Artificial Intelligence in Medicine (2005) **35**, 9—18

ELSEVIER

ARTIFICIAL INTELLIGENCE IN MEDICINE

http://www.intl.elsevierhealth.com/journals/aiim

Finding the biologically optimal alignment of multiple sequences

Hiroshi Mamitsuka *

Institute for Chemical Research, Kyoto University, Gokasho, Uji 611-0011, Japan

Received 8 November 2004; received in revised form 27 December 2004; accepted 12 January 2005

KEYWORDS
Multiple sequence alignment;
Multiple columns model;
Similarity scores;
Deterministic annealing;
Maximum entropy

Summary

Objective: Deterministic annealing, which is derived from statistical physics, is a method for obtaining the global optimum in parameter space. During the annealing process, starting from high temperatures which are then lowered, deterministic annealing *deterministically* find the (global) optimum at each temperature. Thus, deterministic annealing is expected to be more computationally efficient than stochastic sampling strategies to obtain the global optimum. We propose to apply the deterministic annealing technique to the problem of efficiently finding the biologically optimal alignment of multiple sequences.
Methods and material: We take a strategy based on probabilistic models for aligning multiple sequences. That is, we train a probabilistic model using given training sequences and obtain their alignment by parsing, i.e. searching for the most likely parse of each sequence and gaps using the trained parameters of the model. In this scenario, we propose a new stochastic model, which is simple enough to be suited to multiple sequence alignment and, unlike existing stochastic models, say a profile hidden Markov model (HMM), allows us to use similarity scores between symbols (or a symbol and a gap). We further present a learning algorithm for our simple model by combining deterministic annealing with an expectation—maximization (EM) algorithm. We emphasize that our approach is time-efficient, even if the training is done through an annealing process.
Results: In our experiments, we used actual protein sequences whose three-dimensional (3D) structures are determined and which are all aligned based on their 3D structures. We compared the results obtained by our approach with those by other existing approaches. Experimental results clearly showed that our approach gave the best performance, in terms of the similarity to the structurally determined alignment, among the approaches tested. Experimental results further indicated that our approach was ten times more efficient in terms of actual computation time than a competing method.
© 2005 Elsevier B.V. All rights reserved.

* Tel.: +81 774 31 4901; fax: +81 774 31 4904.
E-mail address: mami@kuicr.kyoto-u.ac.jp.

0933-3657/$ — see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.artmed.2005.01.007

1. Introduction

Multiple sequence alignment has been one of the most important problems in bioinformatics. Finding the global optimum of aligning two sequences has been successfully solved by a dynamic programming algorithm [1]. However, this algorithm cannot be effectively extended for the problem of aligning multiple sequences, because this is an NP-complete problem [2].

Thus, mainly the following three types of approaches are taken in practical situations [3,4]. The first approach is a so-called 'progressive alignment' strategy, which was proposed in 1980s [5]. Briefly, the progressive alignment strategy repeats the following three steps until all given sequences are aligned. First, two sequences are chosen from given multiple sequences, and then the two are aligned by the dynamic programming algorithm and are replaced with the resulting pairwise alignment. Though it is certain that this strategy allows us to align a large number of given sequences in a practical amount of computation time, the resulting alignment is often not necessarily optimal, because the result is affected by the local (partial) information produced by each pairwise alignment.

The second is the strategy based on probabilistic models (in particular hidden Markov models (HMMs)), which were proposed in mid-1990s [6], and the third is other heuristic strategies, which have been partly based on either or both of the above two strategies (e.g. [7]). Our method belongs to the second strategy, which is currently the most popular approach for multiple sequence alignment and related areas (see e.g. [8]). As given sequences are dealt with at a time, i.e. by a batch updating manner, this approach, say a method of learning an HMM, is not affected by peculiarities of particular pairs in a given set of data, and is expected to produce a better alignment than that obtained with progressive alignment. However, a popular learning algorithm of the HMM, called 'Baum—Welch' (or 'forward—backward'), is a local optimization algorithm, and the resulting alignment obtained is often far from the global optimum. To cover the weakness of the local optimization algorithm, priors on the probability parameters are used [9]. In practice, however, the priors have to be obtained by other methods such as progressive alignment, and essentially the same problem occurs with the prior-based approach as with the progressive alignment strategy. Thus, to overcome the problem of local optimization, it is important to train a probabilistic model using a learning algorithm which can avoid falling into a local optimum. Another weakness of the existing approaches based on learning a probabilistic model is that they do not allow us to use similarity scores between symbols (or a symbol and a gap), and this feature of the model also makes it difficult to obtain a satisfactory sequence alignment.

In order to overcome the first problem, stochastic sampling, such as Markov chain Monte Carlo, has been generally used (e.g. [10—12]), and the use of simulated annealing [13] or Gibbs sampling [14] to train probabilistic models has been proposed in the literature of bioinformatics. However, stochastic sampling is a quite time-consuming process, and if it is combined with annealing it becomes even more so, since it requires a great number of iterations. On the other hand, there is a method called *deterministic annealing* (e.g. [15]) for obtaining the global optimum without using stochastic sampling, and hence this algorithm is expected to be more efficient (in terms of computation time) than the methods using stochastic sampling. Deterministic annealing, which is motivated by statistical physics, uses the annealing process, which starts at high temperatures which are then lowered. Unlike the stochastic relaxation process as seen in simulated annealing, deterministic annealing *deterministically* optimizes an energy function which is defined for a given probabilistic model at each temperature. The energy function, which corresponds to the *free energy* in statistical physics, represents the rough (smooth) parameter space (containing only one local minimum) of the given probabilistic model at high temperatures. The minimum is then traced while the temperature is gradually lowered, until finally the free energy is coincident with an expected energy function of the probabilistic model when the temperature is equal to 0. The free energy is derived based on the principle of maximum entropy, and a Lagrange multiplier in the derivation is inversely proportional to the temperature.

In this paper, we propose to apply deterministic annealing to train a probabilistic model in the problem of aligning multiple sequences. Deterministic annealing is more time-efficient than any stochastic relaxation methods, but it still requires a lot of iterations. We then propose a simple probabilistic model, which we call MCM, standing for 'Multiple Columns Model'. The MCM is much simpler than the HMM, and it then can be trained computationally efficiently, even if we train it using an annealing process. Another advantage of the MCM is that it allows us to use any similarity scores between symbols if the score satisfies a simple requirement. When we assume the length of the aligned sequences to be N, an MCM has N columns, each of which has a set of parameters. At each column, a parameter is defined for each symbol (including a

Download English Version:

https://daneshyari.com/en/article/9650356

Download Persian Version:

https://daneshyari.com/article/9650356

Daneshyari.com