |
| | | GUG (Val/V) | 0,36 | GCG (Ala/A) | 0,31 | GAG (Glu/E) | -0,45 | GGG (Gly/G) | -0,51 |

**Tabuľka 4.4** *Kodóny kódujúce jednotlivé aminokyseliny a zodpovedajúce loglikelihood ohodnotenia vychádzajúce z frekvencií výskytu týchto kodónov.*

Každé ORF je ohodnotené súčtom hodnotení kodónov, podobne ako tomu je pri získavaní skóre u pozične špecifických matíc, a následne vyberieme to s najvyšším ohodnotením. Takýmto spôsobom sa nám podarilo správne určiť 3889 čítacích rámcov oproti 405 nesprávne určeným v kompletnom genóme *E. coli*. Mohlo by sa zdať, že takto nebudeme schopní určiť správne hlavne dlhšie ORF, no prekvapujúco najväčším problémom sa stal jav, kedy skutočné ORF nekončil na prvom stop kodóne, k čomu došlo v 268 prípadoch. Zvyšok neidentifikovaných rámcov malo veľmi nízke skóre a len v jednom prípade sa stalo, že kratšie ORF malo o niečo lepšie ohodnotenie než dlhšie skutočné.

Aby sme eliminovali možnosť uprednostňovania kratších ORF pred dlhšími, skúsili sme celkové ohodnotenie rámcov nahradiť priemerným skóre na kodón, čo ale neprinieslo žiadne zlepšenie, práve naopak.

[1] MARTÍNEK, T., Rozpoznávaní genů. Bioinformatika. FIT VUT v Brně (slidy). Dostupné z WWW: < https://www.fit.vutbr.cz/study/courses/BIF/private/presentations/rozpoznavani_genu/rozpoznavani_genu.pdf>

### 4.2.4 Terminácia transkripcie, detekcia terminátorov

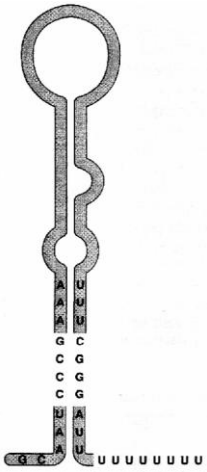
Dostávame sa k poslednej časti predikcie génov, detekcie terminátorov. Terminátor je signál ukončujúci transkripciu uvoľnením väzby medzi RNA a DNA. Hlavným problémom detekcie týchto signálov je fakt, že jeden druh terminátorov, rho-závislé terminátori, sú závislé len na rho-faktore a nemajú žiadnu špecifickú štruktúru a teda ich nie sme schopný z reťazca DNA vyčítať. Takže nám zostávajú rho-nezávislé terminátori, ktoré sa vyznačujú palindromatickou štruktúrou nasledovanou reťazcom bohatým na tymín.

Predstava detekcie terminátorov je nasledovná: po určení správneho ORF budeme hľadať sekvenciu bohatú na tymín a keď ju nájdeme, tak sa pozrieme na oblasť pred ňou a vyhľadáme v nej palindromatické štruktúry. V detekcii terminátorov budeme pokračovať iba do určitej vzdialenosti od posledného stop kodónu, a ak terminátor nenájdeme tak považujeme ukončenie transkripciou spôsobené rho-závislým terminátorom, o tom ale až v ďalšej kapitole.

Palindróm je reťazec, ktorého obsah zostáva rovnaký, či ho čítame spredu alebo odzadu, čo môžeme zapísať ako:

$$p = w.w' \; alebo \; p = w.c.w' \tag{4.1}$$

kde $w$ je reťazec, $c$ je symbolom, písmenom abecedy, a $w'$ je $w$ zapísané v opačnom smere, teda odzadu. V našom prípade budeme $w'$ uvažovať nie len ako reverzné $w$, ale ako reverzné a komplementárne $w$. Taktiež $c$ nebude len jedným symbolom ale viacerými, keďže tu budeme očakávať hlavičku slučky palindromatickej štruktúry terminátora. Ďalej budeme počítať aj s tým, že v oboch reťazcoch $w$ aj $w'$ sa v našom prípade môžu objaviť menšie chyby, napríklad znak navyše. Zrozumiteľnejšie to asi vidno na obrázku nižšie, kde je vyobrazená podoba terminátora.



**Obrázok 4.11** *Terminátor[2], palindromatická štruktúra pripomínajúca slučku, nasledovaná uracilmi, v DNA tymínmi.*

Na tieto účely nám poslúži algoritmus opísaný v [8], respektíve je jednoduchšia upravená podoba. Spomínaný algoritmus patrí medzi algoritmy dynamického programovania, a je postavený na vytvorení distančnej matice $l$ x $l$, kde $l$ je dĺžka skúmaného reťazca. Algoritmus bude najlepšie demonštrovať na príklade. Urobíme tak na reťazci *TATATACTA*, kde očakávame nález palindromatickej štruktúry dĺžky 4 nukleotidov na každej strane (strany sú vzájomne komplementárne) s jednou chybou.

[2] Zdroj: MARTÍNEK, T., Rozpoznávaní genů. Bioinformatika. FIT VUT v Brně (slidy). Dostupné z WWW: < https://www.fit.vutbr.cz/study/courses/BIF/private/presentations/rozpoznavani_genu/rozpoznavani_genu.pdf>

Pri inicializácii matice, označme ju *M*, vyplníme strednú a hneď po nej nasledujúcu diagonálu nulami, tak ako to je zobrazené v tabuľke nižšie, a všetky ostatné bunky naplníme -1, ako doposiaľ nedopočítanou hodnotou.

| | T | A | T | A | T | A | C | T | A |
|---|---|---|---|---|---|---|---|---|---|
| A | 0 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| T | 0 | 0 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| A | X | 0 | 0 | -1 | -1 | -1 | -1 | -1 | -1 |
| T | X | X | 0 | 0 | -1 | -1 | -1 | -1 | -1 |
| A | X | X | X | 0 | 0 | -1 | -1 | -1 | -1 |
| T | X | X | X | X | 0 | 0 | -1 | -1 | -1 |
| G | X | X | X | X | X | 0 | 0 | -1 | -1 |
| A | X | X | X | X | X | X | 0 | 0 | -1 |
| T | X | X | X | X | X | X | X | 0 | 0 |

**Tabuľka 4.5** *Inicializácia distančnej matice, vodorovne pôvodný reťazec, zvislo k nemu komplementárny. S hodnotami pod diagonálou sa nebude ďalej počítať, preto sú nahradené x.*

Obsah matice *M* nad dvoma hlavnými, nulami vyplnenými, diagonálami vyplníme postupne po diagonálach na základe rovnice:

$$M_{i,j} = min\begin{cases} M_{i-1,j} + 1 \\ M_{i,j-1} + 1 \\ M_{i-1,j-1} + \left(w_i == w'_j \,?\, 0 : 1\right) \end{cases} \tag{4.2}$$

kde $w_i$ je *i*-ty symbol reťazca *w* a $w'_j$ je *j*-ty symbol reťazca *w´*, ktorý je reverzný a komplementárny k pôvodnému reťazcu *w*.

| | T | A | T | A | T | A | C | T | A |
|---|---|---|---|---|---|---|---|---|---|
| A | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 2 | 1 |
| T | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 2 |
| A | X | 0 | 0 | 0 | 1 | 0 | 1 | 2 | 1 |
| T | X | X | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| A | X | X | X | 0 | 0 | 0 | 1 | 1 | 0 |
| T | X | X | X | X | 0 | 0 | 1 | 0 | 1 |
| G | X | X | X | X | X | 0 | 0 | 1 | 1 |
| A | X | X | X | X | X | X | 0 | 0 | 0 |
| T | X | X | X | X | X | X | X | 0 | 0 |

**Tabuľka 4.6** *Matica po aplikovaní vyššie uvedených pravidiel, zelenou farbou je vyznačený nájdený približný palindróm dĺžky 4 a s jednou chybou.*

V takto zhotovenej matici jednotlivé bunky obsahujú minimálnu chybu po optimálnej ceste od hlavnej diagonály, teda počet chýb možného palindrómu. Dĺžky približných palindrómov sa rovnajú dĺžkam postupností rovnakých čísiel na severovýchodne orientovaných diagonálach. V tomto prípade je to palindróm dĺžky 4 na každej strane s jednou chybou.

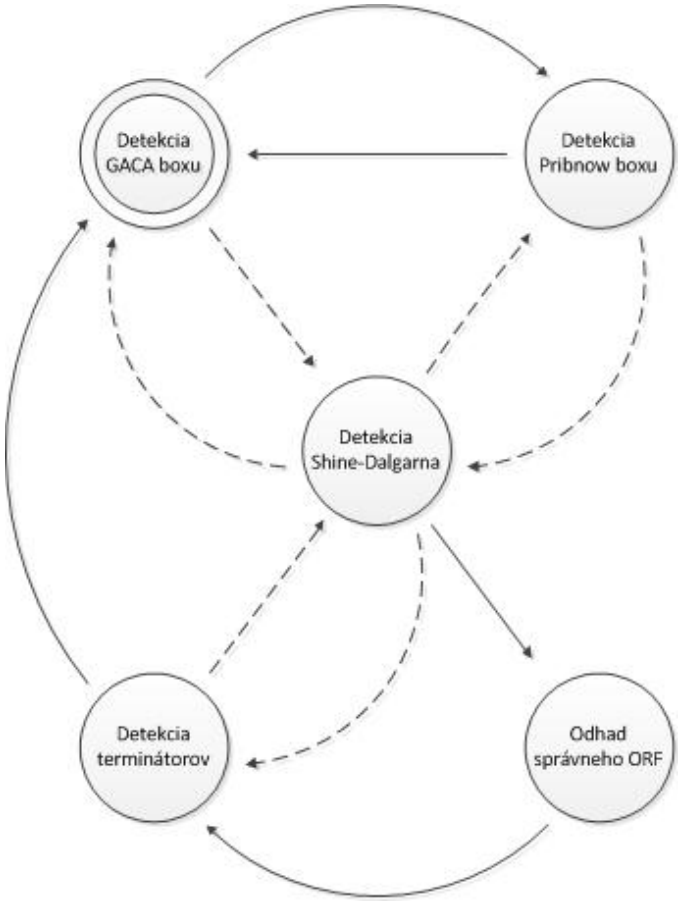
Takto principiálne funguje aj navrhnutá detekcia palindromatických štruktúr, kde však berieme do úvahy, že palindromatická štruktúra má byť nasledovaná sekvenciou tymínov, a tak hľadané palindrómy, respektíve postupnosti, musia začínať v poslednom nanajvýš predposlednom stĺpci našej matice. Taktiež je toto vyhľadávanie obmedzené maximálnou respektíve minimálnou chybou a posledné pravidlo, ktoré musí palindromatická štruktúra spĺňať je určitá minimálna dĺžka. Časová zložitosť základného algoritmu je v najhoršom a na reálnych dátach veľmi nepravdepodobnom prípade

$O(n^2)$, v praxi končí beh oveľa rýchlejšie, a keď navyše pridáme spomenuté pravidlá, tak takejto zložitosti určite nedosiahneme, keďže po palindrómoch pátrame iba v určitej časti matice.

Vhodnú konfiguráciu elementov ako minimálna dĺžka, maximálna a minimálna chyba hľadáme, podobne ako tomu bolo pri promótoroch, 100 nukleotidov za transkripčnými jednotkami. Jednotlivými parametrami nemôžeme len tak voľne hýbať, aby sme zachovali schopnosť zachytiť špecifiká rho-nezávislých terminátorov. Preto napríklad minimálna dĺžka jednej strany palindrómu nesmie klesnúť pod 7 nukleotidov a minimálna chyba zasa pod 2 nukleotidy, aby v strede vznikala nejaká slučka. Najlepšie z testovania vyšli konfigurácie so spomenutými minimami a maximálnou chybou 8, respektíve 10, čo je však už trochu dosť. V prvom prípade sme za 694 transkripčnými jednotkami našli terminátor, v tom druhom prípade dokonca za 774. Nedá mi však nepodotknúť, že kvôli takýmto „lepším“ výsledkom, sme museli zjemniť podmienku, za akých bola sekvencia označená za bohatú na tymín (krok, ktorý predchádza detekcii palindromatických štruktúr), a síce z pôvodných 6 z 8 nukleotidov na 4 z 8. Je to istý kompromis, ku ktorému sme dospeli opierajúc sa o doterajšie poznatky a informácie z [9], kde dospeli k záveru, že na tymín bohatá sekvencia nie je smerodajnou pri detekcii terminátorov. Pre porovnanie, pri pôvodnej podmienke 6 z 8 sme zaregistrovali iba 338 terminátorov za transkripčnými jednotkami.

# 4.3 Zlúčenie

Ako náhle sme sfunkčnili jednotlivé časti prediktora, prichádza čas na ich zlúčenie. Tento proces bol tiež iteratívny, no v tomto prípade obmedzíme výklad už len na záverečné riešenie. Výsledný kompletný konečný automat pozostáva dovedna z piatich stavov: detekcia GACA boxu, Pribnowovho boxu, začiatku, konca translácie a konca transkripcie.




**Obrázok 4.12** *Schéma navrhnutého konečného automatu.*

Počiatočným stavom je detekcia GACA boxu, kedy skenujeme sekvenciu hľadajúc úseky s ohodnotením prekračujúcim spodnú hranicu pre tento signál. Nález každého takéhoto miesta je zaznamenaný. V prípade, že sa posuvné okienko, ktorým vstupnú sekvenciu analyzujeme, nachádza vo vzdialenosti od nejakého zaznamenaného GACA boxu, kde by sa mohol nachádzať Pribnowov box, prechádzame do zodpovedajúceho stavu. Pri detekcii Pribnowovho boxu zároveň detekujeme aj ďalšie prípadné GACA boxy, takže je posuvné okienko analyzované dvakrát. Pozície kompletných promótorov sú taktiež ukladané a kým sa posuvné okienko nachádza vo vhodnej vzdialenosti pre začiatok translácie, teda 4 až 100 nukleotidov od konca niektorého z uložených promótorov, tak je okienko analyzované i voči Shine-Dalgarnovej sekvencii. V týchto stavoch má posuvné okienko dĺžku 15 nukleotidov, práve kvôli možnosti testovania výskytu Shine-Dalgarna. Ak je tento signál detekovaný, tak sa dĺžka okienka zmenší na 3 nukleotidy a začína hľadanie správneho ORF rámca.

Po identifikovaní správneho stop kodónu automat prechádza do stavu detekcie terminátorov. Okrem hľadania týchto ukončovacích signálov prebieha opäť vyhľadávanie Shine-Dalgarnovej sekvencie, keďže sa môžeme nachádzať v operóne. Práve kvôli operónom je pred prechodom do tohto stavu okienko posunuté niekoľko nukleotidov pred nájdený stop kodón, aby sme mohli zachytiť prípadné prekrývajúce sa gény, ktoré sa v operónoch môžu vyskytovať a s najvyššou frekvenciou zvyknú začínať práve jeden nukleotid pred stop kodónom, kedy stop kodón TGA slúži z časti zároveň ako štart kodón nasledujúceho génu ako je to ukázané na obrázku nižšie. O tomto jave pojednáva [10], kde takéto prekrývanie našli až v 16% po sebe nasledujúcich génov v operónoch *E. coli*.


**Obrázok 4.13** *Dva najčastejšie prípady prekrývania génov v operónoch E. coli. V 16% k nasledujúci gén začína jeden nukleotid pred posledným stop kodónom, teda vo vzdialenosti -4, v zhruba 10% prípadov začína štart kodón nasledujúceho génu na poslednom nukleotide stop kodónu toho predchádzajúceho, teda vo vzdialenosti -1.*

V stave detekcie terminátorov zotrváva buď do identifikovania terminátoru alebo štart kodónu nasledujúceho génu v prípadnom operóne alebo do vzdialenosti 100 nukleotidov po poslednom stop kodóne. V takom prípade je skutočnosť, že ku génu nebol detekovaný žiaden terminátor uložená do záznamu o géne, a automat pokračuje ďalej predikciou promótorov. Z behov programu na sekvencii DNA *E. coli* vyplýva, že aj takto označené gény môžu byť predikciou skutočných génov zo záznamov v databázach, čo sa dá pripísať na vrub nedokonalostiam detekcie terminátorov, respektíve rho-závislým terminátorom.

| eschColi_K12_refSeq_b0001 range=chr:90-355 strand=+ | |
|---|---|
| CGTGAGTAAATTAAAATTTTATTGACTTAGGTCACTAAATACTTTAACCA<br>ATATAGGCATAGCGCACAGACAGATAAAAATTACAGAGTACACAACATCC<br>ATGAAACGCATTAGCACCACCATTACCACCACCATCACCATTACCACAGG<br>TAACGGTGCGGGCTGACGCGTACAGGAAACACAGAAAAAAGCCCGCACCT<br>GACAGTGCGGGCTTTTTTTTTCGACCAAAGGTAACGAGGTAACAACCATG<br>CGAGTGTTGAAGTTCG | GACA box 1,81<br>Pribnow box 2,97<br>Shine-Dalgarno 0,17<br>ORF 189-255<br>Terminátor dĺžky 8 |

**Tabuľka 4.7** *Ukážka správne predikovaného génu, šedou farbou sú postupne vyznačené GACA box a Pribnow box vo vzdialenosti 16 nukleotidov od seba, nasleduje Shine-Dalgarno signál žltej farby, zelený štart kodón a ORF ukončené fialovo označeným stop kodónom TGA. Môžeme si všimnúť, že sa pred odhadnutým stop kodónom nachádza ešte jeden TAA, ukončujúci iný čítací rámec. Na konci je červenou vyznačená palindromatická štruktúra terminátora.*

Pri behu na reálnych dátach prehľadá automat najskôr + vlákno zadanej DNA a potom toto vlákno otočí a vytvorí vlákno k nemu komplementárne, čím dostávame – vlákno, na ktorom hľadanie pokračuje.

Takto navrhnutý konečný automat sme otestovali na reálnych dátach, DNA reťazci *E. coli*, použijúc rôzne kombinácie jednotlivých parametrov s cieľom dosiahnuť čo najlepších výsledkov, teda čo najviac správne predikovaných génov a zároveň čo najmenej nesprávne predikovaných génov. Po dokončení predikcie sú z výsledkov vystrihnuté všetky gény kratšie 45 nukleotidom, ktoré v prípade *E. coli* tvorili zhruba polovicu nájdených, keďže najkratší zaznamenaný gén v *E. coli*, ale i vo väčšine prokaryotov je dlhý práve daných 45 nukleotidov.

| Najlepšie dosiahnuté výsledky pri predikcii génov v DNA *E. coli* | |
|---|---|
| Doba behu na + vlákne: | 152,05 s |
| Celková doba behu: | 282,12 s |
| Celkový počet predikovaných génov: | 11817 |
| Celkový počet génov v *E. coli*: | 4294 |
| Počet správne predikovaných génov: | 1905 |
| Počet správne predikovaných stop kodónov: | 3133 |
| Počet správne predikovaných štart kodónov: | 1974 |

**Tabuľka 4.8** *Najlepšie dosiahnuté výsledky navrhnutou metódou predikcie génov na DNA E. coli.*

Najlepších výsledkov, opierajúc sa o dielčie výsledky z predchádzajúcich kapitol, sme dosiahli pri spodných hraniciach -0,9 pre GACA box a 2,0 pre Pribnow box. Povolená vzdialenosť medzi signálmi bola 16 až 18 nukleotidov. Za terminátor boli považované palindromatické štruktúry s minimálnou dĺžkou jednej strany 7, minimálnou chybou 2 nukleotidy, maximálnou 8, nasledované na tymín bohatou sekvenciou (4 z 8 nukleotidov okienka boli tymíny). Pred behom programu je možné ju zmeniť úpravou hodnôt v konfiguračnom súbore *config.ini* umiestnenom v rovnakom priečinku ako spúšťací skript. Pre porovnanie, program GeneMark.hmm-P založený na Skrytých Markovovských modeloch a vyvíjaný na Georgia Institute of Technology v *E. coli* správne identifikoval 3151 génov a v 979 ďalších prípadoch správne detekoval stop kodón, ale netrafil sa do štart kodónu.

Časovo najzložitejšou časťou navrhnutého metódy je detekcia terminátorov, ktorá by sa teoreticky mohla priblížit k $O(n^2)$, čo je však prakticky nereálne, ako to bolo vysvetlené v predchádzajúcej kapitole, navyše automat trávi minimum času v tomto stave. Asi najčastejšie sa nachádza v stave detekcie častí promótorov, kedy je každý nukleotid kontrolovaný minimálne 6-krát (dĺžka jednotlivých signálov promótorov, keďže okienko môžeme posúvať len po jednom nukleotide) a maximálne 24-krát v prípade, že by bol kontrolovaný pri každom posunutí okienka na všetky tri signály, ktoré sa v tomto stave detekujú.

# 5 Záver

Výsledkom práce je metóda predikcie prokaryotických génov aplikovateľná predovšetkým na genóm *E. coli*. Metóda je založená na konečnom automate a skúmanú sekvenciu prechádza v jednom behu. Z podstaty návrhu metódy, ako konečného automatu vyplývajú určité nevýhody, ako napríklad zavádzanie chyby do výsledkov pri nesprávne identifikovanom géne, v ktorého kódujúcej oblasti sa môže nachádzať začiatok skutočného génu, ktorý je týmto pádom preskočený. Na druhej strane je predikcia génov takýmto spôsobom pomerne rýchla a nevyžaduje napríklad presnú identifikáciu terminačných signálov, ktoré v niektorých prípadoch vôbec nie sme schopní detekovať.

Čo sa týka budúceho pokračovania práce, hlavným cieľom by malo byť predovšetkým odstránenie veľkého počtu nesprávne identifikovaných génov. Toho môžeme dosiahnuť spresnením jednotlivých stavov navrhnutého konečného automatu a jeho následnou transformáciou napríklad na Generalizovaný Skrytý Markovovský model, o ktorom síce v práci nebola reč, ale v krátkosti sa jedná o rozšírenie Skrytých Markovovských modelov, kedy každý stav emituje celú sekvenciu na rozdiel od len jedného symbolu emitovaného Skrytými Markovovskými modelmi. Viac informácií nájdete napríklad v [11].

Spresnenie detekcie promótorov by sa možno dalo dosiahnuť použitím pozične špecifických matíc vyššieho rádu, ale to len v prípade, keby existovala výraznejšia spojitosť, väzba, medzi nukleotidmi jednotlivých signálov. Ďalšou zaujímavou možnosťou je rozšíriť detekciu promótorov o tzv. UP element nachádzajúci sa pred GACA boxom od pozície -59 po -38 (GACA box okupuje pozície okolo -35). Tento UP element je úsek bohatý na tymín a adenín, s konvenčnou sekvenciou definovanou v [12] ako 5´-nnAAA(A/T)(A/T)T(A/T)TTTTnnAAAAnnn-3´.

Iné a presnejšie riešenie detekcie Shine-Dalgarnovej sekvencie ponúka napríklad použitie neurónových sietí, ktoré nájdeme v [13], kde sa podarilo dosiahnuť až zhruba 95% úspešnosti pri 1,5% falsepositive nálezov.

S navrhnutým riešením odhadu správnych ORF som pomerne spokojný a dosiahnutá 90% úspešnosť ma pravdu povediac prekvapila. Zlepšenie by mohlo priniesť prípadné použitie trojperiodických Markovovských reťazcov, ktorými sme schopní hodnotiť rámce v širších súvislostiach (napríklad závislosti medzi po sebe idúcimi kodónmi).

Najväčším tŕňom v oku zostáva detekcia terminátorov, ktorej úspešnosť sa pohybovala okolo 50%. Riešením môže byť použitie pozične špecifickej matice a vynechaním na tymín bohatej sekvencie z rovnice, ako to nájdeme v [9]. Problémom však aj naďalej zostanú rho-závislé terminátori, ktoré by sa možno dali odhadovať nejakou simuláciou rho-faktoru, to by ale vyžadovalo hlbšie poznatky o danej problematike.

Za hlavný prínos tejto práce by som označil predovšetkým zozbieranie poznatkov o prokaryotických génoch a analýzu jednotlivých častí prokaryotických génov spolu s príslušnými skriptami, ktoré nájdete na priloženom CD.

# Literatúra


[1] ALBERT, B., BRAY, D., JOHNSON, A., LEIWS, J., RAFF, M., ROBERTS, K. a WALTER, P. *Základy buněčné biologie: Úvod do molekulární biologie buňky.* 2. vyd.: Espero Publishing, 2005. 740 s. ISBN: 80-902906-2-0.

[2] MATHÉ. C., SAGOT, M., SCHIEX, T. a ROUZÉ, P. Current methods of gene prediction, their strengths and weaknesses. *Nucleic Acids Research.* 2002, 30, 19, s. 4103-4117

[3] DURBIN, R., EDDY, S., KROGH, A. a MITCHISON, G. *Biological sequence analysis: Probabilistic models of proteins and nucleic acids.* Cambridge University Press, 2002. 356 s. ISBN: 0-521-63971-3

[4] RABINER, LR. A Tutorial on Hidden Markov Models and Selected Application in Speech Recognition. *Proceedings of the IEEE.* Február 1989, 77, 2, s. 257-286

[5] HARLEY, B. a REYNOLDS, R. Analysis of E. coli promoter sequences. *Nucleic Acids Research.* November 5 1987, 15, s. 2343–2361.

[6] MCCLURE, WR. Mechanism and control of transcription initiation in prokaryotes. *Annu. Rev. Biochem.* 1985, 54, s. 171-204

[7] BARRICK, D., et al. Quantitative analysis of ribosome binding sites in E. coli. *Nucleic Acids Research.* 1994, 22, 7, s. 1287-1295

[8] LLOYD, Allison. *Finding Approximate Palindromes in Strings Quickly and Simply.* [s.l.], 2004. 4 s. Technical report. CSSE Monash University

[9] BRENDEL, V. a TRIFONOV E.N. A computer algorithm for testing potential prokaryotic terminators. *Nucleic Acids Research.* 1984, 12, 10, s. 4411-4427

[10] SALGADO, H., et al. Operons in Escherichia coli: Genomic analyses and predictions. *PNAS.* 2000, 97, 12, s. 6652-6657

[11] MAJOROS, W. H. Methods for Computational Gene Prediction. [s.l.] : Cambridge University Press, 2007. 448 s. ISBN 9780521877510

[12] ESTREM, S.T., et al. Identification of an UP element consensus sequence for bacterial promoters. *PNAS.* August 1998, 95, s. 9761-9766

[13] BISANT, D. a MAIZEL, J. Identification of ribosome binding sites in Escherichia coli using neural network models. *Nucleic Acids Research.* 1995, 23, 9, s. 1632-1639

# Prílohy

Príloha 1: CD so zdrojovými kódmi (priečinok /*src/*)a dokumentáciou (*/doc/*) aplikácie a výsledkami testov (*/stats/*)