ISSN 0006-3509, Biophysics, 2009, Vol. 54, No. 5, pp. 569–573. © Pleiades Publishing, Inc., 2009.
Original Russian Text © M.A. Roytberg, 2009, published in Biofizika, 2009, Vol. 54, No. 5, pp. 791–797.

MOLECULAR BIOPHYSICS

Computation of the Probabilities of Families of Biological Sequences

M. A. Roytberg

Institute of Mathematical Problems of Biology, Russian Academy of Sciences, Pushchino, Moscow Region, 142290 Russia

Received August 11, 2009

Abstract—An algorithm for computing the probabilities of biological sequences is presented. The algorithm is applicable to many problems of bioinformatics, in particular, computing seed sensitivity in the search for local similarities in genomes or estimating the reliability of search for clusters of regulatory sites. It can be also used for distributions of probabilities described by different models, e.g., Bernoulli, Markov, and hidden Markov models. The algorithm is based on the description of probability distribution as well as of the family of sequences using finite automata, whereby the problem of calculating the probabilities is reduced to computing an appropriate generalized partition function. The algorithm can be applied not only to biological sequences but also to symbol sequences of any origin.

Key words: biological sequences, dynamic programming, generalized partition function, probability, *P* value

DOI: 10.1134/S0006350909050029

INTRODUCTION

The task of computing the probability of a set of symbol sequences arises in bioinformatic problems usually when there is a need to assess the reliability of search for some signals or patterns. The query set thereby comprises all sequences of given length m that contain a pattern exceeding the specified quality level Q; the latter can be specified in scalar as well as in vector form (see [1, 2]).

To formally state the problem of computing the probability of a sequence family, one must specify:

—the alphabet A;

—the sequence length m;

—the probability distribution ρ on the set A^m;

—the set of sequences $S \subseteq A^m$.

This paper presents a way of reducing the problem to calculating a so-called generalized partition function (GPF) for an appropriate graph (see [3, 4]), which can be efficiently done by dynamic programming. The graph is built from the finite-automaton (FA) descriptions of the initial set $S \subseteq A^m$ and the probability distribution ρ.

The paper is structured as follows. First the means of FA representation are described for the probability distribution (section 1) and the set of sequences (section 2); section 3 formulates the problem, defines the probabilistic accepting automaton, and describes how such an automaton serves to reduce the problem. Section 4 gives estimates of the complexity of the proposed algorithm for various means of specifying the probabilities. The main ideas of this algorithm as applied to seed sensitivity have been presented elsewhere [4].

Abbreviations: (D)FA, (deterministic) finite automaton; GPF, generalized partition function; HMM, hidden Markov models; PAA, probabilistic accepting automaton; PT, probability transducer.

1. PROBABILITY ASSIGNMENT

Let the probability distribution on A^m be specified with a FA probability transducer (PT) [4]. As regards the possibilities of describing probability distributions on symbol sequence sets, PTs are equivalent to hidden Markov models (HMM) [5].

Informally speaking, PT is a non-deterministic probabilistic FA without finals states, which at every $q \longrightarrow q'$ transition outputs a probability of a, where q and q' are states and a is a symbol of the alphabet. The relation between PTs and HMM is analogous to the equivalence of Mealy and Moore automata [6].

Definition 1. A *probability transducer* over A is a quadruple $G = \langle Q, q^0, A, \rho\rangle$, where Q is a finite set of states, q^0 is the initial state, A is the alphabet, $\rho: Q \times A \times Q \longrightarrow [0, 1]$ is the probability function such that

$$\forall \mathbf{q} \in \mathbf{Q} \sum_{q' \in Q,\, a \in A} \rho(q, a, q') = 1.$$

Definition 2. Let $G = \langle Q, q^0, A, \rho\rangle$ be the probability transducer over alphabet A. A *transition* in G is a triple $e = \langle q, a, q'\rangle$ such that $\rho(q, a, q') > 0$. Letter a is a label of the transition, *label*(e). States q and q' are respec-

tively the start and the end states of the transition, *start*(*e*), *end*(*e*).

The succession $P = (e_1, ..., e_n)$ of transitions is called the *path* in G, if for any $i \in \{1, ..., n-1\}$ it is $start(e_{i+1}) = end(e_i)$. The *label* of path $P = (e_1, ..., e_n)$ is the word $label(e_1)...label(e_n)$. Path P is *initial* if its start state is the q^0 of G. The *probability* $\rho(P)$ of the path is the product $\rho(P) = \Pi_{i=1,n}\rho(e_i)$.

The transducer G is *deterministic* if for any pair $\mathbf{q} \in \mathbf{Q}$, $a \in A$ there can be at most one transition $\langle q, a, q'\rangle$.

Definition 3. Let $G = \langle Q, q^0, A, \rho\rangle$ be the probability transducer over alphabet A. The probability of word w relative to G is $P_G(w)$ equal to the sum probability of all initial paths with label w; if there are no such paths, $P_G(w) = 0$. The probability of finite language $L \subseteq A^*$ is the sum $P_G(L)$ of the probabilities of all words $w \in L$. Obviously, for any n it is true that $P_G(A^n) = 1$.

The conventional means of specifying the probability distributions on words can be expressed in FA-PT terms. A Bernoulli distribution (see, e.g., [7]) is described with a one-state PT. For Markovian distributions of k-th order (e.g. [8]) the probability of the next symbol depends on k preceding symbols; such a distribution can be described with a deterministic PT with maximally $|A|^k$ states. The distributions described by HMM [9] in the general case correspond to nondeterministic PTs; the set of PT states is that of the HMM, plus possibly a special initial state.

Statement 1. Let $G = \langle Q, q^0, A, \rho\rangle$ be a PT over alphabet A. Then for G one can construct a hidden Markov model $H(G)$ that specifies on A^* the same probability distribution as G does. Inversely, for each model H one can construct a $G(H)$ that specifies on A^* the same probability distribution as H does.

The proof follows directly from the definitions of PT and HMM [5].

Note. In biological applications it is sometimes expedient to take as the universum not the entirety of sequences of given length but a finite set U of another type (see example in [10]). In this case it can be taken that the probability distribution is nonetheless specified on set $A^{m(U)}$ such that $U \subseteq A^{m(U)}$, while the significance of set $S \subseteq U \subseteq A^{m(U)}$ can be estimated with a conditional probability $Prob(S)/Prob(U)$. Henceforth it is assumed that the probability distribution is specified on set A^m for appropriate alphabet A and word length m.

2. DESCRIPTION OF SEQUENCE FAMILIES

The set $S \subseteq A^m$ the probability of which is sought for is finite and hence is accepted by DFA [4] with at most $m|S|$ states. However, it is often possible to construct an S-recognizing automaton with much fewer states. As a rule, set S is described as

$$S = A^m \cap S' \quad (1)$$

where $S' \subseteq A^*$ is also recognized by the FA.

Definition 4. Let $B = \langle Q_B, q_B^0, Q_B^F, A, \varphi_B\rangle$ be a finite automaton recognizing set $S' \subseteq A^*$; m is a natural number. Let $B(m) = \langle Q_K, q_K^0, Q_K^F, A, \varphi_K\rangle$ be an automaton specified as follows:

1) $Q_K = \{\langle q_B^0, 0\rangle\} \cup Q_B \times \{1, ..., m\} \cup \{q^T\}$, where q^T is an additional dead-end state;

2) $q_K^0 = \langle q_B^0, 0\rangle$;

3) $Q_K^F = Q_K^F \times \{m\}$;

4) $\varphi_K(q^T, a) = q^T$ for arbitrary $a \in A$;

$\varphi_K(\langle q, k\rangle, a) = \langle \varphi_B(q, a), k+1\rangle$ if $k + 1 \le m$,

$\varphi_K(\langle q, k\rangle, a) = q^T$ otherwise.

Note that $B(m)$ can be regarded as a Cartesian product of B and the A^m-recognizing automaton. For the $|Q_K|$ states of $B(m)$ it is

$$|Q_K| = |Q_B| \cdot m + 2 = O(|Q_B| \cdot m) \quad (2)$$

Statement 2. Let set $S' \subseteq A^*$ be recognized by FA B, m a natural number. Then $B(m)$ recognizes $S = A^m \cap S'$.

The proof is obvious.

In many applications $S'$is specified by the following condition (where L is a finite number of words in alphabet A):

$$S' = \{w \in A^* | \exists u \in L \ (u \text{ is a subword of } w)\} \quad (3)$$

Condition (3) will be called the Aho–Corasick condition (see [11] presenting an algorithm of building for set L an automaton recognizing $S'(L)$ specified by such a condition).

3. FORMULATION OF THE PROBLEM AND ALGORITHM

Thus, the problem of interest can be formulated as follows.

Given:

1) alphabet A,

2) word length m;

3) transducer $G = \langle Q, q^0, A, \rho\rangle$ specifying a probability distribution ρ_G on A^m;

4) language $L \subseteq A^m$ and DFA K recognizing L.

Find the probability $P_G(L)$ of set L relative to ρ_G.

Solution of the problem is based on the notion of a probabilistic accepting automaton (PAA). Informally speaking, PAA is a nondeterministic PT with a subset of accepting states. Let PAA be defined as the Cartesian product of an automaton without output and a PT over the same alphabet.

Definition 5. Let $K = \langle Q_K, q_K^0, Q_K^F, A, \varphi\rangle$ be a DFA without output, $G = \langle Q_G, q_G^0, A, \rho\rangle$ a PT over alphabet A. The probabilistic accepting automaton (PAA) $W = K \times G$ is a pentuple $W = \langle Q_W, q_W^0, Q_W^F, A, \rho_W\rangle$ where

$$Q_W = Q_K \times Q_G;$$

$$q_W^0 = q_K^0 \times q_G^0;$$

$$Q_W^F = \{\langle k, g\rangle = Q_K \times Q_G | k \in Q_K^F\};$$

$$\rho_W(\langle k, g\rangle, a, \langle k', g'\rangle) = \rho(g, a, g') \quad \text{if} \quad \varphi(k, .\, a) = k',$$

$$\rho_W(\langle k, g\rangle, a, \langle k', g'\rangle) = 0 \quad \text{otherwise.}$$

Definition 6. An initial path in PAA is called *full* if it ends in an accepting state.

Statement 3. Let G be a PT, $L \subseteq A^m$ a finite language and K an acyclic DFA accepting L.

Then

1. The $W = K \times G$ graph is acyclic.

2. The probability $P_G(L)$ of language L relative to transducer G is equal to the sum of probabilities of all full paths in PAA $W = K \times G$.

Proof:

1. Follows from that K is acyclic.

2. Since K is a DFA, then

1) for each path in G there is a single path in $W = K \times G$ and inversely, at that the path in G is a projection of the path in W onto G;

2) for arbitrary $w \in A^*$ either any path with label w leads to an accepting state (at $w \in L$) or none of them does (at $w \notin L$).

Therefore the set of all paths in G with labels from L is in one-to-one correspondence with the set of full paths in $W = K \times G$, and the probabilities of the corresponding paths coincide. This concludes the proof of Statement 3.

Consequence. Computation of $P_G(L)$ boils down to calculating the GPF (see [3]) in the W graph.

Note. Since L is a finite language, it is always possible to construct an acyclic FA accepting L.

4. ALGORITHM COMPLEXITY

The consequence of Statement 3 gives the algorithm of computing the probability of the finite set $S \subseteq A^m$ assuming that the transducer and the recognizing automaton are known.

Statement 4. Consider alphabet A and a finite set $L \subseteq A^m$. Let $K = \langle Q_K, q_K^0, Q_K^F, A, \varphi_K\rangle$ be a DFA recognizing L, and $G = \langle Q_G, q_G^0, A, \rho_G\rangle$ a PT specifying ρ_G on A^m.

Then the probability $P_G(L)$ of set L relative to ρ_G can be found by dynamic programming in time $O(|Q_G|^2 \cdot |Q_K| \cdot |A|)$ using memory $O(|Q_G| \cdot |Q_K|)$.

Proof. By Statement 3, $P_G(L)$ can be calculated as the sum of probabilities of all full paths in $W = K \times G$. This sum can be found by dynamic programming (see [12]) in time $O(|D_W|)$ with memory $O(|Q_W|)$, where D_W is the set of edges in the W graph, Q_W is the set of W states. Note that [3] reviews the application of this technique to various problems of bioinformatics.

By construction, the PAA W has $|Q_G| \cdot |Q_K|$ states and for each pair of the state q of W and the symbol $a \in A$ there are at most $|Q_G|$ edges outgoing from q. This concludes the proof.

Statement 5. Consider alphabet A and a finite set $L \subseteq A^m$. Let $K = \langle Q_K, q_K^0, Q_K^F, A, \varphi_K\rangle$ be a DFA recognizing L, and $G = \langle Q_G, q_G^0, A, \rho_G\rangle$ a deterministic PT specifying ρ_G on A^m.

Then the probability $P_G(L)$ of set L relative to ρ_G can be found by dynamic programming in time $O(|Q_G| \cdot |Q_K| \cdot |A|)$ using memory $O(|Q_G| \cdot |Q_K|)$.

Proof. Identical to that for Statement 4, on the strength of that for a deterministic G it is true that $|D_W| \le |Q_G| \cdot |Q_K| \cdot |A|$.

Consequence 1. Let the probability distribution on A^m be Bernoulian. Then for the time and memory needed to compute $P_G(L)$ we have $Time_{\text{Bern}} \le O(|Q_K| \cdot |A|)$ and $Space_{\text{Bern}} \le O(|Q_K|)$.

Proof follows from that in the Bernoullian case, $|Q_G| = 1$.

Consequence 2. Let the probability distribution on A^m be Markovian of order r. Then for the time and memory needed to compute $P_G(L)$ we have $Time_{\text{Mark}} \le O(|Q_K| \cdot |A^{r+1}|)$ and $Space_{\text{Mark}} \le O(|Q_K| \cdot |A^r|)$.

Proof follows from that in the r-th order Markovian case, $|Q_G| = O(|A^r|)$.

An important case when the estimate of Statement 4 can be strengthened is described below.

Statement 6. Consider alphabet A and a finite set $L \subseteq A^m$. Let $B = \langle Q_B, q_B^0, Q_B^F, A, \varphi_B\rangle$ be an automaton recognizing set L' such that $L' \cap A^m = L$, and $G = \langle Q_G, q_G^0, A, \rho_G\rangle$ be a transducer specifying ρ on A^m. Then the probability $P_\rho(L)$ of set L relative to ρ can be found by dynamic programming in time $O(|Q_B| \cdot |Q_G|^2 \cdot |A| \cdot m)$ using memory $O(|Q_B| \cdot |Q_G|)$.

Proof. On the strength of Statement 2, automaton $B(m) = \langle Q_K, q_K^0, Q_K^F, A, \varphi_K\rangle$ (Definition 5) recognizes set $L = L \cap A^m$. The number of states $|Q_K| = |Q_B| \cdot m + 2 = O(|Q_B| \cdot m)$, whence the time bound. Proving the memory bound requires closer consideration of the algorithm [3] used to calculate $P_\rho(L)$. For $k \in Q_K$ and $a \in A$, we define set

$$Pred(k) = \{k' \in Q_K | \exists a \in A(\varphi_K(k', a) = k).$$

Recall that the algorithm for GPF inspects the state set $Q_W = Q_K \times Q_G$ in the topological order. For every state $(k, g) \in Q_W$ the sum probability $P(k, g)$ is calculated for all initial paths ending in (k, g), using the previously calculated $P(k', g')$ only for such (k', g') that $k' \in Pred(k)$.

Denote as the r-th layer in $B(m)$ the set $Layer(B, r)$ of states $\langle b, r\rangle$ where $b \in Q_B$, $r = 0, 1, \ldots, m$. From the definition of $B(m)$ ensue the following auxiliary state-

ments. Analogously, the r-th layer in PAA $W = B(m) \times G$ is the set of all pairs (x, g) such that $x \in Layer(B, r)$.

(i) Let the topological order $<_B$ be specified on the state set Q_B. Then the relation $<_K$ on Q_K, where

$$(b, r) <_K (b', r') \Leftrightarrow (b <_B b') \vee ((b = b') \,\&\, (r < r'),$$

specifies the topological order on Q_K. In other words, states $(k, g) \in Q_W$ can be processed layer-wise.

(ii) The topological order on Q_K together with an arbitrary order on Q_G lexicographically specify the topological order on Q_W.

(iii) Let $k \in Q_K$ and $k \in Layer(B, r)$. Then $Pred(k) \subseteq Layer(B, r-1)$.

From statements (i)–(iii) it follows that states $(k, g) \in Q_W$ can be processed layer-wise and consequently at each moment it is enough to store the $P(k, g)$ values for only two layers: current and preceding. Considering that every layer of W contains $|Q_B| \cdot |Q_G|$ states, the required memory estimate is obtained. Statement 6 is proven.

Statement 7. Consider alphabet A and a finite set $L \subseteq A^m$. Let $B = \langle Q_B, q_B^0, Q_B^F, A, \varphi_B\rangle$ be an automaton recognizing set L' such that $L' \cap A^m = L$, and $G = \langle Q_G, q_G^0, A, \rho_G\rangle$ be a deterministic transducer specifying ρ on A^m. Then the probability $P_\rho(L)$ of set L relative to ρ can be found by dynamic programming in time $O(|Q_B| \cdot |Q_G| \cdot |A| \cdot m)$ using memory $O(|Q_B| \cdot |Q_G|)$.

Proof is analogous to those for Statements 5 and 6.

Consequence 1. Let the probability distribution on A^m be Bernoullian. Then for the time and memory needed to compute $P_G(L)$ we have $Time_{\text{Bern}} \le O(|Q_B| \cdot |A| \cdot m)$ and $Space_{\text{Bern}} \le O(|Q_B|)$.

Proof follows from that in the Bernoullian case, $|Q_G| = 1$.

Consequence 2. Let the probability distribution on A^m be Markovian of order r. Then for the time and memory needed to compute $P_G(L)$ we have $Time_{\text{Mark}} \le O(|Q_B| \cdot |A^{r+1}| \cdot m)$ and $Space_{\text{Mark}} \le O(|Q_B| \cdot |A^r|)$.

Proof follows from that in the r-th order Markovian case, $|Q_G| = O(|A^r|)$.

The last statement of the paper concerns an important special case when set S is specified by the Aho–Corasick condition and the probability distribution is Markovian of the order $r \ge 1$. In this case the estimates of consequence 2 from Statement 7 can be substantially improved.

Statement 8. Consider alphabet A and a finite set $L \subseteq A^m$. Let M be a finite set of words, set

$$L' = \{w \in A^* | w \text{ contains a subword from } M\}$$

and at that it is

$$L = A^m \cap L'.$$

Let further $B = \langle Q_B, q_B^0, Q_B^F, A, \varphi_B\rangle$ be an Aho–Corasick automaton for M that recognizes L', and ρ a probability distribution on A^m specified by a Markov model of order r.

Then the probability $P_\rho(L)$ of set L relative to ρ can be found by dynamic programming in time $O((|Q_B| + |A^r|) \cdot |A| \cdot m)$ with memory $O(|Q_B| + |A^r|)$.

Proof. Let $G = \langle Q_G, q_G^0, A, \rho_G\rangle$ be a deterministic PT specifying a Markovian ρ on A^m. On the strength of Statement 7, it is sufficient to prove that the layer of $B(M) \times G$ contains $O(|Q_B| + |A^r|)$ vertices.

According to Section 1, $|Q_G| = O(|A^r|)$, with Q_G elements being words in alphabet A, of length less than r. The states of automaton B are words $v \in Pref(M)$, see Section 2. Therefore the states of PAA $W = B(m) \times G$ are triples $\langle v, w, s\rangle$, where $v, w \in A^*$; $s \in \{0, 1, ..., m\}$; v is the state of transducer G; $\langle w, s\rangle$ is the state of K, and the initial state of W is $\langle \varepsilon, \varepsilon, 0\rangle$.

Let λ be the W transition function. By the design of Aho–Corasick automaton B, transducer G, and PAA W, if for a certain word $u \in A^*$ it is

$$\lambda(\langle \varepsilon, \varepsilon, 0\rangle, u) = \langle v, w, s\rangle,$$

then words v and w are suffixes of u.

Hence for every state $\langle v, w, s\rangle$ attained from the initial state of W it is true that

$$v \text{ is suffix of } w \quad \text{or} \quad w \text{ is suffix of } v \qquad (4)$$

From (4) it follows that the number of states in a layer of W is estimated at $O(|Q_B| + |Q_G|)$, quod erat demonstrandum.

5. CONCLUSIONS

The proposed algorithm can be used in various bioinformatic problems. To deploy it one should build finite automata representing the probability distribution and the query sequence family. A probability transducer is easily constructed for the main models (Section 1). Constructing an automaton recognizing the sequence family may prove a more difficult task. As a rule, use can be made of the Aho–Corasick construct [11], as exemplified [2, 4]. Sometimes it is useful to present the set under study as a difference $S = S_1 - S_2$, where S is part of S_1. In this case the probability of S is equal to the difference of probabilities of S_1 and S_2. the idea of such decomposition is that S_1 and S_2 automata may in sum have fewer states than the automation for S, which improves the algorithm performance. An example of this is given elsewhere [13].

ACKNOWLEDGMENTS

The author is grateful to A.V. Finkelstein, I.I. Tsitovich, G. Kucherov, L. Noé, V.Yu. Makeev, V. Boeva, and M. Regnier for fruitful discussions.

The work was supported by the Russian Foundation for Basic Research (08-01-92496-NTsNIL, 09-04-01053) and Migec-Inria (France).

REFERENCES

1. M. A. Roytberg, M. N. Simeonenkov, and O. Yu. Tabolina, Biofizika **43**, 581 (1998).
2. V. Boeva, J. Clement, M. Regnier, et al., Algorithms Mol. Biol. **2** (1) (2007).
3. A. V. Finkelstein and M. A. Roytberg, BioSystems **30** (1-3), 1 (1993) (spec. vol. Computer Genetics, Ed. by P. A. Pevzner and M. S. Gelfand).
4. G. Kucherov, L. Noé, and M. A. Roytberg, J. Bioinform. Comput. Biol. **4** (2), 553 (2006)
5. R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis. Probabilistic Models of Proteins and Nucleic Acids* (Cambridge University Press, 1998).
6. B. A. Trakhtenberg and Ya. M. Barzdin', *Finite Automata. Behavior and Synthesis* (Nauka, Moscow, 1970) [in Russian].
7. M. Li, B. Ma, D. Kisman, and J. Tromp, J. Bioinform. Comput. Biol. **2** (3), 417 (2004).
8. J. Buhler, U. Keich, and Y. Sun, in *Proceedings of the 7th Annual International Conference on Computational Molecular Biology (RECOMB03)* (ACM Press, Berlin, 2003), pp. 67–75.
9. B. Brejova, D. Brown, and T. Vinar, in *Proceedings of the 14th Symposium on Combinatorial Pattern Matching, Morelia (Mexico)*, Ed. by M.C.R. Baeza-Yates and E. Chavez (Springer, 2003), pp. 42–54.
10. G. Kucherov, L. Noé, and M. Roytberg, IEEE/ACM Transact. Comput. Biol. Bioinf. **2**, 51 (2005).
11. A. V. Aho and M. J. Corasick, Commun. ACM **18**, 6 (1975).
12. A. V. Aho, J. E. Hopcroft, and J. Ullman, *The Design and Analysis of Computer Algorithms* (Addison-Wesley, Reading, 1974).
13. M. Regnier, Z. Kirakosyan, E. Furletova, and M. Roytberg, in *London Algorithmics 2008: Theory and Practice*, Ed. by J. Chan, J. W. Daykin, and M. S. Rahman (2009), pp. 10–43.