Katedra Informatiky
Fakulta Matematiky, Fyziky a Informatiky
Univerzita Komenského, Bratislava

Pravdepodobnostné modely pre alternatívny zostrih génov

(Diplomová práca)

Martin Králik

Vedúci: Mgr. Tomáš Vinař, PhD

Bratislava, 2011

Katedra Informatiky
Fakulta Matematiky, Fyziky a Informatiky
Univerzita Komenského, Bratislava

Pravdepodobnostné modely pre alternatívny zostrih génov

(Diplomová práca)

Martin Králik

kód: 1b9c1ecb-47f4-4735-b5ec-b59a9c06f908

Študijný program: Informatika

Študijný odbor: 9.2.1 Informatika

Školiace pracovisko: Katedra informatiky

Školiteľ: Mgr. Tomáš Vinař, PhD

Miesto a rok vydania: Bratislava, 2011

Čestne prehlasujem, že som túto diplomovú prácu vypracoval samostatne s použitím citovaných zdrojov.

.....

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Martin Králik
Študijný program: informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: 9.2.1. informatika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: slovenský

Názov: Pravdepodobnostné modely pre alternatívny zostrih génov

Cieľ: Cieľom práce je štúdium vlastností takzvaných zostrihových grafov, ktoré sa používajú na charakterizáciu alternatívneho zostrihu génov. Konkrétnejšie, hlavnou úlohou je vytvoriť jednoduché pravdepodobnostné modely charakterizujúce zostrihové grafy ako náhodné grafy a ich analýza.

Vedúci: Mgr. Tomáš Vinař, PhD.

Dátum zadania: 20.11.2009

Dátum schválenia: 18.02.2011

prof. RNDr. Branislav Rovan, PhD.
garant študijného programu

.....
študent

.....
vedúci

Obsah

Abstrakt

Autor: Martin Králik

Názov práce: Pravdepodobnostné modely pre alternatívny zostrih génov

Typ práce: Diplomová práca

Škola: Univerzita Komenského v Bratislave

Fakulta: Fakulta matematiky, fyziky a informatiky

Katedra: Katedra informatiky

Školiteľ: Mgr. Tomáš Vinař, PhD

Bratislava, 2011

V tejto práci sa venujeme štúdiu pravdepodobnostných modelov alternatívneho zostrihu DNA. Našim cieľom bolo jednak vytvoriť pravdepodobnostné modely popisujúce vlastnosti alternatívneho zostrihu ako náhodných grafov, ako aj navrhnúť metódy, ktorými je možné takéto modely spätne využiť na predikciu alternatívneho zostrihu. V budúcnosti môže naša práca viesť k praktickým metódam na predikciu alternatívneho zostrihu v DNA sekvencii, ako aj k lepšiemu pochopeniu fungovania alternatívneho zostrihu v živých organizmoch.

Kľúčové slová: alternatívny zostrih DNA, hľadanie génov, pravdepodobnostné modelovanie, inferencia

Abstract

Author: Martin Králik

Thesis title: Pravdepodobnostné modely pre alternatívny zostrih génov

Thesis type: Diplomová práca

School: Univerzita Komenského v Bratislave

Faculty: Fakulta matematiky, fyziky a informatiky

Department: Katedra informatiky

Advisor: Mgr. Tomáš Vinař, PhD

Bratislava, 2011

In this thesis, we study probabilistic models of alternative splicing in DNA. Our goal was to create probabilistic models describing alternative splicing in terms of random graphs. Using these probabilistic models, we can design new algorithms for alternative splicing prediction. In future, this work can lead to practical methods for predicting alternative splicing in DNA sequences as well as better understanding of general concepts of alternative splicing.

Keywords: alternative splicing of DNA, gene finding, probabilistic modeling, inference

Kapitola 1

Úvod

V ľudskej bunke prebieha veľa procesov potrebných pre život organizmu. Medzi ne patrí aj výroba proteínov podľa kódujúcich úsekov DNA sekvencie. Sekvencia obsahuje okrem kódujúcich aj nekódujúce úseky a jej rozdeleniu na tieto úseky sa hovorí zostrih. Určenie zostrihu je jedným z problémov, ktorý sa v bioinformatike rieši pomocou štatistických modelov a je už veľmi dobre preskúmaný. Biologickými pozorovaniami sa ukázalo, že toto rozdelenie sekvencie na kódujúce a nekódujúce úseky nie je pre väčšinu génov jednoznačné. Tento jav sa nazýva alternatívny zostrih.

V súčasnosti existuje niekoľko programov, ktoré sa snažia alternatívny zostrih hľadať, no každý z nich má určité obmedzenia a žiaden z nich neposkytuje kompletný model tohto javu. Preto sme sa rozhodli zostrojiť model alternatívneho zostrihu pre úsek celého génu, pomocou ktorého by sme vedeli alternatívny zostrih v DNA sekvencii odhaľovať.

V práci používame grafovú reprezentáciu alternatívneho zostrihu [HAS+02]. Prinášame nový pohľad, pomocou ktorého sa na tieto grafy pozeráme ako na výsledok jednoduchého generatívneho procesu. Vytvárame dva rozdielne pravdepodobnostné modely tohto grafu. Nakoniec navrhujeme algoritmy, ktoré pomocou navrhnutých modelov dokážu vo vstupnej sekvencii odhaliť alternatívny zostrih.

Táto práca je rozdelená do piatich kapitol. V druhej predstavíme biologické procesy, a problémy, ktoré vďaka nim vznikli. Tretia kapitola obsahuje prehľad nástrojov, ktoré sa

v súčastnosti používajú na odhaľovanie zostrihu a alternatívneho zostrihu. Pohľad na alternatívny zostrih ako na graf a stochastické modely tohto grafu sú popísané v štvrtej kapitole. Piata kapitola sa zaoberá trénovaním a porovnaním vybraných vlastností modelu so skutočnými dátami. V šiestej kapitole navrhujeme inferenčné algoritmy pre jednotlivé modely. Sumár našej práce je v siedmej kapitole.

Kapitola 2

Gény a alternatívny zostrih

V tejto kapitole uvedieme základné biologické pojmy a procesy, ktoré motivujú problémy riešené v tejto práci.

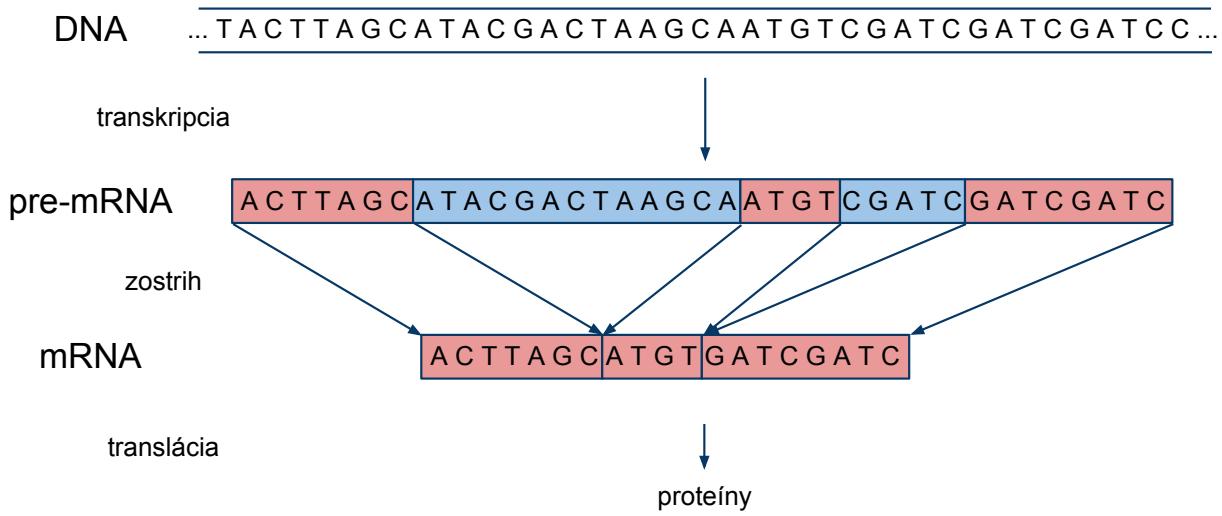
2.1 DNA sekvencia a jej zostrih

DNA obsahuje všetku informáciu potrebnú pre funkčný chod živého organizmu. Je to dvojzávitnica zložená z dvoch dlhých polymérov, ktoré sú pospájané navzájom komplementárnymi dusíkatými bázami. Tieto bázy sa nazývajú adenín, cytozín, guanín a tymín. Ich postupnosťou je kódovaná genetická informácia.

Pred tým, ako sa DNA interpretuje, si bunka najprv danú časť sekvencie skopíruje v procese nazvanom transkripcia. Tento proces nepozostáva len z priamočiareho vytvorenia kópie, pretože niektoré časti DNA nič nekódujú. DNA sa dá rozdeliť na regióny troch typov – exóny, intróny a medzigénové oblasti.

Medzigénové oblasti, už ako ich názov napovedá, nekódujú proteíny. Vieme ich veľmi dobre odlíšiť od zvyšku sekvencie, pretože po ich konci vždy nasleduje exón začínajúci špeciálnou trojicou báz, ktorá sa nazýva štart kodón a pred ich začiatkom sa vždy nachádza exón končiaci stop kodónom, čo je znovu konkrétna trojica báz. Medzi týmito dvomi exónmi s význačnými kodónmi sa môže nachádzať ľubovoľne veľa ďalších exónov a intrónov. Hlavný

rozdiel medzi exónmi a intrónmi je v tom, že intróny priamo nekódujú genetickú informáciu. Donedávna sa o intrónoch myslelo, že nemajú žiadnu funkciu, no ukazuje sa, že to nie je pravda a napríklad pomáhajú pri regulácií transkripcie [FF03].



Obr. 2.1: Pri transkripcii sa najskôr časť DNA skopíruje na pre-mRNA. Z tej sa následne vystrihnú intróny, čím vznikne mRNA (mediátorová RNA).

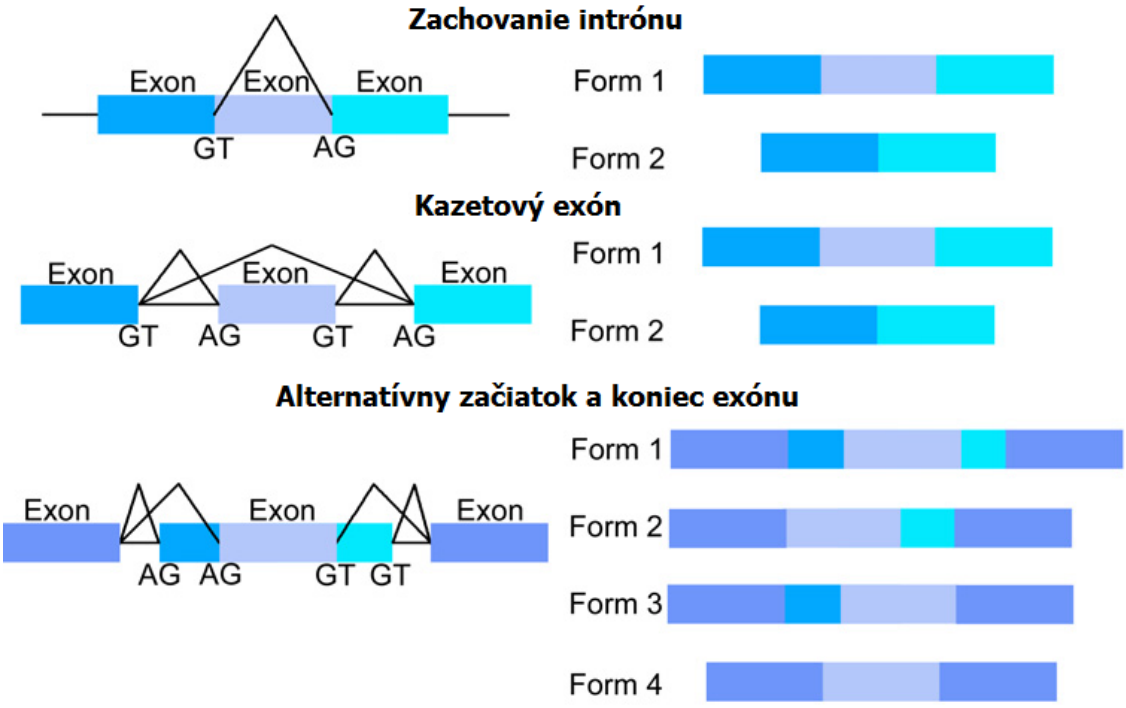
Pri transkripcii DNA sa úseky zodpovedajúce intrónom vystrihnú, tak ako je to naznačené na obrázku 2.1. Rozdelenie DNA na jednotlivé regióny nazývame *zostrih*, hranicu intrónu a exónu *miesto zostrihu*. Tento proces je regulovaný mnohými vplyvmi vnútri bunky, a preto nedokážeme pri pohľade na sekvenciu s úplnou istotou povedať, ktoré úseky budú ako intróny vystrihnuté [CPL97].

2.2 Alternatívny zostrih génov

Reálna situácia je od doposiaľ popísaného procesu zložitejšia. Za určitých podmienok sa zostrih konkrétneho génu môže vykonať iným spôsobom. Tento jav sa nazýva alternatívny zostrih a príčiny jeho vzniku sú rôznorodé. V ľudskom genóme k nemu prichádza až v 74% prípadoch [J$^+$03]. Jednotlivé zostrihy sa nazývajú *transkripty* alebo *izoformy*.

Prejavy alternatívneho zostrihu vieme začleniť do niekoľkých hlavných tried, ktoré sú

zobrazené na obrázku 2.2. Najčastejší je výskyt kazetového exónu, ktorý v ľudskej DNA spôsobuje až 38% všetkého alternatívneho zostrihu. Po ňom nasleduje alternatívny koniec exónu, ktorý má 18% zastúpenie a alternatívny začiatok s 8% zastúpením. Zachovanie intrónu tvorí 3% všetkých výskytov alternatívneho zostrihu. Zvyšných 33% sa nedá kategorizovať do žiadnej z týchto tried[Ast04].



Obr. 2.2: Najčastejšie formy alternatívneho zostrihu. Exóny sú naznačené obdĺžnikmi, intróny úsečkou. Na ľavej strane je vyobrazená DNA sekvencia aj možnosťami zostrihu, na pravej strane sú všetky izoformy, ktoré z nej vieme dostať. Zdroj: [AS06]

2.3 Vzniknuté problémy

Pri sekvenovaní DNA organizmu dostávame sekvenciu bez informácie, ktoré oblasti sú intróny, a ktoré exóny. Informácia o tom, ako sa ktorá časť DNA zostrihne, nám určuje, ktoré proteíny sa vyprodukujú. Taktiež nám môže pomôcť pochopiť vnútorné procesy v bunke,

ktoré ešte stále nie sú úplne zmapované. Preto je pre nás zaujímavým problémom určiť, ako bude zostrih pre daný gén vyzerať. Ešte zaujímavejšia situácia je pri hľadaní alternatívneho zostrihu. Tu sa navyše k dôvodom, prečo hľadať zostrih, pripája aj fakt, že zatiaľ nevieme, ako tento jav vznikol a čo ho spôsobuje [KLMA10]. Nasledujúca kapitola ponúka prehľad metód, ktorými sa tieto problémy v súčasnosti riešia.

Kapitola 3

Hľadanie génov a skryté Markovove modely

V tejto kapitole si najprv bližšie predstavíme skryté Markovove modely, ktoré sa používajú na hľadanie zostrihu. Ďalej ich budeme označovať HMM[1]. Potom priblížime dva programy, ktoré sa snažia odhaľovať alternatívny zostrih a majú v súčastnosti najlepšiu úspešnosť.

3.1 HMM na hľadanie génov

Používaným kritériom na predikciu kódujúcich častí DNA je jej zloženie. Intróny majú trochu inú distribúciu použitých nukleotidov ako exóny, a na ich začiatku (resp. konci) sa väčšinou nachádza dvojica báz GT (resp. AG)[RW80]. Na hľadanie zostrihu pomocou tejto znalosti sa používajú skryté Markovove modely, ktoré si pomocou Markovových reťazcov definujeme v nasledujúcich častiach. Definícia je podľa [Sta03].

3.1.1 Markovove reťazce

Postupnosť náhodných premenných $X_1, X_2, \ldots$ nadobúdajúcich hodnoty zo spočítateľnej množiny Q, sa nazýva Markovov reťazec rádu $k \geq 1$, ak pre všetky $i > k$ a všetky $x_1, x_2, \ldots, x_i \in Q$

[1] z anglického názvu hidden Markov Model

platí

$$P(X_i = x_i \mid X_1 = x_1, \ldots, X_{i-1} = x_{i-1}) = P(X_i = x_i \mid X_{i-k} = x_{i-k}, \ldots, X_{i-1} = x_{i-1})$$

Táto postupnosť sa nazýva homogénny Markovov reťazec, ak $P(X_i = x_{k+1} \mid X_{i-k} = x_i, \ldots, X_{i-1} = x_k)$ nezávisí od i $(x_1, \ldots, x_{k+1} \in Q)$. V opačnom prípade je nehomogénna.

Pre homogénny Markovov reťazec rádu 1 sa matica $A = (a_{r,s})_{r,s \in Q}$ s hodnotami $a_{r,s} = P(X_i = s \mid X_{i-1} = r)$ volá matica prechodov. Množina Q sa nazýva stavový priestor. Ak $X_i = q$, tak sa hovorí, že proces je v čase i v stave q.

K úplnému definovaniu distribúcie Markovovho reťazca nám ešte chýba distribúcia prvej premennej v postupnosti – X_1. Zavedieme si nový štartovací stav q_{init} a ďalšiu konštantnú náhodnú premennú $X_0 \equiv q_{init}$. Potom bude distribúcia Markovovho reťazca plne určená prechodovou maticou. Keďže v reálnom svete nepracujeme s nekonečnými sekvenciami, tak pridáme aj špeciálny ukončovací stav q_{term}, z ktorého sa už nedá odísť. Nech

$$Q^+ = Q \cup \{q_{init}, q_{term}\} \tag{3.1}$$

Rozšírená matica prechodov $A = (a_{r,s})_{r,s \in Q^+}$ musí potom spĺňať

$$a_{q,q_{init}} = 0 \text{ (pre } \forall q \in Q^+) \tag{3.2}$$

$$a_{q_{init},q_{term}} = 0 \tag{3.3}$$

$$a_{q,q_{term}} > 0 \text{ (pre aspoň jedno } q \in Q) \tag{3.4}$$

$$a_{q_{term},q_{term}} = 1 \tag{3.5}$$

3.1.2 Skryté Markovove modely

Majme priestor stavov Q^+ a maticu prechodov A, definovanú rovnako ako v predchádzajúcej sekcii. Nech Σ je spočítateľná množina, nazývaná emisná abeceda. Ďalej, nech sú definované pravdepodobnosti $e_i(\sigma)$ pre $i \in Q^+$ a $\sigma \in \Sigma \cup \varepsilon$. Skrytý Markovov model so stavmi Q^+, maticou prechodov A a emisnými pravdepodobnosťami $e_i(\sigma)$ je postupnosť

$$(X_0, Y_0), (X_1, Y_1), (X_2, Y_2), \ldots$$

kde $X_0 \equiv q_{init}$, postupnosť $X_0, X_1, X_2, \ldots$ je homogénny Markovov reťazec so stavmi Q^+ a maticou prechodov A, a kde $Y_0, Y_1, \ldots$ je postupnosť náhodných premenných z $\Sigma \cup \varepsilon$ takých, že platí $Y_0 \equiv \varepsilon$ a

$$e_{x_i}(y_i) = P(Y_i = y_i \mid X_i = x_i)$$

pre všetky $i > 0$, pre všetky $x_i \in Q^+, y_i \in \Sigma \cup \varepsilon$. Od počiatočného a konečného stavu vyžadujeme, aby neemitovali žiadne písmeno. Zároveň ε nemôže emitovať žiaden iný stav:

$$e_s(\varepsilon) = 0 \text{ (pre } \forall s \in Q)$$
$$e_{q_{init}}(\varepsilon) = 1, e_{q_{term}}(\varepsilon) = 1$$

Označme si $\mathbf{X}$ postupnosť stavov $X_0, X_1, \ldots$ a $\mathbf{Y}$ postupnosť pozorovaní $Y_0, Y_1, \ldots$. Nech $x_1, \ldots, x_n \in Q$ a $d_1, \ldots, d_n \geq 1$. Potom vektor

$$\varphi = (x_1, \ldots, x_n) \tag{3.6}$$

budeme nazývať rozbor končiaci v x_n a pozostávajúci z n krokov. Rozbor $\varphi(\mathbf{X}, \mathbf{Y})$ indukovaný dvojicou ($\mathbf{X}$, $\mathbf{Y}$) je definovaný ako

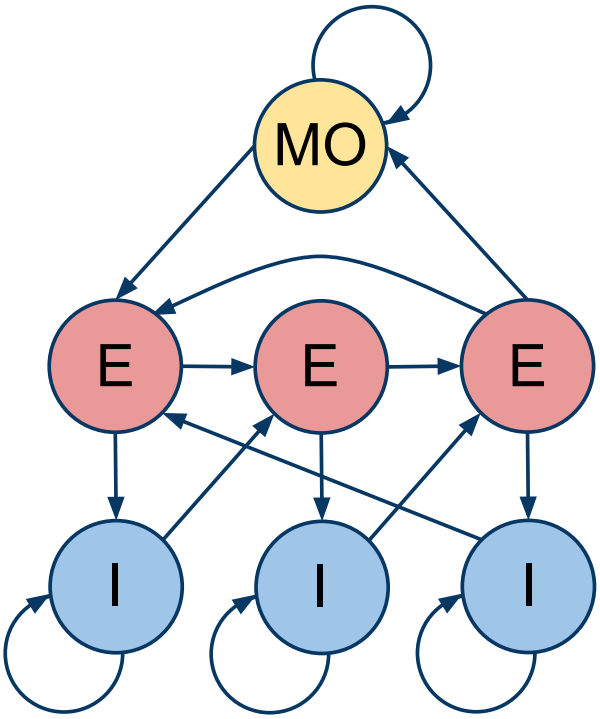
$$\varphi(\mathbf{X}, \mathbf{Y}) := (X_1, \ldots, X_n) \tag{3.7}$$

Označme $\sigma(\mathbf{Y})$ reťazec, ktorý dostaneme spojením reťazcov $Y_0, Y_1, \ldots$.

Konečne sa dostávame k tomu, ako nám tieto modely dokážu pomôcť. $Y_1, \ldots, Y_n$ sa nazývajú emisie, $\sigma(\mathbf{Y})$ je pozorovateľné. To, čo nepoznáme je v ktorom stave bolo ktoré písmeno z $\sigma(\mathbf{Y})$ emitované. V tomto prípade je skrytý rozbor $\varphi(\mathbf{X}, \mathbf{Y})$ a pomocou HMM a $\sigma(\mathbf{Y})$ sa ho snažíme uhádnuť.

V našom prípade potrebujeme označiť časti sekvencie pomocou HMM ako intróny, exóny a medzigénové úseky. Týmto značkám by sme síce mohli priradiť stavy HMM jedna k jednej, ale potom by sme nedokázali obsiahnuť informáciu, že napríklad na začiatku intrónu sú iné pravdepodobnosti výskytu nukleotidov ako na jeho konci.

Jednoduchý model, ktorý by mohol generovať reťazce podobajúce sa na DNA by mohol vyzerať tak ako na obrázku 3.1. Jednotlivé stavy v ňom reprezentujú aká časť DNA sa práve



Obr. 3.1: Jednoduchá schéma HMM pre odhaľovanie zostrihu. Nie je na nej zaznačený špeciálny počiatočný ani konečný stav. Ku kompletnému HMM ešte chýba matica prechodov a emisné pravdepodobnosti pre stavy.

generuje – medzigénová oblasť, intrón alebo exón. Obsahuje tri stavy pre exóny a intróny, pretože aminokyseliny sú kódované pomocou trojíc nukleotidov (kodónov). Keď v strede exónu začne intrón, tak si pomocou stavu zapamätáme, ktorý nukleotid v poradí sme práve vygenerovali. Vďaka tomu budeme po skončení intrónovéhu úseku vedieť, kde v ktorej fáze sme prestali. Medzigénové úseky nemôžu predeliť kodón na dve časti, a preto sa do stavu pre medzigénový úsek dá dostať len po poslednom treťom nukleotide kodónu.

3.2 Viterbiho algoritmus

S pomocou skrytého Markovovho modelu dokážeme k jednotlivým znakom reťazov určiť najpravdepodobnejší stav, v ktorom by sa pri ich generovaní nachádzal. Slovo „skrytý“ sa používa preto, že nevidíme stavy ktorými model prešiel počas generovania, vidíme len výsledok. Čím lepšie model modeluje DNA sekvencie, tým je vyššia šanca, že nájdený zostrih sa podobá reálnemu. Na nájdenie tejto najpravdepodobnejšej postupnosti stavov, ktorou prešiel skrytý Markovov model, sa používa Viterbiho algoritmus. Túto postupnosť nazveme

Viterbiho cesta.

Najpravdepodobnejšiu postupnosť stavov si môžeme označiť ako

$$\pi* = \underset{\pi}{\operatorname{argmax}} P(x, \pi)$$

kde P je pravdepodobnosť vygenerovania cesty π a reťazca x. Nech $v_s(i)$ je pravdepodobnosť najpravdepodobnejšej cesty končiacej v stave s generujúcej prvých i znakov z x. Túto hodnotu vieme rekurzívne vyrátať ako

$$v_s(i+1) = e_s(x_{i+1}) \max_t(v_t(i)a_t(s))$$

kde x_{i+1} je $(i+1)$-vý znak reťazca x. Začiatočná pravdepodobnosť $v_s(0)$ bude rovná 1 pre počiatočný stav a pre ostatné s rovná 0. Našim cieľom je zistiť hodnotu $\max_t v_t(|x|)$. Okrem vyrátania pravdepodobnosti najpravdepodobnejšej cesty nás zaujíma aj samotná cesta. Preto počas rátania si budeme pre každé $i+1$ pamätať, ktorý stav generujúci x_i sme pri výbere maxima použili. Na koniec pomocou tejto informácie vieme zrekonštruovať hľadanú Viterbiho cestu.

Celý tento postup sa dá pomocou paradigmy dynamického programovania naprogramovať nerekurzívne. Najprv sa vyriešia menšie podproblémy, a potom pomocou nich väčšie. Ak si označíme dĺžku sekvencie N a počet stavov modelu M, tak časová zložitosť tohoto algoritmu je $O(NM^2)$.

3.3 Forward a backward algoritmus

Ďalšou možnosťou ako hľadať zostrih, je zisťovať pre jednotlivé pozície sekvencie najpravdepodobnejší stav modelu, ktorý ich mohol vygenerovať. Pre pozíciu i a stav k je táto pravdepodobnosť rovná

$$P(\pi_i = k \mid x) = \frac{P(\pi_i = k, x)}{P(x)} \tag{3.8}$$

podľa vzorca pre podmienenú pravdepodobnosť. Pravdepodobnosť reťazca, pričom i-ty znak bol vygenerovaný stavom k, môžme ďalej rozpísať ako

$$P(\pi_i = k, x) = P(x_1 \dots x_i, \pi_i = k) P(x_{i+1} \dots x_n \mid x_1 \dots x_i, \pi_i = k)$$

Pravdepodobnosť vygenerovania $i+1$ znaku (a všetkých ďalších) závisí len od stavu, v ktorom model generoval i-ty znak. Preto môžeme písať

$$P(x_1 \dots x_i, \pi_i = k)P(x_{i+1} \dots x_n \mid x_1 \dots x_i, \pi_i = k) = P(x_1 \dots x_i, \pi_i = k)P(x_{i+1} \dots x_n \mid \pi_i = k)$$

Označme si $P(x_1 \dots x_i, \pi_i = k)$ ako $f_k(i)$. Túto hodnotu vieme rátať rekurzívne:

$$f_k(i+1) = e_k(x_{i+1}) \sum_l f_l(i) a_l(k)$$

Algoritmus, ktorý by túto hodnotu počítal, by vyzeral podobne ako Viterbiho. Hlavný rozdiel je v tom, že miesto maxima sa berie súčet, pretože nás zaujíma celková pravdepodobnosť všetkých ciest končiacich v stave k. Jeho časová zložitosť by bola rovnaká, $O(NM^2)$, kde N je dĺžka časti sekvencie, ktorá nás zaujíma, a M je počet stavov modelu. Tento algoritmus sa nazýva forward algoritmus.

$P(x_{i+1} \dots x_n \mid \pi_i = k)$ si označíme ako $b_k(i)$. Túto hodnotu vieme rátať analogicky ako pri forward algoritme, s tým rozdielom, že rekurzia by išla v opačnom smere. Nech špeciálny ukončovací stav má číslo 0, potom by algoritmus vyzeral takto:

inicializácia

$b_k(n) = a_k(0)$ pre všetky k

rekurzia

$b_k(i) = \sum_l b_l(i+1) e_k(x_{i+1}) a_k(l)$ pre $i < n$

Túto variantu nazývame backward algoritmus.

Pre vyrátanie (3.8) nám chýba už len poznať pravdepodobnosť celej sekvencie, $P(x)$. Tú môžme vyrátať forward algoritmom tak, že ho necháme prejsť komplet celú sekvenciu:

$$P(x) = \sum_s f_s(n) a_k(0)$$

Časová zložitosť tohto algoritmu ju rovnaká ako pri Viterbiho – $O(NM^2)$.

Tento postup pomocou maximalizácie počtu správnych pozícií sa oplatí použiť v tom prípade, keď má viac ciest podobnú pravdepodobnosť ako najpravdepodobnejšia, pretože

pri rátaní len s Viterbiho by sme prišli o významné informácie. Nevýhodou tohto postupu je fakt, že takto nájdená cesta modelom vôbec nemusí existovať – môžu sa v nej nachádzať také prechody medzi stavmi, ktoré v modeli nie sú, pretože stavy pre jednotlivé pozície sa vyberajú nezávisle na sebe.

3.4 Trénovanie HMM

Ako už bolo spomenuté, náš model by mal čo najpresnejšie simulovať realitu, aby boli výsledky relevantné. S tým sa viaže problém ako zvoliť pravdepodobnosti v stavoch a hranách.

Existuje viacero metód, ako trénovať tieto parametre. V každom prípade potrebujeme nejaký počet vstupných sekvencií. Priaznivý prípad je ten, keď k nim vieme aj postupnosť stavov, teda rozbor. Potom stačí zrátať, ako často sa vo všetkých emisiách zo stavu k objavil ten ktorý znak:

$$e_k(b) = \frac{E_k(b)}{\sum_{b'} E_k(b')}, \tag{3.9}$$

kde $E_k(b)$ označuje počet emisií znaku b v stave k na trénovacích dátach. A ak označíme počet prechodov zo stavu k do stavu l ako A_{kl}, tak podobným spôsobom viem vyrátať aj maticu prechodov:

$$a_{kl} = \frac{A_{kl}}{\sum_{l'} A_{kl'}}. \tag{3.10}$$

Problém môže nastať, ak dát máme príliš málo, pretože sa model môže preučiť. Keby sme napríklad nemali žiadnu informáciu o prechodoch z nejakého vrcholu k, tak o a_{kl} pre ľubovolné l nevieme nič povedať. Vo vzorci by sme totiž delili nulou a výsledok by nebol definovaný. Tomuto stavu môžeme predísť pričítaním dopredu zvolených pseudopočtov ku každej premennej. Ich voľba by mala odrážať naše znalosti o výsledných pravdepodobnostiach.

Druhý prípad je ten, kedy nepoznáme rozbor. Štandardne sa v tomto prípade používa Baum-Welchov algoritmus. Zjednodušene funguje tak, že najskôr odhadne A_{kl} a $E_k(b)$ pomocou pravdepodobných ciest na dátach s použitím nejakých a_{kl} a $e_k(b)$. Potom z (3.10) a (3.9) vyráta nové hodnoty a_{kl} a $e_k(b)$. Tento proces sa iteratívne opakuje dokým sa nedosiahne

ukončovacie kritérium.

A_{kl} a $E_k(b)$ sú odhadované ako očakávaný počet použití daného prechodu, respektíve emisie, z tréningových dát. Používa sa na to veľmi podobný prístup ako pri forward-backward algoritme. Pravdepodobnosť, že prechod a_{kl} je použitý na pozícii i v sekvencii x je

$$P(\pi_i = k, \pi_{i+1} = l \mid x, \theta) = \frac{f_k(i)a_{kl}e_l(x_{i+1})b_l(i+1)}{P(x)}, \tag{3.11}$$

kde θ sú všetky momentálne parametre modelu. Pomocou zosumovania tohto vzorca cez všetky pozície a trénovacie sekvencie dokážeme vytátať očakávaný počet použití prechodu a_{kl}

$$A_{kl} = \sum_j \frac{1}{P(x^j)} \sum_i f_k^j(i)a_{kl}e_l(x_{i+1}^j)b_l^j(i+1) \tag{3.12}$$

pričom horný index j označuje, že sa jedná o dáta pre j-tu sekvenciu. f a b sú premenné z forward-backward algoritmu.

Podobným spôsobom vieme nájsť očakávaný počet emitovania b v stave k:

$$E_k(b) = \sum_j \frac{1}{P(x^j)} \sum_{i|x_i^j=b} f_k^j(i)b_k^j(i) \tag{3.13}$$

Vnútorná suma ide len cez tie pozície, kde je b emitované.

Na začiatku zvolíme pre model ľubovoľné parametre a A_{kl} a $E_k(b)$ nastavíme na pseudopočty. Ďalej iterujeme už popísaný postup, dokým sa nám súčet logaritmov pravdepodobností nášho modelu $\sum_{j=1}^n \log P(x^j \mid \theta)$ nezanedbateľne zvyšuje. Tento proces postupne konverguje do lokálneho maxima. Ktoré to bude závisí od počiatočných parametrov.

Používanou alternatívou k Baum-Welch algoritmu je Viterbiho trénovanie[RT03]. Pri ňom sa pomocou Viterbiho algoritmu nájdu rozbory pre trénovacie dáta a spraví sa nový odhad $e_k(b)$ a a_{kl} pomocou algoritmu pre sekvencie so známym rozborom. Tento proces pokračuje iteratívne ďalej ako predchádzajúci algoritmus. Prvotné rozbory sa nastavia ako rovnomerné rozdelenie sekvencie.

3.5 Hľadanie alternatívneho zostrihu pomocou vzorkovania

Jedna z možností, ako hľadať alternatívne zostrihy, je náhodné vzorkovanie (sampling) pomocou skrytého Markovového modelu. Takýto prístup využíva napríklad program AUGUSTUS [SKG+06].

Vzorkovanie je proces, pri ktorom sa z množiny všetkých možných stavov náhodne vyberie dostatočne veľká podmnožina stavov, pomocou ktorej sa usudzuje o celej množine. Pre hodnovernosť úsudkov sa stavy musia z množiny všetkých stavov vyberať tak, aby pravdepodobnosť ich vybratia zodpovedala ich distribúcii.

AUGUSTUS týmto spôsobom podľa distribúcie danej HMM vyberie n vzoriek. Potom sa odhadnú aposteriórne pravdepodobnosti pre jednotlivé exóny, intróny a cesty pomocou ich relatívnej frekvencie ich výskytu vo vzorkách. Nakoniec sa odhodia tie cesty, v ktorých

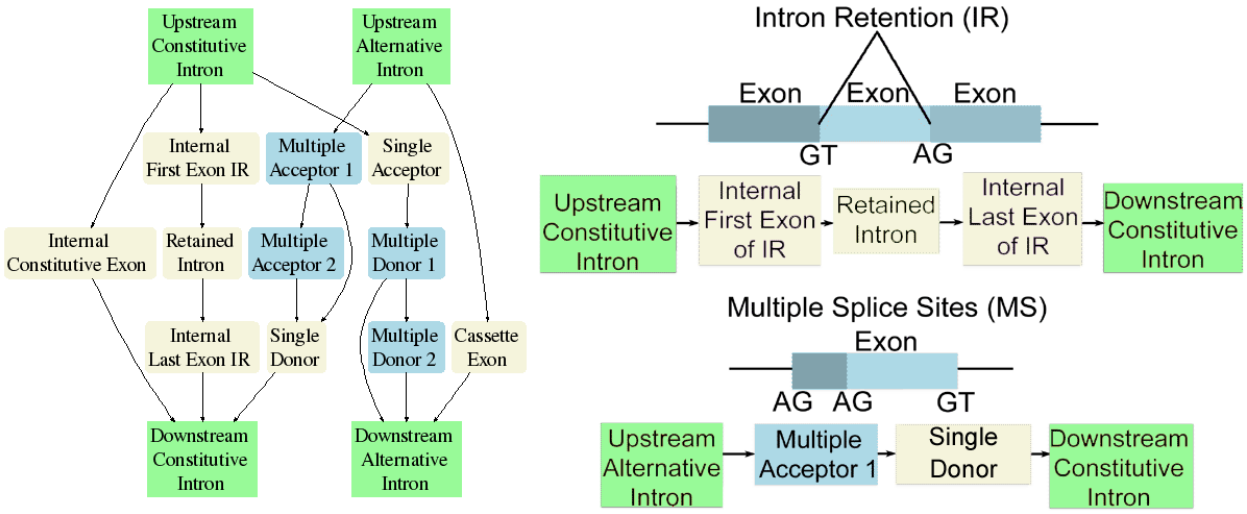
- ľubovoľný intrón alebo exón má nižšiu aposteriórnu pravdepodobnosť ako daná konštanta
- alebo geometrický priemer pravdepodobností všetkých intrónov a exónov je nižší ako daná (iná) konštanta

A navyše ak sa viac ako $T_{\max}$ ciest prekrýva na rovnakom mieste, tak sa usporiadajú podľa priemernej pravdepodobnosti intrónov a exónov a ponechá sa len $T_{\max}$ najlepších z nich. Cesty ktoré zostali predstavujú alternatívne zostrihy.

Najväčšou výhodou AUGUSTUSa je jeho rýchlosť, ktorá vyplýva z nenáročnosti vzorkovania. Na druhej strane, AUGUSTUS neobsahuje žiaden model alternatívneho zostrihu. Snaží sa ho odhaliť len pomocou vzorkovania z modelu pre normálny zostrih, čo nie je biologicky vierohodné. To by totiž v konečnom dôsledku znamenalo, že tento proces je spôsobovaný chybami pri vykonávaní zostrihu, čo nie je pravda.

3.6 Lokálny model alternatívneho zostrihu

Iný prístup na odhaľovanie alternatívneho zostrih používa program ExAlt [AS06]. Nesnaží sa hľadať zostrih pre ľubovoľnú vstupnú DNA, ale špecializuje sa len na detekciu niekoľkých prejavov alternatívneho zostrihu v krátkych úsekoch DNA.



Obr. 3.2: Naľavo je vyobrazená časť stavov HMM používaného ExAltom pre hľadanie alternatívneho zostrih. Napravo sú dva ukážkové prechody modelom aj s kusom alternatívneho zostrihu, ku ktorému daný prechod zodpovedá.

Pre nás relevantnou časťou ExAltu je skrytý Markovov model, ktorého stavy reprezentujú všetky typy sekvencie, ktoré môžu nastať pri alternatívnom zostrihu. To znamená, že jeho stavy nereprezentujú priamo exón a intrón, ale napríklad vypustenie intrónu alebo začiatok exónu, ktorý je v niektorých izoformách vystrihnutý, Konktétne tieto dva typy sekvencie sú spolu s časťou použitého HMM vyobrazené na obrázku 3.2.

Najväčšou nevýhodou ExAltu je jeho zameranie len na krátku časť sekvencie. Neobsahuje model pre celý gén, ale len pre krátky úsek, na ktorom dokáže detegovať pár prejavov alternatívneho zostrihu. Na druhú stranu sa tento program aj vďaka použitiu ďalších konceptov z komparatívnej genomiky, ktoré pre našu prácu nie sú zaujímavé, radí medzi najúspešnejšie hľadače alternatívneho zostrihu. Tento úspech môže naznačovať, že detailné rozlišovanie medzi jednotlivými časťami sekvencie je vhodná cesta pre odhaľovanie alternatívneho zostrihu.

Kapitola 4

Pravdepodobnostné modely alternatívneho zostrihu

Doterajšie prístupy k detekcii alternatívneho zostrihu majú svoje nedostatky, ktoré sme spomenuli v predchádzajúcej kapitole. Do dnešného dňa nám nie je známa existencia algoritmu, ktorý by pomocou modelu alternatívneho zostrihu vedel alternatívny zostrih hľadať v DNA sekvencii pre celý gén. Preto sme sa rozhodli taký model aj s algoritmom zostrojiť. Celý tento postup je vysvetlený v dvoch kapitolách. V tejto kapitole vytvoríme dva rozdielne generatívne modely množiny transkriptov. V šiestej kapitole predstavíme algoritmus, ktorý s ich pomocou pre DNA sekvenciu bude vedieť nájsť najpravdepodobnejšiu množinu alternatívnych zostrihov.

4.1 Zostrihový graf

Náš prístup spočíva v pohľade na množinu transkriptov alternatívneho zostrihu ako na graf. Vrcholy budú reprezentovať časti exónov a sú prepojené orientovanými hranami tak, aby každá cesta grafom zodpovedala jednej izoforme.

Grafu pre množinu transkriptov vytvárame podľa nasledovného postupu:

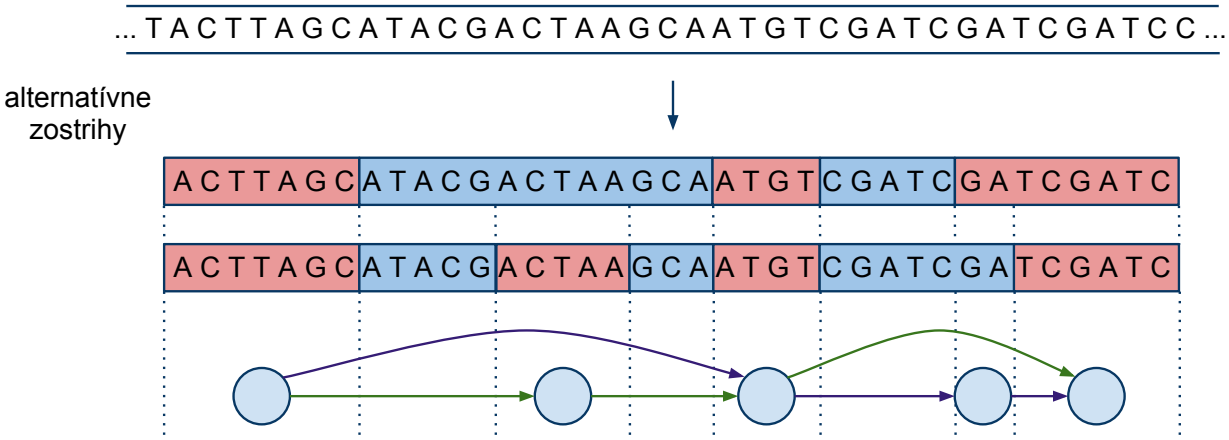
1. Sekvenciu si rozdelíme na menšie časti. Hranice častí budú tam, kde je aspoň v jednej

izoforme hranica medzi intrónom a exónom.

2. Z každej časti, ktorá je aspoň v jednej izoforme exónom, spravíme vrchol.

3. Hranu pridáme z vrcholu A do vrcholu B, pokiaľ existuje taká izoforma zostrihu, v ktorej sú časti sekvencie zodpovedajúce A a B exóny, medzi ktorými neexistuje iná časť sekvencie, ktorá by v danej izoforme bola tiež exónom.

Tento postup je ilustrovaný na obrázku 4.1.



Obr. 4.1: Vytvorenie grafu alternatívneho zostrihu pre sekvenciu s dvoma transkriptami.

Naša grafová reprezentácia má niekoľko nevýhod:

- Neuchováva úplnú informáciu o jednotlivých transkriptoch. Z grafu sa nedajú spätne získať transkripty z ktorých vznikol, iba ich nadmnožina.

- Neobsahuje všetku informáciu o vzájomnej polohe exónov. Dva vrcholy spojené hranou môžu znamenať jeden dlhý exón, ktorý má v jednom transkripte skorší koniec, ale aj dva vzdialené úplne nezávislé exóny.

- Nevyužíva žiadnu informáciu o zložení sekvencie. DNA sekvencia môže svojím zložením napovedať o alternatívnych zostrihoch, pretože biologické procesy v bunke, ktoré zostrih vykonávajú, sú regulované aj samotným zložením sekvencie [WB08].

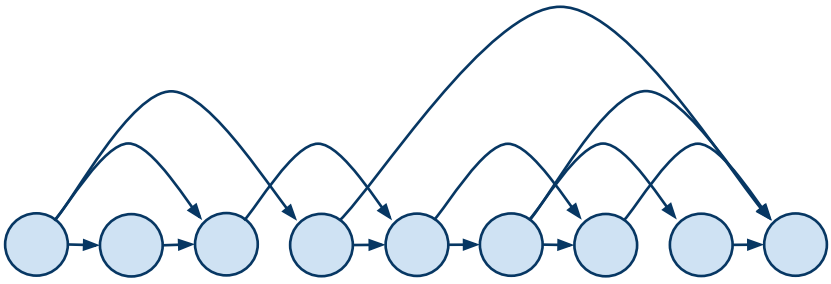
4.2 Zostrihový graf ako pravdepodobnostný model

Na zostrihový graf sa v ďalších častiach budeme pozerať ako na náhodný graf, ktorý vznikol jednoduchým stochastickým procesom. Navrhneme dva generatívne modely, ktorých cieľom bude produkovať grafy alternatívneho zostrihu. Na tieto modely máme dve požiadavky. Prvou je ich jednoduchosť. K modelom chceme vytvoriť inferenčný algoritmus, pomocou ktorého budeme vedieť predpovedať zostrihy vstupnej sekvencie. Prílišná komplikovanosť modelu by mohla viesť k tomu, že tento algoritmus bude mať horšiu zložitosť ako polynomiálnu, čo je pri dĺžkach génov nevyhovujúce. Druhá požiadavka je malý počet parametrov, ktoré by sa dali jednoducho natrénovať pomocou dostupných dát.

4.3 Model so závislosťou na vzdialenosti

Ako prvý sme si zvolili veľmi jednoduchý model, ktorý popisuje grafy generované nasledovným postupom:

1. Zvolíme počet vrcholov V, ktoré uložíme vedľa seba na priamku.
2. Medzi každými dvoma vrcholmi môže existovať hrana. Jej prítomnosť závisí len od vzdialenosti vrcholov a ničoho iného.



Obr. 4.2: Príklad grafu vygenerovaného podľa modelu.

Jeden zástupca tejto triedy je vyobrazený na obrázku 4.2.

Tento model sme pomenovali model so závislosťou na vzdialenosti práve kvôli tomu, že existencia hrán závisí len od vzájomnej vzdialenosti vrcholov. Ďalej ho budeme označovať ako MZV.

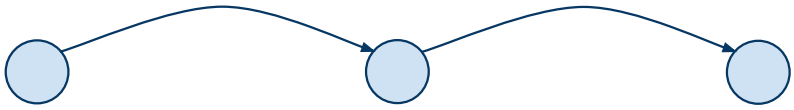
Model je určený dvomi funkciami $P_V(n)$ a $P_H(x)$. Funkcia $P_V(n)$ definuje pravdepodobnostné rozdelenie počtu vrcholov daného grafu. Funkcia $P_H(x)$ definuje pravdepodobnosť s akou sa vyskytne hrana medzi vrcholmi vzdialenými x, pričom vzdialenosť dvoch vrcholov je definovaná ako jedna plus počet vrcholov medzi nimi.

4.4 Biologicky motivovaný model

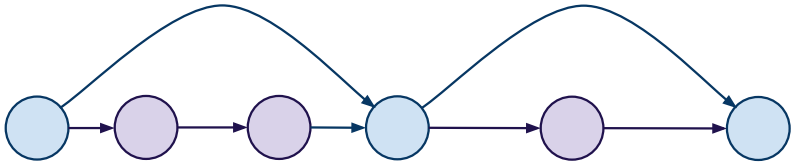
Druhý model sme zvolili s cieľom priblížiť sa skutočným zmenám transkriptov, ktoré vznikajú pri alternatívnom zostrihu. Preto sme tento model nazvali biologicky motivovaným modelom. Ďalej ho budeme uvádzať už len pod skratkou BMM.

Najčastejšie varianty alternatívneho zostrihu sú alternatívny začiatok a koniec exónu, a úplné vypustenie exónu [MPNG08], ktoré sme sa pokúsili zachytiť v rámci jednotlivých krokov generatívneho procesu. Triedu grafov definovaných týmto modelom popisuje nasledujúci postup:

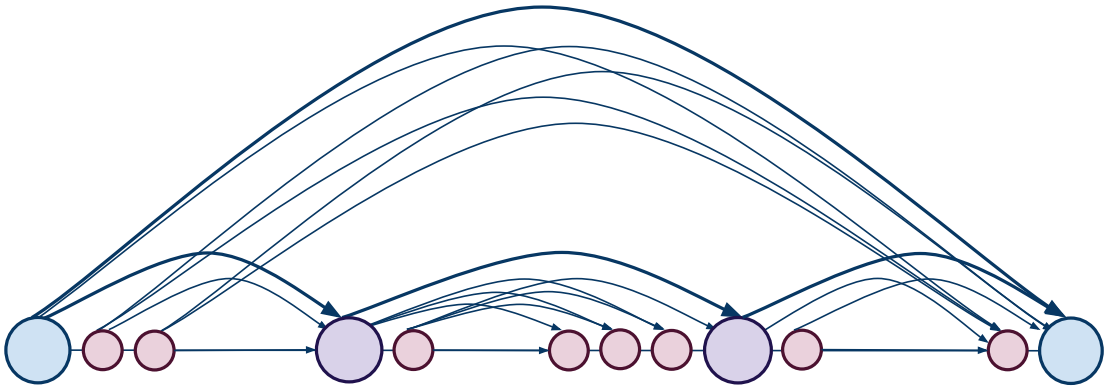
1. Zvolíme počet vrcholov, ktoré uložíme vedľa seba na priamku. Susedné vrcholy spojíme hranou. Tieto vrcholy budú predstavovať exóny, ktoré sú prítomné v každej izoforme. Budeme ich nazývať *stále* vrcholy.



2. Pre každú dvojicu susedných vrcholov zvolíme počet vrcholov, ktoré medzi ne vložím. Tie budú predstavovať exóny, ktorá sa nenachádzajú v každej izoforme. Všetky tieto vložené vrcholy budú ležať na jednej ceste. Budeme ich nazývať *nestále* vrcholy. Môže ich byť aj nula.



3. Na koniec pridáme vrcholy, ktoré budú reprezentovať alternatívne začiatky a konce exónov. Budeme ich vkladať z ľavej a pravej strany už prítomných vrcholov. Ak hrana pred týmto krokom viedla z vrcholu A do vrcholu B, tak nové hrany budú viesť aj zo všetkých koncov A do všetkých začiatkov B. Tieto vrcholy budeme nazývať *okrajové* vrcholy.



Všetky *nestále* a k nim vložené *okrajové* vrcholy medzi dvomi susednými *stálymi* vrcholmi budeme nazývať *skok*.

Tento model je definovaný tromi funkciami:

- distribúcia počtu *stálych* vrcholov pridaných v prvom kroku $P_{V1}(x)$
- distribúcia počtu vkladaných *nestálych* vrcholov v druhom kroku $P_{V2}(x)$
- distribúcia počtu *okrajových* vrcholy vkladaných v treťom kroku $P_{V3}(x)$

Kapitola 5

Trénovanie parametrov

V tejto kapitole sa venujeme natrénovaniu konkrétnych parametrov MZV a následnému porovnaniu vybraných atribútov takéhoto natrénovaného modelu so skutočnými dátami. Na základe výsledkov tejto kapitoly sme sa potom rozhodli vytvoriť model BMM. Koncepty trénovania sú aplikovateľné aj naň, aj keď sa mu v tejto kapitole špecificky nevenujeme.

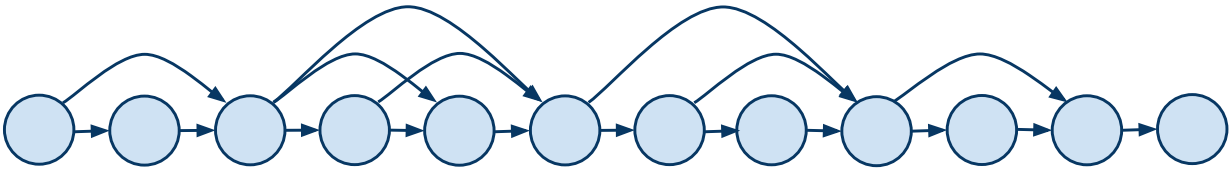
5.1 Zdroj a spracovanie trénovacích dát

Dáta sme získali z databázy UCSC[1], ich zber a kvalitu zaručuje Genome Reference Consortium. Použili sme transkripty ľudských génov zo zostavenia DNA sekvencie verzie „Feb. 2009 (GRCh37/hg19)“ [FRZ+10]. Táto databáza obsahovala 36366 transkriptov, z ktorých sme vytvorili grafy alternatívneho zostrihu pre 22660 génov. Základné štatistické spracovanie týchto dát je v tabuľke 5.1. Na obrázkoch 5.1 a 5.2 sú vyobrazené dva ukážkové grafy z tohto datasetu.

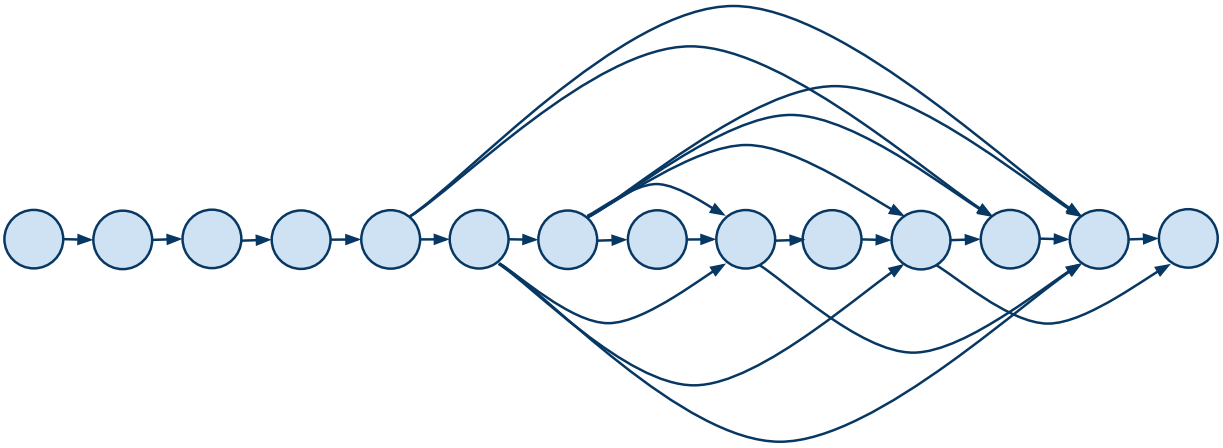
5.2 Parametre pre MZV

MZV určujú dve funkcie – $P_V(n)$, čo je distribúcia počtu vrcholov, a $P_H(x)$, čo je pravdepodobnosť spojenia dvoch vrcholov hranou vo vzdialenosti x. Reálna distribúcia počtu vrcholov

[1] University of California Santa Cruz



Obr. 5.1: Graf pre alternatívny zostrih génu nachádzajúceho sa v ľudskej DNA na pozícii chr1:158323485–158326553. Alternatívny zostrih pre tento gén obsahuje aj kazetový exón aj alternatívny koniec exónu. Takýto graf vieme vytvoriť aj vrámci BMM.



Obr. 5.2: Graf pre alternatívny zostrih génu nachádzajúceho sa v ľudskej DNA na pozícii chr14:94843084–94857029. Táto zložitá štruktúra vznikla najmä vďaka rôznym alternatívnym koncom a začiatkom a exónov, pričom sa nevyskytli všetky možné kombinácie koncov a začiatkov. Tento graf je zároveň príkladom grafu, ktorý BMM nepokrýva, pretože je až príliš komplikovaný.

atribút	minimum	maximum	priemer ± štand. odchýlka	medián
počet vrcholov	1	316	9.85 ± 10.018	7
dĺžka grafu	1	314	10.07 ± 10.146	7
vstupný stupeň vrchola	0	24	0.94 ± 0.417	1
výstupný stupeň vrchola	0	16	0.94 ± 0.415	1
celkový stupeň vrchola	0	25	1.88 ± 0.59	2

Tabuľka 5.1: Dĺžku grafu sme zistili ako počet vrcholov na najdlhšej ceste medzi prvým vrcholom (podľa sekvencie) a posledným. Túto informáciu vieme vyrátať pomocou Bellman-Fordovho algoritmu [CLRS01], pretože sa jedná o orientované acyklické grafy. Ak je v grafe dĺžka najdlhšej cesty menšia ako počet vrcholov, znamená to prítomnosť dvoch vzájomne výlučných exónov. Takýto prípad nastal pri 3443 grafoch, čo je približne 15% všetkých grafov.

je naznačená na obrázku 5.3.

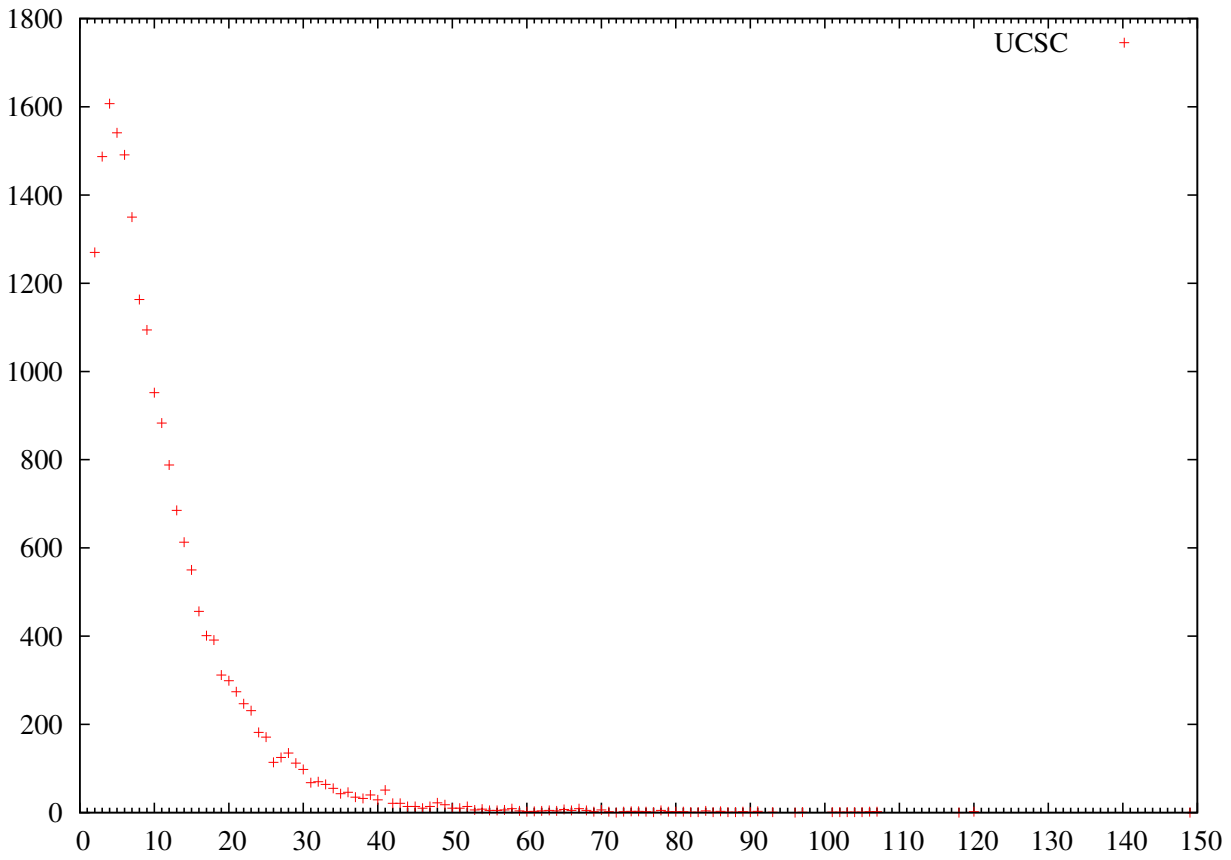
Môžeme ju aproximovať napríklad pomocou geometrického rozdelenia, pričom hodnoty pre jeden a dva vrcholy môžeme napevno určiť. Presné určenie parametrov tohto rozdelenia pre nás nie je zaujímavé, pretože samotné rozdelenie ovplyvňuje len počet vrcholov.

Zaujímavejšie je určenie funkcie P_H, pretože v tomto prípade jej výberom ovplyvňujeme takmer všetky parametre grafu. Ako prvú sme si zvolili jednu z najjednoduchších funkcií, pre ktorú platí, že pravdepodobnosť hrany s rastúcou vzdialenosťou klesá: $P_{H_1}(x) = p^x$, pričom $p \in (0, 1)$. Jej parameter p zistíme princípom maximálnej vierohodnosti.

5.2.1 Princíp maximálnej vierohodnosti

Na natrénovanie parametrov modelu sa používa princíp maximálnej vierohodnosti. Vierohodnosť je pravdepodobnosť pozorovaných výsledkov za zadaných parametrov. Môžme si ju označiť ako $L(\theta \mid X)$[2], kde θ sú parametre modelu a X jednotlivé pozorovania. Jej maximalizáciu dosiahneme takým nastavením parametrov, pri ktorom bude pravdepodobnosť

[2]z anglického likelihood



Obr. 5.3: Počet grafov danej veľkosti v databáze UCSC. Na x-ovej osi je počet vrcholov grafu. Na y-ovej osi je počet grafov.

vygenerovania dát z modelu najvyššia.

Nech X je množina n grafov z databázy, x_i sú jednotlivé grafy, v_i počet vrcholov grafu x_i, E_i je multimnožina tých vzdialeností dvojíc vrcholov, medzi ktorými vedie v grafe x_i hrana a F_i obdobná množina pre vzdialenosti bez hrany.

$$\operatorname*{argmax}_p L(p, P_V \mid X) = \operatorname*{argmax}_p P(X \mid p, P_V) = \operatorname*{argmax}_p \left(\prod_{i=1}^{n} P_{H_1}(v_i) \left(\prod_{e \in E_i} p^e \right) \left(\prod_{f \in F_i} (1 - p^f) \right) \right) =$$

$$\operatorname*{argmax}_p \prod_{i=1}^{n} \left(\prod_{e \in E_i} p^e \prod_{f \in F_i} (1 - p^f) \right)$$

Zjednotenie všetkých vzdialeností dvojíc vrcholov, ktoré spája hrana, cez všetky grafy $\cup_{i=1}^{n} E_i$ si označíme ako E, rovnako si označíme $\cup_{i=1}^{n} E_i$ ako F. Rovnaké p dostaneme aj keď budeme maximalizovať zlogaritmovanú funkciu:

$$\operatorname*{argmax}_p \log \left(\prod_{e \in E} p^e \prod_{f \in F} (1 - p^f) \right) = \operatorname*{argmax}_p \left(\log(p) \sum_{e \in E} e + \sum_{f \in F} \log(1 - p^f) \right)$$

Pre nájdenie maxima funkciu zderivujeme a položíme rovnú nule:

$$\operatorname*{argmax}_p \left(\log(p) \sum_{e \in E} e + \sum_{f \in F} \log(1 - p^f) \right)' = \frac{\sum_{e \in E} e}{p} + \sum_{f \in F} \frac{f p^{f-1}}{p^f - 1} = 0$$

Tento výraz žiaľ nevieme upraviť do uzavretého tvaru pre p. Preto použijeme gradientovú metódu.

5.2.2 Gradientová metóda

Táto numerická metóda spočíva v iteratívnom hľadaní lokálneho minima funkcie pomocou jej derivácie. Nech $p^{(i)}$ je kandidát na lokálne minimum funkcie f v i-tom kroku iterácie. Potom hodnota derivácie funkcie f v bode p' nám povie, ktorým smerom f klesá. Novým kandidátom na minimum bude:

$$p^{(i+1)} = p^{(i)} - \alpha f'(p^{(i)})$$

Konštanta α sa nazýva rýchlosť učenia a je to malé číslo väčšie ako nula. Zabezpečuje nepreskočenie minima, ktoré by mohlo nastať v príliš veľkom skoku v smere klesania.

Nových kandidátov na lokálne minimum vytvárame dokiaľ nenastane ukončovacia podmienka. Vhodnou podmienkou je napríklad poklesnutie rozdielu medzi $p^{(i)}$ a $p^{(i+1)}$ na menej ako vopred vybrané malé nenulové číslo δ. Lokálne minimum, do ktorého sa týmto spôsobom vieme dostať, z veľkej časti závisí od voľby začiatočného bodu $p^{(0)}$.

Pre funkciu $P_{H_1} = p^x$ sme gradientovou metódou sme našli najlepšie $p \cong 0.483658$.

Pre porovnanie sme si zvolili aj druhú funkciu $P_{H_2}(x) = a^{\frac{-1}{x+c}}/b$, ktorú sme vybrali tak, aby čo najlepšie fitovala pravdepodobnosť výskytu hrany pre určenú vzdialenosť. Parametre funkcie P_{H_2} sme z dát určili pomocou Levenberg–Marquardtovho algoritmu [Mar63], implementovaného v programe Gnuplot. Jej výsledná podoba je $P_{H_2}(x) = (1.15512\times10^{-7})^{\frac{-1}{x+0.999999}}/3000$.

5.3 Korešpondencia modelov s dátami

Pre lepší prehľad o vhodnosti MZV sme si vybrali niektoré vhodné štatistiky, ktoré popisujú grafy alternatívneho zostrihu a porovnali ich hodnoty pre MZV a databázu USCS.

Graf 5.4 znázorňuje pravdepodobnosť spojenia dvoch vrcholov hranou v závislosti od ich vzdialenosti. Graf pre MZV s P_{H_2} je bližšie k prevdepodobnostiam pre menšie vzdialenosti hrán, pretože pre ne bolo viacej príkladov v datasete, a každý príklad má pri trénovaní rovnakú váhu. Počet príkladov prudko klesá so zvyšujúcou sa vzdialenosťou, čo je naznačené na obrázku 5.5.

Graf 5.6 znázorňuje priemerný počet hrán v grafe v závislosti od jeho veľkosti. V tomto prípade náš model s oboma funkciami P_H celkom dobre reprezentuje reálne dáta, čo je ale spôsobené tým, že táto vlastnosť je v MZV priamo závislá od pravdepodobnosti pre výskyt hrany.

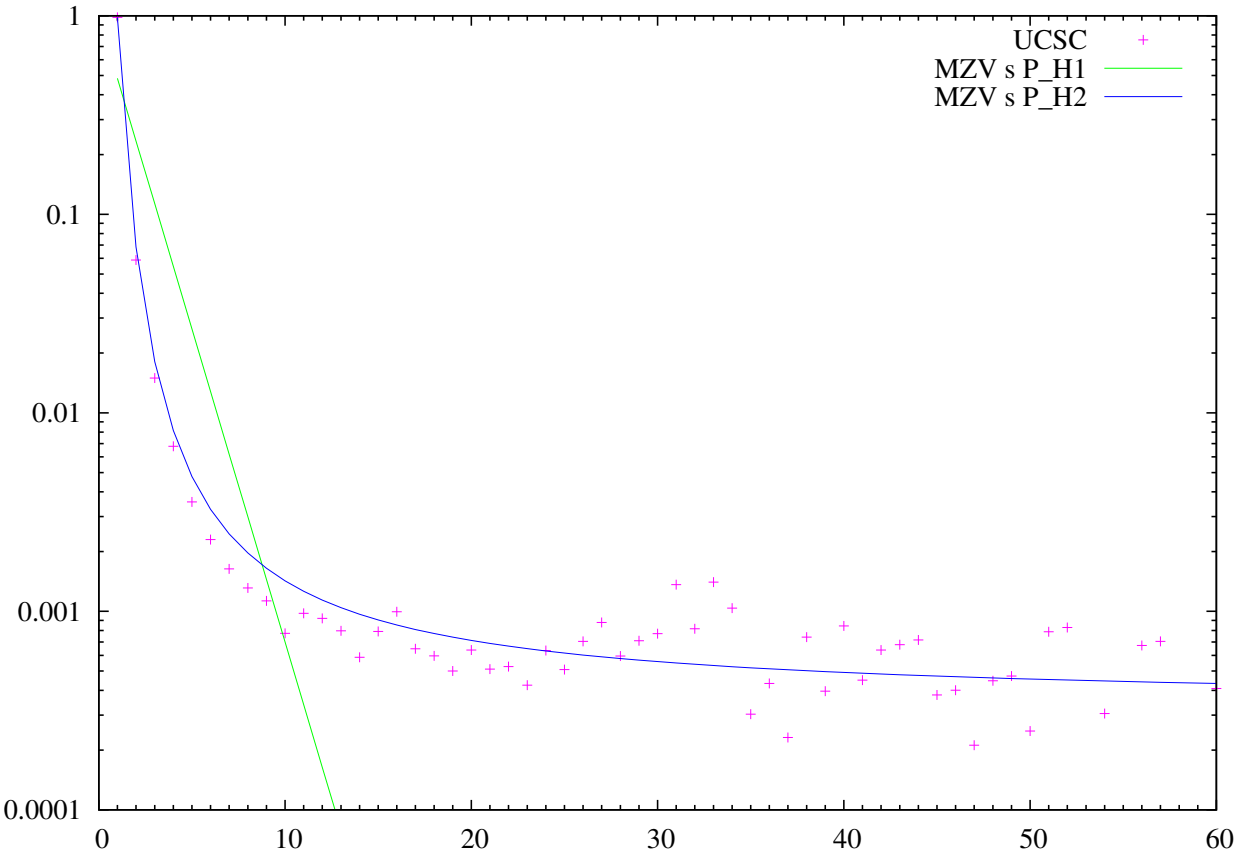
Zaujímavým atribútom je priemerný počet vstupujúcich hrán do vrcholu v závislosti od jeho vzdialenosti od začiatku grafu, respektíve priemerný počet vystupujúcich hrán z vrcholu v závislosti od jeho pozície od konca grafu. V našom modeli majú tieto dve závislosti rovnaký priebeh, pretože pravdepodobnosť pre hrany je symetrická a závisí len od vzdialenosti vrcholov. Porovnanie s reálnymi dátami je na obrázku 5.7. V MZV je viac možností pre vstupujúce hrany pre vrchol, ktorý je od začiatku grafu viac vzdialený, pretože má tie

isté možnosti ako bližší vrchol a ešte ďalšie navyše. V dátach z UCSC databázy je trend skôr opačný, najväčší priemerný počet vstupujúcich hrán majú vrcholy blízke začiatku grafu a opačne. Tento jav je pravdepodobne spôsobený tým, že na začiatku, resp. konci sekvencie sa často objavuje exón s alternatívnym začiatkom, resp. koncom.

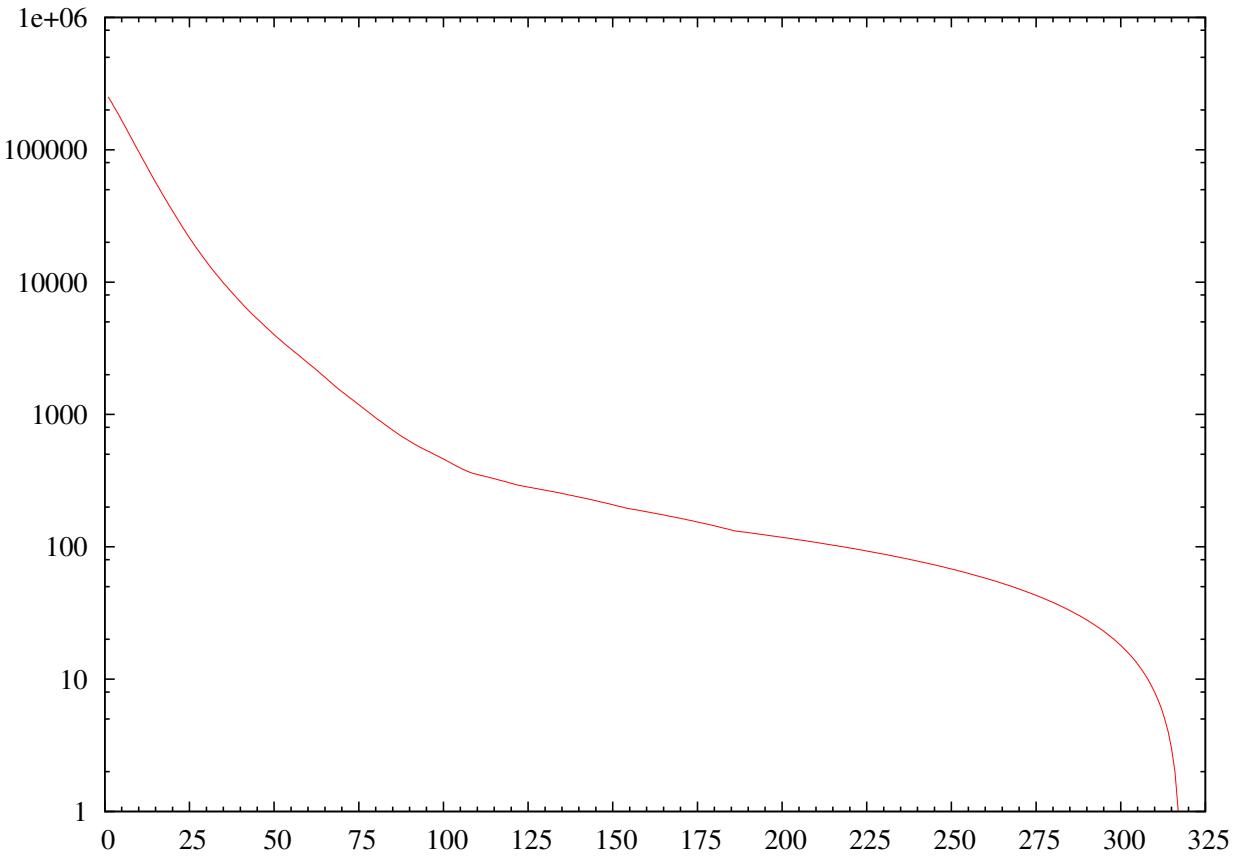
Nakoniec sme vyskúšali MZV použiť ako generátor grafov bez pevného určenia počtu vrcholov. Generovanie prebiehalo tak, že sme postupne pridávali na koniec do grafu nové vrcholy. Po každom pridanom vrchole sme podľa P_H vygenerovali hrany doň vchádzajúce z predchádzajúcich vrcholov. Ak doň po pridaní hrán žiadna hrana nevchádzala, tak sme prehlásili generovanie za ukončené, pridávaný vrchol sme zahodili a ďalšie sme už nepridávali.

Pomocou vzorkovania sme potom zistili frekvenciu výskytu grafov podľa ich veľkosti. Tieto údaje sú zakreslené v grafe na obrázku 5.6. Z neho sa dá usúdiť, že MZV s P_{H_1} nie je na takýto prístup vôbec vhodný. MZV s P_{H_2} je na to o trochu lepšie, ale ani o ňom sa nedá povedať, že by sme ho mohli na takýto spôsob použitia odporučiť.

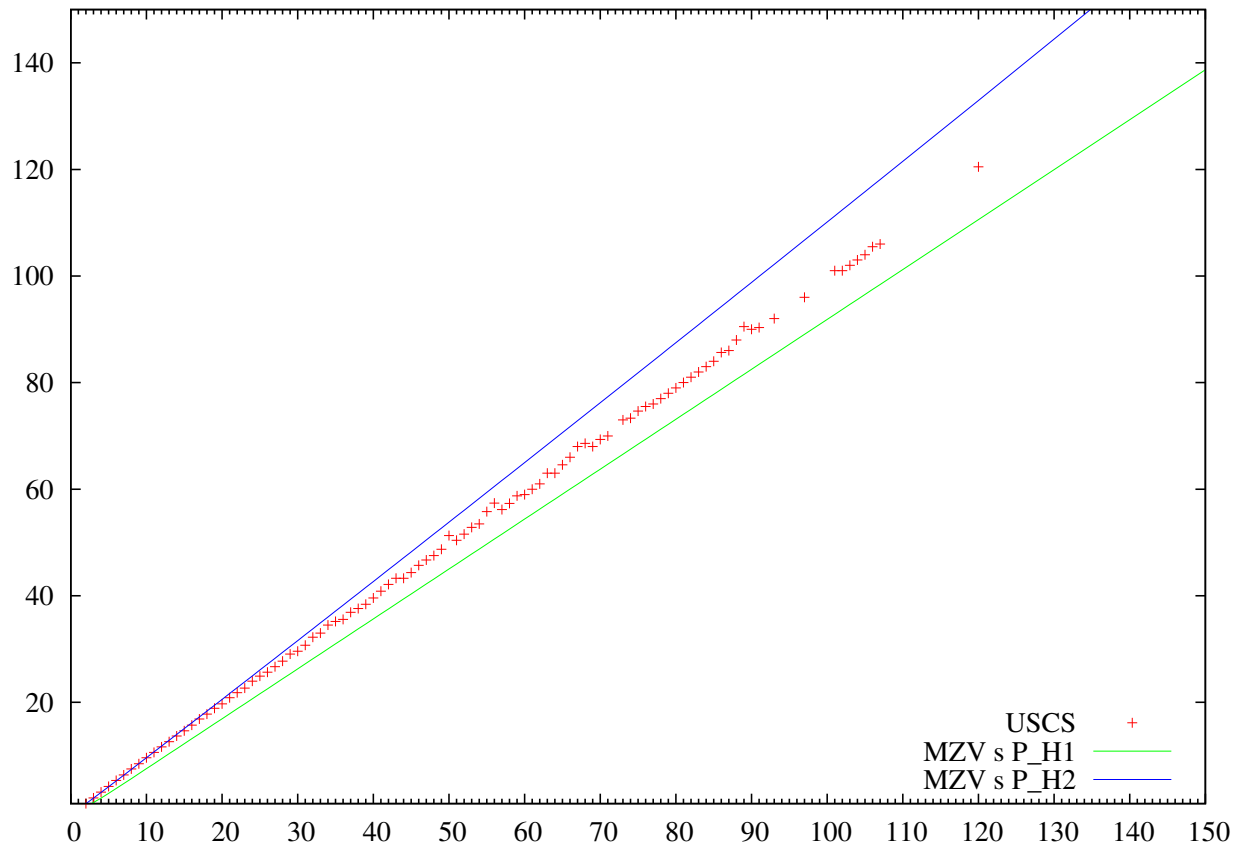
V závere je treba skonštatovať, že MZV je síce model v mnohých aspektoch atraktívny hlavne pre svoju jednoduchosť, no je ho schopnosti dobre popísať javy vyskytujúce sa v grafoch alternatívneho zostrihu sú obmedzené. Preto sme vo zvyšku práce tiež uvažovali s biologicky motivovaný model (BMM).



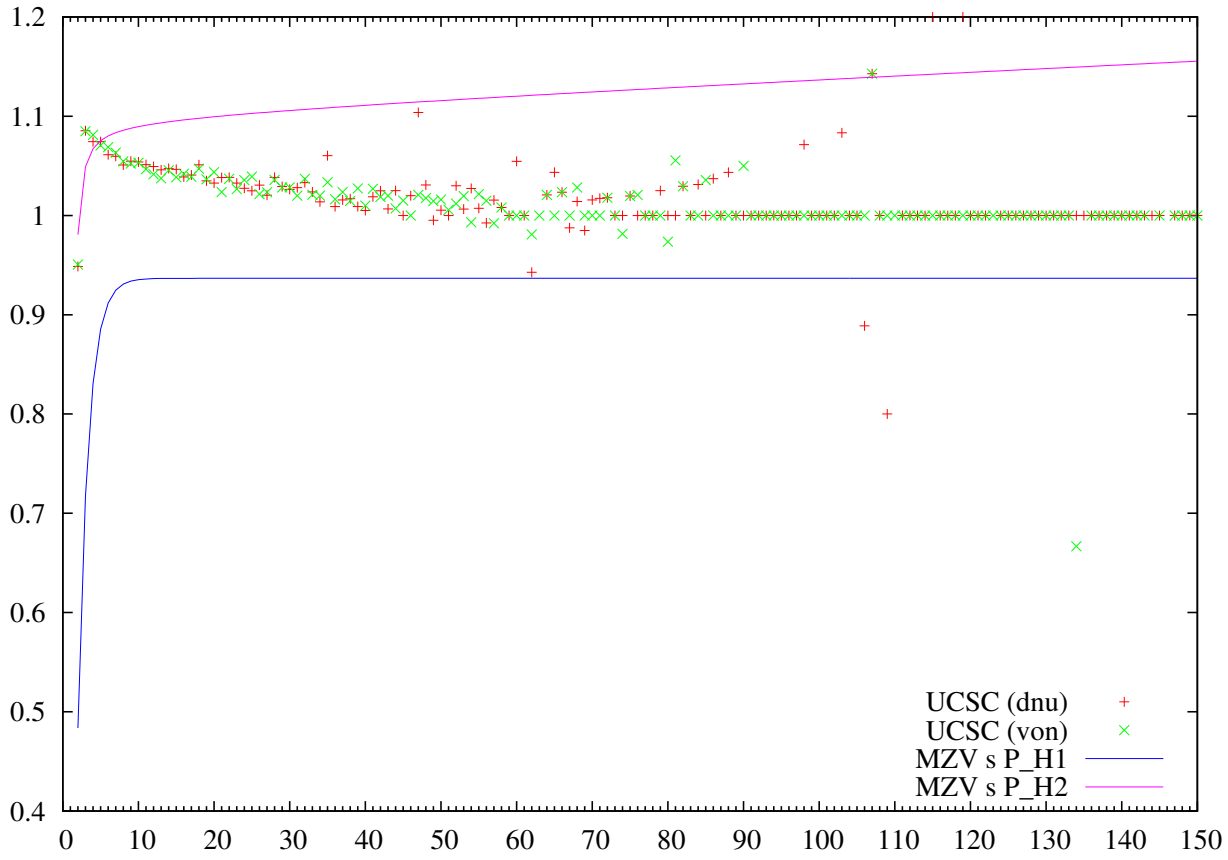
Obr. 5.4: Pravdepodobnosť spojenia dvoch vrcholov hranou v závislosti od ich vzájomnej vzdialenosti. Na x-ovej osi je ich vzdialenosť a na y-ovej pravdepodobnosť hrany. Ružové body predstavujú dáta z UCSC databázy, zelenou čiarou je naznačený stav pre MZV s P_{H_1} a modrá čiara patrí MZV s P_{H_2}.



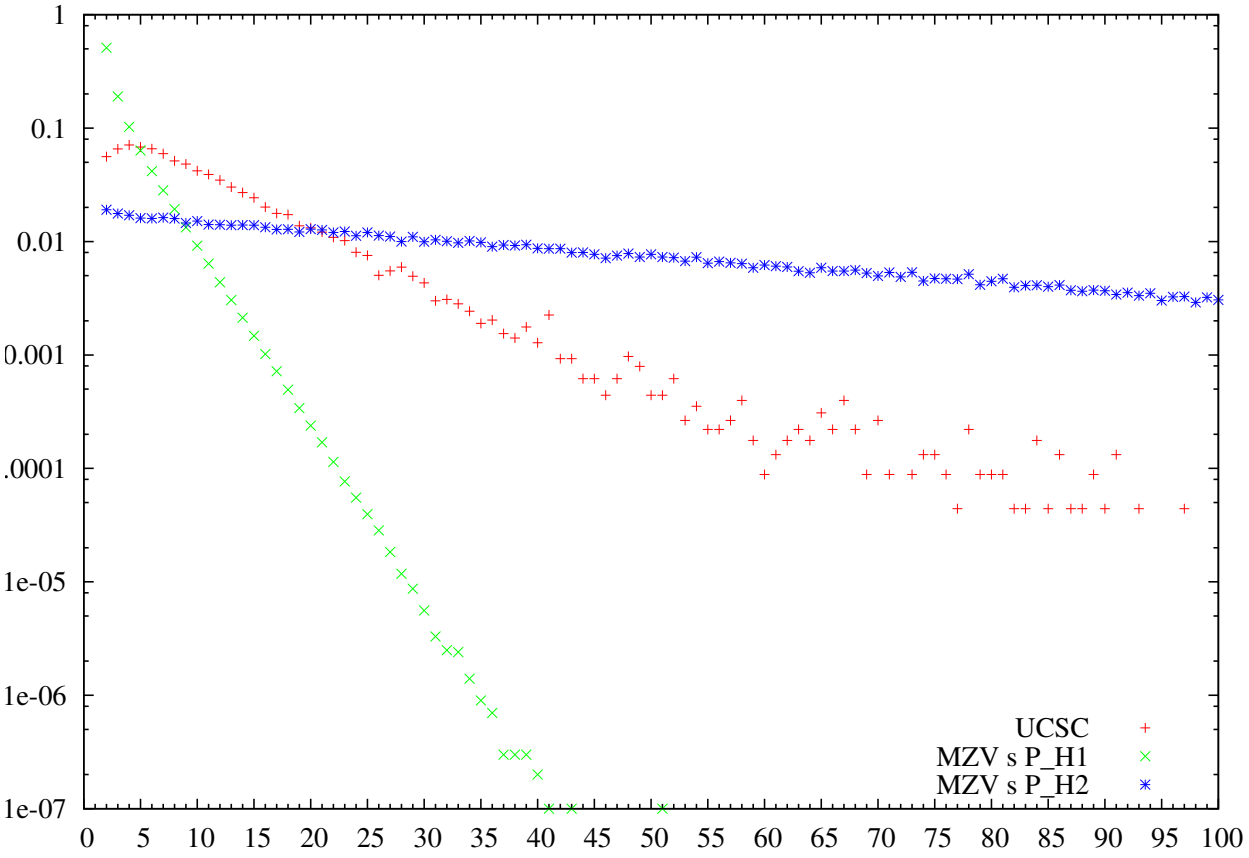
Obr. 5.5: Počet dvojíc vrcholov v UCSC databáze v závislosti od ich vzájomnej vzdialenosti.



Obr. 5.6: Priemerný počet hrán grafu v závislosti od jeho počtu vrcholov. Na x-ovej osi je veľkosť grafu a na y-ovej priemerný počet hrán.



Obr. 5.7: Priemerný počet vstupujúcich hrán do vrcholu v závislosti od pozície vrcholu od začiatku sekvencie podľa dát z USCS databázy je vyznačený červenými bodmi. Zelené body označujú priemerný počet vystupujúcich hrán z vrcholu v závislosti od pozície vrcholu od konca sekvencie. Modrá čiara znázorňuje obe tieto závislosti naraz pre MZV s P_{H_1}, pretože podmienka na prítomnosť hrany je symetrická. Ružová čiara znázorňuje to isté ako modrá, ale pre MZV s P_{H_2}.



Obr. 5.8: Pravdepodobnosť vygenerovania grafu s daným počtom vrcholov. Na x-ovej osi je veľkosť grafu a na y-ovej pravdepodobnosť vygenerovania (pre reálne dáta frekvencia výskytu).

Kapitola 6

Inferencia pomocou modelov alternatívneho zostrihu

Doposiaľ sme vytvorili dva pravdepodobnostné modely grafovej reprezentácie alternatívneho zostrihu. Ukázali sme, ako tieto modely natrénovať. V tejto kapitole ukážeme, ako takýto model spojiť so stochastickými modelmi DNA sekvencie. Na základe tohto spojenia vytvoríme algoritmy, ktoré budú vedieť odvodiť najpravdepodobnejšie rozdelenie vstupnej DNA sekvencie na vrcholy a *nevrcholy*, spolu s hranami spájajúcimi vrcholy. Toto rozdelenie určuje, ktoré časti sekvencie sú exóny (v aspoň jednej izoforme), ktoré intróny a aké sú prípustné izoformy zostrihu (vzhľadom na obmedzenia grafov zostrihu).

6.1 Formálna definícia problému

Najprv predstavíme všeobecný tvar vstupu a výstupu inferenčného algoritmu, ktorý konkrétnejšie špecifikujeme v nasledujúcich sekciách pracujúcich s konkrétnymi modelmi.

Vstupy budú DNA sekvencia, grafový model alternatívneho zostrihu, potencionálne miesta zostrihu sekvencie a stochastické modely častí DNA sekvencie.

Hranice vrcholov a *nevrcholov* nemôžu byť na ľubovolnej pozícii vo vstupnom reťazci, len

na potencionálnych miestach zostrihu [1]. Miesta zostrihu sa dajú získať z databázy SpliceDB [BSS01], v ktorej sú takéto informácie o génoch zhromažďované. Tieto dáta sú biologickými metódami overené. Druhou alternatívou pre prípadnú nedostupnosť dát k skúmanému regiónu je pomocou metód strojového učenia odhadnúť, kde sa tieto miesta zostrihu nachádzajú. Sú známe metódy, ktoré tento problém riešia s chybou menšou ako 5% [SRJM02].

Miesto zostrihu je miestom medzi poslednou bázou jedného typu sekvencie a prvou bázou druhého typu sekvencie. Nech i-te miesto zostrihu leží medzi a-tou a $(a+1)$-vou bázou a j-te miesto zostrihu leží medzi b-tou a $(b+1)$-vou bázu v sekvencii. Potom úsek sekvencie od i-teho po j-te miesto zostrihu bude v ďalšom texte znamenať úsek sekvencie od a-tej bázy po $(b-1)$-vú bázu.

Stochastické modely častí DNA sekvencie sú funkcie popisujúce pravdepodobnosť vygenerovania konkrétnej DNA sekvencie za predpokladu, že daný úsek je určitého typu, napríklad intrón alebo exón. Tieto funkcie je možné rátať napríklad pomocou natrénovaných skrytých Markovových modelov. Budeme o nich predpokladať, že ich hodnotu vieme zistiť v konštantnom čase, respektíve, že ich už máme predrátané pre všetky možné úseky vstupnej sekvencie.

Želaným výstupom je najpravdepodobnejšie rozdelenie vstupnej sekvencie na vrcholy a *nevrcholy*, spolu s hranami spájajúcimi vrcholy. Každá časť sekvencie musí byť buď vrchol alebo *vrchol*, tretia možnosť nie je. Pravdepodobnosť je určená ako súčin pravdepodobnosti grafu v modeli alternatívneho zostrihu a pravdepodobností jednotlivých častí rozdelenia sekvencie.

6.2 Jednoduchý inferenčný algoritmus pre MZV

Pri hľadaní najlepšieho rozdelenia za pomoci MZV budeme využívať dva parametre tohto modelu: pravdepodobnosť počtu vrcholov $P_V(n)$ a pravdepodobnosť výskytu hrany $P_H(x)$. Použijeme dva stochastické modely častí DNA sekvencie, konkrétne pravdepodobnosť $E(i,j)$, že úsek vstupnej sekvencie medzi i-tym a j-tym miestom zostrihu je vrchol (exón aspoň

[1] V ďalšom texte ich budeme pre zjednodušenie nazývať len miesta zostrihu

v jednej izoforme) a pravdepodobnosť $I(i,j)$, že úsek vstupnej sekvencie medzi i-tym a j-tym miestom zostrihu je *nevrchol* (intrón v každej izoforme)

Našim cieľ je nájsť takú segmentáciu vstupnej sekvencie spoločne s konfiguráciou hrán, ktorá maximalizuje pravdepodobnosť vygenerovania tejto sekvencie pomocou nášho modelu a stochastických modelov častí DNA sekvencie. Pravdepodobnosť, ktorú chceme maximalizovať, vieme zapísať takto:

$$P_V(n) \prod_{(k,l)\in \texttt{Hrany}} P_H(l-k) \prod_{(k,l)\in \texttt{Nehrany}} (1-P_H(l-k)) \prod_{(i,j)\in \texttt{Vrcholy}} E(i,j) \prod_{(i,j)\in \texttt{Nevrcholy}} I(i,j)$$

kde n je počet vrcholov, `Hrany` je množina dvojíc poradových čísiel vrcholov, ktoré spája hrana, `Nehrany` je množina dvojíc poradových čísiel vrcholov, ktoré nespája hrana, `Vrcholy` je množina dvojíc indexov miest zostrihu vymedzujúcich jednotlivé vrcholy, a `Nevrcholy` je obdobná množina pre *nevrcholy*.

Nebudeme vyžadovať, aby medzi dvomi vrcholmi bol *nevrcholový* úsek sekvencie, pretože takéto rozdelenie sekvencie vzniká napríklad pri alternatívnom začiatku alebo konci exónu. Na druhej strane, po jednom *nevrchole* nemôže hneď nasledovať ďalší, pretože takéto rozdelenie by vzhľadom na konštrukciu grafu k alternatívnemu zostrihu nemohlo vzniknúť.

Súčin pravdepodobností za hrany nie je ovplyvnený konkrétnou pozíciou vrcholov, závisí len od ich počtu. Preto môžeme optimálne rozloženie hrán riešiť nezávisle od hraníc vrcholov. Najprv objasníme algoritmus, ktorý bude hľadať najpravdepodobnejšie rozdelenie vstupnej sekvencie s konkrétnym počtom vrcholov. Potom predstavíme algoritmus, ktorý bude hľadať najlepšie doplnenie hrán pre konkrétny počet vrcholov. Nakoniec tieto dva algoritmy spojíme do jedného, ktorý už bude hľadať najpravdepodobnejšie rozdelenie sekvencie spolu s hranami bez obmedzenia na počet vrcholov.

6.2.1 Najpravdepodobnejšia segmentácia sekvencie

Rozdelenie sekvencie na vrcholy a *nevrcholy* spravíme pomocou metódy dynamického programovania.

Nech $P[i,n]$ predstavuje pravdepodobnosť najlepšieho riešenia pre podsekvenciu vstupnej sekvencie od prvého až po i-te miesto zostrihu, pričom toto riešenie v sebe zahŕňa práve n vrcholov a končí vrcholom. Ak poznáme $P[j,n-1]$ pre všetky $j < i$, tak pravdepodobnosť $P[i,n]$ môžeme vypočítať výberom najpravdepodobnejšej zo všetkých možností, kde môže končiť predchádzajúci vrchol, a k nim všetkých možností pre dĺžku po ňom nasledujúceho *nevrcholu*, ktorá môže byť aj nulová. Tým je implicitne určená dĺžka práve pridávaného n-tého vrcholu. Pre tieto možnosti vypočítame ich pravdepodobnosti a vyberieme z nich najväčšiu.

Jedným špeciálnym prípadom sú sekvencie pozostávajúce len z jedného vrcholu $P[i,1]$. V tom prípade sa *nevrchol* nikam nevkladá a táto pravdepodobnosť je priamo $E(1,i)$.

Na začiatku a konci celej sekvencie nemusí byť len *nevrchol*, ale môže tam byť aj vrchol. Taktiež po vrchole nemusí vždy nasledovať *nevrchol*. Pre zjednodušenie zápisu výpočtu zavádzame *nevrchol* s nulovou dĺžkou, ktorý začína aj končí v tom istom mieste zostrihu. Jeho pravdepodobnosť bude bude $I(i,i) = 1$. Vďaka tomu nemusíme špeciálne ošetrovať situácie s vypustením *nevrcholu*.

Výpočet $P[i,n]$ môžeme teraz zapísať nasledovným spôsobom:

$$P[i,n] = \begin{cases} \max\{P[j,n-1]I(j,k)E(k,i) \mid n-1 < j \leq k < i\} & \text{ak } n > 1 \\ \max\{I(1,j)E(j,i) \mid 0 < j < i\} & \text{ak } n = 1 \end{cases}$$

Časová zložitosť

Nech počet miest zostrihu na vstupe je N. Potom maximálny počet vrcholov je N, pretože každý úsek medzi dvomi miestami zostrihu môže byť samostatný vrchol. V najhoršom prípade budeme musieť vyrátať pravdepodobnosť pre $P[N,N]$ a všetky bunky s nižšími indexmi, čo je $O(N^2)$ buniek. Pri výpočte každej bunky musíme vyskúšať všetky možné hodnoty pre k a l. Týchto dvojíc je v najhoršom prípade $O(N^2)$. Preto časová zložitosť rozdelenia sekvencie je $O(N^4)$.

Rekonštrukcia riešenia

Pre praktické použitie nie je postačujúce zistiť len pravdepodobnosť najlepšieho riešenia. Preto sme navrhli spôsob, ako zrekonštruovať, kde presne ležia hranice jednotlivých úsekov.

V novej tabuľke $R[i, n]$ si budeme pamätať dvojicu čísiel (j, k) pomocou ktorej sme vybrali maximum pri rátaní $P[i, n]$. V prípade $n = 1$ na hodnote k nezáleží. Po zistení celkového maxima sa vieme vydať spätne podľa hodnôt v R a zistiť rozdelenie vstupnej sekvencie.

Čas potrebný na túto rekonštrukciu je O(N), pretože máme najviac N vrcholov a pre každý sa musíme pozrieť späť k čomu sme ho pridali.

Pamäťová zložitosť

Tabuľka P má N^2 buniek, a pre ďalší výpočet budeme potrebovať každú z nich. Preto pamäťová zložitosť je $O(N^2)$. Ak potrebujeme zrekonštruovať najlepšie riešenie, tak si budeme musieť zapamätať celú tabuľku R. Pamäťovú zložitosť to nezmení, pretože táto tabuľka má rovnaké rozmery ako P.

6.2.2 Najpravdepodobnejšia konfigurácia hrán

Druhá časť algoritmu spočíva vo vyrátaní najpravdepodobnejšej konfigurácie hrán medzi danými vrcholmi. Pre ľubovolný počet vrcholov platí, že hrany budú spájať len tie vrcholy, pre ktoré $P_H(x) \geq 0,5$, pretože v opačnom prípade je výhodnejšie ich nespojiť. Preto použijeme jednoduchý greedy algoritmus, ktorý sa pozrie na každú dvojicu vrcholov, zistí ich vzájomnú vzdialenosť x, a podľa hodnoty $P_H(x)$ rozhodne. Ako presne vypočítať pravdepodobnosť tejto konfigurácie vysvetlíme v nasledujúcich odsekoch.

Pravdepodobnosť najlepšej konfigurácie hrán pre n vrcholov vieme vyrátať z pravdepodobnosti pre $n-1$ vrcholov. Stačí k nim prirátať pravdepodobnosti za $n-1$ potencionálnych hrán s dĺžkami od 1 po $n-1$ ktoré vzniknú pridaním n-tého vrcholu. Pravdepodobnosti za takúto sadu dĺžok si môžme predrátať:

$$H'[n] = \begin{cases} \max\{P_H(1), 1 - P_H(1)\} & \text{ak } n = 1 \\ H'[n-1] \max\{P_H(n), 1 - P_H(n)\} & \text{ak } n > 1 \end{cases}$$

Výpočet doplnenia hrán k n vrcholom potom vieme zapísať nasledovnou rekurenciou:

$$H[n] = \begin{cases} 1 & \text{ak } n = 1 \\ H'[1] & \text{ak } n = 2 \\ H[n-1]H'[n-1] & \text{ak } n > 2 \end{cases}$$

Pre jeden vrchol sme pravdepodobnosť definovali ako 1, neskôr nám to zjednoduší zápis výpočtu celkovej pravdepodobnosti.

Časová zložitosť vyplnenia oboch tabuliek bude $O(N)$, pretože každú bunku oboch tabuliek vieme vyrátať v konštantnom čase a ako N vrcholov nemôžme mať.

Pamäťová zložitosť bude taktiež $O(N)$ kvôli veľkosti H' a H.

Ak by sme potrebovali kvôli rekonštrukcii najpravdepodobnejšieho riešenia vedieť vymenovať všetky hrany, ktoré bude najpravdepodobnejšie riešenie obsahovať, tak by sa časová zložitosť zhoršila na $O(N^2)$, pretože v najhoršom prípade bude hrana medzi každými dvoma vrcholmi.

V tomto prípade sa neoplatí tieto zoznamy hrán vytvoriť pre každý počet hrán, pretože by to v najhoršom prípade zabralo až $O(N^3)$ pamäte. Spravíme to len raz na konci celého algoritmu, kedy už vieme, koľko vrcholov je v najpravdepodobnejšom rozdelení.

Navyše môžeme predpokladať, že funkcia $P_H(x)$ bude od istého X stále menšia ako $0,5$. Tento predpoklad potvrdzujú dáta z UCSC databázy, ktoré sú vyobrazené na obrázku 5.4 v kapitole 5.3.

Nech X je posledné také, pre ktoré $P_H(X) > 0,5$. Potom hrana môže viesť len medzi vrcholmi vzdialenými od seba najviac X, čo by zlepšilo odhad časovej zložitosti vymenovania prítomných hrán na lineárny.

6.2.3 Kompletný algoritmus

Najpravdepodobnejšie dekódovanie vstupnej sekvencie môže mať rôzne formy. Prvá možnosť je, že celá vstupná sekvencia je jeden *nevrchol*. Pravdepodobnosť tejto možnosti nie je v tabuľke P, pretože tá obsahuje len rozdelenia s aspoň jedným vrcholom. Pravdepodobnosť rozdelenia bez vrcholov je $I(1, N)P_V(0)$.

Druhá možnosť je, že najpravdepodobnejšie dekódovanie obsahuje aspoň jeden vrchol. Máme vyrátané pravdepodobnosti pre všetky počty vrcholov a všetky pozície posledného vrcholu, ale za koncom posledného vrcholu do konca sekvencie je ešte *nevrchol*, ktorého pravdepodobnosť musíme k týmto pravdepodobnostiam prinásobiť. Okrem konečného *nevrcholu* k týmto možnostiam ešte prirátame už vyrátanú pravdepodobnosť za hrany a pravdepodobnosť výskytu daného počtu vrcholov podľa modelu, čím dostaneme pravdepodobnosti pre najpravdepodobnejšie rozdelenie celej sekvencie s konkrétnym počtom vrcholov, konkrétnou pozíciou konca posledného vrcholu a s hranami medzi vrcholmi. Pravdepodobnosť, ktorú hľadáme, je maximum z týchto hodnôt:

$$\max\{I(1, N)P_V(0), \max\{P[i, n]I(i, N)H[n]P_V(n) \mid 0 < i \leq N, 0 < n \leq N\}\}$$

Pre rekonštrukciu najpravdepodobnejšieho riešenia si stačí zapamätať ktorú možnosť sme vybrali ako maximum.

Časová zložitosť

Samotný výber maxima zo všetkých možností trvá $O(N^2)$. Časová zložitosť celého algoritmu je $O(N^4)$, kvôli času potrebného na vyplnenie tabuliek P a H.

Rekonštrukcia riešenia

Pre rekonštrukcia najpravdepodobnejšieho riešenia si v tomto kroku stačí zapamätať pomocou ktorej kombinácie i a n sme vybrali maximum. Samotné rozdelenie sekvencie potom nájdeme s pomocou tabuľky R a zoznam prítomných hrán vytvoríme podľa jednoduchého

pravidla spomenutého v kapitole 6.2.2. Časová zložitosť vytvorenia riešenia bude rovnaká ako čas potrebný na vytvorenie zoznamu hrán. Ten v každom prípade nie je menší ako O(N), čo je časová zložitosť rekonštrukcie rozdelenia sekvencie.

Pamäťová zložitosť

Najväčšie nároky na pamäť má algoritmus na rozdelenie sekvencie. V ňom sa nevyhneme použitiu $O(N^2)$ pamäte. Všetky ostatné časti celého algoritmu túto potrebu neprevyšujú. Pri rekonštrukcii najpravdepodobnejšieho rozdelenia taktiež nepotrebujeme viac pamäte.

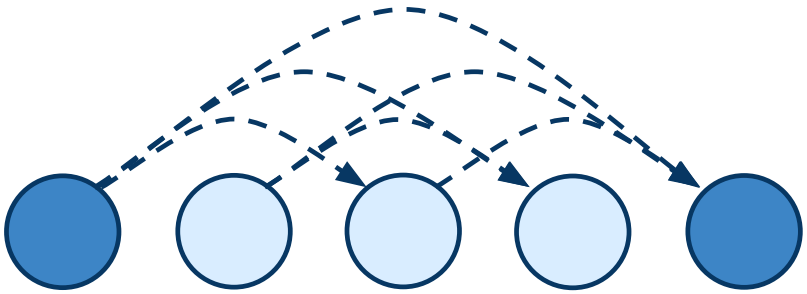
6.3 Komplexný inferenčný algoritmus pre MZV

Jediná informácia, ktorú sme doposiaľ využili pri predpovedaní alternatívnych zostrihov bola obsiahnutá v pravdepodobnostnom modeli grafu alternatívneho zostrihu. Doteraz sme nevyužili informáciu obsiahnutú v samotnej sekvencii, pretože použité stochastické modely DNA sekvencie poskytovali len informáciu o tom, či úsek DNA sekvencie je aspoň v jednej izofome exón (vrchol), alebo vôbec (*nevrchol*). To viedlo k pomerne triviálnej inferencii, ktorej výsledky nedávajú biologicky zmysluplné odpovede.

Aby sme adresovali tento problém, rozšírime použité stochastické modely DNA nasledovným spôsobom:

- pravdepodobnosť $E(i, j)$, že úsek vstupnej sekvencie medzi i-tym a j-tym miestom zostrihu je vrchol cez ktorý prechádza každá cesta grafu (exón prítomný v každej izoforme)
- pravdepodobnosť $E_{\texttt{alt}}(i, j)$, že úsek vstupnej sekvencie medzi i-tym a j-tym miestom zostrihu je vrchol cez ktorý neprechádza každá cesta grafu (exón prítomný len v niektorých izoforme)
- pravdepodobnosť $I(i, j)$, že úsek vstupnej sekvencie medzi i-tym a j-tym miestom zostrihu je *nevrchol* (intrón v každej izoforme)

Najväčší rozdiel oproti predchádzajúcemu problému je v tom, že tentokrát nemôžeme existenciu hrán riešiť úplne oddelene od rozdelenia sekvencie. Prítomnosť hrán totiž mení funkciu, ktorou sa ráta pravdepodobnosť pre vrcholový úsek sekvencie. Označíme si vrcholy nad ktorými ide hrana ako *preskočené* a vrcholy nad ktorými nejde hrana ako *nepreskočené*. Ak si určíme, ktoré vrcholy ležiace vedľa seba budú *preskočené* a ich ľavý a pravý sused bude *nepreskočený* vrchol, tak pre takéto zoskupenie vrcholov už dokážeme nájsť najpravdepodobnejšie rozloženie hrán bez ohľadu na ich pozíciu v sekvencii. Príklad takéhoto zoskupenia vrcholov je na obrázku 6.1.



Obr. 6.1: Príklad zoskupenia vrcholov, pre ktoré budeme vyberať najpravdepodobnejšiu konfiguráciu hrán. Doplnené hrany budú tvoriť takú podmnožinu hrán naznačených prerušovanou čiarou, pre ktorú platí, že nad každým svetlý vrchol bude *preskočený*. Tmavšie vrcholy na krajoch budú *nepreskočené*.

Algoritmus na nájdenie pravdepodobnosti najpravdepodobnejšieho rozdelenia s hranami znovu rozdelíme na dve časti. Prvá z nich bude riešiť rozdelenie sekvencie a druhá dopĺňanie hrán.

6.3.1 Najpravdepodobnejšia segmentácia sekvencie

V bunke tabuľky $P[i, n, m]$ si budeme pamätať pravdepodobnosť najpravdepodobnejšieho rozdelenia vstupnej sekvencie od prvého po i-te potencionálne miesto zostrihu, pričom pre toto rozdelenie platí, že posledný vrchol končí i-tym miestom zostrihu, rozdelenie obsahuje n vr-

cholov a posledných m vrcholov je *preskočených* (pričom $m+1$-vý vrchol od konca je *nepreskočený*).

Rátanie pravdepodobnosti vieme rozdeliť na štyri prípady podľa toho ktorý vrchol pridávame a akého je typu:

1. *preskočený a prvý vrchol:* Takýto vrchol nemôže existovať, pretože pred ním nie je žiaden, z ktorého by mohla vychádzať hrana, ktorá ho preskočí. Pravdepodobnosť tejto možnosti je preto nula.

2. *preskočený a nie prvý vrchol:* V tomto prípade ešte nie je ukončené zoskupenie za sebou idúcich *preskočených* vrcholov a preto pravdepodobnosť za hrany vrámci tohto úseku ešte nezarátame.

3. *nepreskočený a prvý vrchol:* Stále musíme rátať s možnosťou, že prvý vrchol nezačína priamo od prvého miesta zostrihu a nachádza sa tam *nevrchol*.

4. *nepreskočený a nie prvý vrchol:* V tomto prípade ukončujeme zoskupenie predchádzajúcich *preskočených* vrcholov a zarátavame pravdepodobnosť za hrany medzi nimi. Je možné, že pred práve pridávaným *nepreskočeným* vrcholom neboli žiadne *preskočené* vrcholy. Potom sa zaráta pravdepodobnosť len za možnú hranu medzi poslednými dvomi vrcholmi, lebo do práve pridaného vrcholu iná hrana vstupovať nemôže, pretože v opačnom prípade by platilo, že predchádzajúci vrchol už nie je *nepreskočený*.

Nech $H[n]$ označuje pravdepodobnosť najpravdepodobnejšieho rozmiestnenia hrán nad zoskupením $n+2$ vrcholov, kde dva krajné vrcholy sú *nepreskočené* a tie medzi nimi *preskočené*. $H[0]$ bude pravdepodobnosť najpravdepodobnejšieho pridania hrany medzi dvomi susednými *nepreskočenými* vrcholmi. Potom výpočet tabuľky P je definovaný nasledujúcou rekurenciou:

$$P[i,n,m] = \begin{cases} 0 & \text{ak } n = 1, m > 0 \\ \max\{P[j,n-1,m-1]I(j,k)E_{\texttt{alt}}(k,i) \mid 2n-2 < j \leq k < i\} & \text{ak } n > 1, m > 0 \\ \max\{I(1,j)E(j,i) \mid 0 < j < i\} & \text{ak } n = 1, m = 0 \\ \max\left\{ P[j,n-1,l]I(j,k)E(k,i)H(l) \,\middle|\, \begin{array}{l} 2n-2 < j \leq k < i, \\ 0 \leq l < n-1 \end{array} \right\} & \text{ak } n > 1, m = 0 \end{cases}$$

Celkové najpravdepodobnejšie rozdelenie vyberieme ako maximum z tabuľky P.

Znovu ešte musíme prirátať pravdepodobnosť za *nevrchol* na konci sekvencie a pravdepodobnosť za počet vrcholov celého grafu podľa modelu. Posledný vrchol, rovnako ako prvý, nemôže byť *preskočený*. Preto berime do úvahy len tie možnosti, kde $m = 0$. Nech N označuje počet miest zostrihu, ktoré máme k dispozícii. Pravdepodobnosť rozdelenia pre celú sekvenciu vyrátame nasledovným spôsobom:

$$\max\{I(1,N)P_V(0), \max\{P[i,m,0]I(i,N)P_V(m) \mid 0 < i \leq N, 0 < m < N\}\}$$

Časovej a pamäťovej zložitosti sa budeme venovať až spoločne po vysvetlení všetkých častí algoritmu.

6.3.2 Najpravdepodobnejšia konfigurácia hrán

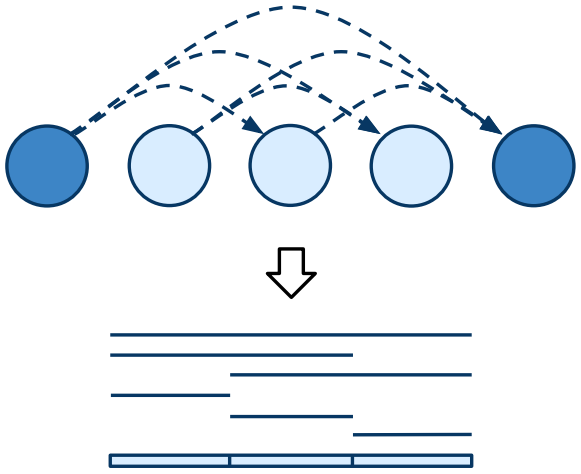
V tejto časti predstavíme algoritmus na zistenie najpravdepodobnejšej konfigurácie hrán pre zoskupenie *preskočených* vrcholov medzi dvomi *nepreskočenými* vrcholmi.

Ak existuje aspoň jedna vzdialenosť vrcholov $x > 1$ pre ktorú platí $P_H(x) \geq 0.5$, tak je zaručené, že nad každým *preskočeným* vrcholom hrana naozaj pôjde, pretože v takom prípade hrana pôjde nad každými x *preskočenými* vrcholmi ležiacimi vedľa seba. Na vyrátanie pravdepodobnosti za hrany potom môžme použiť algoritmus zo sekcie 6.2.2.

Ak neexistuje vzdialenosť $x > 1$ pre ktorú platí $P_H(x) \geq 0.5$, tak musíme pridať hrany tak, aby nad každým *preskočeným* vrcholom viedla aspoň jedna a súčasne aby sme našli najpravdepodobnejšie riešenie.

Transformácia úlohy pridania hrán

Úlohu pridania hrán do zoskupenia vrcholov si upravíme na úlohu, ako pokryť úsečku rozdelenú na dieliky pomocou malých úsečiek, ktoré budeme ďalej volať záplaty. Dĺžky záplat sú od jedného po n dielikov, kde n je počet dielikov celej úsečky. Rovnako dlhé záplaty majú rovnakú cenu. Naším cieľom je nájsť najlacnejšie pokrytie, pričom každý dielik úsečky môže byť pokrytý aj viac ako jednou záplatou. Tento prevod úlohy je znázornený na obrázku 6.2. Ľavý koniec záplaty budeme nazývať jej začiatkom a pravý koniec bude jej koncom.



Obr. 6.2: Transformácia úlohy výberu najpravdepodobnejších hrán medzi n vrcholmi na pokrytie úsečky dĺžky $n-2$ záplatmi. Obrázok naznačuje, aké záplaty máme k dispozícií. Každá z nich zodpovedá jednej hrane v pôvodnej úlohe.

Cena riešenia

Ceny záplat si z pravdepodobností výskytu hrán musíme určiť tak, aby najlacnejšie pokrytie transformovanej úlohy zodpovedalo najpravdepodobnejšiemu výberu hrán.

Cenu záplaty dĺžky x si označíme ako $C(x)$ a definujeme ako $\frac{1-P_H(x+1)}{P_H(x+1)}$. Táto cena je pre každú dĺžku záplaty väčšia ako 1, pretože $P_H(x) < 0.5$ pre všetky $x > 1$. Cena celkového pokrytia $H_{\texttt{pokrytie}}$ bude súčin cien všetkých použitých záplat, a pri použití viacerých záplat narastať.

Následne budeme môcť jednoducho vyrátať pravdepodobnosť za hrany pre celé zoskupenie vrcholov ako cenu celkového pokrytia $H_{\texttt{pokrytie}}$ na mínus prvú vynásobenú s pravdepodobnosťou za žiadne hrany. a s pravdepodobnosťou za hrany medzi susednými vrcholmi,

ktoré na *preskočenosť* vrcholov nemajú vplyv.

Výpočet pre n vrcholov bude vyzerať takto:

$$H[n] = \frac{1}{H_{\texttt{pokrytie}}[n-2]} \prod_{i=1}^{n} (1 - P_H(i+1))^{n-i+1} \max\{P_H(1), 1 - P_H(1)\}^{n-1}$$

Tým sa vykrátia pravdepodobnosti za neprítomnosť hrán, ktoré pridáme, a zároveň sa zarátajú pravdepodobnosti za ich prítomnosť.

Normálny tvar riešenia transformovanej úlohy

Každé najlacnejšie pokrytie sa dá transformovať na také pokrytie, pri ktorom prekryv buď nebude žiaden, alebo ho bude spôsobovať jedine posledná záplata. To dosiahneme postupným pridávaním záplat z pokrytia vedľa seba od ľavého konca pokrývanej úsečky. Pred pridaním poslednej záplaty ešte nemáme pokrytú celú úsečku, pretože v opačnom prípade by pokrytie, ktoré upravujeme, nebolo najlacnejšie, lebo by sme z neho mohli poslednú záplatu vyhodiť a tým ho zlepšiť. Poslednú záplatu umiestnime tak, aby jej koniec ležal na pravom konci úsečky. Pokrytiu v takomto tvare budeme hovoriť pokrytie v normálnom tvare.

Ku každému najlacnejšiemu pokrytiu existuje pokrytie v normálnom tvare s rovnakou cenou, a preto budeme hľadať najlacnejšie pokrytie priamo v normálnom tvare. Ak by sme poslednú záplatu tohto najlacnejšieho pokrytia presunuli tak, aby jej začiatok pokračoval priamo za koncom predposlednej záplaty, tak nepresiahne koniec úsečky o viac ako $n - 1$ dielikov, pretože najdlhšia záplata pozostáva z n dielikov.

Keď zistíme najlacnejšie pokrytia bez prekryvov pre úsečky s dĺžkami od n po $2n - 1$, tak z nich vyberieme to najlacnejšie. Nech je to pokrytie P' pre úsečku dĺžky $n + x$. Potom posunutím jeho poslednej záplaty ho upravíme na pokrytie P s prekryvom pre úsečku dĺžky n. Toto pokrytie bude zároveň aj najlacnejšie, pretože ak by existovalo iné lacnejšie pokrytie Q, tak ho posunom poslednej záplaty upravím na riešenie Q' pre úsečku konkrétnej dĺžky v intervale n až $2n - 1$. Keďže Q' má rovnakú cenu ako Q a Q je lacnejšie ako P, tak sme našli lacnejšie pokrytie ako P', čo je v spore s pravidlom jeho výberu.

Algoritmus pre hľadanie pokrytia

Nech N je dĺžka úsečky, ktorú chceme pokryť a nech $P_{\texttt{pok}}[n]$ označuje cenu najlacnejšieho pokrytia bez prekryvov úsečky dĺžky n. Pokrytie úsečky dĺžky n vieme vytvoriť pridaním záplaty dĺžky x k pokrytiu úsečky dĺžky $n-x$, kde x je prípustná cena záplaty. Najlacnejšie pokrytie úsečky dĺžky n bude potom minimum z cien pokrytí, ktoré dostaneme pridaním záplaty dĺžky x k najlacnejšiemu pokrytiu úsečky dĺžky $n-x$:

$$P_{\text{pok}}[n] = \begin{cases} 1 & \text{ak } n = 0 \\ \min\{P_{\text{pok}}[n-m]C(m) \mid 0 < m \leq \min\{N, n\}\} & \text{ak } n > 0 \end{cases}$$

My hľadáme riešenie úlohy pokrytia s prekryvom pre úsečku s dĺžkou n. Ako sme už ukázali, riešenie, ktoré hľadáme, je najlepšie z pokrytí bez prekryvov pre úsečky dĺžok n až $2n-1$ a jeho cena bude:

$$H_{\texttt{pokrytie}}[n] = \min\{P_{\texttt{pok}}[m] \mid n \leq m \leq 2n-1\}$$

Časová zložitosť

Pravdepodobnosť najpravdepodobnejšieho pridania hrán potrebujem pre všetky počty vrcholov menšie alebo rovné N, kde N je počet miest zostrihu sekvencie.

Vyrátanie tabuľky H za pomoci vyplnenej tabuľky $H_{\texttt{pokrytie}}$ zaberie lineárny čas vzhľadom na N, pretože toľko má buniek a každú z nich viem vyrátať v konštantnom čase, pokiaľ si predrátame zvyšok výrazu bez hodnoty z $H_{\texttt{pokrytie}}$. Toto predrátanie vieme spraviť v čase $O(N^2)$ veľmi podobným spôsobom, ako je použitý v sekcii 6.2.2.

Časová zložitosť vyplnenia $H_{\texttt{pokrytie}}$ bude $O(N^2)$, pretože z nej potrebujeme $N-2$ buniek a hodnotu každej z nich vyberáme ako minimum z N možností.

Posledná tabuľka, ktorú pre výber hrán musíme mať vyrátanú, je P_{pok} Jej celé vyplnenie bude trvať $O(N^2)$, pretože budeme potrebovať hodnoty od 0 po $2N-1$ a každú z nich získavame vybratím minima z najviac N možností.

Najpravdepodobnejšie pridanie hrán pre zoskupenie *preskočených* vrcholov medzi dvomi *preskočenými* vrcholmi potrebujeme pre vedieť pre veľkosti celého zoskupenia do N vrcholov.

Žiadna z tabuliek, ktorá sa na tento výpočet používa, nepotrebuje viac času ako $O(N^2)$. Preto celková časová zložitosť pridania hrán je kvadratická vzhľadom na počet miest zostrihu.

Pamäťová zložitosť

Pamäťová zložitosť je $O(N)$, pretože každá použitá tabuľka je jednorozmerná a najväčšia veľkosť rozmeru je $2N - 1$.

Rekonštrukcia riešenia

Pre zapamätanie si najpravdepodobnejšieho riešenia potrebujeme vedieť, ako sme k nemu dospeli. Stačí si ku každému výberu minima zapamätať ktorú možnosť sme vybrali. To sa týka tabuliek $H_{\texttt{pokrytie}}$ a $P_{\texttt{pok}}$, ku ktorým vytvoríme tabuľky $R_{\texttt{pokytie}}$ a $R_{\texttt{pok}}$ obsahujúce práve túto informáciu o spôsobe výberu minima. Pre jednu bunku tabuliek si stačí zapamätať jedno číslo a preto veľkosť pomocných tabuliek R bude rovnaká ako tabuliek H a P a pamäťová zložitosť sa nezhorší.

Spätným prechodom týchto tabuliek vieme v lineárnom čase vymenovať, ktoré medzi ktorými vrcholmi pôjde hrana.

6.3.3 Kompletný algoritmus

Časová zložitosť

Nech počet miest zostrihu je N. Prvou časťou celého algoritmu je vyplnenie tabuľky H. Nezávisle na tom, ako vyzerá funkcia P_H a ktorý z postupov na vypĺňanie budeme musieť použiť, bude mať táto časť časovú zložitosť $O(N^2)$.

V druhej časti algoritmu, kde rozdeľujeme vstupnú sekvenciu, potrebujeme na nájdenie najväčšej pravdepodobnosti vyplniť trojrozmernú tabuľku P s N^3 bunkami. V najhoršom prípade je potrebné pre vyplnenie jednej bunky prezrieť $O(N^2)$ možností. Dokopy to dáva časovú zložitosť $O(N^5)$.

Rozdeľovanie sekvencie je náročnejšie ako predrátavanie hrán a teda časová zložitosť celého algoritmu je $O(N^5)$.

Rekonštrukcia riešenia

Pri rozdeľovaní sekvencie budeme mať druhú tabuľku R, ktorá pre každú bunku P bude obsahovať maximálne tri čísla. Tie budú určovať aký výpočet sme spravili pri výbere maxima pre danú bunku. Tabuľka R bude preto obsahovať $O(N^3)$ hodnôt.

Jej spätným prechodom vieme v čase O(N) zrekonštruovať celé rozdelenie sekvencie, pretože druhá súradnica do R môže byť najviac N a v každom kroku späť sa zníži aspoň o jedna.

Pre kompletné riešenie potrebujeme aj hrany medzi vrcholmi. Túto informáciu potrebujeme zistiť vždy keď pri výpočte použijeme tabuľku H. V závislosti od funkcie P_H bude táto časť trvať buď $O(N^2)$, alebo $O(N)$. Keďže do tabuľky H sa v najhoršom prípade potrebujeme pozrieť N krát, tak by nám časová zložitosť vypísania hrán mohla stúpnuť na $O(N^2)$.

Stačí si však uvedomiť, že všetkých možných hrán pre N vrcholov je $O(N^2)$ a tým pádom toto vypisovanie hrán dokopy nemôže spraviť viac ako $O(N^2)$ práce, pretože žiadnu z hrán nevypisujeme viac krát.

Pamäťová zložitosť

Najviac pamäte potrebuje tabuľka P a k nej pomocná tabuľka na rekonštrukciu riešenia R. Žiadna iná časť algoritmu nemá takéto pamäťové nároky a preto pamäťová zložitosť celého algoritmu je $O(N^3)$.

6.4 Inferenčný algoritmus pre BMM

Úloha pre BMM ostáva rovnaká, no použité stochastické modely častí DNA sekvencie zmeníme tak, aby dokázali lepšie popísať jednotlivé typy vrcholov v tomto modeli. Budeme preto používať nasledovné modely pre nasledovné typy sekvencie:

- pravdepodobnosť $E_{\texttt{staly,stred}}(i, j)$, že úsek vstupnej sekvencie medzi i-tym a j-tym miestom zostrihu je *stály* vrchol
  (stred exónu prítomného v každej izoforme)

- pravdepodobnosť $E_{\texttt{nestaly,stred}}(i,j)$, že úsek vstupnej sekvencie medzi i-tym a j-tym miestom zostrihu je *nestály* vrchol
  (stred exónu, ktorý nie je prítomný v každej izoforme)

- pravdepodobnosť $E_{\texttt{staly,okraj}}(i,j)$, že úsek vstupnej sekvencie medzi i-tym a j-tym miestom zostrihu je *okrajovým* vrcholom vloženým ku *stálemu* vrcholu
  (alternatívny začiatok alebo koniec exónu nachádzajúceho sa v každej izoforme)

- pravdepodobnosť $E_{\texttt{staly,okraj}}(i,j)$, že úsek vstupnej sekvencie medzi i-tym a j-tym miestom zostrihu je *okrajovým* vrcholom vloženým k *nestálemu* vrcholu
  (alternatívny začiatok alebo koniec exónu nenachádzajúceho sa v každej izoforme)

- pravdepodobnosť $I(i,j)$, že úsek vstupnej sekvencie medzi i-tym a j-tym miestom zostrihu je *nevrchol*
  (intrón v každej izoforme)

Medzi stredovou časťou (*stály* alebo *nestály* vrchol) a *okrajovými* časťami sekvencie nebude môcť byť vo výslednom rozdelení *nevrcholová* časť sekvencie, pretože takéto rozdelenie pomocou BMM nevieme získať. V nasledujúcich častiach sa preto budeme pozerať na *stály* (resp. *nestály*) vrchol ako na kombináciu *stáleho* (resp. *nestáleho*) vrcholu spolu s *okrajovými* vrcholmi, ktoré ho obklopujú.

6.4.1 Najpravdepodobnejšia dekompozícia sekvencie

Náš algoritmus je opäť postavený na dynamickom programovaní. Rozdelili sme ho do troch logických častí, ktoré postupne od najvrchnejších častí objasnia celý algoritmus. Narozdiel od MZV sa teraz nebudeme špeciálne zaoberať konfiguráciou hrán, pretože v tomto modely je prítomnosť hrán medzi vrcholmi určená priamo generatívnym procesom.

Algoritmus

Podproblém, ktorý budeme riešiť, bude nájdenie najpravdepodobnejšieho rozdelenia vstupnej sekvencie od jej začiatku po i-te miesto zostrihu na vrcholy a hrany medzi nimi, pričom

stálych vrcholov bude presne n a posledný z nich bude končiť na i-tom mieste. Pravdepodobnosti budeme mať uložené v tabuľke $P'[i, n]$.

Predpokladajme, že máme vyrátané $P'[j, m]$ pre všetky $j < i$ a $m < n$. Rozdelenie s n stálymi vrcholmi môže vzniknúť jedine z rozdelenia s $n - 1$ *stálymi* vrcholmi pripojením jedného *skoku* a za ním nasledujúceho *stáleho* vrcholu. Na vyrátanie $P'[i, n]$ nám preto stačí odskúšať všetky možnosti na ktorých môže končiť $(n - 1)$-vý *stály* vrchol a *skok* za ním, a vybrať z nich tú s najväčšou pravdepodobnosťou.

Použijeme dve ďalšie tabuľky, $SKOK$ a $V_{\texttt{staly}}$. V $SKOK[i, j]$ budú pravdepodobnosti najpravdepodobnejšieho *skoku*, ktorý začína na i-tom a končí na j-tom mieste potencionálneho zostrihu, pričom na jeho začiatku a konci je *nevrchol*. $V_{\text{staly}}[i, j]$ bude obsahovať pravdepodobnosť najpravdepodobnejšieho rozdelenia sekvencie od i-teho po j-te miesto zostrihu na *stály* vrchol obklopený *okrajovými* vrcholmi.

Špeciálnym prípadom je rozdelenie obsahujúce práve jeden *stály* vrchol. Vtedy toto rozdelenie pozostáva len z úvodného *nevrcholu* nasledovaného týmto vrcholom.

Výpočet P' vieme zapísať nasledovným spôsobom:

$$P'[i, n] = \begin{cases} \max\{P'[j, n-1]SKOK[j, k]V_{\texttt{staly}}[k, i] \mid n < j \leq k < i\} & \text{ak } n > 1 \\ \max\{I(1, j)V_{\texttt{staly}}(j, i) \mid 0 < j \leq i\} & \text{ak } n = 1 \end{cases}$$

Pri výpočte najpravdepodobnejšej možnosti nesmieme zabudnúť na:

- Zatiaľ sme nikde nezapočítali pravdepodobnosť podľa modelu za všetky použité *stále* vrcholy P_{V1}. Tú zohľadníme vo finálnom výpočte pravdepodobnosti rozdelenia vstupnej sekvencie.
- Celkové rozdelenie nemusí končiť vrcholom, ale na konci sekvencie môže byť aj *nevrchol*.
- Prípustný je aj prípad, kedy rozdelenie neobsahuje žiaden vrchol, pretože tabuľka P' obsahuje len pravdepodobnosti pre rozdelenia s aspoň jedným vrcholom.

Pravdepodobnosť najpravdepodobnejšieho rozdelenia vstupnej sekvencie po posledné N-té miesto zostrihu označíme ako P a vypočítame ako maximum z pravdepodobností pre jednotlivé možnosti počtu *stálych* vrcholov a pozície koncu posledného *stáleho* vrchola:

$$P = \max\{I(1,N), \max\{P'[j,n]P_{V1}(n)I(j,N) \mid 0 < n \leq j \leq N\}\}$$

Skoky

Skok rozprestierajúci sa od i-teho po j-te miesto zostrihu pozostáva zo striedajúcich sa *nevcholov* a *nestálych* vrcholov, pričom na začiatku aj na konci je *nevrchol*.

Takýto *skok* sa môže skladať z nula až $j-i$ *nestálych* vrcholov. Ak vyrátame najpravdepodobnejšie pravdepodobnosti pre všetky možnosti vrcholov, zarátame k každej aj pravdepodobnosť za počet vrcholov podľa funkcie P_{V2} z modelu, tam najväčšia z nich je hľadaná pravdepodobnosť celého skoku:

$$SKOK[i,j] = \max\{SKOK'[i,j,n]P_{V2}(n) \mid 0 <= n < j-i\}$$

Hodnoty pre $SKOK'[i,j,n]$ vieme zistiť ako maximum zo všetkých možností, kde môže končiť posledný *nevrchol* predchádzajúceho skoku a kde môže končiť za ním nasledujúci *nestály* vrchol:

$$SKOK'[i,j,n] = \begin{cases} I(i,j) & \text{ak } n = 0 \\ \max\{SKOK'[i,k,n-1]V_{\texttt{nestaly}}[k,l]I(l,j) \mid i \leq k < l \leq j\} & \text{ak } n > 0 \end{cases}$$

Vrcholy

Na predchádzajúce výpočty sme potrebovali mať vyrátané pravdepodobnosti $V_{\texttt{nestaly}}[i,j]$ a $V_{\texttt{staly}}[i,j]$. Obe predstavujú najpravdepodobnejšie rozdelenie časti sekvencie na *nestály*, resp. *stály* vrchol s k nemu pridruženými *okrajovými* vrcholmi.

Výpočet týchto pravdepodobností sa líši len v použitých stochastických modeloch DNA sekvencie, a preto ďalej budeme postupovať spoločne pre *stále* a *nestále* vrcholy s pravdepodobnosťou $V_{\texttt{x}}$. Celý vrchol sa dá rozložiť na tri časti: *okrajové* vrcholy ležiace naľavo od stredu vrcholu, stred vrcholu, ktorý je priamo *nestály* alebo *stály* vrchol a *okrajové* vrcholy ležiace napravo od stredu vrcholu.

Pri konštrukcii celého vrcholu ležiaceho na $j-i$ častiach vstupnej sekvencie máme $j-i-1$ možností[2], kde bude ležať posledný pravý *okrajový* vrchol a koľko ich bude. Ak si označím pravdepodobnosť rozdelenia vrchol s n pravými *okrajovými* vrcholmi ako V_{vpravo}, ktorá ešte neobsahuje pravdepodobnosť $P_{V3}(n)$ za pravé vloženie *okrajových* vrcholov podľa modelu, tak pravdepodobnosť celého vrcholu viem vyrátať nasledovným spôsobom:

$$V_{\mathtt{x}}[i,j] = \max\left\{V_{\text{x,vpravo}}[i,j,n]P_{V3}(n) \mid 0 \leq n \leq k-i-1\right\}$$

Pri rátaní $V_{\text{x,vpravo}}[i,j,n]$ musíme uvažovať s dvomi možnosťami. Prvou je $n > 0$. Vtedy pravdepodobnosť určím ako maximum zo všetkých možností pre koniec predchádzajúceho *okrajového* vrcholu.

Druhou je $V_{\text{x,vpravo}}[i,j,0]$, čo znamená, že tento vrchol už nemá žiadne pravé *okrajové* vrcholy. Vtedy máme $j-i$ možností, kde môže začínať stred vrcholu, a ďalšie možnosti pre počet ľavých *okrajových* vrcholov pred ním. Tu zároveň zarátame aj pravdepodobnosť P_{V3} za počet ľavých *okrajových* vrcholov.

Keď si pravdepodobnosť rozdelenia sekvencie od i-teho po j-te miesto zostrihu na n ľavých *okrajových* vrcholov označíme ako $V_{\text{x,vlavo}}$, tak $V_{\text{x,vpravo}}$ vieme rátať nasledovným spôsobom:

$$V_{\text{x,vpravo}}[i,j,n] = \begin{cases} \max\left\{V_{\text{x,vlavo}}[i,k,m]P_{V3}(m)E_{\mathtt{x,stred}}(k,j) \;\middle|\; \begin{array}{l} i \leq k < j, \\ 0 \leq m < k-i \end{array}\right\} & \text{ak } n = 0 \\ \max\left\{V_{\text{x,vpravo}}[i,k,n-1]E_{\mathtt{x,okraj}}(k,j) \mid i < k < j\right\} & \text{ak } n > 0 \end{cases}$$

Posledná tabuľka, ktorú potrebujeme vyrátať, je $V_{\text{x,vlavo}}[i,j,n]$. To spravíme podobným spôsobom ako pre pravé *okrajové* vrcholy. Pre viac ako jeden ľavý *okrajový* vrchol pravdepodobnosť vyberieme ako maximum z pravdepodobností pre riešenia $V_{\text{x,vlavo}}[i,k,n-1]$, kde $k < j$, vynásobenou pravdepodobnosťou za daný úsek podľa stochastického modelu DNA.

Pre jeden *okrajový* vrchol to bude priamo pravdepodobnosť podľa stochastického modelu DNA. A pre prípad bez ľavých *okrajových* vrcholov túto pravdepodobnosť definujeme ako 1 alebo 0, podľa toho či úsek ktorý chcem pokryť má nejakú dĺžku.

[2] $n-1$, pretože aspoň jedna časť musí byť stred vrcholu, *okrajové* vrcholy nemusia byť prítomné

Výpočet $V_{\text{x,vlavo}}$ potom môžme zapísať nasledovným výrazom:

$$V_{\text{x,vlavo}}[i,j,n] = \begin{cases} 1 & \text{ak } n = 0,\ i = j \\ 0 & \text{ak } n = 0,\ i \neq j \\ E_{\texttt{x,okraj}}(i,j) & \text{ak } n = 1 \\ \max\left\{V_{\text{x,vlavo}}[i,k,n-1]E_{\texttt{x,okraj}}(k,j) \mid i < k < j\right\} & \text{ak } n > 1 \end{cases}$$

Časová a pamäťová zložitosť

Pre nájdenie najpravdepodobnejšieho rozdelenia vstupnej sekvencie potrebujeme vypočítať viacero tabuliek. Nasledovná tabuľka sumarizuje čas na to potrebný, pričom N je stále počet miest zostrihu zo vstupu:

názov tabuľky	rozmery tabuľky	čas vyplnenia bunky v najhoršom prípade	čas vyplnenia celej tabuľky
P	1	$O(N^2)$	$O(N^2)$
P'	N^2	$O(N^2)$	$O(N^4)$
$SKOK$	N^2	$O(N)$	$O(N^2)$
$SKOK'$	N^3	$O(N^2)$	$O(N^5)$
$V_{\texttt{x}}$	N^2	$O(N)$	$O(N^2)$
$V_{\texttt{x,vpravo}}$	N^3	$O(N^2)$	$O(N^5)$
$V_{\texttt{x,vlavo}}$	N^3	$O(N)$	$O(N^4)$

Najviac času zaberie výpočet pravdepodobností v tabuľke $SKOK'$ a preto časová zložitosť je $O(N^5)$.

Pamäťová zložitosť celého algoritmu je $O(N^3)$, pretože takú pamäť niekoľko tabuliek zaberie a žiadna nepotrebuje viac.

Rekonštrukcia riešenia

Rekonštrukciu najpravdepodobnejšieho riešenia vieme spraviť rovnakým spôsobom, ako sme to robili už pri inferencii pre MZZ. Ku každej tabuľke si stačí vytvoriť jej pomocnú kópiu, v ktorej budú uložené informácie identifikujúcu možnosť, pomocou ktorej sme vybrali maximum pre danú bunku.

Týchto informácií je vždy len konštantne veľa, a preto táto úprava nezhorší pamäťovú zložitosť.

Ani časová zložitosť sa nezhorší, pretože pri výpočte bude treba navyše len zapísať tieto informácie, čo trvá konštantný čas.

Rekonštrukcia najpravdepodobnejšieho riešenia spätným prechodom cez pomocné tabuľky bude trvať $O(N)$, pretože každý z N úsekov sekvencie môže byť len jeden typ vrcholu, resp. *nevrchol* a pravdepodobnosť zaň zarátame len raz v práve jednej tabuľke.

6.5 Vlastnosti inferenčných algoritmov

V tejto kapitole sme predstavili inferenčné algoritmy pre dva nami navrhnuté modely MZV a BMM. Oba pracujú v polynomiálnej časovej zložitosti $O(N^5)$, kde N je počet miest zostrihu. Tento čas je akceptovateľný vzhľadom na priemerný počet miest zostrihu v géne.

Nespokojnosť môžu vyvolať predpovedané grafy alternatívneho zostrihu pre MZV. Z tohto modelu generatívnym procesom môže vzniknúť ľubovolný graf, no pomocou navrhnutých algoritmov sme schopný odvodiť len niektoré typy grafov. Najpravdepodobnejšia konfigurácia hrán s použitím jednoduchšej sady stochastických modelov DNA obsahovala hrany medzi každou dvojicou vrcholov, ktoré boli od seba vzdialené vhodnú vzdialenosť. Podobná situácia nastala aj pri použití jemnejších stochastických modelov, kde v závislosti od funkcie určujúcej pravdepodobnosť spojenia dvoch vrcholov hranou dostávame len určitú podtriedu grafov vygenerovateľných pomocou MZV.

Inferenčný algoritmus pre BMM týmto problémom netrpí, no na druhú stranu samotný model nepokrýva všetky možné grafy alternatívneho zostrihu. Druhou nevýhodou algoritmu

pre BMM je závislosť na stochastických modeloch DNA, ktoré sa typom sekvencie, ktorú sa snažia modelovať, líšia len málo. Existuje preto obava, že by tieto stochastické modely nepriniesli takú rozlišovaciu schopnosť, ako by sme si želali. Použitie externých stochastických modelov má aj značnú výhodu – náš algoritmus priamo profituje z ich neustáleho zlepšovania.

Kapitola 7

Záver

Našim cieľom bolo nájsť nový spôsob, ktorým by sa dal hľadať alternatívny zostrih. To sme spravili v štyroch krokoch:

1. Na grafy získané grafovou reprezentáciu alternatívneho zostrihu sme sa pozreli ako na produkt jednoduchého stochastického procesu, čo je doteraz nepoužívaný spôsob pohľadu na tento jav.

2. Navrhli sme dva generatívne modely pre graf alternatívneho zostrihu. Prvý z nich je veľmi jednoduchý. Druhý je komplikovanejší, ale motivovaný reálnymi dátami.

3. Pre jeden z modelov sme uviedli spôsoby ako natrénovať jeho parametre a vyhodnotili sme jeho (ne)vhodnosť porovnaním so skutočnými dátami.

4. Pre obidva modely sme vymysleli netriviálne inferenčné algoritmy, pomocou ktorých dokážeme za splnenia určitých predpokladov predpovedať alternatívny zostrih vo vstupnej sekvencii. Tieto algoritmy sú založené na princípe dynamického programovania a oba pracujú v polynomiálnej časovej zložitosti, čo je vzhľadom na veľkosti dát s ktorými sa pracuje nevyhnutné.

Táto práca otvára možnosti pre ďalší výskum v tejto oblasti. Jedna z možných ciest, ktorou sa dá vydať, je porovnať nami navrhnutý spôsob hľadania alternatívneho zostrihu

s už existujúcimi programami. Je to značne náročná úloha, pretože na jej splnenie je nutné natrénovať použité modely, pripraviť dáta, ktoré naše algoritmy potrebujú, a zvoliť vhodný podproblém všeobecného hľadania alternatívneho zostrihu, ktorý budú vedieť riešiť aj existujúce algoritmy. Druhou možnosťou je hlbšie preskúmať biologicky motivovaný model jeho natrénovaním a porovnaním so skutočnými dátami.

Literatúra

[AS06] Jonathan E. Allen and Steven L. Salzberg. A phylogenetic generalized hidden markov model for predicting alternatively spliced exons. *Algorithms for Molecular Biology*, 1, 2006.

[Ast04] G. Ast. How did alternative splicing evolve? *Nature Reviews Genetics*, 5(10):773–782, 2004.

[BSS01] M. Burset, I. A. Seledtsov, and V. V. Solovyev. SpliceDB: database of canonical and non-canonical mammalian splice sites. *Nucleic acids research*, 29(1):255–259, 2001.

[CLRS01] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Second Edition*, chapter 24, pages 588–592. The MIT Press, 2001.

[CPL97] Jean-Michel Claverie, Olivier Poirot, and Fabrice Lopez. The difficulty of identifying genes in anonymous vertebrate sequences. *Computers & Chemistry*, 21(4):203 – 214, 1997. Open Problems of Computational Molecular Biology.

[FF03] L. Fedorova and A. Fedorov. Introns in gene evolution. *Genetica*, 118(2):123–131, 2003.

[FRZ+10] Pauline A. Fujita, Brooke Rhead, Ann S. Zweig, Angie S. Hinrichs, Donna Karolchik, Melissa S. Cline, Mary Goldman, Galt P. Barber, Hiram Clawson, Antonio Coelho, Mark Diekhans, Timothy R. Dreszer, Belinda M. Giardine, Rachel A.

Harte, Jennifer Hillman-Jackson, Fan Hsu, Vanessa Kirkup, Robert M. Kuhn, Katrina Learned, Chin H. Li, Laurence R. Meyer, Andy Pohl, Brian J. Raney, Kate R. Rosenbloom, Kayla E. Smith, David Haussler, and W. James Kent. The ucsc genome browser database: update 2011. *Nucleic Acids Research*, 2010.

[HAS+02] Steffen Heber, Max Alekseyev, Sing-Hoi Sze, Haixu Tang, and Pavel A. Pevzner. Splicing graphs and est assembly problem. *Bioinformatics*, 18(suppl 1):S181–S188, 2002.

[J+03] Jason M. Johnson et al. Genome-wide survey of human alternative pre-mrna splicing with exon junction microarrays. *Science*, 302(5653):2141–2144, 2003.

[KLMA10] Hadas Keren, Galit Lev-Maor, and Gil Ast. Alternative splicing and evolution: diversification, exon definition and function. *Nature reviews. Genetics*, 11(5):345–345, 2010.

[Mar63] Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):pp. 431–441, 1963.

[MPNG08] Abigail McGuire, Matthew Pearson, Daniel Neafsey, and James Galagan. Cross-kingdom patterns of alternative splicing and splice recognition. *Genome Biology*, 9(3):R50, 2008.

[RT03] Luis Javier Rodríguez and Inés Torres. *Pattern Recognition and Image Analysis*, chapter Comparative Study of the Baum-Welch and Viterbi Training Algorithms Applied to Read and Spontaneous Speech Recognition, pages 847–857. Springer Berlin / Heidelberg, 2003.

[RW80] J. Rogers and R. Wall. A mechanism for rna splicing. *Journal of molecular evolution*, 77(4):1877–1879, 1980.

[SKG+06] Mario Stanke, Oliver Keller, Irfan Gunduz, Alec Hayes, Stephan Waack, and Burkhard Morgenstern. Augustus: ab initio prediction of alternative transcripts. *Nucleic Acids Research*, 34:W435–W439, 2006.

[SRJM02] S. Sonnenburg, G. Rätsch, A. Jagota, and K.R. Müller. New methods for splice site recognition. *Artificial Neural Networks—ICANN 2002*, pages 793–793, 2002.

[Sta03] Mario Stanke. *Gene Prediction with a Hidden Markov Model*. PhD thesis, The Georg-August-Universität Göttingen, 2003.

[WB08] Zefeng Wang and Christopher B. Burge. Splicing regulation: From a parts list of regulatory elements to an integrated splicing code. *RNA*, 14(5):802–813, 2008.