Computational Biology and Chemistry 34 (2010) 328–333

Contents lists available at ScienceDirect

Computational Biology and Chemistry

journal homepage: www.elsevier.com/locate/compbiolchem

Brief communication

Computation of mutual information from Hidden Markov Models

Daniel Reker[a,b], Stefan Katzenbeisser[b], Kay Hamacher[a,*]

[a] Theoretical Biology and Bioinformatics, Institute of Microbiology and Genetics, Department of Biology, TU Darmstadt, Schnittspahnstr. 10, 64287 Darmstadt, Germany
[b] Security Engineering Group, Department of Computer Science, Technische Universität Darmstadt, 64287 Darmstadt, Germany

ARTICLE INFO

Article history:
Received 29 June 2010
Received in revised form 30 August 2010
Accepted 30 August 2010

Keywords:
Hidden Markov Model
Mutual information
Dynamic
Programming
Co-evolutionary signals

ABSTRACT

Understanding evolution at the sequence level is one of the major research visions of bioinformatics. To this end, several abstract models – such as Hidden Markov Models – and several quantitative measures – such as the mutual information – have been introduced, thoroughly investigated, and applied to several concrete studies in molecular biology. With this contribution we want to undertake a first step to merge these approaches (models and measures) for easy and immediate computation, e.g. for a database of a large number of externally fitted models (such as PFAM). Being able to compute such measures is of paramount importance in data mining, model development, and model comparison. Here we describe how one can efficiently compute the mutual information of a homogenous Hidden Markov Model orders of magnitude faster than with a naive, straight-forward approach. In addition, our algorithm avoids sampling issues of real-world sequences, thus allowing for direct comparison of various models. We applied the method to genomic sequences and discuss properties as well as convergence issues.

1. Introduction

Evolutionary pressure enforces correlations in biological sequences. A particularly promising method to reveal the presence of such (co-)evolutionary signals and to investigate general information contained within biological data sets is the computation of the mutual information (MI) between different positions, e.g. in a set of strings of biological codes. The co-evolution of amino acids in a protein, for example, reveals itself by high MI content in a set of homologous sequences from various taxa. MI based studies have become an important tool to understand evolutionary processes in such gene products (Boba et al., 2010; Hamacher, 2008, 2010; Weil et al., 2009).

At the same time, biological sequences are routinely modeled in bioinformatics by Hidden Markov Models (HMM) (Durbin et al., 1998) and large databases of (manually) curated models exist (Finn et al., 2006). Besides these applications in evolutionary and computational biology, signal creating processes in neurobiology, speech synthesis (Dines and Sridharan, 2001; Zen et al., 2007) or biochemistry (Grundy et al., 1997) are frequently modeled by HMMs, too. Such HMMs capture the essentials of the consensus sequence as well as additional "fluctuations" in the individual sequences.

Due to their widespread usage and the importance of evolutionary signals, understanding the ability of HMMs to model the underlying correlation in sequences is of great importance. One has also to concede that HMMs themselves – in particular as a plain collection of probability values – are not instructive at all. They do not provide immediate insight into the non-local effects in the sequences under investigation. The MI on the other hand offers a direct, intuitive, and transmissible interpretation; in particular one can easily visualize it (Bremm et al., 2010).

A generic framework based on an analytical approach to compute MI from HMMs directly is desirable. Such a framework avoids problems of empirical data sets with finite size. The naive approach of computing MI from sequences emitted by an HMM would typically be subject to statistical fluctuations (Weil et al., 2009). In particular, this offers the possibility to use the existing biological knowledge and machine readable information in the form of HMMs in an automated fashion (Stiller and Radons, 1999).

From the algorithmic point of view the MI computation from HMMs poses an interesting problem as one needs to find an algorithm to efficiently compute the combinatorial many paths through the state space of an HMM.

For all of the above mentioned practical and theoretical arguments we want to show in this paper how to compute the MI for (homogeneous) HMMs efficiently. The investigation of homogeneous HMMs is a first step in our final goal of mining general HMM databases for co-evolutionary signals. PFAM models (Finn et al., 2006), for example, model C- and N-termini of proteins by homogeneous sites in the respective HMM. Therefore we present here a first step to finally compute MI of HMMER models.

In particular the finding of functional motifs by HMMs can be facilitated by filtering for information rich regions (Horan et al., 2010). Our results provide for alternative insight into the

* Corresponding author. Tel.: +49 6151 16 5318; fax: +49 6151 16 2956.
E-mail address: kay.hamacher@gmail.com (K. Hamacher).

1476-9271/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compbiolchem.2010.08.005

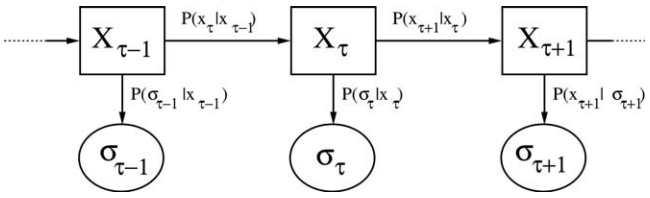

Fig. 1. The structure of a general Hidden Markov Model (HMM) with the emission probabilities $P(\sigma_\tau|x_\tau)$ and the transition probabilities $P(x_{\tau+1}|x_\tau)$.

information contained within HMMs and might therefore open an alternative route to such guided protocols.

2. Approach

2.1. Mutual information in sequences

Let $\Sigma := \{\sigma^1, \ldots, \sigma^m\}$ be the set of the m observed symbols in a biological sequence data set. We consider our data to be strings of these symbols with length $\tau_{end} \in \mathbb{N}$. A position in the string can therefore be referenced by any number $\tau \in \{1, 2, \ldots, \tau_{end}\} =: T \subset \mathbb{N}$. The MI between two positions $\tau, \tau' \in T$ in symbol sequences can be calculated by

$$\mathrm{MI}_{\tau,\tau'} = \sum_{\sigma_\tau, \sigma'_{\tau'} \in \sum} P(\sigma_\tau, \sigma'_{\tau'}) \cdot \log_2 \frac{P(\sigma_\tau, \sigma'_{\tau'})}{P(\sigma_\tau), P(\sigma'_{\tau'})}, \tag{1}$$

where $P(\sigma_\tau)$ is the probability to observe the symbol $\sigma_\tau \in \Sigma$ at position $\tau \in T$ and $P(\sigma_\tau, \sigma'_\tau)$ is the joint probability to observe symbol $\sigma_\tau \in \Sigma$ at position $\tau \in T$ and symbol $\sigma'_{\tau'} \in \sum$ at position $\tau' \in T$. Typically, these probabilities are estimated using frequencies of symbol occurrence in an empirical data set. We will show how to compute the probabilities directly from a given HMM.

2.2. Definition of Hidden Markov Models

We will consider a time-discrete, time-homogeneous HMM shown in Fig. 1 up to a string length of τ_{end}. For such HMMs we define $\chi := \{x^1, \ldots, x^n\}$ to be the set of the n hidden states of our HMM. It is described by three probability functions (Eddy, 1995):

- $P(\sigma_\tau|x_\tau) : \sum \times \chi \to \mathbb{R}$, which is the emission probability for a certain symbol $\sigma_\tau \in \Sigma$ conditioned on an internal state $x_\tau \in \chi$ at a position τ. As we use homogeneous HMMs, the emission probability for any hidden state does not depend on its position, rendering $P(\sigma_\tau > x_\tau)$ to be τ-independent and a universal table.
- $P(x_{\tau+1}|x'_{\tau'}) : \chi \times \chi \to \mathbb{R}$, which reflects the transition probability into a hidden state $x_\tau \in \chi$ from a state $\chi'_{\tau'} \in \chi$ at position τ'. Again, because of the time homogeneity of our model, these probabilities do not change with τ and need only to be defined for $\tau' = \tau + 1$ due the Markov property employed in HMMs.
- $\pi(x) : \chi \to \mathbb{R}$, which represents the starting probabilities at the (virtual) starting position $\tau = 0$.

To calculate MI from the HMM we need to find a way to calculate the values of $P(\sigma_\tau, \sigma'_\tau)$ and $P(\sigma_\tau)$ for any $\sigma_\tau, \sigma'_{\tau'} \in \sum$ and for any $\tau, \tau' \in T$ with the help of the given probability functions.

2.3. An analytic solution

In the following we will only consider the case $\tau < \tau'$. The case $\tau > \tau'$ is symmetric to $\tau < \tau'$ in Eq. (1) due to the symmetry in joint probabilities. The case $\tau = \tau'$ leads to the entropy for position τ and is not relevant for our investigation. To calculate $P(\sigma_\tau, \sigma'_\tau)$ and $P(\sigma_\tau)$, we use the probability $P(x_\tau)$ to be in a certain hidden state $x_\tau \in \chi$ for a given $\tau \in T$ and the joint probability $P(x_\tau|x'_{\tau'})$ to be in hidden state $x_\tau \in \chi$ for a given $\tau \in T$ and in hidden state $\chi'_{\tau'} \in \chi$ for a given $\tau' \in T$:

$$P(\sigma_\tau) = \sum_{x_\tau \in \chi} P(\sigma_\tau|x_\tau) P(x_\tau) \tag{2}$$

$$P(\sigma_\tau \sigma'_{\tau'}) = \sum_{x_\tau, x'_{\tau'} \in \chi} P(\sigma_\tau|x_\tau) P(\sigma'_{\tau'}|x'_{\tau'}) P(x_\tau|x'_{\tau'}). \tag{3}$$

Therefore the computation of the MI reduces to the determination of $P(x_\tau|x'_{\tau'})$ for every x_t, $\chi'_{\tau'} \in \chi$ and $\tau, \tau' \in T$ with $\tau < \tau'$ and the probabilities in Eqs. (2) and (3). The joint probability can be obtained using

$$P(x_\tau|x'_{\tau'}) = P(x'_{\tau'}|x_\tau) P(x_\tau). \tag{4}$$

We use the Markov property and the law of alternatives to evaluate the conditional probability $P(x'_{\tau'}|x_\tau)$ as

$$P(x'_{\tau'}|x_\tau) = \sum_{x''_{\tau'-1} \in \chi} P(x'_{\tau'}|x''_{\tau'-1}) P(x''_{\tau'-1}|x_\tau). \tag{5}$$

This equation can be used to build a recursive formula for the joint probability (4) for $\tau < \tau'$:

$$P(x_\tau, x'_{\tau'}) = \sum_{x''_{\tau'-1} \in \chi} P(x'_{\tau'}|x''_{\tau'-1}) P(x_\tau, x''_{\tau'-1}). \tag{6}$$

Implementing this formula, we can calculate the required values of $P(x_\tau|x'_{\tau'})$ iteratively. Thereby, we use the values of $P(x_\tau|x'_{\tau'})$ for every $\chi'_{\tau'} \in \chi$ to calculate $P(x_\tau|x''_{\tau'+1})$ for any $\chi''_{\tau'+1} \in \chi$.

2.4. Dynamic programming

The computation of $P(x_\tau|x''_{\tau'+1})$ can be performed efficiently following a dynamic programming approach (Bellman, 1952). To this end we express the sum in (6) as a matrix multiplication. Let $\mathbf{A}_\tau := (P(x_{\tau+1}|x'_\tau))_{x_{\tau+1}, x'_\tau \in \chi}$ be the matrix containing all transition probabilities for position τ. As we use a homogeneous HMM these matrices are the same for all $\tau \in T$, which we will simply call $\mathbf{A}$. With the matrix $\mathbf{A}$ we compute matrices $(P(x_\tau, x'_{\tau'}))_{x_\tau, x'_{\tau'} \in \chi} =: \mathbf{X}_{\tau,\tau'} \in \mathbb{R}^{|\chi| \times |\chi|}$ for all $\tau, \tau' \in T$ with $\tau \le \tau'$. These matrices will contain all required values and can simply be calculated by:

$$\mathbf{X}_{t,t'} = \mathbf{A} \cdot \mathbf{X}_{t,t'-1} \tag{7}$$

for $\tau < \tau'$. With the help of this equation we can calculate all $\mathbf{X}_{\tau,\tau'}$ for $\tau < \tau'$ iteratively as long as we know $\mathbf{X}_{\tau,\tau}$.

$\mathbf{X}_{1,1}$ is simply initialized with

$$\mathbf{X}_{1,1} := \mathrm{diag}(\pi(x_1), \pi(x_2), \ldots, \pi(x_n)). \tag{8}$$

To compute $\mathbf{X}_{\tau,\tau}$ for $\tau \in T, 1 < \tau$ we can use the values of $\mathbf{X}_{\tau-1,\tau}$ and the law of alternatives to compute $\mathbf{X}_{\tau,\tau}$ as

$$\mathbf{X}_{\tau,\tau} = \mathrm{diag}\left(\sum_{i \in \{1,\ldots,n\}} \mathbf{X}_{\tau-1,\tau}[i,1], \ldots, \sum_{i \in \{1,\ldots,n\}} \mathbf{X}_{\tau-1,\tau}[i,n] \right) \tag{9}$$

with $\mathbf{X}_{\tau,\tau'}[i,j]$ being the element in the i-th row and the j-th column of the matrix $\mathbf{X}_{\tau,\tau'}$. We can therefore simply collapse the columns of $\mathbf{X}_{\tau-1,\tau}$ to calculate $\mathbf{X}_{\tau,\tau}$. Using these equations we can calculate the required values of $P(x_\tau|x'_{\tau'})$ by first initializing $\mathbf{X}_{1,1}$ and then successively calculating $\mathbf{X}_{1,\tau}$ for all $\tau \in \{2, \ldots, \tau_{\mathrm{end}}\}$. After one step of this calculation, we obtain $\mathbf{X}_{1,2}$ with which we can calculate $\mathbf{X}_{2,2}$. Using $\mathbf{X}_{2,2}$ we can then successively calculate $\mathbf{X}_{2,\tau}$ for $\tau \in \{3, \ldots, \tau_{\mathrm{end}}\}$ and so on. After having calculated all necessary $\mathbf{X}_{\tau,\tau'}$, $P(\sigma_\tau, \sigma'_\tau)$ and $P(\sigma_\tau)$ can be calculated with help of (2) and (3). Again, these sums are expressible in matrix–vector multiplication which enables fast