

Odkrywanie *cis*-regulatorowych RNA w prokariotach

Aleksander Jankowski

Uniwersytet Warszawski
Wydział Matematyki, Informatyki i Mechaniki

6 grudnia 2007 roku

Plan prezentacji

- 1 Źródła
- 2 Wstęp biologiczny
- 3 Zastosowane algorytmy
 - Filogenetyczny footprinting
 - CMfinder
 - RaveNnA
- 4 Materiał i metody
- 5 Wyniki

Plan prezentacji

- 1 Źródła
- 2 Wstęp biologiczny
- 3 Zastosowane algorytmy
 - Filogenetyczny footprinting
 - CMfinder
 - RaveNnA
- 4 Materiał i metody
- 5 Wyniki

Źródła

- Yao Z, Barrick J, Weinberg Z, Neph S, Breaker R, et al. (2007) *A Computational Pipeline for High-Throughput Discovery of cis-Regulatory Noncoding RNA in Prokaryotes*. PLoS Comput Biol 3(7): e126
- Blanchette M, Tompa M (2002) *Discovery of Regulatory Elements by a Computational Method for Phylogenetic Footprinting*. Genome Res. 12: 739-748.
- Zasha W, Walter LR (2006) *Sequence-based heuristics for faster annotation of non-coding RNA families*. Bioinformatics 22(1): 35-39

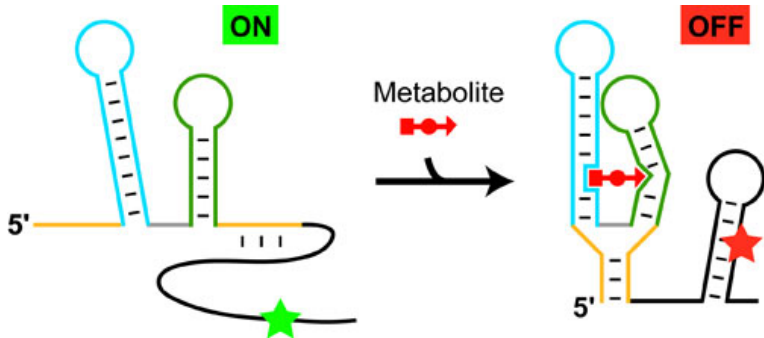
Plan prezentacji

- 1 Źródła
- 2 Wstęp biologiczny**
- 3 Zastosowane algorytmy
 - Filogenetyczny footprinting
 - CMfinder
 - RaveNnA
- 4 Materiał i metody
- 5 Wyniki

Ryboprzełączniki

- *Ryboprzełącznik* (przełącznik RNA) to to fragment łańcucha mRNA, długości ok. 200 par zasad, który reguluje ekspresję kodowanego przez ten łańcuch białka.
- Zazwyczaj składa się on z dwóch części:
 - *aptameru* wiążącego metabolit
 - *platformy ekspresyjnej*, która po związaniu metabolitu z aptamerem zmienia swoją strukturę przestrzenną, wpływając w ten sposób na ekspresję genu.
- Związanie metabolitu powoduje zmianę ekspresji genu (zahamowanie lub pobudzenie).
- Istnienie ryboprzełączników w organizmach żywych zostało doświadczalnie potwierdzone dopiero w 2002 roku.
- Najwięcej ryboprzełączników odnaleziono jak dotąd u bakterii.

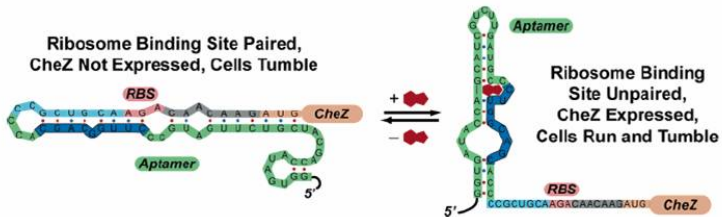
Ryboprzełączniki



Źródło: <http://www.nsls.bnl.gov/newsroom/science/2006/08-Serganov.htm>

Ryboprzełączniki

- Zmiana struktury przestrzennej platformy ekspresyjnej może mieć różne konsekwencje:
 - zamaskowanie lub odsłonięcie miejsca wiązania rybosomu (jak poniżej)
 - utworzenie struktury „szpilki do włosów”, której obecność wymusza zakończenie translacji
 - samoprzecięcie łańcucha RNA.

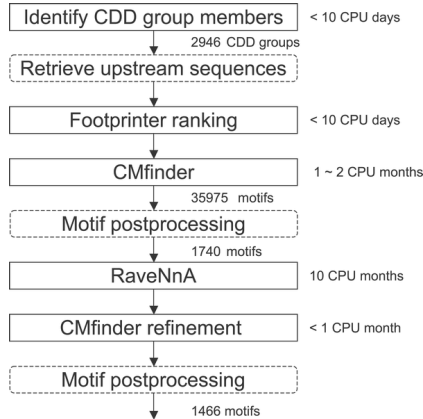


Źródło: <http://www.osel.cz/index.php?zprava=205>

Plan prezentacji

- 1 Źródła
- 2 Wstęp biologiczny
- 3 Zastosowane algorytmy**
 - Filogenetyczny footprinting
 - CMfinder
 - RaveNnA
- 4 Materiał i metody
- 5 Wyniki

Zastosowane algorytmy



Źródło: Yao et al., *A Computational Pipeline for High-Throughput Discovery of cis-Regulatory Noncoding RNA in Prokaryotes*.

Filogenetyczny footprinting

- Filogenetyczny footprinting jest techniką używaną do znajdowania miejsc wiązania czynników transkrypcyjnych w blisko spokrewnionych gatunkach.
- Pomysł opiera się na dwóch spostrzeżeniach:
 - Działanie czynników transkrypcyjnych u różnych, ale blisko spokrewnionych gatunków, jest bardzo podobne. W szczególności dotyczy to ich sposobu wiązania.
 - Miejsca wiązania czynników transkrypcyjnych, jako istotne fragmenty niekodującego DNA, są mocno konserwatywne i ulegają wolniejszym zmianom ewolucyjnym niż inne części kodu genetycznego.
- Przedmiotem analizy są sekwencje homologiczne, czyli pochodzące od wspólnego przodka.

Filogenetyczny footprinting – wejście i wyjście

- Wejście:
 - n homologicznych sekwencji: S_1, \dots, S_n
 - drzewo filogenetyczne T opisujące pochodzenie sekwencji
 - długość poszukiwanych motywów k
 - maksymalna dopuszczalna wielkość kary d .
- Wyjście:
 - zbiór rozwiązań, w którym każde rozwiązanie jest określone przez wskazanie motywu długości k w każdej z sekwencji S_1, \dots, S_n .
- Oznaczenia:
 - $C(u)$ – zbiór synów wierzchołka u w drzewie T
 - $h(s, t)$ – liczba pozycji, na których różnią się k -mery s i t
 - $\Sigma = \{A, C, G, T\}$.

Filogenetyczny footprinting – algorytm

- Przechodzimy przez drzewo filogenetyczne T od liści do korzenia.
- Dla każdego wierzchołka u z drzewa T wyznaczamy tablicę W_u zawierającą 4^k elementów.
- Dla każdego k -meru s , wartość $W_u[s]$ będzie najmniejszą karą osiąganą dla poddrzewa zaczepionego w u , przy założeniu, że motyw dla wspólnego przodka u ma postać s .
- Tablicę W_u wypełniamy rekurencyjnie:

$$W_u[s] = \begin{cases} 0, & \text{gdy } u \text{ jest liściem i } s \text{ jest podstwowem } S_u \\ +\infty, & \text{gdy } u \text{ jest liściem i } s \text{ nie jest podstwowem } S_u \\ \sum_{v \in C(u)} \min_{t \in \Sigma^k} \{W_v[t] + h(s, t)\}, & \text{gdy } u \text{ nie jest liściem.} \end{cases}$$

Filogenetyczny footprinting – algorytm

- Niech r będzie korzeniem drzewa T .
- Każda wartość tablicy W_r nie większa niż d prowadzi do jednego lub wielu rozwiązań.
- Pojedyncze rozwiązanie jest określone przez wskazanie motywu długości k w każdej z sekwencji S_1, \dots, S_n .
- Algorytm wydaje się być bardzo złożonym obliczeniowo, ale w typowych zastosowaniach jest wystarczająco szybki.

Filogenetyczny footprinting – słabości

- Nie wszystkie miejsca wiązania czynników transkrypcyjnych są znajdowane.
 - Niektóre z nich są specyficzne tylko dla niewielkiej grupy gatunków.
 - Bardzo krótkie motywy mogą występować przypadkowo.
 - Niektóre czynniki transkrypcyjne są mniej wrażliwe na mutacje w miejscach wiązania.
- Aby uniknąć fałszywych wyników pozytywnych, należy się upewnić, że znalezione motywy mają istotnie mniejszą częstość występowania mutacji niż otaczające je fragmenty sekwencji.

CMfinder

- CMfinder jest iteracyjnym algorytmem opierającym się na modelu kowariancji.
- Algorytm jednocześnie poprawia model, opisujący występowanie motywów w ustalonym zbiorze sekwencji, oraz koryguje oszacowanie położenia motywów w tych sekwencjach.
- Dwie fazy, wykonywane przemiennie:
 - (krok E) poprawienie oszacowania występowania motywów w sekwencjach i położenia kandydatów na motywy
 - (krok M) uaktualnienie motywu, przez rozważenie możliwych złączeń dwóch nici RNA i zastosowanie modelu termodynamicznego.

RaveNnA

- RaveNnA jest heurystycznym algorytmem wyszukiwania fragmentów niekodującego RNA, homologicznych do podanego, w dużych genomach.
- Jest on znacznie szybszy od algorytmów wykorzystujących modele kowariancji.
- Idea polega na dobraniu odpowiedniego do sekwencji wejściowej ukrytego modelu Markowa.
- Jego użycie do znajdowania motywów będzie miało charakter uzupełniający.

Plan prezentacji

- 1 Źródła
- 2 Wstęp biologiczny
- 3 Zastosowane algorytmy
 - Filogenetyczny footprinting
 - CMfinder
 - RaveNnA
- 4 Materiał i metody**
- 5 Wyniki

Źródło danych

- Do analizy wybrano *Firmicutes* – typ pospolitych bakterii.
- Użyto bazy danych *NCBI RefSeq*, w której znajdują się 44 kompletne genomy bakterii typu *Firmicutes*.
- Sekwencje opisanych białek występujących w tych gatunkach pobrano z bazy danych *NCBI Conserved Domain Database*. Spośród nich 92% miało przypisaną co najmniej jedną grupę CCD.
- Grupa CCD zawiera homologiczne białka, w których znajduje się dobrze zachowana wspólna domena, czyli fragment cząsteczki zdolny do samodzielnego zachowania kształtu.
- Ograniczono się do 145 grup CCD zawierających od 5 do 70 elementów.

Przygotowanie danych

- Dla każdego genu z bazy, pobrano jego sekwencję 5' upstream o długości nie większej, niż 600 nukleotydów.
 - Niektóre geny występują w operonach, co powoduje, że czynniki regulacyjne nie znajdują się bezpośrednio przed genem, ale przed całym operonem.
 - Jeśli następny region kodujący w kierunku upstream znajdował się mniej niż 100 nukleotydów dalej i był zorientowany w tą samą stronę, to łączono go z wybranym genem.
- Zbiór sekwencji upstream związany z białkami z jednej grupy CCD nazywać będziemy zestawem danych.
- W celu zwiększenia czułości, usuwano przy użyciu BLASTa grupy bardzo podobnych sekwencji, oraz usunięto sekwencje kodujące tRNA i rRNA.

Analiza filogenetyczna

- Użyto filogenetycznego footprintingu, aby wybrać zestawy danych, w których najprawdopodobniej znajdują się miejsca występowania motywów.
- Potrzebne do tego drzewo filogenetyczne przybliżono przez analizę sekwencji białek, których sekwencje upstream rozważamy.
- Funkcjonalne RNA, takie jak ryboprzełączniki, ma niską zachowawczość, ale zazwyczaj zawiera kawałki, które ulegają bardzo małej zmienności.
- Zauważmy, że do tej pory wykorzystywano tylko informacje o zachowawczości sekwencji RNA.

Użycie CMfindera

- Przy użyciu CMfindera wyszukujemy motywy w każdym z zestawów danych.
- CMfinder jest zorientowany na strukturę, jaką tworzy nić RNA.
- Do oceniania motywów użyto funkcji oceniającej postaci

$$r = sp \cdot \sqrt{lc \cdot bp/sid} \cdot (1 + \log(mc)),$$

gdzie

- sp – liczba gatunków, w których motyw występuje
- mc – średnia liczba wystąpień motywu na gatunek
- bp – (ważona) liczba par zasad w strukturze konsensusowej
- lc – lokalna zachowawczość sekwencji
- sid – średnie (parami) podobieństwo sekwencji.

Szukanie dodatkowych wystąpień motywów

- Jedną z zalet omawianej metody jest połączenie odkrywania motywów z ich wyszukiwaniem.
- Wyszukiwanie motywów jest skupione na grupach CCD, gdyż sekwencje upstream zbliżonych białek najprawdopodobniej będą miały wspólne *cis*-regulatorowe RNA.
- Okazuje się jednak, że wiele *cis*-regulatorowych elementów, takich jak ryboprzełączniki, można znaleźć w pobliżu różnych operonów, regulujących powiązane procesy.
- Dlatego też potrzebne jest dodatkowe wyszukiwanie wystąpień motywów w całym genomie. Następuje po nim kolejny przebieg CMfindera, aby uaktualnić oceny motywów.

Plan prezentacji

- 1 Źródła
- 2 Wstęp biologiczny
- 3 Zastosowane algorytmy
 - Filogenetyczny footprinting
 - CMfinder
 - RaveNnA
- 4 Materiał i metody
- 5 Wyniki

Identyfikacja znanych motywów

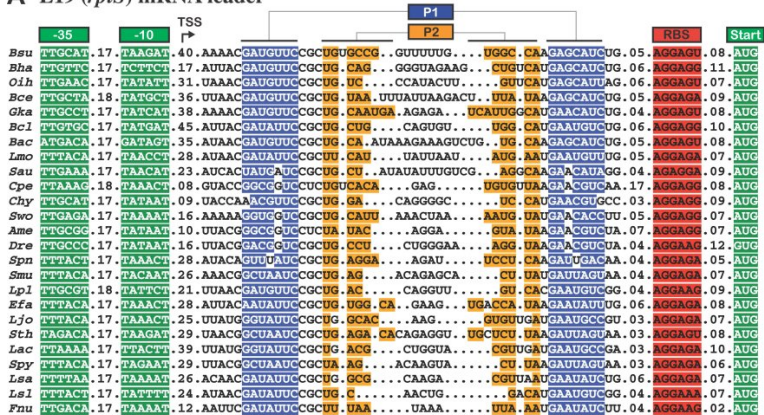
- Przewidywane motywy zostały porównane z motywami występującymi w *RNA Family Database* (Rfam).
- W bazie Rfam znajduje się 21 rodzin niekodującego RNA występującego w *Bacillus subtilis*, spośród nich cztery zostały wygaszone na etapie przygotowywania danych.
- Spośród 17 pozostałych rodzin, 13 występowało wśród 50 najlepiej ocenionych kandydatów na motywy.

Nowe, nieznanne dotąd motywy

- Ze szczególną uwagą przebadano 200 najlepiej ocenionych motywów.
- Dla 116 nie udało się utrzymać hipotezy stwierdzającej, że stanowią one *cis*-regulatorowe RNA.
- Spośród pozostałych 84, 20 odpowiada istniejącym w Rfam rodzinom, a 11 najprawdopodobniej powstało wskutek transpozycji.
- Pozostałe 53 oceniono jako kandydatów na *cis*-regulatorowe RNA.

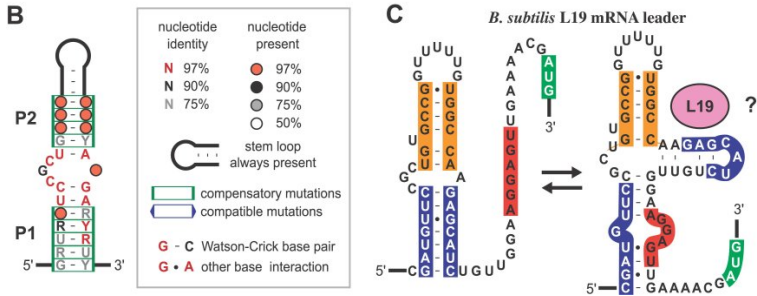
Wyniki

A L19 (*rplS*) mRNA leader



Źródło: Yao et al., *A Computational Pipeline for High-Throughput Discovery of cis-Regulatory Noncoding RNA in Prokaryotes*.

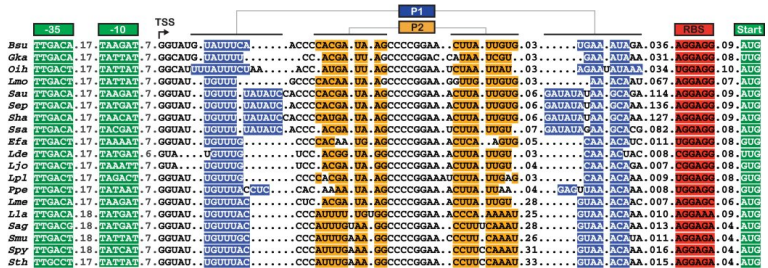
Wyniki



Źródło: Yao et al., *A Computational Pipeline for High-Throughput Discovery of cis-Regulatory Noncoding RNA in Prokaryotes.*

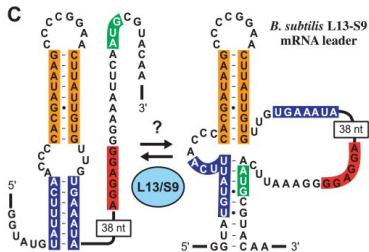
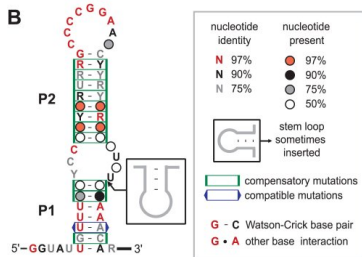
Wyniki

A L13-S9 (*rplM-rpsI*) mRNA leader



Źródło: Yao et al., *A Computational Pipeline for High-Throughput Discovery of cis-Regulatory Noncoding RNA in Prokaryotes.*

Wyniki



Źródło: Yao et al., *A Computational Pipeline for High-Throughput Discovery of cis-Regulatory Noncoding RNA in Prokaryotes.*

Ukryte modele Markowa i algorytm Bauma-Welcha

Aleksander Jankowski

Uniwersytet Warszawski
Wydział Matematyki, Informatyki i Mechaniki

13 grudnia 2007 roku

Ukryte modele Markowa

- Łańcuch Markowa jest określony przez zbiór stanów wraz z prawdopodobieństwami przejść

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

gdzie x_i – stan w chwili czasu i .

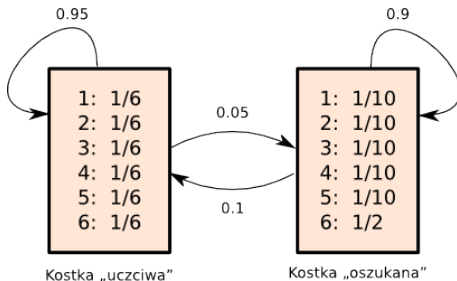
- W ukrytym modelu Markowa mamy do czynienia z:
 - π_i – stanem w chwili czasu i
 - x_i – obserwacją w chwili czasu i .
- Oprócz prawdopodobieństw przejść, wprowadzamy prawdopodobieństwa emisji

$$e_k(b) = P(x_i = b | \pi_i = k).$$

- Przyjmijmy, że stanem początkowym i końcowym modelu jest stan 0.

Przykład sporadycznie nieuczciwego kasyna

- Gramy z krupierem wiele razy w prostą grę:
 - każde z nas rzuca kostką
 - stawkę wygrywa ten, kto wyrzuci więcej oczek.
- Krupier sporadycznie podmienia kostkę „uczciwą” na kostkę „oszukaną”.



Ukryte modele Markowa

- W przypadku łańcuchów Markowa, zawsze wiemy, któremu stanowi odpowiada każda obserwacja.
- Obserwując ukryty model Markowa, sekwencja stanów, którą podążaliśmy, pozostaje dla nas nieznana.
- Nawet jeśli dokładnie znamy wszystkie parametry ukrytego modelu Markowa oraz ciąg obserwacji od stanu początkowego, to nie jesteśmy w stanie dokładnie odtworzyć ciągu stanów.
- Powód jest prozaiczny – zazwyczaj możliwych ciągów stanów jest wiele.
- Możliwe jest jednak znalezienie najbardziej prawdopodobnego ciągu stanów dla ustalonej sekwencji obserwacji.

Algorytm Viterbi'ego

- Znamy ciąg obserwacji $x = (x_1, \dots, x_n)$. Szukamy ciągu stanów $\pi^* = (\pi_1^*, \dots, \pi_n^*)$ takiego, że

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

- Niech $v_k(i)$ – prawdopodobieństwo zajścia najbardziej prawdopodobnego ciągu stanów kończącego się w stanie k obserwacją i .
- Wiadomo, że $v_0(0) = 1$ i $v_k(0) = 0$ dla $k \neq 0$.
- Prawdopodobieństwa $w_k(i)$ dla $i > 0$ można wyznaczyć rekurencyjnie:

$$v_l(i) = e_l(x_i) \cdot \max_k \{v_k(i-1)a_{kl}\}.$$

Algorytm Viterbi'ego

- Na koniec wyliczamy

$$P(x, \pi^*) = \max_k \{v_k(n)a_{k0}\}.$$

- Zapamiętując podczas przebiegu algorytmu informację o tym, które częściowe ciągi stanów były najbardziej prawdopodobne, możemy odtworzyć π^* .
- Ze względu na ograniczenia arytmetyki zmiennoprzecinkowej, w praktyce obliczenia prowadzi się po zlogarytmowaniu, to znaczy wyznacza się $\log v_l(i)$ i $\log P(x, \pi^*)$.
- Algorytm Viterbi'ego jest bardzo wydajny, ale niestety nie zawsze znamy niezbędne do jego zastosowania parametry ukrytego modelu Markowa: $e_l(x_i)$ i a_{kl} .

Algorytm Bauma-Welcha – wielkości pomocnicze

- Oznaczmy przez $f_k(i)$ prawdopodobieństwo ciągu obserwacji x_1, \dots, x_i , takiego że ostatnim stanem jest k :

$$f_k(i) = P(x_1 \dots x_i, \pi_i = k).$$

- Można je wyznaczyć rekurencyjnie, wiedząc że $f_0(0) = 1$ i $f_k(0) = 0$ dla $k \neq 0$.
- Dla $i > 0$ zachodzi

$$f_l(i) = e_l(x_i) \sum_k f_k(i-1) a_{kl}.$$

Algorytm Bauma-Welcha – wielkości pomocnicze

- Rozważając sytuację odwrotną, oznaczmy przez $b_k(i)$ prawdopodobieństwo wystąpienia ciągu obserwacji x_{i+1}, \dots, x_n *pod warunkiem*, że przed wyemitowaniem tego ciągu obserwacji byliśmy w stanie k :

$$b_k(i) = P(x_{i+1} \dots x_n | \pi_i = k).$$

- Można je wyznaczyć rekurencyjnie, wiedząc że $b_k(n) = a_{k0}$.
- Dla $i < n$ zachodzi

$$b_k(i) = \sum_l a_{kl} e_l(x_{i+1}) b_l(i+1).$$

Algorytm Bauma-Welcha – wejście i wyjście

- Przez θ oznaczać będziemy punkty w przestrzeni parametrów ukrytego modelu Markowa, to znaczy zestawy parametrów $(a_{kl}), (e_k(b))$.
- Wejście:
 - N ciągów obserwacji, nazywanych *sekwencjami treningowymi*: x^1, \dots, x^N .
- Wyjście:
 - Punkt θ w przestrzeni parametrów, maksymalizujący $\log P(x^1, \dots, x^N | \theta) = \sum_{j=1}^N \log P(x^j | \theta)$.
- Innymi słowy, szukamy takiego ukrytego modelu Markowa (o ustalonej liczbie stanów), dla którego prawdopodobieństwo zaobserwowania sekwencji treningowych jest największe.

Algorytm Bauma-Welcha – algorytm

- 1 Ustal dowolnie parametry $(a_{kl}), (e_k(b))$ dla modelu początkowego.
- 2 Korzystając z posiadanego modelu, dla każdego stanu k, l wyznacz A_{kl} – wartość oczekiwaną liczby przejść ze stanu k do stanu l podczas emisji sekwencji treningowych x^1, \dots, x^N .
- 3 Korzystając z posiadanego modelu, dla każdego stanu k i symbolu b wyznacz $E_k(b)$ – wartość oczekiwaną liczby emisji symbolu b w stanie k podczas emisji sekwencji treningowych x^1, \dots, x^N .
- 4 Re-estymuj parametry $(a_{kl}), (e_k(b))$ modelu przy użyciu wartości oczekiwanych $(A_{kl}), (E_k(b))$.
- 5 Jeśli $\log P(x^1, \dots, x^N | \theta)$ zauważalnie wzrosło, to wróć do punktu 2.

Algorytm Bauma-Welcha – wyznaczanie A_{kl} i $E_k(b)$

- Ustalmy sekwencję treningową x . Wówczas

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

- Sumując te wielkości po wszystkich sekwencjach i po wszystkich pozycjach w tych sekwencjach, stwierdzamy że

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

- Analogicznie stwierdzamy, że

$$E_k(b) = \sum_j \sum_{\{i: x_i^j=b\}} \frac{f_k^j(i) b_k^j(i)}{P(x^j)}.$$

Algorytm Bauma-Welcha – re-estymacja parametrów

- Wystarczy prosta normalizacja:

$$a_{kl} = \frac{A_{kl}}{\sum_m A_{km}}, \quad e_k(b) = \frac{E_k(b)}{\sum_c E_k(c)}.$$

Algorytm EM (expectation maximization)

- Rozważmy model statystyczny określony przez parametry θ .
- Obserwowane wielkości oznaczmy przez x .
- Prawdopodobieństwo wystąpienia x jest określone przez pewne ukryte dane y .
- Dla ukrytych łańcuchów Markowa, θ jest zestawem parametrów (a_{kl}) , $(e_k(b))$, zaś y reprezentuje ciąg stanów.
- Naszym celem jest znalezienie modelu, który maksymalizuje logarytm szans

$$\log P(x|\theta) = \log \sum_y P(x, y|\theta).$$

Algorytm EM (expectation maximization)

- Załóżmy, że mamy poprawny model θ^t . Chcielibyśmy estymować nowy, lepszy model θ^{t+1} .
- Jako że $P(x, y|\theta) = P(y|x, \theta)P(x|\theta)$, to

$$\log P(x|\theta) = \log P(x, y|\theta) - \log P(y|x, \theta).$$

- Mnożąc obie strony przez $P(y|x, \theta^t)$ i sumując po y otrzymujemy

$$\log P(x|\theta) = \sum_y P(y|x, \theta^t) \log P(x, y|\theta) - \sum_y P(y|x, \theta^t) \log P(y|x, \theta).$$

- Określmy $Q(\theta|\theta^t) = \sum_y P(y|x, \theta^t) \log P(x, y|\theta)$.

Entropia względna (odległość Kullbacka-Leiblera)

- Powiedzmy, że p i q są dwoma dyskretnymi rozkładami prawdopodobieństwa na pewnej przestrzeni.
- Ich entropię względną określamy następująco:
$$d_{KL}(p, q) = \sum_i p(i) \log \frac{p(i)}{q(i)}.$$
- Wykorzystywać będziemy fakt, że entropia względna jest nieujemna.
- Wystarczy to wykazać dla logarytmu naturalnego, wykorzystując nierówność $\ln x \leq x - 1$.
- Istotnie, $\ln \frac{q(i)}{p(i)} \leq \frac{q(i)}{p(i)} - 1$, zatem $\ln \frac{p(i)}{q(i)} \geq 1 - \frac{q(i)}{p(i)}$, a więc

$$\sum_i p(i) \ln \frac{p(i)}{q(i)} \geq \sum_i p(i) \frac{p(i) - q(i)}{p(i)} = \sum_i p(i) - \sum_i q(i) = 0.$$

Algorytm EM (expectation maximization)

- Zauważmy, że

$$\begin{aligned} \log P(x|\theta) - \log P(x|\theta^t) &= \\ &= Q(\theta|\theta^t) - Q(\theta^t|\theta^t) + \sum_y P(y|x, \theta^t) \log \frac{P(y|x, \theta^t)}{P(y|x, \theta)} \geq \\ &\geq Q(\theta|\theta^t) - Q(\theta^t|\theta^t). \end{aligned}$$

- Wybierając $\theta^{t+1} = \arg \max_{\theta} Q(\theta|\theta^t)$, zapewniamy, że logarytm szans nowego modelu będzie nie mniejszy niż poprzedniego.

Dowód poprawności algorytmu Bauma-Welcha

- Wystarczy wykazać, że sposób wyboru θ^{t+1} dla ustalonego θ^t spełnia warunek $\theta^{t+1} = \arg \max_{\theta} Q(\theta|\theta^t)$.
- Dla ustalonego ciągu obserwacji oraz ustalonego ciągu stanów π , oznaczmy przez $A_{kl}(\pi)$ liczbę przejść z k do l , zaś przez $E_k(b, \pi)$ liczbę emisji symbolu b w stanie k . Wówczas

$$P(x, \pi|\theta) = \prod_{k=1}^M \prod_b e_k(b)^{E_k(b, \pi)} \cdot \prod_{k=1}^M \prod_{l=1}^M a_{kl}^{A_{kl}(\pi)}.$$

- Zauważmy, że wartości A_{kl} i $E_k(b)$ zdefiniowane dla algorytmu Bauma-Welcha wyrażają się następująco:

$$E_k(b) = \sum_{\pi} P(\pi|x, \theta^t) E_k(b, \pi), \quad A_{kl} = \sum_{\pi} P(\pi|x, \theta^t) A_{kl}(\pi).$$

Dowód poprawności algorytmu Bauma-Welcha

- Z definicji Q , $Q(\theta|\theta^t) = \sum_{\pi} P(\pi|x, \theta^t) \log P(x, \pi|\theta)$. Ponadto

$$\log P(x, \pi|\theta) = \sum_{k=1}^M \sum_b E_k(b, \pi) \log e_k(b) + \sum_{k=1}^M \sum_{l=1}^M A_{kl}(\pi) \log a_{kl}.$$

- Po przekształceniu uzyskujemy postać

$$Q(\theta|\theta^t) = \sum_{k=1}^M \sum_b E_k(b) \log e_k(b) + \sum_{k=1}^M \sum_{l=1}^M A_{kl} \log a_{kl}.$$

- Pozostało do wykazania, że Q przyjmuje maksimum dla

$$a_{kl} = \frac{A_{kl}}{\sum_m A_{km}}, \quad e_k(b) = \frac{E_k(b)}{\sum_c E_k(c)}.$$

Dowód poprawności algorytmu Bauma-Welcha

- Pokażemy, że $\sum_{k=1}^M \sum_{l=1}^M A_{kl} \log a_{kl}$ dla $a_{kl}^0 = \frac{A_{kl}}{\sum_m A_{km}}$ przybiera wartość nie mniejszą niż dla dowolnych innych a_{kl} .
- Interesuje nas różnica

$$\begin{aligned} \sum_{k=1}^M \sum_{l=1}^M A_{kl} \log \frac{a_{kl}^0}{a_{kl}} &= \sum_{k=1}^M \sum_{l=1}^M \left(\sum_m A_{km} \right) a_{kl}^0 \log \frac{a_{kl}^0}{a_{kl}} = \\ &= \sum_{k=1}^M \left(\sum_m A_{km} \right) \sum_{l=1}^M a_{kl}^0 \log \frac{a_{kl}^0}{a_{kl}} \geq 0. \end{aligned}$$

- Analogiczne rozumowanie przeprowadzamy dla składnika $\sum_{k=1}^M \sum_b E_k(b) \log e_k(b)$.