Porovnanie gene finderov

Metódy v bioinformatike - projekt

Peter Kováč
kovac77@compbio.fmph.uniba.sk

11. januára 2010

Abstrakt

Programy hľadajúce gény hrajú dôležitú úlohu pri skúmaní genómov. Porovnal som štyri gene findery na DNA sekvencii z drozofily (*Drosophila melanogaster*): Augustus [4], Exonhunter [1], GeneID [6] a GeneMark [8][9]. Tri z nich, Augustus, ExonHunter a GeneID, som porovnal aj s použitím zarovnania EST sekvencií na genóm.

1 Úvod

V mojej bakalárskej práci sa venujem programu ExonHunter. Prirodzene ma zaujíma, ako si ExonHunter vedie v porovnaní s inými programami. Navyše, po vykonaní každej zmeny si treba overiť, či táto zmena priniesla nejaký úžitok. Jednou takouto zmenou je aj možnosť použitia programu Blat na spracovanie EST sekvencií. Týmto projektom som sa pokúsil zodpovedať prvú otázku a získal skúsenosti a základný rámec pre neskoršie porovnanie so zmeneným ExonHunterom.

Hľadanie génov (alebo predpovedanie) sa vo všeobecnosti dá rozdeliť na tzv. *ab initio* hľadanie a hľadanie pomocou externých dát. Oba typy programov majú ako vstup DNA sekvenciu a ich výstupom je anotácia: kde sú exóny, intróny, štart/stop kodóny, niekedy aj acceptor/donor signály. Pravdepodobnostné modely (skryté Markovovské modely - Augustus, ExonHunter, GeneMark; *positional weight matrices* - GeneID) v týchto programoch vyžadujú trénovanie na už oanotovaných kúskoch genómu - niekoľko stoviek génov (Augustus, ExonHunter, GeneID). Výnimkou je GeneMark, ktorý používa samotrénovanie (self-training), vstupná DNA sekvencia však musí byť dostatočne dlhá (aspoň 10MB).

Programy pracujúce aj s externými dátami fungujú podobne, no svoje predikcie dokážu korigovať na základe externých informácií. Takými to informáciami sú napr. zarovnanie genómu s EST sekvenciami, zarovnanie genómu s proteínmi, zarovnanie genómu k inému genómu. Nevýhodu je, že

tieto externé dáta nie sú dostupné pre novosekvenované organizmy, výhodou je, že dokážu byť presnejšie.

2 Dáta

Pracoval som dátami predtým použitými pri trénovaní programu ExonHunter [17]. Boli to DNA sekvencie chromozómu 2 (L a R) drozofily [18]. Na vyhodnotenie bola použitá anotácia z UCSC genome annotation databázy [19]. Podľa tejto anotácie, 2L obsahuje 2449 génov s 3630 transkriptmi obsahujúcimi 16594 kódujúcich úsekov, 2R obsahuje 2712 génov s 4240 transkriptmi obsahujúcimi 21898 kódujúcich úsekov. Tieto dáta boli rozdelené na testovaciu množinu (tú som použil pri porovnaní všetkých programov) a trénovaciu množinu pre ExonHunter. Testovacia množina (sekvencia [22], anotácia [23]) obsahuje 1796 génov s 2681 transkriptami obsahujúcimi 7681 exónov (16 MB).

Pri porovnaní výsledkov programov s použitím EST boli použité dáta z DFCI Gene Index [20], dokopy 42538 EST sekvencii [24] vo formáte FASTA (62 MB).

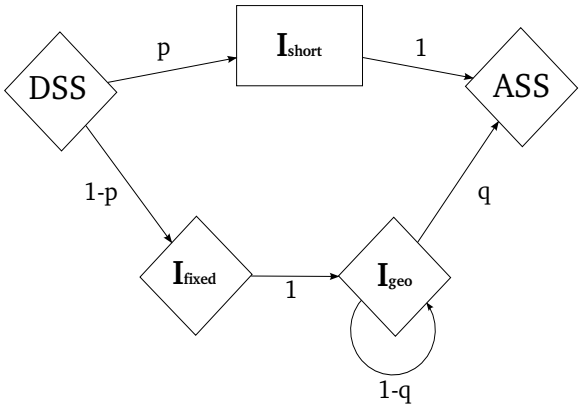
3 Programy

Porovnával som s programami na hľadanie génov Augustus, ExonHunter, GeneID, GeneMark a pri porovnaní s EST dátami aj s programami RepeatMasker [12] na maskovanie opakujúcich sa úsekov v genóme a Blat [13] na zarovnanie EST sekvencií ku genómu. Pôvodne som medzi gene findery chcel zaradiť aj program GeneZilla [10][11], avšak autori na webe neuvádzajú natrénované parametre pre drozofilu a na žiadosť o ich poskytnutie neodpísali. Namiesto GeneZilly som do porovnania zahrnul GeneID.

3.1 Augustus

Augustus [4] stavia na viacerých matematických modeloch. Autori zdôrazňujú hlavne modelovanie dĺžok intrónov. Ich skrytý Markovovský model v každom stave emituje reťazec DNA v podstate náhodnej dĺžky. Rozdelenie dĺžok a prechodové pravdepodobnosti sa určia trénovaním. Používajú pri tom ďalšie matematické modely: Markovovské reťazce, *windowed weight array model*, interpolované Markovovské modely a ich vlastnú metódu nazvanú *similarity weighting of sequence patterns*.

Intróny modelujú stavy na obrázku 1. Autori kombinujú stav s daným rozdelením dĺžok kratšími ako d a stavy modelujúce dlhšie intróny, kde dĺžka časti dlhšej ako d má geometrické rozdelenie. Dosahujú tým väčšiu presnosť, pričom príliš nestrácajú výpočtový výkon.



Obrázok 1: Modelovanie intrónov v HMM programu Augustus [4]. Stav DSS znamená *donor splice site*, ASS *acceptor splice site*, I_{short} modeluje krátke intróny a I_{fixed} s I_{geo} dlhé intróny.

Ak má intrón dĺžku najviac d, zodpovedajúca cesta vedie cez stav I_{short}, ak má dĺžku viac ako d, cesta vedie najskôr cez stav I_{fixed} a potom $l - d$ krát cez stav I_{geo} do stavu ASS.

Autori udávajú, že pre drozofilu sú zodpovedajúce hodnoty parametrov približne $q \approx 1/4894, p \approx 0.78, d = 929$.

Výstupom programu Augustus je anotácia s najväčšou *a-posteriori* pravepodobnosťou, nájdená Viterbiho algoritmom.

3.2 ExonHunter

ExonHunter [1] sa snaží využiť čo najviac dostupných dát – okrem samotnej DNA sekvencie aj EST zarovnania, proteíny, vzájomné porovnanie s iným genómom a repetitívne sekvencie. Kombinuje ich do hierarchie ,,radcov", pravdepodobnostného rozdelenia štrukturálnych prvkov genómu. Používa pri tom kvadratické programovanie, ktoré pri kombinovaní dokáže nájsť konsenzus aj v prípadoch, keď si dáta z rôznych zdrojov odporujú.

Program využíva skrytý Markovovský model. Od iných programov sa odlišuje modelovaním GC obsahu na základe ,,okna" veľkosti 1000 báz okolo aktuálnej pozície (keďže aj v rámci jedného génu sa podiel GC obsahu môže výrazne líšiť v kódujúcich a nekódujúcich úsekoch); modelovaním *donor* a *acceptor* signálov pomocou HOT rádu 2 [2]; modelovaním rozdelenia dĺžok exónov a intrónov ako hlavu (s ľubovoľým rozdelením) a geometricky zmenšujúcim sa chvostom [3] (podobná metóda je použitá v programe Augustus na modelovanie dĺžok intrónov) a modelovaním medzigénových úsekov stavom negeometricky generujúcim počet k-tic a stavom rovnomerne generujúcim dĺžky od 1 po k, čím sa dosahuje negeometrické rozdelenie dĺžok medzigénových úsekov.

Dáta z externých zdrojov spojí ExonHunter do superradcu, ktorý potom ovplyvňuje Viterbiho algoritmus hľadajúci najpravdepodobnejšiu anotáciu v HMM prenásobením emisných pravdepodobností.

Ak nie sú k dispozícii žiadne externé dáta, ExonHunter sa správa ako štandardný *ab initio* program, predikujúci štruktúru sekvencie len na základe skrytého Markovovského modelu.

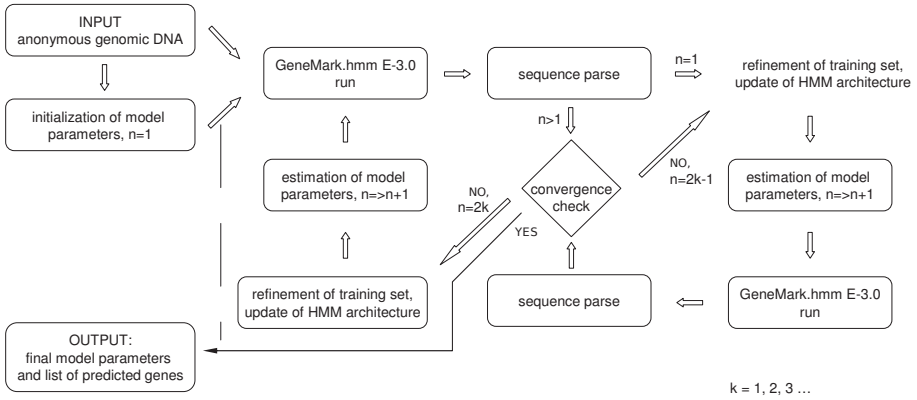
3.3 GeneID

GeneID [6] bol jedným z prvých programov umožňujúcich predikciu štruktúry sekvencie DNA stavovcov. Používal hierarchickú štruktúru – najskôr predikoval signály (štart a stop kodóny, miesta zostrihu) pozdĺž celej sekvencie, potom na základe tejto predikcie zostavil exóny a napokon optimálnu anotáciu. Novšia verzia Funguje tiež hierarchicky avšak skóre pre signály počíta v logaritmickej mierke, signály pre daný exón sčíta, a potom dynamickým programovaním [7] prehľadá priestor exónov a zostaví z nich anotáciu (pričom maximalizuje celkové skóre).

Autori zdôrazňujú, že výhodou ich programu je veľmi dobrá časová a pamäťová náročnosť.

3.4 GeneMark

GeneMark.hmm ES-3.0 [8][9] sa od ostatných programov významne líši najmä spôsobom, akým získava parametre pre svoj skrytý Markovovský model. Zatiaľ čo napríklad ExonHunter sa sústredí na čo najlepšie využitie všetkých zdrojov informácií, GeneMark sa spolieha len na minimum informácií – vstupnú sekvenciu. To ho predurčuje na hľadanie génov v čerstvo osekvenovaných organizmoch, pri ktorých dodatočné údaje absentujú. Vďaka it-



Obrázok 2: Diagram popisujúci jednotlivé kroky trénovacieho algoritmu programu GeneMark.hmm ES-3.0 [9].

eratívnemu trénovaniu spojenému s predikciou génov GeneMark nepotrebude extra oanotovanú trénovaciu sekvenciu. V prvom kroku algoritmu inicializuje parametre skrytého polo-Markovovského modelu (HSMM); v druhom kroku použije GeneMark.hmm E-3.O na anotáciu podľa aktuálnych parametrov; v treťom kroku použije vybrané anotácie na úpravu parametrov modelu. Kroky dva a tri sa opakujú, až kým sa nedosiahne konvergencia v biologicky relevantnom stave. Práve obmedzenie v treťom kroku zaručuje, že konvergencia nastane. Schéma celého algoritmu je zachytená na Obrázku 2.

Autori uvádzajú, že tento prístup vyžaduje veľkosť vstupnej sekvencie zhruba 10 MB, hoci presná veľkosť záleží od konkrétneho organizmu a štruktúry jeho genómu. Sekvencie výrazne presahujúce 10 MB už nepriniesli výrazné zlepšienie, ale vyžadovali väčší počet iterácií.

3.5 RepeatMasker

RepeatMasker [12] je program, ktorý v DNA sekvenciách vyhľadáva opakujúce sa úseky a ,,maskuje" ich písmenom `N` (ostatné programy v sekvenciách ignorujú znaky rôzne od `A`, `C`, `G`, `T`). V tomto projekte je použitý len ako podprogram procedúry `prepare-evidence`, upravujúcej EST sekvencie do vstupov pre programy Augustus, ExonHunter a GeneID.

3.6 Blat

Blat [13] je program slúžiaci na zarovnávanie sekvencií. V tomto projekte je použitý len ako podprogram procedúry `prepare-evidence`, upravujúcej EST sekvencie do vstupov pre programy Augustus, ExonHunter a GeneID.

3.7 Sim4

Sim4 [14] rovnako ako Blat slúži na zarovnávanie sekvencií. V tomto projekte je použitý len ako podprogram procedúry `prepare-evidence`, upravujúcej EST sekvencie do vstupu pre ExonHunter.

4 Postup

V nasledujúcom popíšem príklady použitia jednotlivých programov. Ešte predtým však uvediem postup, ktorým som získal zoznam génov v testovacej množine - ak by sa totiž testovacia množina významne prekrývala s trénovacou množinou niektorého z programov, mohlo by to až príliš ovplyvniť výsledky v prospech daného programu a porovnanie by stratilo výpovednú hodnotu.

4.1 Gény v testovacej množine

Anotácia testovacej množiny [23] je vo formáte GTF. Posledným poľom je *gene_name*, ktorého hodnota je názov génu, ktorému prislúcha daný záznam. Jednoduchým skriptom som získal zoznam týchto názvov [25]:

```
cd /projects/eh-dev/peter/gene-finders/trainsets/
perl -nle 'print $1 if /gene_name "(.+)"/' ../test.gtf | sort |
        uniq >gene_names
```

Keďže program Augustus mal dáta o génoch v trénovacích množinách ako FlyBase identifikátory, použil som nástroj FlyBase ID Converter [21] a exportovaním voľbou *file, uniq IDs only* som získal zoznam FlyBase identifikátorov pre tieto gény [26].

4.2 Hľadanie s pomocou EST databázy

Tri zo štyroch programov dokážu zlepšovať svoje predikcie s pomocou externých dát (Augustus, ExonHunter, GeneID). Zaujímalo ma, o koľko sa ich výsledky zlepšia, keď im „pomôžem" informáciami z EST. EST sekvencie sa získavajú sekvenovaním cDNA komplementárnej k mRNA reprezentujúcej exprimovaný gén, zväčša pozostávajúci z viacerých exónov. Aby s týmito informáciami vedeli gene findery pracovať, treba ich časti zodpovedajúce exprimovaným exónom zarovnať k pôvodnému genómu – použitím programu Blat – a previesť jeho výstup do formátu GTF. Aby však Blat nezarovnával opakujúce sa sekvencie, treba ich zamaskovať - programom RepeatMasker. Keďže rovnakú procedúru treba vykonať pri každom programe, využil som už existujúcu implementáciu, ktorá je súčasťou ExonHuntera – `prepare-evidence`. Jej použitie v prípade ExonHuntera je priamočiare. Pre potreby programov Augustus a GeneID je ešte potrebné jej výstup (vo formáte GTF) upraviť do GFF, kde posledný stĺpec má tvar *source=E* (čo znamená, že informácia určuje exón):

```
cd /projects/eh-dev/peter/gene-finders/results/
prepare-evidence --dir hints ../test.fa drosophila repeat
prepare-evidence --dir hints --set PROGRAM_EST=blat ../test.fa drosophila
        dromel-est
perl -nle 'print "$1\t$2\t$3\t$4\t$5\t$6\t$7\t$8\tsource=E"
        if /(\S+)\s+(\S+)\s+(\S+)\s+(\S+)\s+(\S+)\s+(\S+)\s+(\S+)\s+(\S+)\s+info_name/'
        hints/dromel-est.gtf >hints/dromel-est-hint.gff
```

4.3 Augustus

4.3.1 Prienik trénovacej a testovacej množiny

Autori uvádzajú, že pre drozofilu Augustus trénovali na množinách *fly* a *adh* [5]. Zoznam génov z týchto množín som získal dvomi spôsobmi. Prvým som získal FlyBase identifikátory:

```
cd /projects/eh-dev/peter/gene-finders/trainsets/augustus
wget http://augustus.gobics.de/datasets/fly.train.gb.gz
wget http://augustus.gobics.de/datasets/adh.train.gb.gz
gunzip fly.train.gb.gz
gunzip adh.train.gb.gz
perl -nle 'print $1 if /LOCUS\s+(\w+)/' adh.train.gb
        fly.train.gb | sort | uniq >augustus_flybase_ids
```

Druhým som sa pokúsil získať priamo názvy génov (aktuálny priečinok a vstupné dáta sú rovnaké ako v predchádzajúcom skripte):

```
perl -nle 'print $1 if /gene="(.+)"/' adh.train.gb fly.train.gb
        | sort | uniq >augustus_genes
```

Zatiaľ čo prvý zoznam obsahuje 414 FlyBase identifikátorov, druhý obsahuje 232 názvov. Niektoré záznamy (napr. FBgn0000052) nemajú vo svojom popise explictne uvedený názov génu, čo vysvetľuje menší počet názvov génov ako FlyBase identifikátorov. Zaujímavé je, že autori uvázdajú, že *fly* obsahuje 320 génov a *adh* 400 génov. Vysvetlenie je však jednoduché, *fly* aj *adh* sa výrazne prekrývajú - obsahujú až 306 záznamov so spoločnými FlyBase indentifikátormi.

Následne som oba zoznamy porovnal so zodpovedajúcimi zoznamami z testovacej množiny. Výsledkom sú dva spoločné zoznamy, jeden pre FlyBase identifikátory [28] (84 spoločných identifikátorov), druhý pre názvy génov [29] (44 spoločných názov).

4.3.2 *Ab initio* predikcia

Nastavenia parametrov povoľovali predikciu neúplných génov na hraniciach sekvencií a nepovoľovali predikciu prekrývajúcich sa génov na opačných vláknach (východzie hodnoty). Ukážkové spustenie programu vyzeralo takto:

```
augustus --species=fly /projects/eh-dev/peter/gene-finders/test.fa
        >out_augustus.gff
```

Takto spustený program bežal na jednom z CPU servrov zhruba 52 minút (čas behu je len orientačný, pretože som nemohol zabezpečiť rovnaké podmienky pre beh každého programu – kolegovia mali na servroch spustené vlastné programy).

4.3.3 Predikcia s EST dátami

Pre správne fungovanie Augustus potrebuje nastavíť konfiguračný súbor [30]. Východzí konfiguračný súbor je nastavený tak, že ignoruje externé zdroje dát (aj keby sme mu ich v parametroch zadali). Ja som v ňom spravil jedinú zmenu – ak Augustus predpovie exón s rovnakými súradnicami ako

daný hint, takýto exón dostane trojnásobné skóre. Keďže skóre je tu vyjadrené ako pravdepodobnosť, treba ešte vykonať drobnú zmenu v súbore s EST dátami a predeliť skóre 100. Ukážkové spustenie programu potom vyzerá takto:

```
augustus --species=fly --hintsfile=hints/dromel-est-hint.gff
        --extrinsicCfgFile=../augustus/config/extrinsic.cfg
        ../test.fa >out_augustus_hint.gtf
```

Za povšimnutie stojí čas behu (hoci opäť len orientačný): 1707 minút!

4.4 ExonHunter

Keďže ExonHunter bol natrénovaný inými časťami chromozómu 2 [17], prienik s testovacou množinou je prázdny.

4.4.1 *Ab initio* predikcia

Ukážka použitia programu ExonHunter:

```
exonhunter --dir ehtemp ../test.fa drosophila >out_exonhunter.gtf
```

Program bežal približne 121 minút.

4.4.2 Predikcia s EST dátami

Pred spustením s informáciami z externých zdrojov dát treba spustiť procedúru `prepare-evidence`, ktorá spracuje tieto dáta do vhodnej podoby. V tomto projekte som EST sekvencie spracoval v programoch RepeatMasker a potom Blat alebo Sim4.

```
prepare-evidence --dir ehtemp-est ../test.fa drosophila repeat
prepare-evidence --dir ehtemp-est --set PROGRAM_EST=sim4 ../test.fa
        drosophila dromel-est
exonhunter ../test.fa drosophila >out_exonhunter_hint.gtf
# alebo
prepare-evidence --dir ehtemp-est ../test.fa drosophila repeat
prepare-evidence --dir ehtemp-est --set PROGRAM_EST=blat ../test.fa
        drosophila dromel-est
exonhunter ../test.fa drosophila >out_exonhunter_hint.gtf
```

Program Blat pracuje výrazne rýchlejšie ako Sim4.

4.5 GeneID

4.5.1 Prienik trénovacej a testovacej množiny

Na overenie prieniku trénovacej a testovacej množiny som použil podobný postup ako v prípade programu Augustus. Bohužiaľ, nepodarilo sa mi s istotou získať aktuálne trénovacie dáta. Môžem len dúfať, že jediné dáta, ktoré autori zverejnili [6] (z roku 2000), boli použité aj na trénovanie aktuálnej verzie programu (z roku 2009).

```
cd /projects/eh-dev/peter/gene-finders/trainsets/geneid
wget http://www.fruitfly.org/seq_tools/datasets/Drosophila/multi_exon_GB.dat.gz
wget http://www.fruitfly.org/seq_tools/datasets/Drosophila/single_exon_GB.dat.gz
gunzip multi_exon_GB.dat.gz
gunzip single_exon_GB.dat.gz
perl -nle 'print $1 if /gene="(.+)"/' multi_exon_GB.dat single_exon_GB.dat
        | sort | uniq >geneid_genes
```

Výsledkom je 40 spoločných génov [31].

4.5.2 *Ab initio* predikcia

```
../geneid/bin/geneid -3 -P ../geneid/param/drosophila.param
        ../test.fa >out_geneid.gff3
gff3_to_gtf.pl out_geneid.gff3 >out_geneid.gtf
```

GeneID naozaj bežal výrazne rýchlejšie ako ostatné programy, stačili mu necelé tri minúty.

4.5.3 Predikcia s EST dátami

Rovnaký súbor ako pre Augustus [27] mal vylepšiť predikcie aj v prípade programu GeneID.

```
cd /projects/eh-dev/peter/gene-finders/results/
../../geneid/bin/geneid -3 -R dromel-est-hint.gff -P
        ../../geneid/param/drosophila.param ../../test.fa
        >out_geneid_hint.gff3
```

Avšak GeneID zjavne ignoruje EST dáta, pravdepodobný dôvod je ich zlý formát – riadky označujúce exóny majú ako tretie pole hodnotu *exon*, namiesto konkrétnejších hodnôt *Initial, Internal, Terminal, Single*, ktoré používa GeneID.

4.6 GeneMark

GeneMark ako jediný porovnávaný program nepotrebuje extra trénovaciu množinu. Navyše, nepodporuje predikciu s externými zdrojmi dát, preto jediné použitie bolo nasledovné:

```
../genemark/gm_es.pl ../test.fa >& log_genemark
```

5 Porovnanie predikcií

Na porovnanie som použil skript `evaluate.pl` [32]. Všetky výsledky sú v nezmenenej podobe v prílohe. Z porovnania v tabuľke 1 vyšiel ako ,,víťaz"

Tabuľka 1: Porovnanie predikcií

	Aug	Aug. +EST	EH	EH +EST (Blat)	EH +EST (Sim4)	GID	GID +EST	GM
Gene Sn.	49.50%	**63.86%**	41.54%	51.95%	54.34%	32.35%	32.35%	46.71%
Gene Sp.	51.99%	**65.45%**	45.57%	49.29%	50.28%	32.86%	32.86%	37.60%
Transcript Sn.	38.64%	**51.59%**	31.97%	39.43%	40.77%	25.85%	25.85%	36.55%
Transcript Sp.	51.99%	**59.10%**	45.57%	49.29%	50.28%	32.86%	32.86%	37.60%
Exon Sn.	69.12%	**82.52%**	72.53%	77.37%	77.22%	65.58%	65.58%	71.72%
Exon Sp.	79.81%	**82.64%**	70.44%	73.11%	73.30%	66.08%	66.08%	64.23%
Nucleotide Sn.	88.26%	94.89%	95.22%	**96.34%**	94.81%	88.92%	88.92%	94.29%
Nucleotide Sp.	**96.93%**	94.40%	89.56%	92.95%	92.86%	95.07%	95.07%	88.31%

Aug - Augustus, *EH* - ExonHunter, *GID* - GeneID, *GM* - GeneMark. Maximum v riadku je zvýraznené.

Augustus. Keďže však existuje istý prienik s jeho trénovacou množinou s testovacou množinou, jeho výsledky sú napríklad v porovnaní s programom ExonHunter, ktorý nemá spoločné gény trénovacej a testovacej množiny, trochu skreslené.

Zaujímavo vyzerá porovnanie použitia iných programov pri spracúvaní externých dát pri predikcii ExonHunter-om. Blat je rýchlejší, ale výsledky má trochu horšie. V tomto prípade je možné, že vplyv na výsledky má aj konverzia z výstupných formátov programov Sim4 a Blat do formátu GTF.

Hoci som v prípade programu GeneID tiež našiel spoločné gény trénovacej a testovacej množiny, na výsledkoch sa to neprejavilo. Je preto možné, že dáta zverejnené autormi programu GeneID v roku 2000 nepoužili na trénovanie najnovšej verzie.

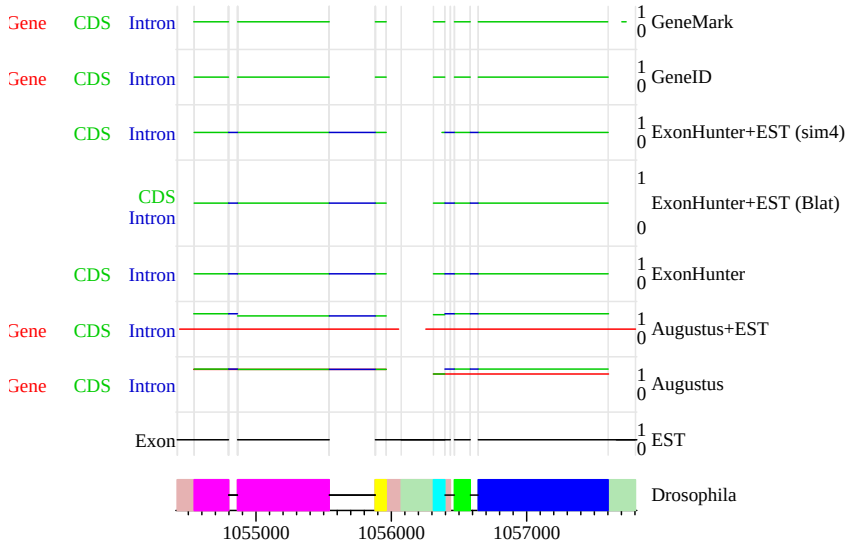
Výsledky programu GeneMark ukazujú, že *ab-initio* predikciu anotácie je možné vykonať aj bez použitia experimentálne overených génov na trénovanie. Odpoveď na otázku ako sa zmení predikcia, keď máme o genóme informácie

Tabuľka 2: Zmena výsledkov pri použití EST

	Augustus +EST	ExonHunter +EST (Blat)	ExonHunter +EST (Sim4)
Gene Sn.	+14.36%	+10.41%	+12.80%
Gene Sp.	+13.46%	+3.72%	+4.71%
Transcript Sn.	+12.95%	+7.46%	+8.80%
Transcript Sp.	+7.11%	+3.72%	+4.71%
Exon Sn.	+13.40%	+4.84%	+4.69%
Exon Sp.	+2.83%	+2.67%	+2.86%
Nucleotide Sn.	+6.63%	+1.12%	-0.41%
Nucleotide Sp.	-2.53%	+3.39%	+3.30%

Najviac z dodatočných dát vyťažil Augustus.

v podobe EST databázy dáva tabuľka 2. Do porovnania som nezahrnul program GeneID, keďže sa mi ho nepodarilo nakonfigurovať tak, aby pracoval s externými zdrojmi dát. Z dvojice Augustus–ExonHunter si väčšie prírastky pripísal Augustus. Najviac je rozdiely vidieť pri špecificite predikcie génov a senzitivite predikcie exónov.



Obrázok 3: V tomto prípade ani dodatočná informácia nestačila na preklenutie medzery.

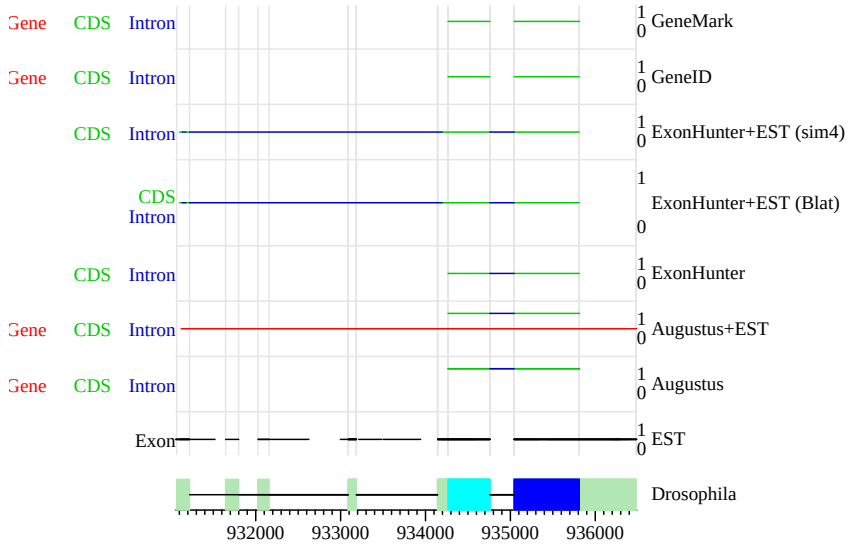
Pre iný pohľad na anotácie som použil program Mikroskop [15]. Je zaujímavé sledovať, ako dokáže dodatočná informácia napomôcť predikcii (obrázok 5), hoci nie vždy (obrázok 4).

Za povšimnutie stojí aj fakt, že zatiaľ čo anotácia podľa UCSC databázy má počet transkriptov na gén v priemere 1.49, všetky programy majú vo výstupoch jeden transkript na jeden gén (s výnimkou programu Augustus pri predikcií s EST sekvenciami, kedy je tento pomer 1.12). Programy teda nepredikujú alternatívne transkripty.

6 Záver

Porovnal som niekoľko programov na hľadanie génov metódami *ab initio* a použitím externých zdrojov dát. Najlepšie výsledky dosahoval program Augustus, no mohli byť ovplyvnené trénovacou množinou. Porovnanie ukázalo, že keď máme o genóme informácie v podobe EST sekvencií, dokážeme predpovedať s väčšou presnosťou.

Ako problém sa pri projekte ukázalo veľké množstvo odlišných formátov používaných programami. Vyhodnocovací skript totiž dokáže spracovať len



Obrázok 4: Viaceré EST zarovnania označil ExonHunter ako intrón.

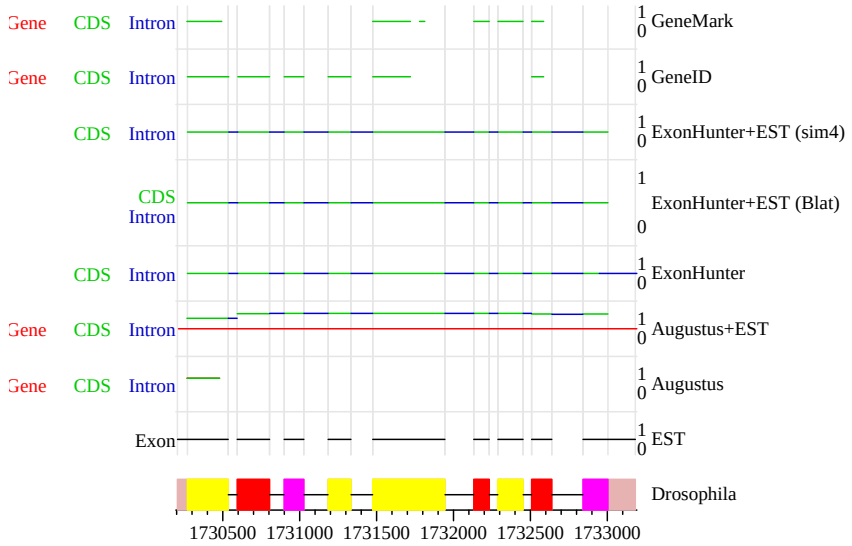
dobre formátované GTF (so správne nastavenými poliami *gene_id* a *transcript_id*). Napr. program GeneID takýto výstupný formát nemá, musí sa spraviť konverzia z GFF3 alebo iného (jeden z nich produkoval súbory veľkosti 1.6 GB!). Pri každej konverzii sa môžu vyskytnúť chyby, prípadne sa nejaké dáta zahodia. To tiež môže skresliť výsledky.

Asi najvypuklejšie sa tento problém prejavil pri mojej snahe poskytnúť programu GeneID dáta z EST sekvencií. Keďže však v spracovanom súbore s EST dátami neboli exóny rozdelené na začiatočné, vnútorné, koncové a samostatné, GeneID tieto dáta ignoroval.

Konkrétne čísla tiež závisia od vyhodnocovacieho programu. Pokúsil som sa nájsť aj alternatívy, no neúspešne (narazil som len na jednu diplomovú prácu, avšak priložený program na mojich dátach padal).

Ďalšia práca môže byť zameraná na zlepšenie celkového sledu práce (napríklad automatizácia pri porovnaní viacerých sekvencií, resp. organizmov), vychytanie múch pri konverzii a nastavení súborov s parametrami (napr. Augustus), a zahrnutie aj iných zdrojov dát (napr. informácia o homológoch).

Prácou na projekte som si zlepšil znalosti o Linuxovom prostredí (napríklad pri prvotnom zisťovaní prienikov trénovacej množiny programu Augustus a testovacej množiny som používal vlastné Pythonovské skripty, neskôr som prišiel na to, že rovnaky efekt sa dá docieliť jednoriadkovým príkazom v Perl-e) a o bioinformatických databázach (pri zisťovaní FlyBase identifikátorov génov).



Obrázok 5: EST dáta výrazne pomohli programu Augustus, ExonHunter tu dodatočné informácie nepotreboval.

Referencie

[1] Brejová, B., Brown, D., Li, M., Vinař, T., (2005) *ExonHunter: a comprehensive approach to gene finding*, Bioinformatics, Vol. 21, Suppl. 1 2005, i57-i65, DOI: 10.1093/bioinformatics/bti1040

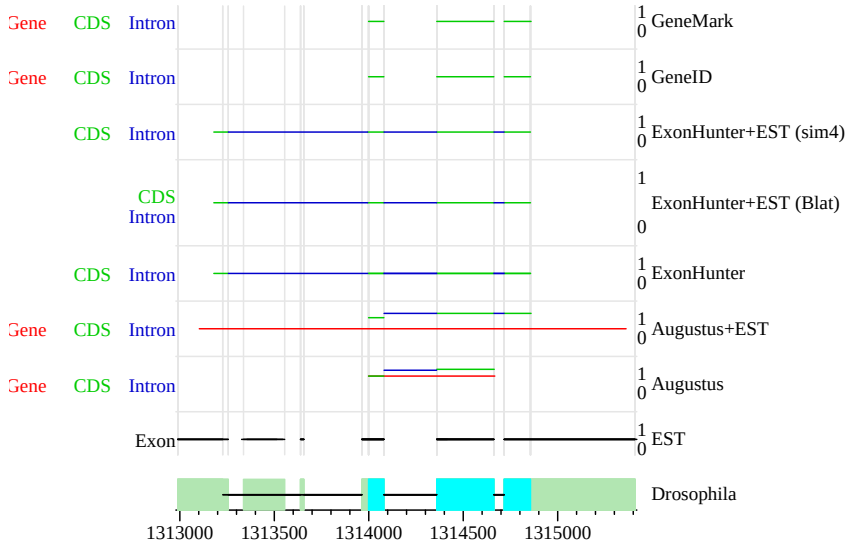
[2] Brejová, B., Brown, D., Vinař, T., (2003) *Optimal DNA signal recognition models with a fixed amount of intrasignal dependency*, Algorithms and Bioinformatics: 3rd International Workshop (WABI), vol. 2812 of LNBI, pp. 78-94

[3] Brejová, B., Vinař, T., (2003) *A better method for length distribution modeling in HMMs and its application to gene finding*, Combinatorial Pattern Matching (CPM), vol. 2373 of LNCS, pp. 190-202

[4] Stanke, M. a Waack, S., (2003) *Gene prediction with a hidden Markov model and new intron submodel*, Bioinformatics, Vol. 19, Suppl. 2 2003, ii215-ii225, DOI: 10.1093/bioinformatics/btg1080

[5] Augustus: datasets
http://augustus.gobics.de/datasets/

[6] Parra G., Blanco E., Guigó R., (2000) *GeneID in Drosophila*, Genome Res. 2000 10: 511-515 DOI: 10.1101/gr.10.4.511



Obrázok 6: V tomto prípade Augustus po pridaní dodatočnej informácie označil celý úsek ako gén, ale nevytýčil hranice exónov a intrónov.

[7] Guigó R., (1998) *Assembling genes from predicted exons in linear time with dynamic programming*, Journal of Computational Biology, 5: 681-702

[8] Lomsadze, A., Ter-Hovhannisyan V., Chernoff, Y., Borodovsky, M., (2005) *Gene identification in novel eucaryotic genomes by self-training algorithm*, Nucleic Acids Research, 2005, Vol. 33, No. 20, 6494-6506, DOI: 10.1093/nar/gki937

[9] Ter-Hovhannisyan V., Lomsadze, A., O.Chernoff, Y., Borodovsky, M., (2008) *Gene prediction withc in novel fungal genomes using an ab initio algorithm with unsupervised training*, Genome Res. 2008 18: 1979-1990 DOI: 10.1101/gr.081612.108

[10] Majoros W., Pertea M., Salzberg S., (2004) *TIGRscan and GlimmerHMM: two open-source ab initio eukaryotic gene finders*, Bioinformatics, Vol. 20 no. 16 2004, pages 2878–2879 DOI:10.1093/bioinformatics/bth315

[11] Majoros W., Pertea M., Delcher A., Salzberg S., (2005) *Efficient decoding algorithms for generalized hidden Markov model gene finders*, BMC Bioinformatics 2005, 6:16, DOI:10.1186/1471-2105-6-16

[12] Smit, AFA, Hubley, R & Green, P. RepeatMasker Open-3.0. 1996-2004 http://www.repeatmasker.org

[13] Kent, J. (2002) *BLAT – The BLAST-Like Alignment Tool*, Genome Res. 2002 12: 656-664, DOI:10.1101/gr.229202

[14] Florea L., Hartzell, G., Zhang, Z., Rubin, G., Miller, W., (1998) *A Computer Program for Aligning a cDNA Sequence with a Genomic DNA Sequence*, Genome Research 1998, 8: 967-974, DOI: 10.1101/gr.8.9.967

[15] Mikroskop, flexible tool for creating diagrams of sequence annotations http://www.bioinformatics.uwaterloo.ca/downloads/mikroskop/

[16] Drosophila melanogaster draft assembly, Apr. 2006 (Berkeley Drosophila Genome Project Release 5.12, Oct. 2008), http://genome.ucsc.edu/cgi-bin/hgGateway?clade=insect&org=0&db=0

[17] Preparation of parameters and testing data from Drosophila melanogaster (fruit fly) http://biowiki.brejovci.net/index.php?title=EH:Brona-2009-11-02

[18] UCSC genome annotation database for the Apr. 2006 assembly of the D. melanogaster genome (dm3, BDGP Release 5), The assembly sequence in one file per chromosome. http://hgdownload.cse.ucsc.edu/goldenPath/dm3/bigZips/chromFa.tar.gz

[19] UCSC genome annotation database for the Apr. 2006 assembly of the D. melanogaster genome (dm3, BDGP Release 5), RefGene database tables http://hgdownload.cse.ucsc.edu/goldenPath/dm3/database/refGene.txt.gz

[20] The Gene Index Databases, Dana-Farber Cancer Institute. Boston, MA 02115 USA, DFCI gene index for Drosophila Release 11.0 (June 14, 2006), ftp://occams.dfci.harvard.edu/pub/bio/tgi/data/Drosophila_melanogaster/DGI.release_11.zip

[21] FlyBase ID Converter http://flybase.org/static_pages/downloads/IDConv.html

[22] Testovacia množina - sekvencia, adresa na Compbiu /projects/eh-dev/peter/gene-finders/test.fa

[23] Testovacia množina - anotácia, adresa na Compbiu /projects/eh-dev/peter/gene-finders/test.gtf

[24] EST databáza, adresa na Compbiu /projects/eh-dev/peter/gene-finders/dromel-est.fa

[25] Gény z testovacej množiny, adresa na Compbiu /projects/eh-dev/peter/gene-finders/trainsets/gene-names

[26] FlyBase identifikátory génov z testovacej množiny, adresa na Compbiu /projects/eh-dev/peter/gene-finders/trainsets/flybase_ids.txt

[27] EST dáta pre Augustus a GeneID, adresa na Compbiu /projects/eh-dev/peter/gene-finders/results/hints/dromel-est-hint.gff

[28] FlyBase identifikátory spoločných génov z testovacej množiny a trénovacej množiny pre Augustus (spolu 84 génov), adresa na Compbiu /projects/eh-dev/peter/gene-finders/trainsets/augustus/ common_augustus.txt

[29] Názvy spoločných génov z testovacej množiny a trénovacej množiny pre Augustus (spolu 44 génov), adresa na Compbiu /projects/eh-dev/peter/gene-finders/trainsets/augustus/ common_genes_augustus.txt

[30] Konfiguračný súbor programu Augustus pre predikciu s externými zdrojmi dát, adresa na Compbiu /projects/eh-dev/peter/gene-finders/augustus/config/extrinsic.cfg

[31] Názvy spoločných génov z testovacej množiny a trénovacej množiny pre GeneID (spolu 40 génov), adresa na Compbiu /projects/eh-dev/peter/gene-finders/trainsets/geneid/ common_geneid

[32] Porovnávací skript, adresa na Compbiu /projects/eh/bin/evaluate.pl