Modelowanie motywów łańcuchami Markowa wyższego rzędu

Aleksander Jankowski

Uniwersytet Warszawski
Wydział Matematyki, Informatyki i Mechaniki

23 października 2008 roku

Plan prezentacji

Plan prezentacji

Źródła

- Paulo G. S. da Fonseca, Katia S. Guimarães, Marie-France Sagot, *Efficient representation and P-value computation for high-order Markov motifs*, Bioinformatics 24(16): i160-i166
- H. Touzet, J.-S. Varre *Efficient and accurate P-value computation for position weight matrices*, Algorithms Mol. Biol, 2 (2007) : 15

Plan prezentacji

Motywy i ich znaczenie

- Podobne fragmenty sekwencji kwasów nukleinowych, szeroko rozpowszechnione, mające zbliżone własności i funkcjonalne biologicznie, są nazywane *motywami*.
- Mają one istotne znaczenie w mechanizmach regulacyjnych DNA, i są silnie ewolucyjnie zachowywane.
- Zazwyczaj znajdują się one we fragmentach DNA, które nie kodują białek.

Position Weight Matrix

- *Position Weight Matrices* (PWM), zwane również *Position-Specific Score Matrices* (PSSM) są powszechnie wykorzystywane do modelowania miejsc wiązania czynników transkrypcyjnych.

Źródło: http://weblogo.berkeley.edu/

Łańcuchy Markowa wyższego rzędu

- Łańcuch Markowa:

$$P(X_{n+1} = x | X_n = x_n, \ldots, X_1 = x_1) = P(X_{n+1} = x | X_n = x_n)$$

„przyszłość zależy od przeszłości tylko przez teraźniejszość”

- Łańcuch Markowa rzędu K:

$$P(X_{n+1} = x | X_n = x_n, \ldots, X_1 = x_1) =$$

$$= P(X_{n+1} = x | X_n = x_n, \ldots, X_{n-k+1} = x_{n-k+1})$$

Plan prezentacji

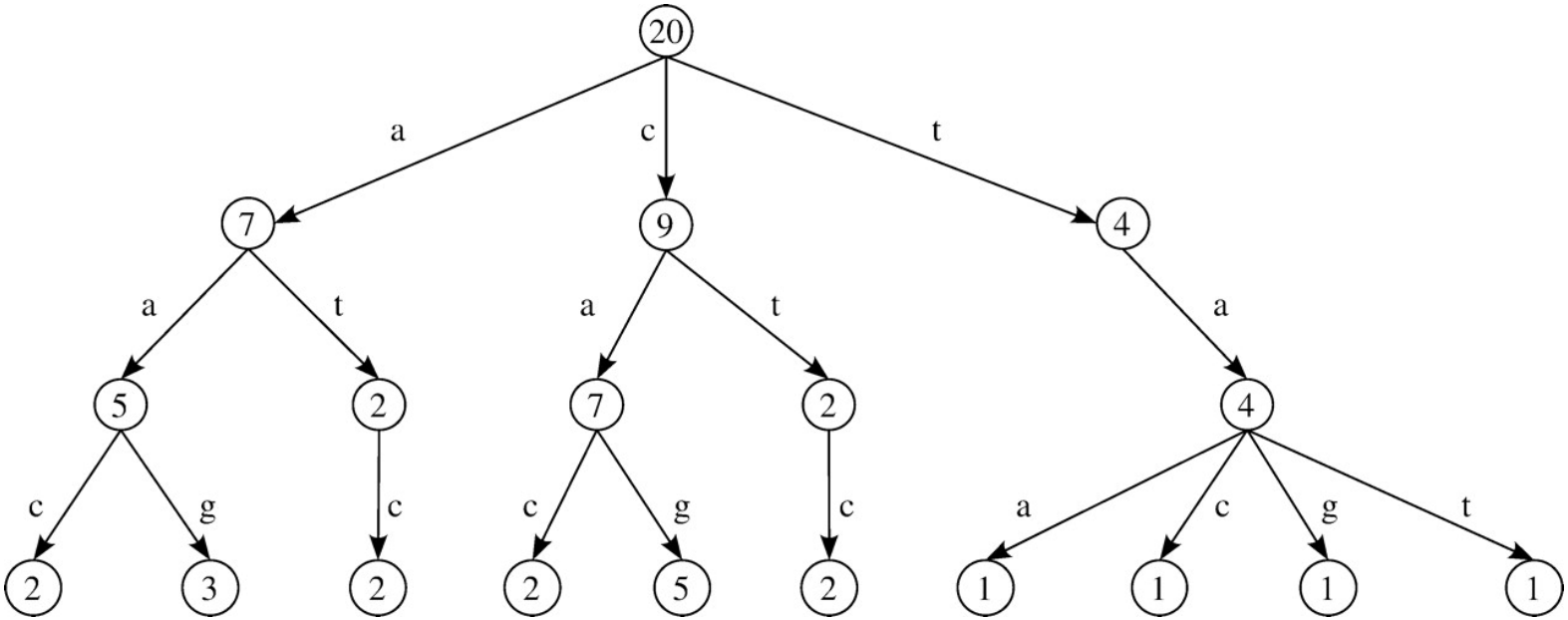
Reprezentacja zbioru wystąpień motywu

- Parametry modelu motywu są zazwyczaj ustalane na podstawie pewnego zbioru wystąpień, tak aby podsumowywały informację w tym zbiorze zawartą.
- Metody maszynowego uczenia się pozwalają dopasować parametry modelu tak, aby zmaksymalizować prawdopodobieństwo zaobserwowania danego zbioru wystąpień.
- Jak wygodnie reprezentować (duży) zbiór wystąpień motywu?

Drzewa trie

- *Drzewem trie* (czyt. *traj*) nad alfabetem $\mathcal{A}$ nazywamy drzewo ukorzenione, takie że
  - (i) każda krawędź jest etykietowana symbolem z $\mathcal{A}$
  - (ii) dla każdego $a \in \mathcal{A}$, każdy węzeł ma co najwyżej jedną krawędź wychodzącą opatrzoną etykietą a.
- Istnieje wzajemna odpowiedniość między wierzchołkiem v oraz odpowiadającym mu słowem powstałym z połączenia etykiet krawędzi prowadzących od korzenia do v.
- Definicję drzewa trie można uogólnić, tak że mogą one reprezentować multizbiory słów. Wystarczy w odpowiednim liściu przechowywać liczbę wystąpień słowa, a w węzłach wewnętrznych – sumę etykiet liści z odpowiedniego poddrzewa. Wszystkie te liczby nazywamy *licznikami liści*.

Przykład drzewa trie



Źródło: P. G. S. da Fonseca et al., *Efficient representation and P-value computation for high-order Markov motifs*

Własności liczników liści

- Licznik liści wierzchołka v, któremu odpowiada słowo label(v), równy jest ilości słów reprezentowanych w drzewie mających prefiks label(v).
- Częstość występowania symbolu a na pozycji j można wyznaczyć dzieląc sumę liczników liści w węzłach na poziomie j do których wchodzi krawędź etykietowana przez a przez licznik liści korzenia drzewa.
- Podążając wzdłuż drzewa można łatwo również wyliczać prawdopodobieństwa warunkowe.

Plan prezentacji

Obliczanie *P*-wartości

- Ustalmy $\mathbf{m}$ – model motywu o długości W będący łańcuchem Markowa rzędu K.
- *Logarytm prawdopodobieństwa* słowa $\mathbf{x}$ względem $\mathbf{m}$ definiujemy jako

$$\ell(\mathbf{x}, \mathbf{m}) = \log P_{\mathbf{m}}(\mathbf{x}).$$

  Będziemy pisać krótko $\ell(\mathbf{x})$.
- Gdy mamy także model tła $\mathbf{m_0}$, to *P-wartość* logarytmu prawdopodobieństwa l definiujemy jako

$$P\text{-}value(l, \mathbf{m}) = P_{\mathbf{m_0}}(\{\mathbf{x} \in \mathcal{A}^W \mid \ell(\mathbf{x}, \mathbf{m}) \geqslant l\}).$$

Obliczanie *P*-wartości c.d.

- Innymi słowy, *P*-wartość jest prawdopodobieństwem zdarzenia, że losowa sekwencja wygenerowana z modelu tła $\mathbf{m_0}$ będzie miała logarytm prawdopodobieństwa względem $\mathbf{m}$ większy lub równy l.
- *P*-wartość można liczyć na wiele sposobów, m.in.:
  - bezpośrednio z definicji – sprawdzając wszystkie $\mathbf{x} \in \mathcal{A}^W$
  - metodą Monte Carlo – szybko i całkiem dokładnie
  - sprytnie – o tym za chwilę.

Algorytm podziału i ograniczeń – wejście i wyjście

- Wejście:
  - $\mathbf{m}$ – model motywu o długości W będący łańcuchem Markowa rzędu K
  - $\mathbf{m_0}$ – model tła
  - I – próg dla logarytmu prawdopodobieństwa
  - $\mathbf{y}$ – prefiks długości $j \leqslant W$, początkowo pusty
- Wyjście:

$$P_{\mathbf{m_0}}(\{\mathbf{z} \in \mathcal{A}^W : \mathbf{z}_{1\ldots j} = \mathbf{y} \wedge \ell(\mathbf{z}, \mathbf{m}) \geqslant I\})$$

Algorytm podziału i ograniczeń – schemat

- Jeśli $|\mathbf{y}| = W$, to
  - jeśli $\ell(\mathbf{y}, \mathbf{m}) \geqslant l$, to zwróć $P_{\mathbf{m_0}}(\mathbf{y})$
  - jeśli $\ell(\mathbf{y}, \mathbf{m}) < l$, to zwróć 0.
- Jeśli $|\mathbf{y}| < W$, to
  - oszacuj $L^{\min}(\mathbf{y})$ oraz $L^{\max}(\mathbf{y})$ takie, że dla każdego $\mathbf{z} \in \mathcal{A}^W$ takiego, że $\mathbf{z}_{1\ldots j} = \mathbf{y}$, zachodzi $L^{\min}(\mathbf{y}) \leqslant \ell(\mathbf{z}, \mathbf{m}) \leqslant L^{\max}(\mathbf{y})$
  - jeśli $L^{\min}(\mathbf{y}) \geqslant l$, to zwróć $P_{\mathbf{m_0}}(\{\mathbf{z} \in \mathcal{A}^W : \mathbf{z}_{1\ldots j} = \mathbf{y}\})$
  - jeśli $L^{\max}(\mathbf{y}) < l$, to zwróć 0
  - w przeciwnym przypadku zwróć sumę

$$\sum_{a \in \mathcal{A}} \mathit{BranchAndBoundPvalue}(\mathbf{m}, \mathbf{m_0}, l, \mathbf{y}a)$$

Znajdowanie mocnych ograniczeń

- Zauważmy, że

$$\ell(\mathbf{x}, \mathbf{m}) = \sum_{j=1}^{W} \log P(x_j | x_{j-K} \dots x_{j-1}).$$

- Oznaczmy $\ell_j(\mathbf{x}, \mathbf{m}) = \log P(x_j | x_{j-K} \dots x_{j-1})$.
- Mając dany prefiks **y** słowa **x**, chcemy znaleźć ograniczenie górne i dolne na $\ell(\mathbf{x}, \mathbf{m})$.
- Nie jest konieczne rozważanie wszystkich słów **x** spełniających ten warunek (dlaczego)?

Algorytm *max suffix log-score* – wejście i wyjście

- Wejście:
  - $\mathbf{m}$ – model motywu o długości W będący łańcuchem Markowa rzędu K
  - u – początkowa pozycja sufiksu ($1 \leqslant u \leqslant W$)
  - $\mathbf{k} \in \mathcal{A}^{\min\{K, u-1\}}$ – słowo, które poprzedza sufiks
- Wyjście:

$$\max_{* \in \mathcal{A}^{W-u+1}} \sum_{j=u}^{W} \ell_j(\mathbf{k}*, \mathbf{m})$$

Algorytm *max suffix log-score* – schemat

- Inicjacja: $S[u-1,k] \leftarrow 0$.
- Dla każdego $j = u, \ldots, W$:
  - dla każdego $\mathbf{y}$ takiego, że $S[j-1,\mathbf{y}]$ jest określone, i dla każdego $a \in \mathcal{A}$:

$$\tilde{S}[j,\mathbf{y}a] \leftarrow S[j-1,\mathbf{y}] + \ell_j(\mathbf{y}a, m)$$

  - jeśli $j \leqslant K$, to skopiuj $\tilde{S}$ na S
  - jeśli $j > K$, to dla każdego $\mathbf{z} \in \mathcal{A}^K$

$$S[j,\mathbf{z}] \leftarrow \max_{a \in \mathcal{A}} \tilde{S}[j,a\mathbf{z}].$$

Algorytm iteracyjnego poprawiania modelu

- W algorytmie podziału i ograniczeń obcinanie kolejnych gałęzi obliczeń byłoby łatwiejsze, gdyby zbiór wartości przyjmowanych przez P-wartość był niewielki.
- Pomysł: określamy *wartość zaokrągloną* x przy dokładności ϵ jako

$$[x]_\epsilon = \frac{1}{\epsilon}\lfloor \epsilon \cdot x \rfloor$$

gdzie np. $\epsilon = 10^k$ dla $k \geqslant 0$.

- Modyfikacja algorytmu polega na użyciu

$$\ell(\mathbf{x}, \mathbf{m}_\epsilon) = \sum_{j=1}^{W} \log[P(x_j | x_{j-K} \ldots x_{j-1})]_\epsilon.$$

Plan prezentacji

Wyniki wydajnościowe

- Do testów użyto motywów długości $W = 6, \ldots, 12$ par zasad zawierających miejsca wiązania czynników transkrypcyjnych z bazy TRANSFAC.
- Rozważano rząd łańcucha Markowa $K = 0, 1, 2$.
- Jak należało się spodziewać, czas działania algorytmu „naiwnego” niezbyt zależał od rzędu łańcucha Markowa.
- Algorytm podziału i ograniczeń oraz algorytm iteracyjnego poprawiania modelu działały o rząd wielkości szybciej.