Uniwersytet Warszawski
Wydział Matematyki, Informatyki i Mechaniki

Adam Fuksa
Nr albumu: 209240

Wyszukiwanie Rho-niezależnych terminatorów w genomach bakterii i bakteriofagów

Praca magisterska
na kierunku INFORMATYKA

Praca wykonana pod kierunkiem
prof. dra hab. Jerzego Tiuryna
Instytut Informatyki

Wrzesień 2007

Oświadczenie kierującego pracą

Potwierdzam, że niniejsza praca została przygotowana pod moim kierunkiem i kwalifikuje się do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data Podpis kierującego pracą

Oświadczenie autora (autorów) pracy

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data Podpis autora (autorów) pracy

Streszczenie

Tematem tej pracy jest opis metod i programów stosowanych do automatycznego wyszukiwania Rho-niezależnych terminatorów—miejsc w DNA kończących transkrypcję przez polimerazę RNA. Wadą istniejących programów jest niska skuteczność mierzona parametrami takimi jak czułość i specyficzność. W ramach pracy starałem się napisać program lepszy od istniejących. Mój program ritfind porównuję z popularnymi programami TranstermHP i rnall na genomie Escherichii Coli oraz bakteriofaga P1.

Słowa kluczowe

Rho-niezależne, terminator, terminacja, operon, RegulonDB

Dziedzina pracy (kody wg programu Socrates-Erasmus)

11.3 Informatyka

Klasyfikacja tematyczna

J. Computer Applications
J.3. Life and Medical Sciences: Biology and genetics

Tytuł pracy w języku angielskim

Finding Rho-independent terminators in bacteria and bacteriophage genomes

Dziękuję dr hab. Małgorzacie Łobockiej za pomoc w przygotowaniu tej pracy.

Spis treści

Rozdział 1

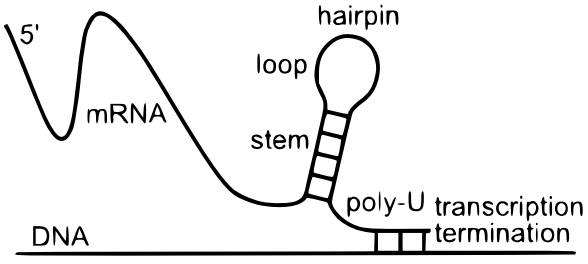
Wprowadzenie

1.1. Wstęp biologiczny

Jednym z często stosowanych podziałów organizmów jest podział na organizmy prokariotyczne (bezjądrowe) i eukariotyczne (posiadające jądro komórkowe). Większość prokariotów stanowią bakterie. Najczęściej DNA prokariotów przechowywane jest w formie pojedynczej kolistej makrocząsteczki. DNA eukariotów zawarte jest w wielu chromosomach. Geny u prokariotów transkrybowane są grupami. U eukariotów każdy gen transkrybowany jest na osobne mRNA. [1]

Grupa (zbiór) wspólnie transkrybowanych genów zwana jest *operonem*. W konkretnym operonie, nie wszystkie geny muszą być transkrybowane jednocześnie. Przykładowo, jeśli na operon składają się geny A, B i C, to w komórce mogą pojawiać się tylko *jednostki transkrypcyjne* (ang. *transcription units*) A-B i A-B-C. Końce jednostek transkrypcyjnych wyznaczają *terminatory transkrypcji*.

U prokariotów istnieją co najmniej dwa rodzaje terminacji. W terminacji Rho-zależnej mRNA zostaje oddzielone na skutek interakcji z białkiem (czynnikiem transkrypcyjnym) Rho. Terminacja Rho-niezależna następuje gdy polimeraza RNA napotka na transkrybowanym DNA na sekwencję palindromową (terminator), która w syntetyzowanym RNA przybiera strukturę spinki (ang. *hairpin loop*). Prawdopodobieństwo terminacji wzrasta, gdy za strukturą palindromową występuje ciąg nukleotydów T [2]. Schemat Rho-niezależnej terminacji pokazany jest na rys. 1.1.



Rysunek 1.1: Schemat Rho-niezależnej terminacji. Hybryda RNA:DNA, struktura spinki z wyróżnionymi nogami i pętlą, ogon z wieloma zasadami U (źródło: [EKWSS00]).

Znalezienie miejsc terminacji pozwala określić strukturę operonów i jednostek transkrypcyjnych w genomie. Wiedza o operonach z kolei jest pomocna w określaniu funkcji składających się na operon genów. [KAS07]

Obecnie, jedyną pewną metodą sprawdzenia, czy dany fragment DNA jest Rho-niezależnym terminatorem, jest eksperyment biologiczny. Często używanymi technikami detekcji są *hybrydyzacja northern* (ang. *northern blotting*) i *mapowanie nukleazą S1* (ang. *S1 nuclease mapping*). Wysokie koszty eksperymentów biologicznych są zachętą do opracowywania i używania programów komputerowych, które mogłyby częściowo zastępować, uzupełniać lub wskazywać nowe, warte przeprowadzenia eksperymenty.

Tematem tej pracy było opracowanie programu do automatycznego wyszukiwania Rho-niezależnych terminatorów w genomach bakterii o większej skuteczności od istniejących programów.

1.2. Historia programów do znajdowania Rho-niezależnych terminatorów

W 1984 powstał program Terminator [BT84], dostępny w komercyjnym pakiecie GCG. Do detekcji terminatorów Terminator używa macierzy z częstością występowania dinukleotydów w poszczególnych miejscach znanych terminatorów. Najnowszym programem jest TranstermHP [KAS07]. Na uwagę zasługują też Transterm [EKWSS00] (poprzednik TranstermHP) oraz rnall [WD05]. Zostało opublikowanych kilka innych prac (m.in. [LSLHME01], [CBT90], [HMN05]), ale programy w nich opisane nie zostały udostępnione.

Rozdział 2

Program ritfind, a inne programy — budowa i sposób działania

Nowy program do wyszukiwania Rho-niezależnych terminatorów nazwałem *ritfind* od ang. *R*ho *i*ndependent *t*erminator *find*er, czyli wyszukiwarka Rho-niezależnych terminatorów. Nazwa ta w przeciwieństwie do niektórych wcześniejszych nazw programów tego typu (np. Terminator) będzie pozwalać na znalezienie programu w popularnych wyszukiwarkach internetowych — przykładowo Google na zapytanie „ritfind" zwraca 0 wyników.

2.1. Struktura programu

Program ritfind jest napisany głównie w języku Python[3]. Do parsowania plików z genomami wykorzystywana jest biblioteka BioPython[4]. Program składa się z kilku części odpowiedzialnych za następujące funkcje: wyszukiwania kandydatów na terminatory, wyliczanie charakterystyk kandydatów, ocena jakości kandydatów, interfejs użytkownika (tekstowy i WWW). Schemat działania znajduje się na rysunku 2.1.

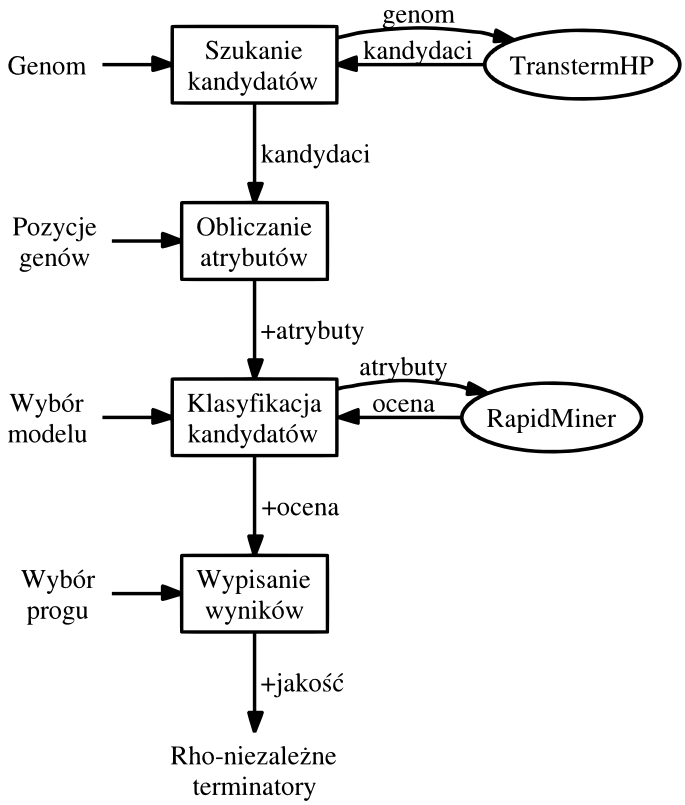
2.2. Szukanie kandydatów na Rho-niezależne terminatory

Pierwsza faza działania programu polega na wyszukaniu w określonym przez użytkownika genomie kandydatów na Rho-niezależne terminatory. Genom znajduje się w pliku w formacie FASTA[1] do którego ścieżka podawana jest jako parametr programu.

Do wyszukiwania kandydatów na terminatory użyłem fragmentu programu TranstermHP. Wybrałem takie rozwiązanie, ponieważ TranstermHP znajduje kandydatów na terminatory wystarczająco szybko i dokładnie. Kod źródłowy TranstermHP jest dostępny na licencji GNU, co pozwala na jego zmianę i użycie w programach udostępnianych na tej samej licencji, a takim programem jest ritfind. Dzięki użyciu fragmentu programu TranstermHP mogłem bardziej skupić się na ocenie kandydatów na terminatory, czyli na zadaniu z którym istniejące programy radzą sobie słabiej.

Szybkość TranstermHP wynika z zastosowania algorytmu o niewysokiej złożoności obliczeniowej i zaimplementowania go w języku C++, dla którego istnieją kompilatory generujące szybki kod maszynowy. Działanie algorytmu polega na przesuwaniu okienka długości 59 nukleotydów po genomie i wyliczaniu dla sekwencji w okienku wartości funkcji energii spinki

[1] prosty format tekstowy mogący przechowywać nazwaną sekwencję nukleotydów lub aminokwasów



Rysunek 2.1: Schemat działania programu ritfind.

(patrz 2.3.3) dla każdej pozycji okienka. Ograniczenie długości okienka pozwala na przyśpieszenie wyszukiwania, znane terminatory mają długość mniejszą niż 59 nukleotydów.

Funkcja jest wyliczana przy pomocy algorytmu dynamicznego, przyjmuje niskie wartości dla biologicznych palindromów. Na kandydatów na terminatory brane są sekwencje dla których wyliczona wartość jest mniejsza niż -2. Sekwencje będące fragmentami sekwencji o mniejszych energiach są pomijane. Dodatkowo pomijane są sekwencje dla których długość nogi (patrz 2.3.1) jest mniejsza niż 4 nukleotydy lub waga ogona (patrz 2.3.4) większa niż 2.5. Przeszukiwana jest cała sekwencja genomu, oraz jej odwrócone dopełnienie.

Przykładowe dane wejściowe (genom) dla tej części programu znajdują się w dodatku C. Wyjściem jest lista kandydatów na terminatory zawierająca pozycję kandydata w genomie (nr pierwszego i ostatniego nukleotydu) oraz kierunek (1, jeśli 5'–3'; -1 jeśli przeciwny). Dodatkowo zwracana jest wartość funkcji wagi ogona, energii spinki i odpowiadające parowanie nukleotydów. Oto przykład takiej listy:

```
 47,  62,  1, -3.3, -3.09, 'CTGTA AATCCA TACAG'
 77,  93,  1, -2.2, -2.60, 'AGGTCA AATAG TGACCA'
 88, 101, -1, -3.1, -2.63, 'TGACC ACTT GATCA'
289, 307,  1, -6.8, -3.29, 'TGATAGTT GCC AACTGTCA'
330, 360, -1, -4.7, -4.66, 'TAG-AACGCACGCG CGGT CGCTGGCGTTTCTA'
335, 354, -1, -4.8, -4.08, 'CGCACGCG CGGT CGCTGGCG'
```

2.3. Charakterystyki kandydatów na Rho-niezależne terminatory

Po wyznaczeniu kandydatów na Rho-niezależne terminatory, ritfind wylicza dla każdego kandydata pewne charakterystyki (atrybuty), na podstawie których zostanie wyznaczona jakość (prawdopodobieństwo bycia Rho-niezależnym terminatorem) każdego kandydata. Następująca tabela porównuje programy do wyszukiwania terminatorów pod względem atrybutów branych pod uwagę przy ocenie kandydatów na terminatory.

program	energia spinki	waga ogona	energia ogona
Transterm	+	+	-
TranstermHP	+	+	-
rnall	+	+	+
TermIT	+	+	-
ritfind	+	+	+
RNAMotif	+	-	+

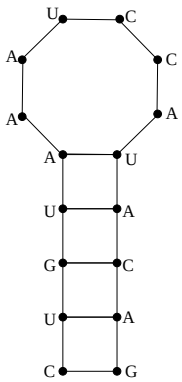
Tabela 2.1: Charakterystyki używane do wyznaczania jakości kandydatów na terminatory (oznaczenia: + — używane, - — nie używane)

Atrybuty takie jak długość nogi, pętli i zawartość GC mają wpływ na 3 atrybuty z tabeli. Program ritfind wykorzystuje te atrybuty także bezpośrednio.

W dalszej części rozdziału znajduje się dokładny opis atrybutów używanych przez program ritfind. Dodatek A zawiera fragment kodu w języku Python liczący najważniejsze atrybuty.

2.3.1. Długość nogi

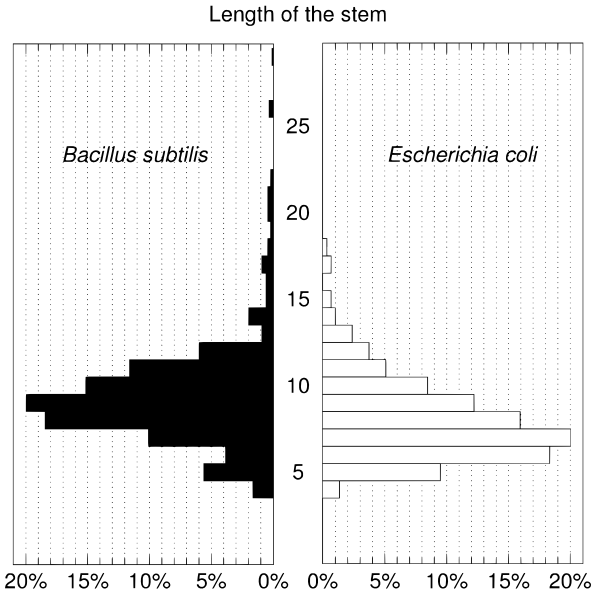
Długość nogi (ang. *stem length*) oznacza liczbę sparowanych nukleotydów w przybliżonym biologicznym palindromie. Przykładowo w sekwencji CTGTA AATCCA TACAG jest 5 sparowanych nukleotydów: C-G, T-A, G-C, T-A i A-T. Sposób parowania w cząstecze RNA odpowiadającej tej sekwencji przedstawia rysunek 2.2. Sparowane nukleotydy przy transkrypcji DNA na RNA odpowiadają nodze w tworzącej się strukturze spinki (ang. *hairpin*).



Rysunek 2.2: Schemat struktury spinki odpowiadającej sekwencji CTGTA AATCCA TACAG. Nukleotydy T zamienione są na U.

Wśród opublikowanych Rho-niezależnych terminatorów transkrypcji *E. Coli* 75% terminatorów ma długość nogi od 5 do 9 nukleotydów, natomiast 75% terminatorów *B. Subtilis*

ma długość nogi od 7 do 11 nukleotydów [HMN05]. Najczęściej programy do wyszukiwania terminatorów nie używają tego atrybutu bezpośrednio, ale razem z długością pętli, w celu obliczenia długości struktury spinki terminatora (liczby nukleotydów w strukturze spinki), która może służyć np. do obliczenia ilorazu energii spinki i długości spinki. Struktura spinki składa się z nogi i pętli, na nogę składają się sparowane nukleotydy, dlatego długość struktury spinki to suma długości pętli i podwojonej długości nogi. Rysunek 2.3 podaje rozkład długości nogi dla dwóch bakterii.



Rysunek 2.3: Rozkład długości nogi obliczony na podstawie postulowanych 147 Rho-niezależnych terminatorów *E. Coli* i 425 *B. Subtilis* (źródło: [HMN05]).

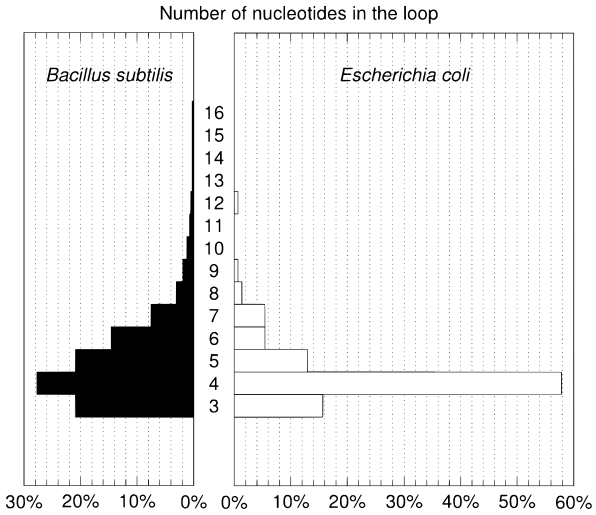
2.3.2. Długość pętli

Długość pętli (ang. loop length) to liczba nukleotydów między sparowanymi nukleotydami w przybliżonym biologicznym palindromie. Przykładowo w sekwencji CTGTA AATCCA TACAG jest 6 nukleotydów między sparowanymi nukleotydami. Nukleotydy te odpowiadają pętli w strukturze spinki tworzącej się przy transkrypcji DNA na RNA. Większość poznanych terminatorów ma pętle długości 4. Rysunek 2.4 przedstawia rozkład długości pętli dla poznanych terminatorów *E. Coli* i *B. Subtilis*.

2.3.3. Energia spinki

Energia spinki (ang. hairpin energy, hairpin score) to atrybut mający modelować stabilność struktury spinki. Program ritfind liczy energię spinki w sposób zaproponowany w pracy [KAS07] oraz [EKWSS00]. Założenia tego modelu są następujące: najbardziej stabilną parą nukleotydów występującą w nodze spinki jest para G-C. Nieco mniej stabilna jest para A-U. Istnieje też pewne słabe oddziaływanie między parą G-U. Inne pary nukleotydów nie tworzą wiązań wodorowych i osłabiają nogę. Energia spinki dla danej struktury określona jest wzorem:

$$E = \mathsf{gc}n_1 + \mathsf{au}n_2 + \mathsf{gu}n_3 + \mathsf{mm}n_4 + \mathsf{gp}n_5 + (n_6 - 2),$$



Rysunek 2.4: Rozkład długości pętli obliczony na podstawie postulowanych 147 Rho-niezależnych terminatorów *E. Coli* i 425 *B. Subtilis* (źródło: [HMN05]).

gdzie $n_1 \dots n_5$ to liczba odpowiednio par G-C, A-U, G-U, innych par i dziur (nukleotydów bez pary) w strukturze nogi, a n_6 to długość pętli.

Parametry $\mathsf{gc} \dots \mathsf{gp}$ zostały obliczone drzewem decyzyjnym OC1 na podstawie 70 terminatorów z pracy [CBT90][2] oraz 70 sekwencji wewnątrzgenowych nie będących terminatorami, lecz o podobnych charakterystykach. Obliczone wartości parametrów znajdują się w następującej tabeli.

parametr	wartość
gc	-2.3
au	-0.9
gu	1.3
mm	3.5
gp	6.0

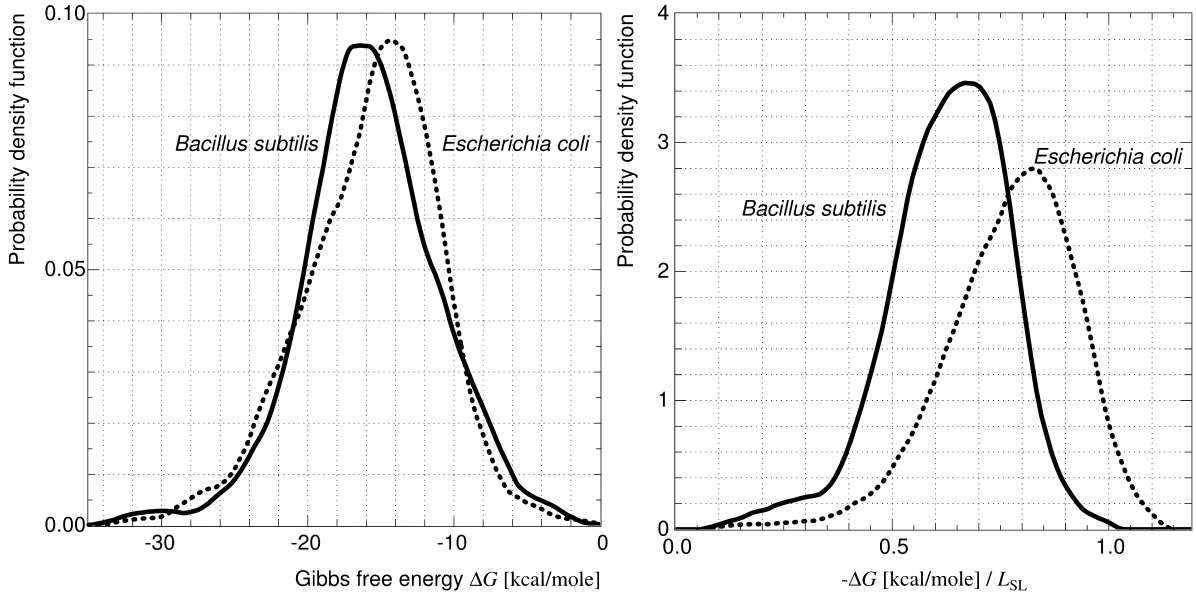
Tabela 2.2: Wartość parametrów używanych do obliczenia energii spinki dla potencjalnego Rho-niezależnego terminatora (źródło [EKWSS00]).

Daną sekwencję można zinterpretować jako strukturę spinki na wiele sposobów, w zależności od rozmieszczenia dziur i długości nogi. Przykładowo dla sekwencji CGCGCGAATATCGCGCG struktura spinki może mieć postać np. CGCGCG AATAT CGCGCG, CGCGCGA ATA TCGCGCG lub inną. Do obliczeń używana jest konfiguracja o najmniejszej energii. Aby ją wyznaczyć stosowany jest algorytm dynamiczny o następującym równaniu rekurencyjnym:

$$E[i,j] = \mathsf{min} \begin{cases} j-i-1, \quad j-i \geqslant 2 & \text{pętla} \\ \mathsf{e}(s[i],s[j]) + E[i+1,j-1] & \text{parowanie nukleotydów} \\ \mathsf{gp} + E[i+1,j] & \text{dziura na lewej nodze} \\ \mathsf{gp} + E[i,j-1] & \text{dziura na prawej nodze} \end{cases},$$

[2]W tej pracy znajduje się lista 148 Rho-niezależnych terminatorów *E. Coli*. Lista składa się z dwóch części: 66 terminatorów zweryfikowanych eksperymentalnie oraz 82 postulowanych na podstawie struktury i pozycji w genomie.

gdzie $E[i,j]$ to energia fragmentu sekwencji od i-tego do j-tego nukleotydu, a $\mathsf{e}(s[i], s[j])$ to energia pary nukleotydów z sekwencji (tabela 2.2). Poniższy rysunek przedstawia rozkład energii spinki dla terminatorów *E. Coli* i *B. Subtilis*.



Rysunek 2.5: Rozkład energii spinki oraz rozkład energii spinki podzielonej przez długość spinki obliczony na podstawie postulowanych 147 Rho-niezależnych terminatorów *E. Coli* i 425 *B. Subtilis* (źródło: [HMN05]).

2.3.4. Waga ogona

Waga ogona (ang. *U-tail weight*) zdefiniowana została w pracy [CBT90]. Parametr ten przyjmuje niską wartość dla ogonów terminatorów z dużą liczbą nukleotydów U (czyli nukleotydów T w DNA). Nukleotydy U bliższe nodze mają większą wagę. Parametr ten jest stosowany, ponieważ eksperymentalnie zweryfikowane Rho-niezależne terminatory mają ogony z wieloma nukleotydami U. Waga ogona jest parametrem sztucznym, nie wyraża żadnej konkretnej wartości fizykochemicznej. Oto definicja wagi ogona:

$$T(s) = -\sum_{n=1}^{15} x_n,$$

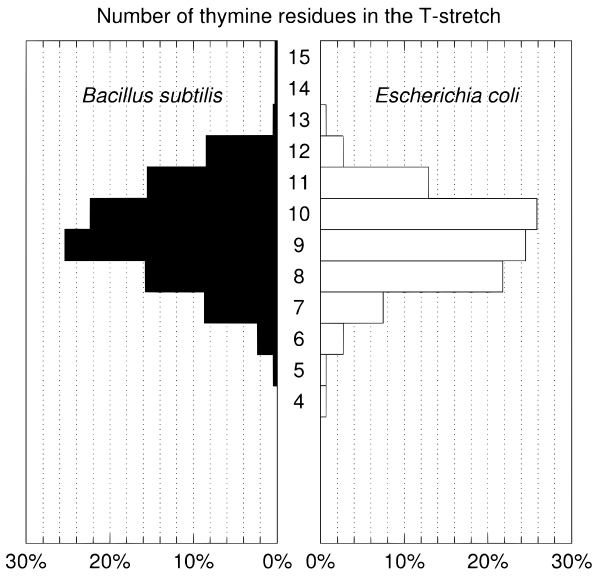
gdzie $x_n = 0.9x_{n-1}$ gdy $s[n]$ (n-ty nukleotyd ogona s) to U, $0.6x_{n-1}$ w przeciwnym przypadku, $x_0 = 1$.

Rysunek 2.6 przedstawia rozkład liczby nukleotydów T w ogonach terminatorów.

2.3.5. Energia ogona

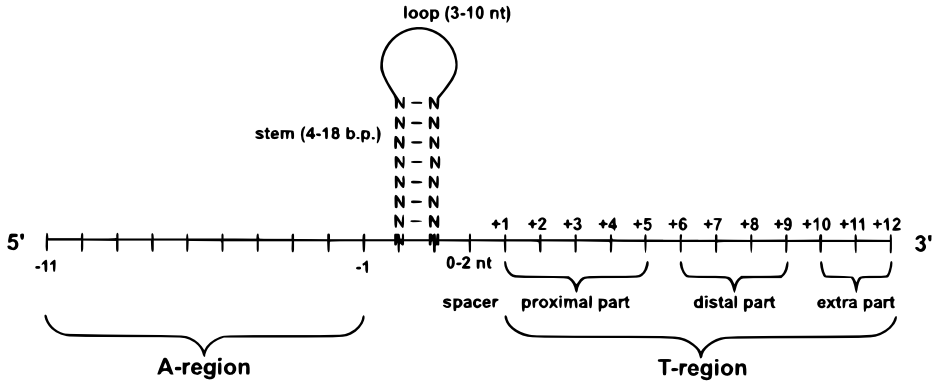
Energia ogona (ang. *U-tail hybridization energy*) została zdefiniowana w [LSLHME01] na podstawie modelu Rho-niezależnej terminacji z [GN99]. Model zakłada, że terminacja zachodzi na skutek konkurencji między stabilnością struktury spinki i hybrydy RNA:DNA nogi terminatora. Energia ogona wyraża stabilność hybrydy następującym równaniem:

$$H(s) = \Delta G^\circ_{37}(s_{\text{spacer}}) + \Delta G^\circ_{37}(s_{\text{proximal}}) + 0.5\Delta G^\circ_{37}(s_{\text{distal}}) + 0.01\Delta G^\circ_{37}(s_{\text{extra}}),$$



Rysunek 2.6: Rozkład liczby nukleotydów T w 15 nukleotydowych ogonach terminatorów, obliczony na podstawie postulowanych 147 Rho-niezależnych terminatorów *E. Coli* i 425 *B. Subtilis* (źródło: [HMN05]).

gdzie $\Delta G^{\circ}_{37}(x)$ oznacza energie swobodną Gibbsa obliczoną poprzez sumowanie energii poszczególnych sąsiadujących par nukleotydów z x przy użyciu parametrów z [SNKMNOYS95] (tab. 2.3). Składniki sumy oznaczają energię liczoną dla poszczególnych fragmentów ogona terminatora (rys. 2.7).



Rysunek 2.7: Rho-niezależny terminator, podział na części składowe. spacer to fragment długości 0-2 nukleotydów innych niż T (źródło: [LSLHME01]).

Table 3: Thermodynamic Parameters for Hybrid Duplex Initiation and Propagation in 1 M NaCl Buffer[a]

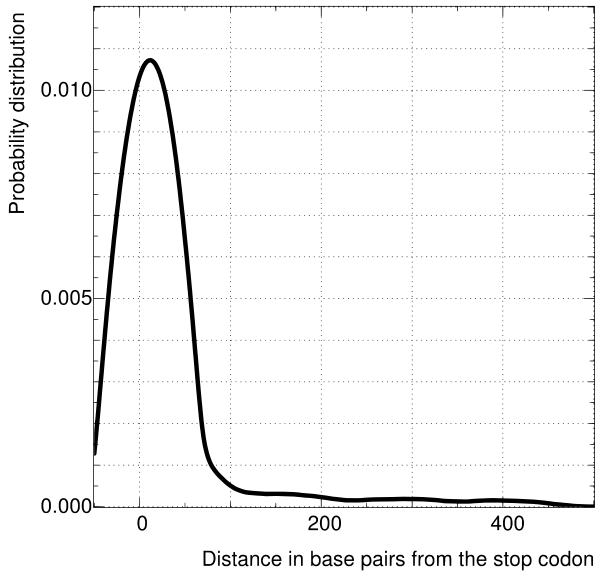
Sequence	ΔH° / kcal mol^{-1}	ΔS° / cal mol^{-1} K^{-1}	ΔG°_{37} / kcal mol^{-1}
rAA dTT	-7.8	-21.9	-1.0
rAC dTG	-5.9	-12.3	-2.1
rAG dTC	-9.1	-23.5	-1.8
rAU dTA	-8.3	-23.9	-0.9
rCA dGT	-9.0	-26.1	-0.9
rCC dGG	-9.3	-23.2	-2.1
rCG dGC	-16.3	-47.1	-1.7
rCU dGA	-7.0	-19.7	-0.9
rGA dCT	-5.5	-13.5	-1.3
rGC dCG	-8.0	-17.1	-2.7
rGG dCC	-12.8	-31.9	-2.9
rGU dCA	-7.8	-21.6	-1.1
rUA dAT	-7.8	-23.2	-0.6
rUC dAG	-8.6	-22.9	-1.5
rUG dAC	-10.4	-28.4	-1.6
rUU dAA	-11.5	-36.4	-0.2
initiation	1.9	-3.9	3.1

[a] Estimated errors in ΔH°, ΔS°, and ΔG_{37}° are ± 0.3 kcal mol^{-1}, ± 1.3 cal mol^{-1} K^{-1}, and ± 0.1 kcal mol^{-1}, respectively.

Tabela 2.3: Wartość parametrów używanych do obliczenia energii swobodnej Gibbsa hybryd RNA:DNA metodą sąsiadujących nukleotydów (źródło [SNKMNOYS95]).

2.3.6. Odległość od kodonu STOP

Odległość terminatora od kodonu STOP oznacza liczbę nukleotydów między najbliższym końcem genu na tej samej nici na której jest terminator, a środkiem terminatora (nukleotydem w połowie odległości między początkami nóg terminatora). Większość terminatorów odległa jest o mniej niż 100 nukleotydów, jednak zdarzają się terminatory zawarte częściowo lub całkowicie w genach (terminatory wewnątrzgenowe). Na rysunku 2.8 znajduje się rozkład odległości od kodonu STOP dla terminatorów *B. Subtilis*.



Rysunek 2.8: Rozkład odległości od kodonu STOP dla 425 Rho-niezależnych terminatorów *B. Subtilis* (źródło: [HMN05]).

2.3.7. Zawartość nukleotydów G i C w nodze

Parametr ten jest równy liczbie nukleotydów G i C podzielonej przez długość nogi. Podczas prac nad modelem okazało się jednak, że atrybut ten nie wpływał na klasyfikację kandydatów na terminatory.

2.3.8. Korelacje między atrybutami

Wyżej opisane atrybuty nie są od siebie niezależne. Przykładowo terminatory z mniejszą wagą ogona (z ogonami z licznymi nukleotydami T), będą miały większą energię ogona (wysoki współczynnik dla pary nukleotydów TT). Zależność ta widoczna jest w poniższej tabeli (współczynnik korelacji -0.89 między atrybutami).

	dł. nogi	dł. pętli	e. spinki	w. ogona	e. ogona	zaw. GC	odl. STOP
dł. nogi	1	0.152	0.464	0.296	-0.247	-0.461	-0.039
dł. pętli	0.152	1	0.583	0.208	-0.182	-0.216	0.342
e. spinki	0.464	0.583	1	0.212	-0.200	-0.150	0.295
w. ogona	0.296	0.208	0.212	1	-0.890	-0.480	0.048
e. ogona	-0.247	-0.182	-0.200	-0.890	1	0.383	-0.014
zaw. GC	-0.461	-0.216	-0.150	-0.480	0.383	1	-0.147
odl. STOP	-0.039	0.342	0.295	0.048	-0.014	-0.147	1

Tabela 2.4: Macierz korelacji atrybutów dla 151 Rho-niezależnych terminatorów *E. Coli* z bazy RegulonDB v.5.6.

2.4. Klasyfikacja kandydatów

Po obliczeniu atrybutów kandydatów na Rho-niezależne terminatory, następuje ocena jakości każdego kandydata. Do oceny ritfind używa programu *RapidMiner* [MWKSE06] oraz

wcześniej przygotowanego (wytrenowanego) modelu. Program RapidMiner stosując model na zbiorze kandydatów przypisuje każdemu kandydatowi liczbę (atrybut nazwany *confidence(yes)*) od 0 do 1 oznaczającą jakość terminatora. Im liczba bliższa 1, tym większa wg. modelu szansa na to, że kandydat jest Rho-niezależnym terminatorem. Użytkownik uruchamiając ritfind może wybrać model stosowany do oceny, albo użyć domyślnego.

2.5. Trenowanie modelu

Do oceny jakości kandydatów na Rho-niezależne terminatory ritfind używa modelu przygotowanego w programie RapidMiner. RapidMiner to rozbudowany pakiet do dataminingu obejmujący środowisko graficzne. Praca z programem polega na budowaniu eksperymentu z drzewa operatorów takich jak rozmaite źródła danych (CSV, bazy danych itd.), filtry, operatory do klasyfikacji, oceny jakości zbudowanego modelu. Wygenerowane modele można zapisać i używać we własnych programach. RapidMiner pozwala używać operatorów z pakietu WEKA [WF05]. WEKA i RapidMiner napisane są w języku Java. WEKA dostępna jest na licencji GPL, RapidMiner dostępny jest na licencji komercyjnej oraz GPL.

RapidMiner udostępnia wiele algorytmów do klasyfikacji danych. Podczas opracowywania modelu wykonywałem próby z użyciem różnych typów klasyfikatorów. Bez specjalnego dostrajania parametrów najlepsze rezultaty dawał klasyfikator W-BayesNet z pakietu WEKA. Ponieważ nie udało mi się uzyskać lepszych wyników przy pomocy innych klasyfikatorów, użyłem go do budowy modelu. Jego dodatkową zaletą jest szybkość trenowania i klasyfikacji.

Przygotowałem dwa modele, obydwa trenowane na genomie *E. Coli*. Budując pierwszy model (ec151t151f) wzorowałem się na innych programach, do trenowania użyłem małej i nieproporcjonalnej do spotykanej w praktyce liczby przykładów pozytywnych i negatywnych (151:151). Powoduje to sztuczne zawyżenie szacunkowej skuteczności modelu — czułość, specyficzność, dodatnia i ujemna wartość predykcyjna (patrz 2.8) ok. 98% (tab. 2.5).

	praw. nie	praw. tak	dokładność
pred. nie	148	3	98%
pred. tak	3	148	98%
kompletność	98%	98%	

Tabela 2.5: Skuteczność modelu ec151t151f oszacowana metodą 10 krotnej walidacji krzyżowej.

Drugi model (ecnormal), wytrenowany na proporcjonalnej do spotykanej w danych wejściowych liczbie przykładów pozytywnych i negatywnych (wg. moich szacunków, na 1 Rho-niezależny terminator w *E. Coli* przypada 456 kandydatów znajdowanych przez *TranstermHP*), ma dużo mniejszą skuteczność — czułość 29% przy wysokiej dodatniej wartości predykcyjnej (tab. 2.6).

	praw. nie	praw. tak	dokładność
pred. nie	68822	107	99.8%
pred. tak	1	44	97.7%
kompletność	100.0%	29.1%	

Tabela 2.6: Skuteczność modelu ecnormal oszacowana metodą 10 krotnej walidacji krzyżowej.

Skuteczność obydwu modeli oszacowałem metodą 10-krotnej walidacji krzyżowej. Oto definicja eksperymentu programu RapidMiner trenującego model ecnormal.

```
<?xml version="1.0" encoding="UTF-8"?>
<process version="4.0">
  <operator name="Root" class="Process">
      <parameter key="logfile" value="log"/>
      <parameter key="logverbosity" value="warning"/>
      <parameter key="random_seed" value="2000"/>
      <operator name="ExampleSource" class="ExampleSource">
          <parameter key="attributes" value="ecoli.aml"/>
      </operator>
      <operator name="XValidation" class="XValidation">
          <parameter key="create_complete_model" value="true"/>
          <parameter key="keep_example_set" value="true"/>
          <operator name="W-BayesNet" class="W-BayesNet">
          </operator>
          <operator name="OperatorChain" class="OperatorChain">
              <operator name="ModelApplier" class="ModelApplier">
                  <list key="application_parameters"> </list>
              </operator>
              <operator name="ThresholdFinder" class="ThresholdFinder">
                  <parameter key="misclassification_costs_second" value="455.78"/>
              </operator>
              <operator name="ThresholdApplier" class="ThresholdApplier">
              </operator>
              <operator name="PerformanceEvaluator" class="PerformanceEvaluator">
                  <parameter key="accuracy" value="true"/>
                  <list key="additional_performance_criteria"> </list>
                  <list key="class_weights"> </list>
                  <parameter key="main_criterion" value="accuracy"/>
                  <parameter key="precision" value="true"/>
                  <parameter key="recall" value="true"/>
              </operator>
          </operator>
      </operator>
      <operator name="ModelWriter" class="ModelWriter">
          <parameter key="model_file" value="../apply/model.mod"/>
      </operator>
  </operator>
</process>
```

Model ecnormal trenowany jest na 456 razy większej liczbie przykładów negatywnych niż pozytywnych, dlatego aby oszacować czułość i specyficzność przy zbliżonych pozytywnych i negatywnych wartościach predykcyjnych, znajdowany jest próg dla jakości kandydata (confidence(yes)) zwracanej przez klasyfikator, od którego kandydat uznawany jest za terminator, dla którego przy karze za nieprawdziwe wskazanie terminatora 456 razy większej od kary za nieprawdziwe wskazanie nie-terminatora, sumy kar dla dwóch możliwych klasyfikacji są równe.

Ponieważ ten próg może być nieco inny w każdej iteracji walidacji krzyżowej, oszacowana przez walidację krzyżową skuteczność może się różnić od faktycznej skuteczności klasyfikatora (obliczonej dla z góry ustalonego progu). Czułość pełnego modelu (trenowanego na pełnym zbiorze pozytywów i negatywów) testowanego na pełnym zbiorze danych (czyli naiwnie obliczona), po zastosowaniu znalezionego progu wynosi 34.4% dla specyficzności i wartości predykcyjnych bliskich 100%. Dlatego prawdopodobnie można postulować, że wynik walidacji krzyżowej (czułość 29% przy wysokich wartościach predykcyjnych) jest wiarygodny.

2.6. Przykłady pozytywne

Do trenowania klasyfikatora potrzebne są zweryfikowane w eksperymentach biologicznych Rho-niezależne terminatory. Wyniki takich eksperymentów ukazują się w publikacjach w czasopismach biologicznych. Najczęściej w jednej publikacji znajdują się informacje o pojedynczych terminatorach. Podejmowane są próby zbierania informacji z wielu publikacji i tworzenia baz danych z terminatorami pozwalającymi na maszynowe badanie genomów organizmów.

Tabela 2.7 przedstawia programy do wyszukiwania Rho-niezależnych terminatorów w kontekście wykorzystania przykładów pozytywnych.

program	organizm	źródło przykładów	opis
Transterm	*E. Coli*	[CBT90]	70 terminatorów użyto do trenowania, 61 do testowania programu
(nienazwany) [HMN05]	*B. Subtilis*	[MNON04]	463 terminatory użyte do wyznaczenia współczynników równania z [CBT90] metodą regresji logistycznej
TranstermHP	*E. Coli*	[CBT90]	70 terminatorów do trenowania, dodatkowo testy tak jak w [HMN05]
rnall	*E. Coli*	[CBT90]	129 terminatorów użyto do wyznaczenia optymalnych progów odcięcia
TermIT	*E. Coli*	[CBT90]	nieopisany sposób użycia
ritfind	*E. Coli*	[REGULON06]	151 terminatorów, walidacja krzyżowa
RNAMotif	*E. Coli*	[CBT90]	trenowanie na 135 terminatorach

Tabela 2.7: Wykorzystanie przykładów pozytywnych

Najwięcej informacji o Rho-niezależnych terminatorach zgromadzono dla *E. Coli* i *B. Subtilis*. Najczęstszym źródłem pozytywów dla programów jest praca [CBT90], w której znajduje się lista 66 terminatorów *E. Coli* zweryfikowanych eksperymentalnie oraz 82 sekwencji nie zweryfikowanych biologicznie, które prawdopodobnie też są terminatorami. Autorzy programów wykorzystywali więcej niż 66 terminatorów, ale mniej niż 148, ponieważ części sekwencji nie udało się odnaleźć w nowszych (dokładniejszych) wersjach genomu *E. Coli*.

Nowszym zbiorem informacji o *E. Coli* jest baza RegulonDB [REGULON06], zawierająca m.in. listę genów, operonów, terminatorów, promotorów i czynników transkrypcyjnych. Baza jest uaktualniana kilka razy w roku, wersja 5.7 (kwiecień 2007) zawiera listę 154 Rho-niezależnych i 8 Rho-zależnych terminatorów.

Niestety, kontrola wykonana przez dr hab. Małgorzatę Łobocką wykazała, że prawie połowa terminatorów z listy RegulonDB 5.6 (kwiecień 2006) faktycznie nie została eksperymentalnie zweryfikowana. Kontrola polegała na analizie publikacji na podstawie których terminatory zostały umieszczone w bazie. Moja pomoc w kontroli polegała na zgromadzeniu publikacji. Większość dostępna była online, a pozostałe odnalazłem w bibliotece. Lista 60 niezweryfikowanych eksperymentalnie terminatorów znajduje się w dodatku D.

Dla *B. Subtilis* publikacja [HMN05] zawiera listę 425 eksperymentalnie zweryfikowanych Rho-niezależnych terminatorów. Związana nazwiskami z publikacją jest baza DBTBS [MNON04] (baza dot. *B. Subtilis*, podobna do RegulonDB), w której obecnie (wersja 5.0, październik 2007) znajdują się 463 terminatory. Niestety, inaczej niż w RegulonDB, autorzy

DBTBS nie udostępniają zbiorczych danych w formie łatwej do przetwarzania przez programy. Do skorzystania z DBTBS konieczne wydaje się być napisanie automatu wyciągającego dane ze stron WWW DBTBS (lub kontakt z autorami bazy).

Program ritfind do trenowania i testowania klasyfikatora korzysta z listy terminatorów udostępnianej przez RegulonDB.

2.7. Przykłady negatywne

Istniejące programy do wyszukiwania terminatorów są zróżnicowane pod względem źródeł przykładów negatywnych (tab. 2.8). Na potrzeby programu ritfind przykłady negatywne zgromadziłem w następujący sposób. Najpierw wyszukałem kandydatów na terminatory, w ten sam sposób w jaki są wyszukiwani kandydaci podczas działania programu (punkt 2.2). Następnie odrzuciłem wszystkie sekwencje, które były znanymi terminatorami, lub należały do operonów, dla których znane są terminatory. Użyłem danych o operonach (składające się na nie geny), pozycjach genów i terminatorów z bazy RegulonDB. Powstała w ten sposób lista liczy 68823 pozycje. Lista znajduje się w pakiecie z programem ritfind.

program	organizm	opis
Transterm	*E. Coli*	trenowanie na 70 sekwencjach wewnątrzgenowych o parametrach podobnych do terminatorów
(nienazwany) [HMN05]	*B. Subtilis*	trenowanie na 567 sekwencjach międzygenowych, wewnątrzoperonowych
TranstermHP	brak	parametry z Transterm + losowe sekwencje o ustalonej zawartości GC do trenowania, dodatkowo testy na negatywach z [HMN05]
rnall	*E. Coli*	trenowanie na 129 sekwencjach wewnątrzgenowych
TermIT	*E. Coli*	trenowanie i testy na nieopisanej liczbie sekwencji wewnątrzgenowych
ritfind	*E. Coli*	trenowanie i walidacja krzyżowa na 68823 sekwencjach wyszukanych podczas normalnej pracy programu, o cechach: wewnątrzoperonowe, różne od terminatorów, z operonów ze znanymi terminatorami
RNAMotif	*E. Coli*	nieużywane

Tabela 2.8: Źródła przykładów negatywnych

W przypadku innych programów, najczęściej przykłady negatywne używane są w niewielkiej liczbie, zbliżonej do liczby przykładów pozytywnych. Dodatkowo przykłady pochodzą z niereprezentatywnych miejsc, np. tylko z genów lub tylko z obszarów międzygenowych, albo z generatora liczb losowych. Publikacje o innych programach, a także pakiety dystrybucyjne z programami nie zawierają listy używanych przykładów negatywnych.

2.8. Skuteczność programu ritfind

Kluczowymi pojęciami potrzebnymi do oceny jakości programów wyszukujących Rho-niezależne terminatory (i innych klasyfikujących) są pojęcia informujące jak często ma miejsce określona sytuacja:

czułość (*ang. sensitivity*) program za terminator uznaje sekwencję, o której wiadomo że jest terminatorem,

specyficzność (*ang. specificity*) program za nie-terminator uznaje sekwencję, o której wiadomo że nie jest terminatorem,

pozytywna wartość predykcyjna (*ang. positive predictive value*) sekwencja, którą program uznał za terminator, jest terminatorem,

negatywna wartość predykcyjna (*ang. negative predictive value*) sekwencja, którą program uznał za nie-terminator, nie jest terminatorem.

Biolog określa przydatność programu wyszukującego terminatory przede wszystkim czułością i pozytywną wartością predykcyjną, czyli parametrami mówiącymi o tym jaką część terminatorów program wykrywa, oraz na ile pewne są terminatory wykryte przez program. W przypadku modelu ecnormal programu ritfind wartości te wynoszą odpowiednio ok. 29% i 97%. Dla innych programów wartości te są nieznane. W publikacjach opisujących programy podawane są czułości i specyficzności, ale dla eksperymentów polegających na stosowaniu danego klasyfikatora na niewielkim, zbliżonym licznością, zbiorze pozytywów i negatywów, często użytym wcześniej do trenowania klasyfikatora. Ponieważ liczba kandydatów na terminatory znajdowane przez programy wyszukujące terminatory jest znacznie większa niż liczba odpowiadających im Rho-niezależnych terminatorów, nie można podawanych czułości i specyficzności przełożyć na faktyczną skuteczność programów.

2.9. Wypisanie wyników

Ostatnią fazą działania programu jest wypisanie wyników. Wypisywane są podstawowe dane (pozycja, kierunek, parowanie nukleotydów, wartości atrybutów oraz jakość) każdego kandydata, którego jakość jest większa niż próg określony przez użytkownika. Wyniki zwracane są na standardowe wyjście lub do pliku albo poprzez stronę WWW lub w formie emaila, w zależności od sposobu korzystania z programu (linia komend lub interfejs WWW).

Modele w programie ritfind jakość kandydatów wyrażają liczbą z zakresu 0 (najgorsza) – 1 (najlepsza). Aby ułatwić użytkownikowi wybór odpowiedniego progu oraz interpretację wyniku, program ritfind umożliwia wypisanie tabeli osiągów modelu. Każdy wiersz tabeli pokazuje wyniki zastosowania modelu z konkretnym progiem dla atrybutu confidence(yes) zwracanego przez model do klasyfikacji zbioru treningowego. Można się więc spodziewać że faktyczna skuteczność przy danych progach jest mniejsza. Kolumny tabeli oznaczają:

thre próg dla atrybutu confidence(yes) zwracanego przez model, dla którego kandydat jest uznawany z Rho-niezależny terminator,

pncn liczba kandydatów ocenionych przez model (przy danym progu) za nie-terminatory, będących faktycznie nie-terminatorami (liczba prawdziwych negatywów),

pncy liczba kandydatów ocenionych jako nie-terminatory, będących faktycznie terminatorami (liczba fałszywych negatywów),

pycn liczba kandydatów ocenionych jako terminatory, będących faktycznie nie-terminatorami (liczba fałszywych pozytywów),

pycy liczba kandydatów ocenionych jako terminatory, będących faktycznie terminatorami (liczba prawdziwych pozytywów).

Na podstawie wartości w kolumnach pncn, pncy, pycn, pycy wyliczane są wartości w pozostałych kolumnach:

spec stosunek liczby kandydatów ocenionych jako terminatory będących faktycznie terminatorami do liczby faktycznych terminatorów (pycy / (pycy + pncy)),

sens stosunek liczby kandydatów ocenionych jako nie-terminatory będących faktycznie nie-terminatorami do liczby faktycznych nie-terminatorów (pncn / (pncn+pycn)),

ppv stosunek liczby kandydatów ocenionych jako terminatory będących faktycznie terminatorami do liczby kandydatów ocenionych jako termiantory (pycy / (pycy + pycn)),

npv stosunek liczby kandydatów ocenionych jako nie-terminatory będących faktycznie nie-terminatorami do liczby kandydatów ocenionych jako nie-termiantory (pncn / (pncn + pncy)).

Warto podkreślić, że wartości w kolumnach spec, sens, ppv, npv nie oznaczają skuteczności modelu w rozumieniu punktu 2.8, ponieważ nie można wnioskować o skuteczności modelu na podstawie wyników testu modelu na zbiorze użytym do trenowania modelu. Można się za to spodziewać, że faktyczna skuteczność modelu nie będzie większa.

Poniżej znajduje się tabela dla modelu ec151t151f.

thre	spec	sens	npv	ppv	pncn	pncy	pycn	pycy
0.00004	0.13245	1.00000	1.00000	0.53546	20	0	131	151
0.00015	0.23179	1.00000	1.00000	0.56554	35	0	116	151
0.00019	0.38411	1.00000	1.00000	0.61885	58	0	93	151
0.00026	0.45033	1.00000	1.00000	0.64530	68	0	83	151
0.00067	0.54305	1.00000	1.00000	0.68636	82	0	69	151
0.00089	0.70861	1.00000	1.00000	0.77436	107	0	44	151
0.00114	0.74172	1.00000	1.00000	0.79474	112	0	39	151
0.00151	0.74834	1.00000	1.00000	0.79894	113	0	38	151
0.00233	0.75497	1.00000	1.00000	0.80319	114	0	37	151
0.00350	0.76159	1.00000	1.00000	0.80749	115	0	36	151
0.00396	0.77483	1.00000	1.00000	0.81622	117	0	34	151
0.00397	0.80132	1.00000	1.00000	0.83425	121	0	30	151
0.01284	0.82781	1.00000	1.00000	0.85311	125	0	26	151
0.01365	0.83444	1.00000	1.00000	0.85795	126	0	25	151
0.02303	0.84106	1.00000	1.00000	0.86286	127	0	24	151
0.04353	0.86093	1.00000	1.00000	0.87791	130	0	21	151
0.04618	0.87417	1.00000	1.00000	0.88824	132	0	19	151
0.07154	0.88079	1.00000	1.00000	0.89349	133	0	18	151
0.07738	0.88742	0.99338	0.99259	0.89820	134	1	17	150
0.17747	0.90066	0.99338	0.99270	0.90909	136	1	15	150
0.21231	0.92053	0.98675	0.98582	0.92547	139	2	12	149
0.22684	0.94702	0.98675	0.98621	0.94904	143	2	8	149
0.33184	0.95364	0.98675	0.98630	0.95513	144	2	7	149
0.61426	0.97351	0.98013	0.98000	0.97368	147	3	4	148
0.63469	0.98013	0.96689	0.96732	0.97987	148	5	3	146
0.64136	0.98675	0.96689	0.96753	0.98649	149	5	2	146
0.74582	0.98675	0.96026	0.96129	0.98639	149	6	2	145
0.80646	0.98675	0.94040	0.94304	0.98611	149	9	2	142
0.81935	0.98675	0.92715	0.93125	0.98592	149	11	2	140
0.86190	0.98675	0.92053	0.92547	0.98582	149	12	2	139
0.93470	0.98675	0.91391	0.91975	0.98571	149	13	2	138
0.93566	0.98675	0.90728	0.91411	0.98561	149	14	2	137
0.93580	0.98675	0.90066	0.90854	0.98551	149	15	2	136
0.95528	0.98675	0.89404	0.90303	0.98540	149	16	2	135
0.96036	0.98675	0.88742	0.89759	0.98529	149	17	2	134
0.96096	0.98675	0.88079	0.89222	0.98519	149	18	2	133
0.96411	0.98675	0.85430	0.87135	0.98473	149	22	2	129
0.98326	0.98675	0.84768	0.86628	0.98462	149	23	2	128
0.98680	0.98675	0.81457	0.84181	0.98400	149	28	2	123
0.98786	0.98675	0.80795	0.83708	0.98387	149	29	2	122
0.98852	0.98675	0.80132	0.83240	0.98374	149	30	2	121
0.98928	0.98675	0.79470	0.82778	0.98361	149	31	2	120
0.98945	0.98675	0.78808	0.82320	0.98347	149	32	2	119
0.98947	1.00000	0.70199	0.77041	1.00000	151	45	0	106
0.99279	1.00000	0.69536	0.76650	1.00000	151	46	0	105
0.99364	1.00000	0.68212	0.75879	1.00000	151	48	0	103
0.99374	1.00000	0.66225	0.74752	1.00000	151	51	0	100
0.99582	1.00000	0.65563	0.74384	1.00000	151	52	0	99
0.99792	1.00000	0.64901	0.74020	1.00000	151	53	0	98
0.99793	1.00000	0.63576	0.73301	1.00000	151	55	0	96
0.99800	1.00000	0.62252	0.72596	1.00000	151	57	0	94
0.99817	1.00000	0.59603	0.71226	1.00000	151	61	0	90
0.99820	1.00000	0.52980	0.68018	1.00000	151	71	0	80
0.99859	1.00000	0.52318	0.67713	1.00000	151	72	0	79
0.99949	1.00000	0.48344	0.65939	1.00000	151	78	0	73
0.99954	1.00000	0.47020	0.65368	1.00000	151	80	0	71
0.99965	1.00000	0.45695	0.64807	1.00000	151	82	0	69
0.99975	1.00000	0.45033	0.64530	1.00000	151	83	0	68
0.99978	1.00000	0.43709	0.63983	1.00000	151	85	0	66
0.99979	1.00000	0.43046	0.63713	1.00000	151	86	0	65
0.99980	1.00000	0.41722	0.63180	1.00000	151	88	0	63
0.99988	1.00000	0.39735	0.62397	1.00000	151	91	0	60
0.99991	1.00000	0.37086	0.61382	1.00000	151	95	0	56
0.99992	1.00000	0.35762	0.60887	1.00000	151	97	0	54

0.99994	1.00000	0.33775	0.60159	1.00000	151	100	0	51
0.99996	1.00000	0.30464	0.58984	1.00000	151	105	0	46
0.99997	1.00000	0.19205	0.55311	1.00000	151	122	0	29
0.99998	1.00000	0.18543	0.55109	1.00000	151	123	0	28
0.99999	1.00000	0.06623	0.51712	1.00000	151	141	0	10
1.00000	1.00000	0.00000	0.50000	1.00000	151	151	0	0

Tabela zawiera kolejne progi, dla których wartości w pozostałych kolumnach się zmieniają. Z tabeli wynika, że prawdopodobnie najbardziej przydatnym progiem (thre) jest 0.61, dla którego wartości w kolumnach sens i ppv wynoszą 0.98 i 0.97.

Poniżej znajduje się fragment (co trzeci wiersz, za wyjątkiem ostatnich wierszy) analogicznej tabeleli dla modelu ecnormal.

thre	spec	sens	npv	ppv	pncn	pncy	pycn	pycy
0.00002	0.82172	1.00000	1.00000	0.01216	56553	0	12270	151
0.00005	0.86021	1.00000	1.00000	0.01545	59202	0	9621	151
0.00008	0.86298	1.00000	1.00000	0.01576	59393	0	9430	151
0.00011	0.88231	1.00000	1.00000	0.01830	60723	0	8100	151
0.00016	0.88715	1.00000	1.00000	0.01907	61056	0	7767	151
0.00019	0.89402	1.00000	1.00000	0.02028	61529	0	7294	151
0.00024	0.90403	1.00000	1.00000	0.02235	62218	0	6605	151
0.00030	0.90415	1.00000	1.00000	0.02238	62226	0	6597	151
0.00036	0.90506	1.00000	1.00000	0.02259	62289	0	6534	151
0.00040	0.90907	1.00000	1.00000	0.02356	62565	0	6258	151
0.00044	0.90962	1.00000	1.00000	0.02370	62603	0	6220	151
0.00049	0.91063	1.00000	1.00000	0.02396	62672	0	6151	151
0.00057	0.92024	0.99338	0.99998	0.02660	63334	1	5489	150
0.00062	0.93180	0.99338	0.99998	0.03097	64129	1	4694	150
0.00073	0.93271	0.99338	0.99998	0.03137	64192	1	4631	150
0.00088	0.93331	0.99338	0.99998	0.03165	64233	1	4590	150
0.00114	0.94296	0.96689	0.99992	0.03585	64897	5	3926	146
0.00118	0.94330	0.96689	0.99992	0.03607	64921	5	3902	146
0.00135	0.94418	0.96689	0.99992	0.03661	64981	5	3842	146
0.00154	0.94441	0.96689	0.99992	0.03676	64997	5	3826	146
0.00169	0.94454	0.96689	0.99992	0.03684	65006	5	3817	146
0.00181	0.94540	0.96689	0.99992	0.03740	65065	5	3758	146
0.00202	0.94548	0.96689	0.99992	0.03746	65071	5	3752	146
0.00220	0.94601	0.96689	0.99992	0.03780	65107	5	3716	146
0.00235	0.94666	0.96689	0.99992	0.03825	65152	5	3671	146
0.00280	0.95910	0.96026	0.99991	0.04899	66008	6	2815	145
0.00339	0.96270	0.95364	0.99989	0.05312	66256	7	2567	144
0.00418	0.96305	0.95364	0.99989	0.05359	66280	7	2543	144
0.00519	0.96575	0.94702	0.99988	0.05720	66466	8	2357	143
0.00652	0.96609	0.94702	0.99988	0.05773	66489	8	2334	143
0.00656	0.96642	0.94040	0.99986	0.05789	66512	9	2311	142
0.00716	0.96658	0.94040	0.99986	0.05815	66523	9	2300	142
0.00834	0.96673	0.94040	0.99986	0.05839	66533	9	2290	142
0.00872	0.96694	0.93377	0.99985	0.05836	66548	10	2275	141
0.00995	0.96738	0.93377	0.99985	0.05909	66578	10	2245	141
0.01020	0.96830	0.93377	0.99985	0.06070	66641	10	2182	141
0.01213	0.96875	0.93377	0.99985	0.06152	66672	10	2151	141
0.01286	0.96909	0.92715	0.99984	0.06176	66696	11	2127	140
0.01677	0.97571	0.90066	0.99978	0.07522	67151	15	1672	136
0.02104	0.97584	0.90066	0.99978	0.07560	67160	15	1663	136
0.02311	0.97601	0.90066	0.99978	0.07611	67172	15	1651	136
0.02906	0.97616	0.90066	0.99978	0.07653	67182	15	1641	136
0.03033	0.97790	0.90066	0.99978	0.08208	67302	15	1521	136
0.03509	0.97825	0.89404	0.99976	0.08272	67326	16	1497	135
0.03739	0.97940	0.89404	0.99976	0.08693	67405	16	1418	135
0.04502	0.97954	0.88742	0.99975	0.08690	67415	17	1408	134
0.05300	0.98110	0.87417	0.99972	0.09211	67522	19	1301	132
0.05490	0.98150	0.86755	0.99970	0.09330	67550	20	1273	131
0.06729	0.98197	0.86755	0.99970	0.09548	67582	20	1241	131
0.09588	0.98269	0.85430	0.99967	0.09773	67632	22	1191	129
0.10641	0.98294	0.85430	0.99967	0.09900	67649	22	1174	129
0.14224	0.98320	0.84768	0.99966	0.09969	67667	23	1156	128
0.14754	0.98734	0.80132	0.99956	0.12198	67952	30	871	121
0.17693	0.98872	0.73510	0.99941	0.12514	68047	40	776	111
0.20963	0.98886	0.73510	0.99941	0.12642	68056	40	767	111
0.25759	0.98938	0.71523	0.99937	0.12872	68092	43	731	108
0.33603	0.98952	0.71523	0.99937	0.13028	68102	43	721	108
0.38412	0.99282	0.62914	0.99918	0.16129	68329	56	494	95
0.43649	0.99321	0.62252	0.99917	0.16756	68356	57	467	94
0.44843	0.99353	0.60927	0.99914	0.17132	68378	59	445	92
0.53240	0.99423	0.59603	0.99911	0.18480	68426	61	397	90
0.65013	0.99458	0.57616	0.99907	0.18913	68450	64	373	87
0.67007	0.99470	0.57616	0.99907	0.19248	68458	64	365	87
0.73986	0.99491	0.56954	0.99905	0.19725	68473	65	350	86
0.78757	0.99555	0.55629	0.99902	0.21538	68517	67	306	84
0.89444	0.99653	0.53642	0.99898	0.25312	68584	70	239	81
0.91698	0.99776	0.47682	0.99885	0.31858	68669	79	154	72
0.96502	0.99818	0.46358	0.99882	0.35897	68698	81	125	70
0.97103	0.99821	0.46358	0.99882	0.36269	68700	81	123	70
0.97914	0.99908	0.43709	0.99877	0.51163	68760	85	63	66
0.98088	0.99930	0.41722	0.99872	0.56757	68775	88	48	63
0.98914	0.99930	0.39735	0.99868	0.55556	68775	91	48	60
0.98955	0.99930	0.38411	0.99865	0.54717	68775	93	48	58
0.99104	0.99930	0.37086	0.99862	0.53846	68775	95	48	56
0.99350	1.00000	0.34437	0.99856	1.00000	68823	99	0	52
0.99413	1.00000	0.33113	0.99853	1.00000	68823	101	0	50
0.99455	1.00000	0.32450	0.99852	1.00000	68823	102	0	49
0.99647	1.00000	0.31788	0.99851	1.00000	68823	103	0	48
0.99709	1.00000	0.30464	0.99848	1.00000	68823	105	0	46
0.99802	1.00000	0.27815	0.99842	1.00000	68823	109	0	42
0.99936	1.00000	0.25828	0.99838	1.00000	68823	112	0	39
0.99967	1.00000	0.23841	0.99833	1.00000	68823	115	0	36
0.99973	1.00000	0.23179	0.99832	1.00000	68823	116	0	35
0.99982	1.00000	0.22517	0.99830	1.00000	68823	117	0	34
0.99994	1.00000	0.19868	0.99824	1.00000	68823	121	0	30
0.99995	1.00000	0.19205	0.99823	1.00000	68823	122	0	29
0.99996	1.00000	0.18543	0.99822	1.00000	68823	123	0	28
0.99997	1.00000	0.17881	0.99820	1.00000	68823	124	0	27

```
0.99998  1.00000  0.13245  0.99810  1.00000          68823    131      0        20
0.99999  1.00000  0.11921  0.99807  1.00000          68823    133      0        18
1.00000  1.00000  0.00000  0.99781  1.00000          68823    151      0        0
```

Tym razem widać, że interesującym progiem jest 0.993, wtedy wartość w kolumnie ppv jest dużo większa niż 0.5 i wynosi 1.0. Odpowiadająca wartość z kolumny sens wynosi ok. 0.3. Wartości te są zbliżone, chociaż nieco lepsze od wartości określających skuteczność modelu (pozytywna wartość predykcyjna 0.97, czułość 0.29) oszacowanych metodą walidacji krzyżowej.

Program ritfind używa tabeli osiągów, w celu określenia progu dla atrybutu confidence(yes) zwracanego przez model, dla którego wartość z kolumny ppv będzie większa od wartości podanej przez użytkownika przy uruchamianiu programu (parametr threshold).

Wypisywaną przez program przy każdym kandydacie wartość ppv należy interpretować jako jakość terminatora.

2.10. Obsługa programu ritfind

Standardowym trybem pracy z programem ritfind jest praca z linii komend / powłoki systemowej. Uruchomienie programu ritfind z parametrem `--help` lub bez żadnych parametrów powoduje wypisanie krótkiej informacji na temat sposobu uruchamiania:

```
$ ritfind --help
usage: ritfind [options]

options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -m MODEL, --model=MODEL
                        Classifier model: ecnormal, ec151t151f [default:
                        ecnormal]
  -t THRESHOLD, --threshold=THRESHOLD
                        Set ppv threshold
  --performance         Show model performance table
  --fasta=FILENAME      Sequence to scan for terminators
  --ptt=FILENAME        Genes in sequence
  -c, --classify        Show putative terminators
```

Aby wyszukać terminatory w genomie z pliku `genom.fasta` i opisem pozycji genów z pliku `geny.ptt` należy wydać polecenie:

```
$ ritfind --classify --fasta genom.fasta --ptt geny.ptt
```

Aby wybrać inny niż domyślny (ecnormal) model używany przez program ritfind polecenie należy uzupełnić parametrem `--model`. Dostępne są dwa modele — ecnormal (domyślny), oraz ec151t151f. Parametrem `--threshold` można ograniczyć liczbę wypisywanych na wyjściu kandydatów. Jego domyślna wartość to 0.75. Wartość bliższa 1 zmniejszy liczbę wypisywanych kandydatów, wartość bliższa 0 zwiększy liczbę wypisywanych kandydatów.

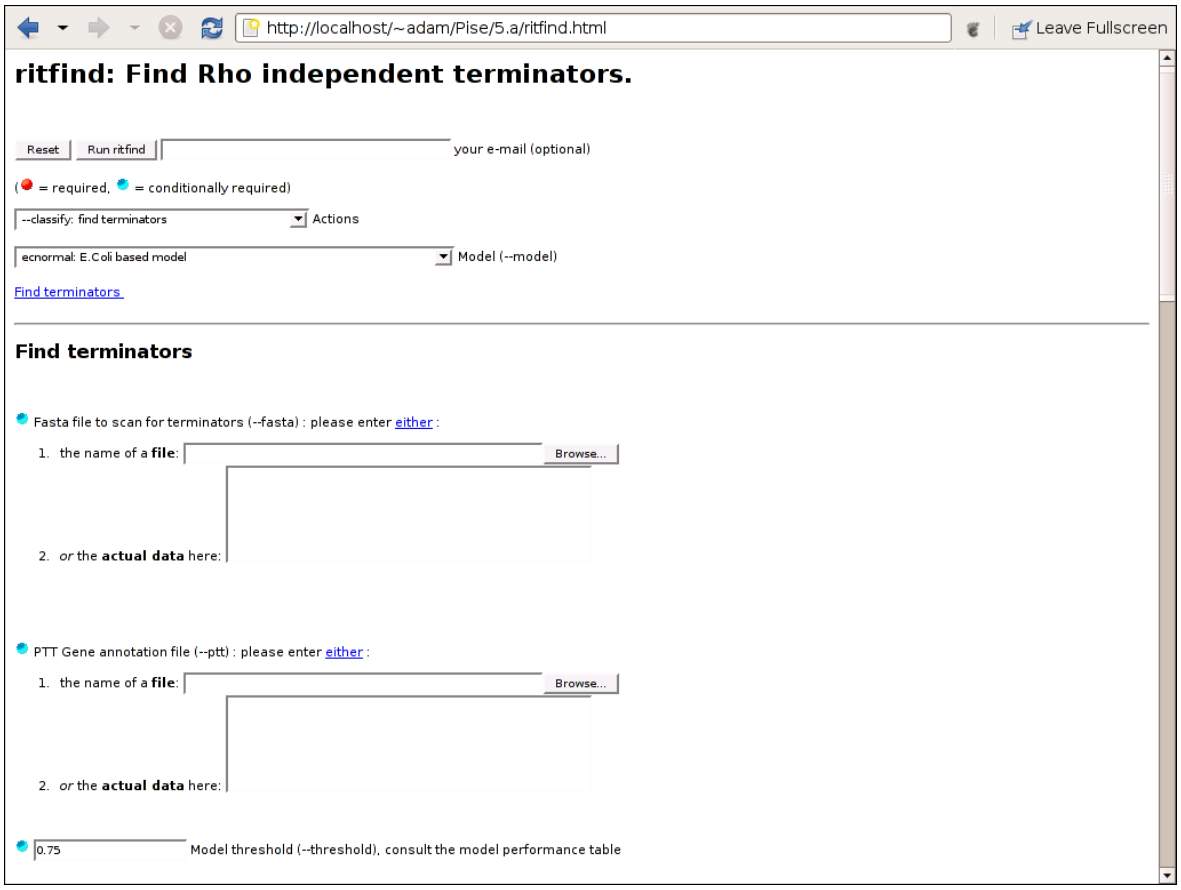
Opcja `--performance` pozwala na wypisanie tablicy osiągów dla wybranego modelu.

Alternatywnym sposobem używania programu ritfind jest korzystanie z interfejsu WWW. Interfejs WWW dla programu przygotowałem przy użyciu pakietu Pise [L01] — generatora interfejsów WWW dla programów uruchamianych z linii komend. Pise na wejściu przyjmuje

definicje interfejsu WWW (definicja dla programu ritfind znajduje się w dodatku B), na wyjściu generuje odpowiednie pliki .html oraz .cgi, które należy umieścić wraz z Pise na serwerze WWW. Pise i pliki .cgi generowane przez Pise są programami w języku programowania perl.

Rysunek 2.9 przedstawia wygląd strony WWW, przez którą można uruchomić ritfind. Strona umożliwia ustawienie wszystkich parametrów dostępnych z linii komend. Pliki wejściowe można wysłać na serwer, zawartość krótszych plików można wkleić w pola tekstowe. Tryb pracy (wyszukiwanie terminatorów, wypisywanie tabeli osiągów) i model (ecnormal, ec151t151f) można wskazać w polu wyboru. Próg dla wypisywanych terminatorów (paramter `--threshold`) można wpisać w odpowiednie pole tekstowe (lub pozostawić domyślną wartość).

Kliknięcie guzika *Run ritfind* powoduje przesłanie wybranych plików i parametrów na serwer i uruchomienie programu ritfind. Po zakończeniu obliczeń wyniki dostępne są na stronie WWW. Jeśli zostanie wypełnione pole *email*, na podany adres zostanie wysłane zawiadomienie o gotowych wynikach, jeśli obliczenia będą trwały dłuższy czas.



Rysunek 2.9: Interfejs WWW programu ritfind wygenerowany przez Pise.

Rozdział 3

Eksperyment: *E. Coli*, bakteriofag P1

W tym rozdziale znajdują się wyniki eksperymentów — szukania Rho-niezależnych terminatorów w genomie *E. Coli* oraz, w genomie bakteriofaga P1, atakującego m.in. bakterię *E. Coli*. Do szukania terminatorów użyłem 3 programów: ritfind, TranstermHP oraz rnall.

3.1. Bakteriofag P1

Przedmiotem eksperymentu był genom bakteriofaga P1 długości 94800 nukleotydów, wraz z listą 109 genów.

Uruchomienie programu rnall z domyślnymi parametrami na tym genomie zwraca listę 58 kandydatów. Program rnall nie dokonuje oceny jakości zwracanych kandydatów, jego działanie polega na wyszukaniu kandydatów których parametry są większe od określonych progów.

Program ritfind uruchomiony ze standardowymi parametrami, czyli z modelem ecnormal oraz wypisywaniem kandydatów dla których pozytywna wartość predykcyjna jest większa od 0.50 wskazuje 1 ze 1567 kandydatów, jednak jego jakość (ppv) jest bliska 0.50.

```
ppv, conf(yes): 0.511627906977 0.97914
dir, first, last: 1 83736 83760
term: CAGTGTCACTTTTAA TGCTGGTGG AGTGCGC CCACCAGCA TTTTTTTCGTCCAAT
gene dist: (33, 'PmgS ')
hp_e, t_e, hyb_e: -11.5 -5.52986 -5.371
```

Program TranstermHP uruchomiony na bakteriofagu P1 znajduje 38 Rho-niezależnych terminatorów, jakość 11 oceniona jest na 100 (najwyższa). Jakość kandydata o energii spinki H i wadze ogona T znalezionego w genomie o zawartosci nukleotydów GC równej gc w programie TranstermHP określona jest wzorem:

$$C_{\mathsf{gc}}(H,T) = \frac{-100}{\log(L)} \log\left(\frac{\max(1, R_{\mathsf{gc},L}(H,T))}{L}\right),$$

gdzie $R_{\mathsf{gc},L}(H,T)$ to liczba kandydatów o energii spinki mniejszej lub równej H i wadze ogona mniejszej lub równej T w losowym genomie długości L o zawartości nukleotydów GC równej gc. W programie TranstermHP wartość $C_{\mathsf{gc}}(H,T)$ obliczana jest w sposób przybliżony na podstawie prekalkulowanych tablic obliczonych dla L długości 20M nukleotydów.

Aby sprawdzić, czy ritfind zwraca podobny zestaw terminatorów do zestawu zwracanego przez TranstermHP uruchomiłem ritfind z progiem obniżonym do 0.1. Powoduje to wypisanie

22 kandydatów, poniżej znajduje się 5 najlepiej ocenionych przez ritfind, oraz odpowiadające im wyniki z rnall (wiersz zaczynający się od StrNo i kolejny), następnie z TransTermHP (wiersz zaczynający się od TERM i kolejny).

Można zauważyć, że ritfind/ecnormal zwraca podobne wyniki (najlepiej oceniani kandydaci znajdują się w liscie kandydatów zwracanych przez TranstermHP), lecz znacznie niżej ocenia ich jakość.

```
ppv, conf(yes): 0.511627906977 0.97914
dir, first, last: 1 83736 83760
term: CAGTGTCACTTTTAA TGCTGGTGG AGTGCGC CCACCAGCA TTTTTTTCGTCCAAT
gene dist: (33, 'PmgS ')
hp_e, t_e, hyb_e: -11.5 -5.52986 -5.371

StrNo:54 start=83734 stop=83762 dG=-17.6 MPS=-56 Twt=4.7373 hbG=-7.888
seq=ACAGUGU%CACUUUU/AAUGCUGGUGG AGUGCGC CCACCAGCAUU/UUUUUCGU#CCAAUGA

TERM 34       83736 - 83760    + F   100 -11.5 -5.52986 |
CAGTGTCACTTTTAA          TGCTGGTGG AGTGCGC CCACCAGCA          TTTTTTTCGTCCAAT

-----------------------
ppv, conf(yes): 0.253125 0.89444
dir, first, last: -1 52912 52950
term: ATGTGTAGAAAGAGC GATCGAACCCGATACATA GCAA TATGTGTCGGGTTC-AGC TTTTTATGTCCCAAG
gene dist: (144, 'PmgF ')
hp_e, t_e, hyb_e: -10.5 -4.96848 -5.742

[brak odpowiednika]

TERM 20       52950 - 52912    - T    93 -10.5 -4.96848 |
CTTGGGACATAAAAA  GCT-GAACCCGACACATA TTGC TATGTATCGGGTTCGATC   GCTCTTTCTACACAT

-----------------------
ppv, conf(yes): 0.243975903614 0.85817
dir, first, last: 1 2046 2063
term: CTCCTTGACATCATT GGCGGCC ATTA GGCCGCC TTTTTTTTGCCATAT
gene dist: (224, 'C8 ')
hp_e, t_e, hyb_e: -14.1 -5.79526 -4.93

StrNo:2 start=2046 stop=2063 dG=-15 MPS=-39 Twt=5.79526 hbG=-5.074
seq=UCCUUGA%CAUCAUU/GGCGGCC AUUA GGCCGCC/UUUUUUUU#GCCAUAU

TERM 1         2046 - 2063     + F   100 -14.1 -5.79526 |
CTCCTTGACATCATT             GGCGGCC ATTA GGCCGCC             TTTTTTTTGCCATAT

-----------------------
ppv, conf(yes): 0.23275862069 0.81387
dir, first, last: 1 34320 34343
term: AACAACCTGAATGAG ACAAGGCCCG ATAG CGGGCCTTAA TTTTTATTCAGGCTT
gene dist: (12, 'gpU prime ')
```

```
hp_e, t_e, hyb_e: -6.6 -5.08088 -5.247

StrNo:20 start=34318 stop=34345 dG=-14.7 MPS=-49 Twt=4.19974 hbG=-8.656
seq=GAACAAC%CUGAAUG/AGACAAGGCCCG AUAG CGGGCCUUAAUU/UUUAUUCA#GGCUUUU

TERM 12       34320 - 34343    + T    84  -6.6 -5.08088 |
AACAACCTGAATGAG          ACAAGGCCCG ATAG CGGGCCTTAA           TTTTTATTCAGGCTT

-----------------------
ppv, conf(yes): 0.23275862069 0.81387
dir, first, last: -1 22550 22569
term: TAATTTCGTATTATC AGGGTGCT CTCT GGCACCCC TTTTTATTCTAGTAA
gene dist: (83, 'InsB ')
hp_e, t_e, hyb_e: -5.6 -5.22006 -5.224
[brak odpowiednika]

TERM 7        22569 - 22550    - T    83  -5.6 -5.22006 |
TTACTAGAATAAAAA            GGGGTGCC AGAG AGCACCCT             GATAATACGAAATTA
```

W publikacji o bakteriofagu P1 [ŁRPRSLYB04] znajduje się lista 36 potencjalnych Rho-niezależnych terminatorów. Lista zawiera terminatory znalezione przez programy Transterm oraz Terminator, które dodatkowo spełniały pewne warunki (określone zakresy długości nogi, pętli i kilka innych). Co najmniej 11 (dokładna liczba trudna do ustalenia, ze względu na inną wersję genomu w publikacji) terminatorów z tej listy znajduje się w wynikach TranstermHP. Jakość 5 terminatorów z tych 11 TranstermHP ocenił na 100.

Skoro ritfind/ecnormal znajduje tylko 1 terminator, a TranstermHP znajduje 38 terminatory, w tym co najmniej 11 jest zgodnych z listą sporządzoną przez 2 programy ocenioną dodatkowo przez biologów, a 5 z 11 zostało ocenionych najwyżej, można mieć wątpliwości, czy ritfind działa poprawnie.

Ale program ritfind uruchomiony z modelem ec151t151f na genomie faga P1 zwraca 1567 kandydatów z jakością większą od 0.5, w tym 134 z większą od 0.95 i 35 z jakością równą 1.0. Wśród 35 kandydatów z jakością 1.0 znajduje się 24 kandydatów, które należą też do listy 38 zwracanej przez TranstermHP. Natomiast 9 z tych 24 kandydatów należy do listy 11 wspólnych terminatorów z publikacji i wyników TranstermHP.

Czyli jeśli chodzi o najwyżej ocenionych kandydatów, 9 z 35 kandydatów znalezionych przez ritfind/ec151t151f i 5 z 11 kandydatów znalezionych przez TranstermHP znajduje się w publikacji.

Podsumowując — wskazywane są podobni kandydaci, ale jakość kandydatów zwracana przez model ecnormal jest dużo niższa, niż przez TranstermHP i ec151t151f. Model ec151t151f zwraca jakość kandydatów wyższą niż TranstermHP.

Jeśli genom faga P1 ma podobny charakter jeśli chodzi o terminatory transkrypcji co genom *E. Coli* i prawdziwe jest założenie, że na 456 kandydatów analizowanych przez ritfind przypada 1 Rho-niezależny terminator, to liczba Rho-niezależnych terminatorów powinna wynosić ok. 3-4, zaś terminatory zwracane przez TranstermHP, rnall i ritfind/ec151t151f to w większości fałszywe pozytywy. Ponieważ czułość programu ritfind/ecnormal dla wysokiej wartości predykcyjnej wynosi ok. 29% brak wyników programu ritfind/ecnormal dla genomu faga P1 jest zrozumiały.

3.2. *Escherichia Coli*

W kolejnym eksperymencie badany był genom *E. Coli* długości 4639675 nukleotydów, wraz z listą 4243 genów.

Z założenia o liczbie Rho-niezależnych terminatorów w stosunku do liczby kandydatów znajdowanych przez ritfind wynika, że w genomie *E. Coli* znajduje się ok. 186 Rho-niezależnych terminatorów.

Program ritfind/ecnormal wypisuje 146 kandydatów z jakością większą od 0.5.

Jakość 84604 kandydatów oceniona została na mniej niż 0.5. Wśród 146 kandydatów z jakością większą od 0.5, 58 ma jakość 1.0, 17 ma jakość 0.57, pozostałe 71 ma jakość 0.51.

Program rnall znalazł 2515 kandydatów.

TranstermHP znajduje 84747 kandydatów, w tym 35012 kandydatów niepokrywających się nawet częściowo z innymi kandydatami. 1186 kandydatów ma jakość 100. 37828 ma jakość większą od 50, 1321 od 95.

Wszystkie 58 kandydatów z jakością 1.0 znalezionych przez ritfind znajduje się na liście 1186 kandydatów ocenionych na 100 zwracanej przez TranstermHP. 17 kandydatów z jakością 0.57 również znajduje się na tej liście. 61 kandydatów ocenionych przez ritfind na 0.51 przez TranstermHP ocenianych jest na 100, pozostałe 11 kandydatów reszty znalezionych przez ritfind zostało ocenionych przez TranstermHP na mniej niż 100.

Program ritfind/ec151t151f wypisuje 84750 kandydatów z jakością większą od 0.50. 1933 ma jakość 1.0. 7007 ma jakość większą od 0.95.

Wyniki programu TranstermHP na zbiorze przykładów pozytywnych i negatywnych na których trenowany był ritfind są następujące: dla 151 przykładów pozytywnych TranstermHP podaje 26 terminatorów z jakością 100, wśród 68823 przykładów negatywnych znajduje 883 terminatory, w tym 3 terminatory z jakością 100, 35 z jakością większą od 95, 548 z jakością większą od 50. Biorąc pod uwagę tylko terminatory ocenione na 100, czułość TranstermHP na tym zbiorze danych wynosi 0.17, przy pozytywnej wartości predykcyjnej 0.90. Uwzględnienie wszystkich kandydatów zwracanych przez TranstermHP obniża pozytywną wartość predykcyjną do 0.15.

3.3. Podsumowanie

Program ritfind używa tych samych charakterystyk terminatorów co inne programy, ale klasyfikuje kandydatów używając sieci bayesowskich, a nie korzystając z modelu typu termodynamicznego (próba modelowania reakcji terminacji) czy poprzez obliczanie kombinacji liniowych atrybutów.

Wyniki zwracane przez program ritfind są zbliżone do wyników zwracanych przez konkurencyjne programy, chociaż ritfind surowiej ocenia jakość znajdowanych terminatorów.

Do zalet programu ritfind można zaliczyć przynajmniej teoretyczne wykorzystanie większej liczby atrybutów, co powinno zwiększyć skuteczność klasyfikacji. Program ritfind wyróżnia się też sposobem oceny skuteczności (walidacja krzyżowa na dużej i proporcjonalnej liczbie przykładów pozytywnych i negatywnych). Zaletą może być modułowa budowa, pozwalająca na łatwiejsze eksperymenty i optymalizację klasyfikatora w zewnętrznym narzędziu do dataminingu.

Wadą programu ritfind jest zajmujowanie znacznie więcej przestrzeni dyskowej od innych programów, instalacja jest trudniejsza ze względu na konieczność instalacji wymaganych przez ritfind programów i bibliotek.

Jeśli dane na których trenowany był program ritfind są zgodne z wynikami doświadczeń biologicznych, a przy implementacji nie popełniłem dużych błędów, to możliwe że program ritfind/ecnormal cechuje się większą czułością (.29) przy wysokiej pozytywnej wartości predykcyjnej (.97), niż konkurencyjne programy (TranstermHP 0.17/0.90).

Dodatek A

Fragment programu ritfind wyliczający główne atrybuty

```
#energia spinki
def split_stemloop(seq):
  cache = {}

  def loop_pen(n):
    return n < 3 and 1000 or n - 2

  def pair_score(n1, n2):
    if '-' in n1+n2:
      return 6.0
    elif 'A' in n1+n2 and 'T' in n1+n2:
      return -0.9
    elif 'C' in n1+n2 and 'G' in n1+n2:
      return -2.3
    elif 'G' in n1+n2 and 'T' in n1+n2:
      return 1.3
    else:
      return 3.5

  def hps(i, j):
    if j <= i:
      return 1000, ''
    try:
      return cache[(i, j)]
    except KeyError:
      nogap = hps(i+1, j-1)
      lgap = hps(i+1, j)
      rgap = hps(i, j-1)
      cache[(i, j)] = min((loop_pen(j - i + 1), ','+seq[i:j+1]+','),
          (pair_score(seq[i], seq[j]) + nogap[0], seq[i]+nogap[1]+seq[j]),
          (pair_score('-', seq[j]) + lgap[0], '-'+lgap[1]+seq[j]),
          (pair_score(seq[i], '-') + rgap[0], seq[i]+rgap[1]+'-'))
      return cache[(i, j)]

  score, seqs = hps(0, len(seq)-1)
  return seqs.split(',') + [score]

#waga ogona
def tail_score(tail):
    if len(tail) != 15:
        raise Exception('tail length != 15')
```

```
    sum = 0.0
    prev = 1.0
    for c in tail:
        if c == 'T':
            prev *= 0.9
        else:
            prev *= 0.6
        sum += prev
    return -sum

#energia ogona
def hybridization(tail):
  def deltaG(seq):
    dict = {'AA':-1.0, 'GC':-2.7, 'TA':-0.6, 'CA':-0.9,
            'AC':-2.1, 'GT':-1.1, 'TG':-1.6, 'CT':-0.9,
            'AG':-1.8, 'GA':-1.3, 'TC':-1.5, 'CG':-1.17,
            'AT':-0.9, 'GG':-2.9, 'TT':-0.2, 'CC':-2.1}
    dG = 0
    for i in xrange(len(seq)-1):
      dG += dict[seq[i:i+2]]
    return dG

  i = tail.find('T', 0, 2)
  if i == -1:
    spacer = tail[:2]
    i = 2
  else:
    spacer = tail[:i]

  proximal = tail[i:i+5]
  distal = tail[i+5:i+9]
  extra = tail[i+9:i+12]

  return deltaG(spacer) + deltaG(proximal) +
     0.5*deltaG(distal) + 0.01*deltaG(extra) - 3.1
```

Dodatek B

Definicja interfejsu WWW programu ritfind dla Pise

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE pise SYSTEM "PARSER/pise.dtd">

<pise>
<head>
 <title>ritfind</title>
 <version>0.01</version>
 <description>Find Rho independent terminators.</description>
</head>

<command>ritfind</command>
<parameters>
 <parameter ismandatory="1" issimple="0" type="Excl">
  <name>actions</name>
  <attributes>
   <prompt>Actions</prompt>
   <format>
    <language>perl</language>
    <code> " $value" </code>
   </format>
   <vdef><value>--classify</value></vdef>
   <group>1</group>
   <vlist>
    <value>--classify</value>
    <label>--classify: find terminators</label>
    <value>--performance</value>
    <label>--performance: show model performance table</label>
   </vlist>
  </attributes>
 </parameter>

 <parameter ismandatory="1" type="Excl">
 <name>model</name>
 <attributes>
  <prompt>Model (--model)</prompt>
  <format>
   <language>perl</language>
   <code> " --model=$value" </code>
  </format>
  <vdef><value>ecnormal</value></vdef>
  <group>2</group>
```

```
  <vlist>
   <value>ecnormal</value>
   <label>ecnormal: E.Coli based model</label>
   <value>ec151t151f</value>
   <label>ec151t151f: E.Coli based toy model (just 151 true, 151 false
       terminators)</label>
  </vlist>
 </attributes>
 </parameter>

 <parameter type="Paragraph">
  <paragraph>
   <name>classify</name>
   <prompt>Find terminators</prompt>
   <precond>
    <language>perl</language>
    <code>($actions =~ /classify/ )</code>
   </precond>
   <group>3</group>
   <parameters>
    <parameter type="InFile" ismandatory="1" issimple="1" ishidden="0">
     <name>fasta</name>
     <attributes>
      <prompt>Fasta file to scan for terminators (--fasta)</prompt>
      <format>
       <language>perl</language>
       <code>" --fasta=$value"</code>
      </format>
      <group>1</group>
     </attributes>
    </parameter>
    <parameter type="InFile" ismandatory="1" issimple="1" ishidden="0">
     <name>ptt</name>
     <attributes>
      <prompt>PTT Gene annotation file (--ptt)</prompt>
      <format>
       <language>perl</language>
       <code>" --ptt=$value"</code>
      </format>
      <group>2</group>
     </attributes>
    </parameter>
    <parameter type="Float" ismandatory="1" issimple="1" ishidden="0">
     <name>threshold</name>
     <attributes>
      <prompt>Model threshold (--threshold), consult the model performance
          table</prompt>
      <format>
       <language>perl</language>
       <code>" --threshold=$value"</code>
      </format>
     <vdef><value>0.75</value></vdef>
      <group>3</group>
     </attributes>
    </parameter>
   </parameters>
  </paragraph>
 </parameter>
</parameters>
</pise>
```

Dodatek C

Przykładowe dane wejściowe — fragment genomu bakteriofaga P1

>gi|46401626|ref|NC_005856.1| Enterobacteria phage P1 virion, complete genome
ACATTATACGAAGTTATATTAAGGGTTATTGAATATGATCAATTTACCTGTAAATCCATACAGTTCAATA
CCTTAGCAGGTCAAATAGTGACCACTTGATCATTTGATCAAGGTTGCGCTACGTAAAATCTGTGAAAAAT
TGGCGGTGTTAGTCCTACAGATTTCGCGTAGCGCACTTAGCACCACCAATCAATCAGAGGTGAAAAATGG
GATATTCAACTGCTAAAGTGTCCACTCATCTTGAGCTTGAGAAAAACCGTGGTTACTGGCGGGCAAAAGG
GTTTGATCGTGATAGTTGCCAACTGTCATTATCGCGCGGTGAAGAGAAAATAGAACGCACGCGCGGTCGC
TGGCGTTTCTATGACGAGAACCATAAACAGGTAAAGGCAGAGCCGATCCTGTACACTTTACTTAAAACCA
TTATCTGAGTGTTAAATGTCCAATTTACTGACCGTACACCAAAATTTGCCTGCATTACCGGTCGATGCAA
CGAGTGATGAGGTTCGCAAGAACCTGATGGACATGTTCAGGGATCGCCAGGCGTTTTCTGAGCATACCTG
GAAAATGCTTCTGTCCGTTTGCCGGTCGTGGGCGGCATGGTGCAAGTTGAATAACCGGAAATGGTTTCCC
GCAGAACCTGAAGATGTTCGCGATTATCTTCTATATCTTCAGGCGCGCGGTCTGGCAGTAAAAACTATCC
AGCAACATTTGGGCCAGCTAAACATGCTTCATCGTCGGTCCGGGCTGCCACGACCAAGTGACAGCAATGC
TGTTTCACTGGTTATGCGGCGGATCCGAAAAGAAAACGTTGATGCCGGTGAACGTGCAAAACAGGCTCTA
GCGTTCGAACGCACTGATTTCGACCAGGTTCGTTCACTCATGGAAAATAGCGATCGCTGCCAGGATATAC
GTAATCTGGCATTTCTGGGGATTGCTTATAACACCCTGTTACGTATAGCCGAAATTGCCAGGATCAGGGT
TAAAGATATCTCACGTACTGACGGTGGGAGAATGTTAATCCATATTGGCAGAACGAAAACGCTGGTTAGC
ACCGCAGGTGTAGAGAAGGCACTTAGCCTGGGGGTAACTAAACTGGTCGAGCGATGGATTTCCGTCTCTG
GTGTAGCTGATGATCCGAATAACTACCTGTTTTGCCGGGTCAGAAAAAATGGTGTTGCCGCGCCATCTGC
CACCAGCCAGCTATCAACTCGCGCCCTGGAAGGGATTTTTGAAGCAACTCATCGATTGATTTACGGCGCT
AAGGATGACTCTGGTCAGAGATACCTGGCCTGGTCTGGACACAGTGCCCGTGTCGGAGCCGCGCGAGATA
TGGCCCGCGCTGGAGTTTCAATACCGGAGATCATGCAAGCTGGTGGCTGGACCAATGTAAATATTGTCAT
GAACTATATCCGTAACCTGGATAGTGAAACAGGGGCAATGGTGCGCCTGCTGGAAGATGGCGATTAGCCA
TTAACGCGTAAATGATTGCTATAATTATTTGATATTTATGGTGACATATGAGAAAGGATTTCAACATCGA

Dodatek D

Lista niezweryfikowanych eksperymentalnie terminatorów z bazy RegulonDB

ECK120015459 9199 9226 forward taatcattctTAGCGTGACCGGGAAGTCGGTCACGCTAcctcttctga
ECK120010860 65800 65821 reverse ccatcaaaaaACCAGGCTTGAGTATAGCCTGGtttcgtttga
ECK120015446 138801 138816 reverse tccggcatgaACAACGCGCACGTTGTcagcaatctg
ECK120010788 360436 360460 reverse gaacttgtagGCCTGATAAGCGCAGCGTATCAGGCaatttttata
ECK120010809 423710 423732 forward ccgtttaacaCGTTCTGGATGAAATCCATATCGcgatagcgca
ECK120010810 423754 423777 forward cagccactgcTCTGACCACAAGTAATTGTTCAGAttgataaaac
ECK120010830 426980 427010 forward gctgttcagaTCACTGGTGCGCGCGGAGTCGCCGCCAGTGAgcaaacgctg
ECK120010826 473487 473525 forward caagcactgcAAAAAACAGCCGGACGGTTTTCACCTCCGGCTATTTTTTTtaattgtgat
ECK120010853 496222 496253 forward ttcctgatgtAATGCCGGATGACCTTCGTGTCATCCGGCATTtttcttttca
ECK120010804 498169 498204 reverse aaaaatatgaATATATTCCGGCGCTTAATGCCACGCCGGAACATATcgaaatgatg
ECK120015460 602600 602632 forward caaccatccgAAACCGCTCTCATCCATTCGATGAGAGCGGTTTttttaattac
ECK120010866 709341 709357 reverse aacaaataagTGAGAGCTGTAACTCTCgcttttctta
ECK120015444 714434 714465 forward acatttaataAAAAAAGGGCGGTCGCAAGATCGCCCTTTTTTacgtatgaca
ECK120010859 786020 786041 forward gcaacaaaaaAGCCGACTCACTTGCAGTCGGCtttctcattt
ECK120015462 898871 898888 reverse agcaataaaaAAGCCGCATGTGCGGCTTcagattgctg
ECK120015461 899918 899948 reverse tgaaataaaaAAGACGGACAACTTAGTGGGTTGTCCGTCTTcattataaga
ECK120010842 1036831 1036857 forward ccgacgtaaaAAGACGGTAAGTATCGCTTTCAGTCTTatgaatatcg
ECK120015445 1066056 1066080 forward ataacagaaaACTCCCCCGCGAGAAGCGGGGGAGTcgctggttaa
ECK120010829 1085294 1085318 forward gttatcggtgCAGAGCCCGGGCGAACCGGGCTTTGttttgggtgt
ECK120010783 1112755 1112783 forward acgagccaatAAAAATACCGGCGTTATGCCGGTATTTTTTttacgaaaga
ECK120010782 1158548 1158569 reverse tggggagactAAGGCAGCCAGATGGCTGCCTTttttacaggt
ECK120010828 1332869 1332898 forward tgcgttatttTCGGCACCTTTTATGTAGCGAAGGTGCCGGaatatattct
ECK120010787 1386291 1386325 forward agtaagcgcgAATATGCCTGATGGTGCAACACCATCAGGCATATTaaattatgct
ECK120015453 1814282 1814306 forward aaacacgtagGCCTGATAAGCGAAGCGCATCAGGCagttttgcgt
ECK120015526 1942243 1942272 reverse gaacacatttGTCGGATGCGGCGCGAGCGCCTTATCCGACctacggttcg
ECK120015457 1942297 1942327 reverse ctctggtagtCCTGGTAAGACGCGAACAGCGTCGCATCAGGcatattgcca
ECK120015439 1944839 1944878 reverse tgcggagggaGTGGAAACGCCTCAGCCGGAACTGACCGAGGCGGTATAACttaacgcagt
ECK120015458 1946686 1946717 reverse ccttcgctcaTCCCATTCACATAGTTATCTATGCTCATGGGAgttcactcag
ECK120010840 1956471 1956497 reverse ggataaaaaaAGGCCCCTGTTGAAATTGCAGGGGCCTggtacgagca
ECK120010837 2004135 2004189 forward
cgaataatccGATTACGGCTACGCTTCTAATGTTCCCCTTGAATGGAGTCGAAGAATGCGTAATCccacgctgtt
ECK120015448 2005666 2005698 forward gaggtgatttAAATTCATCCCCGGCGGCAAGCCGGGGAGATTTcattacggca
ECK120015450 2217682 2217704 reverse ccggcctacaAAGGGCGCAAACGTCGCGCCCTTactaaagcat
ECK120010781 2309632 2309646 reverse caatgaaaaaAGGGCCCGCAGGCCCtttgttcgat
ECK120015169 2354745 2354774 forward ctcctttctgATGCCCGGTAAGCATGTGGTTACCGGGCATttttgcgtac
ECK120015447 2506446 2506469 reverse tttaaaagatTATCGGGAGAGTTACCTCCCGATAtaaaaggaag
ECK120010867 2520720 2520742 reverse tatcataaaaGCAGCTCTGAAGAGCAGAGCCGCgaatccttt
ECK120010784 2595825 2595846 reverse atcagaaaaaAGGGCCGGATGATTCCAGCCCTgtatttttac
ECK120015451 2840499 2840523 reverse aaccatgaagGCCCGATAAGCGCAGCGCATCGGGCaatttagcat
ECK120010835 2871256 2871306 reverse
cgcaaataacCAGGAGATAAAACCGACCACGGCACCAGGCAGTGACCATGTGGTTTCTTCAtcctcagtaa
ECK120010839 3089111 3089134 forward ttaaccctaCCCCACGCGTACAACCGCGTGGGGagacgacgcg
ECK120010813 3148574 3148598 reverse gttaaaaaaaTGCCCGGTTGTGAAAAGCAACCGGGcatcattgtg
ECK120010851 3322964 3322991 reverse ttgataagaaAAAACCCCGGAGCACGCCCCGGGGTTTTcggtacaaat
ECK120010811 3346283 3346311 forward tgacatacaaCTCCCTTCAAACTCCCCCCGACAATAAGAaaatcacgta
ECK120015455 3553817 3553842 forward gtgtaagtttAGCCGGATAACGCGCCAGATCCGGCTtacatctctg
ECK120010816 3557776 3557853 reverse
cagctttttgACGTTTCGCGCCAGGGCAAAGCCAGCGTCAATGACGCGGATCGGCTACGCAAATACCTGGCGCGTTTTGGTCTGACGTgggaagccgt

ECK120010846 3638593 3638614 forward cgtgttcctgAACGCCCGCATATGCGGGCGTTtttgctttttg
ECK120010847 3737553 3737593 forward aacagtaactTTTCCGGCTTCCCGTTCGTCAGTACCTCGGGAAGCCGCCAAccaggataaa
ECK120010831 3752929 3752964 reverse tcaggttcccGTAGGCATGATAAGACGCGTAAGCGTCGCATCAGGCaatgaatacc
ECK120010808 3904616 3904657 reverse tataaacaaaAAAACCCGACTTCACCAGTATTCTCTGGTTATGTCAGGTTTTgcctgcgaat
ECK120010864 3909833 3909853 reverse aggaagaaaaATGCCCCGCTTACGCAGGGCAtccatttatt
ECK120010790 4104384 4104402 forward tccctgtcttCCCCCACATGCTGTGGGGGttttttttat
ECK120015449 4112550 4112571 reverse aacaaaaaaaTTCCAGTCCCGAAGGACTGGAAggctcaatcg
ECK120010819 4198605 4198624 forward agttttaacgAAGGGGTGGTTTCACCCCTTtttgtctttct
ECK120010836 4275950 4275980 forward aagatgaacaAAACTAAAGCGCCACAAGGGCGCTTTAGTTTgttttccggt
ECK120010785 4311824 4311850 forward agaaattgagATTCATCCACTACTTGCATGGATGAGTaatgattaat
ECK120010827 4336233 4336260 reverse aaggccaggcGGCGTAATGTTATTTAAACAATTACGCCttcagcggaa
ECK120010856 4343654 4343673 reverse aaaagaggtgGCCAGGGGGATCACCTGGCAgcatgctgcc
ECK120010858 4364867 4364887 reverse acaagaaaaaAGGCACGTCATCTGACGTGCCttttttattt
ECK120015957 4464307 4464321 reverse tcacaatttaTCGTTCAGGAAACGAtcaggacagg
ECK120010876 4637561 4637595 forward taaggttgaaAAATAAAAACGGCGCTAAAAAGCGCCGTTTTTTTTgacggtggta

Bibliografia

[1] Wikipedia, *Prokaryote — Wikipedia, The Free Encyclopedia*, `http://en.wikipedia.org/w/index.php?title=Prokaryote&oldid=158095909`, accessed 17-September-2007.

[2] Wikipedia, *Transcription (genetics) — Wikipedia, The Free Encyclopedia*, urlhttp://en.wikipedia.org/w/index.php?title=Transcription_(genetics)&oldid=157069321, accessed 17-September-2007.

[KAS07] Carleton L Kingsford, Kunmi Ayanbule, Steven L Salzberg, *Rapid, accurate, computational discovery of Rho-independent terminators illuminates their relationship to DNA uptake*, Genome Biology, 8:R22, February 2007.

[BT84] Brendel, V. and Trifonov, E.N., *A computer algorithm for testing potential prokaryotic terminators*, Nucl. Acids Res. 12, 4411-4427, 1984.

[EKWSS00] Maria D. Ermolaeva, Hanif G. Khalak, Owen White, Hamilton O. Smith and Steven L. Salzberg, *Prediction of Transcription Terminators in Bacterial Genomes*, J Mol Biol 301, (1), 27-33, 2000.

[WD05] Wan, X-F., and D. Xu, *Intrinsic terminator prediction and its application in Synechococcus sp. WH8102*, International Journal of Computer Science and Technology, 20: 465-482, 2005.

[HMN05] De Hoon MJL, Makita Y, Nakai K, Miyano S, *Prediction of transcriptional terminators in Bacillus subtilis and related species*, PLoS Comp Biol 1(3): e25, 2005.

[3] `http://www.python.org`

[4] `http://www.biopython.org`

[CBT90] d'Aubenton Carafa Y, Brody E, Thermes C, *Prediction of Rho-independent Escherichia coli transcription terminators*, J Mol Biol 216: 835–858, 1990.

[LSLHME01] Elena A. Lesnik, Rangarajan Sampath, Harold B. Levene, Timothy J. Henderson, John A. McNeil and David J. Ecker, *Prediction of rho-independent transcriptional terminators in Escherichia coli*, Nucleic Acids Research, Vol. 29, No. 17 3583-3594, 2001.

[GN99] Ivan Gusarov and Evgeny Nudler, *The Mechanism of Intrinsic Transcription Termination*, Molecular Cell, Vol. 3, 495504, April 1999.

[SNKMNOYS95] Sugimoto,N., Nakano,S., Katoh,M., Matsumura,A., Nakamuta,H., Ohmichi,T., Yoneyama,M. and Sasaki,M., *Thermodynamic parameters to predict stability of RNA/DNA hybrid duplexes*, Biochemistry, 34, 11211–11216, 1995.

[REGULON06] Salgado H, Gama-Castro S, Peralta-Gil M, Diaz-Peredo E, Sanchez-Solano F, Santos-Zavaleta A, Martinez-Flores I, Jimenez-Jacinto V, Bonavides-Martinez C, Segura-Salazar J, Martinez-Antonio A, Collado-Vides J. *RegulonDB (version 5.0): Escherichia coli K-12 transcriptional regulatory network, operon organization, and growth conditions*, Nucleic Acids Res., 34(Database issue):D394-7, January 2006.

[MNON04] Makita Y, Nakao M, Ogasawara N, Nakai K., *DBTBS: database of transcriptional regulation in Bacillus subtilis and its contribution to comparative genomics*, Nucleic Acids Res., 32,D75-77 (NAR online), 2004.

[MWKSE06] Mierswa, Ingo and Wurst, Michael and Klinkenberg, Ralf and Scholz, Martin and Euler, Timm: *YALE: Rapid Prototyping for Complex Data Mining Tasks*, in Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-06), 2006.

[WF05] Ian H. Witten and Eibe Frank *Data Mining: Practical machine learning tools and techniques*, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[ŁRPRSLYB04] Małgorzata B. Łobocka, Debra J. Rose, Guy Plunkett III, Marek Rusin, Arkadiusz Samojedny, Hansjörg Lehnherr, Michael B. Yarmolinsky, and Frederick R. Blattner, *Genome of Bacteriophage P1*, Journal of Bacteriology, p. 7032-7068, Vol. 186, No. 21, November 2004.

[L01] Letondal C., *A Web interface generator for molecular biology programs in Unix*, Bioinformatics, 17(1), pp 73-82., 2001.