BIOINFORMATICS APPLICATIONS NOTE Vol. 15 no. 12 1999
Pages 1064–1065

Sausage: protein threading with flexible force fields

Thomas Huber[1,*], Anthony J. Russell[2], Daniel Ayers[2] and Andrew E. Torda[2]

[1]ANU Supercomputing Facility and [2]Research School of Chemistry, Australian National University, Canberra ACT 0200, Australia

Received on March 28, 1999; accepted on June 30, 1999

Abstract

Summary: *Sausage is a protein sequence threading program, but with remarkable run-time flexibility. Using different scripts, it can calculate protein sequence-structure alignments, search structure libraries, swap force fields, create models from alignments, convert file formats and analyse results. There are several different force fields which might be classed as knowledge-based, although they do not rely on Boltzmann statistics. Different force fields are used for alignment calculations and subsequent ranking of calculated models.*

Availability: *Freely available to academics at ftp://ftp.rsc.anu.edu.au/pub/torda/sausage/README*

Contact: *Andrew.Torda@anu.edu.au*

There are many approaches to protein structure prediction from sequence information, but protein threading has achieved some popularity when there is no significant homology between the sequence and any known structure (Westhead and Thornton, 1998; Sternberg, 1996). The general approach involves taking a sequence and testing it on each member of a library of known protein structures. On each template, one must find the optimal sequence to structure alignment according to some score or force field. These alignments are then ranked and the best ones taken as candidate predictions (Jones *et al.*, 1992; Sippl, 1993). Sausage (Sequence-structure Alignment Using a Statistical Approach Guided by Experiment) is primarily a protein threading program, but differs from others in its force fields and alignment methods.

The best known functions (knowledge-based) are constructed by surveying known protein structures and assuming that the inter-particle distances follow a Boltzmann distribution. Logarithms of frequencies then give a table-driven score function (Jones *et al.*, 1992; Sippl, 1990).

The sausage score functions are based on a different philosophy. Generally, one defines the aim of a score function (fold recognition) and casts it as a function of parameters (and calibration proteins). If one optimises this function with respect to parameters, one is effectively optimising force field quality.

Typically the force fields have five interaction sites per amino acid (four backbone atoms and one side-chain site) with all the identity of the amino acid represented by the side-chain site. Pair-wise interaction terms are represented by hyperbolic tan functions scaled by adjustable parameters and a simple neighbour-counting environment term is added mainly to account for solvation effects. The final score is the sum over all pairwise interactions and environment terms (Huber and Torda, 1998). The methodology has also been used to optimise score functions based on other functional forms (such as table-driven) and various measures of force field quality (Ayers *et al.*, 1999a).

The next issue in protein threading is calculating a sequence to structure alignment. For a conventional pair-wise score function, the problem is NP-complete (Lathrop, 1994) so one needs some approximation. The approach adopted in sausage is to split the prediction calculation into separate steps of sequence-structure alignment and structure ranking (Huber and Torda, 1999). A force field approximation is used in the first step which allows an optimal alignment to be calculated in polynomial time. This approximation is then removed and the final score for each model calculated using a force field with no approximations. The NP-complete nature of the problem arises since one must score each residue of the sequence in the field due to its neighbours. Unfortunately, the identity of the neighbours is not known since they have not been aligned. To avoid this problem, score functions have been built which use the identity of only one member of each interaction pair. For example a conventional neighbour-specific score function for three amino acid types would have parameters for pairs AA, AB, AC, BB, BC, CC. In an alignment (neighbour non-specific) score function, there are parameters for AX, BX and CX where X is a generic amino acid type. The X residue is conceptually an average amino acid, but its parameters result from

*To whom correspondence should be addressed.

numerical optimisation and not averaging over an existing score function.

Regardless of score function, sausage provides two methods for calculating sequence to structure alignments, namely the Needleman and Wunsch (1970) and Gotoh (1982) methods. The latter method is attractive because of computational speed, but the Needleman and Wunsch algorithm has an interesting feature. In sequence-structure alignments, one can use geometric gap penalties of the form $k_{\mathrm{gap}}(d_{\mathrm{CN}^2} - d_{0^2})$ where d_{CN} is the distance between the carbonyl carbon of residue i and the backbone nitrogen of residue $i + 1$ and k_{gap} is scales the overall penalty. d_0 is the ideal distance ($\approx$ 1.3Å). While this is intuitively appealing, it is computationally expensive since the gap penalty must be calculated in the inner loop of the dynamic programming algorithm.

As a practical compromise between speed and elegance, initial alignments can be calculated using the fast method, but final scores for ranking can be calculated using geometric gap penalties. Finally, alignments can be refined in a conventional force field using Monte Carlo/simulated annealing. The code can even calculate alignments using the popular frozen approximation (Godzik *et al.*, 1992; Sippl, 1993).

Sausage has been constructed so it can be a test bed for different methods or a tool for production calculations. A run might consist of looping over a template library, calculating alignments in the neighbour non-specific score function and ranking them in a neighbour specific force field. This is not, however, hard-coded in the program. One might instead compare score functions or build models from alignments. To achieve this flexibility, sausage is written in C, but as a 'Tcl extension'. The run time script completely controls the calling of the inner C-coded functions. Similarly, control parameters are installed as Tcl variables which can be manipulated and looped over at run time. For example, the following excerpt would select a score function, open a parameter file, set the geometric gap penalty to 1000, calculate an alignment and print it out.

```
set scr_func score_tanh_cxa
set param [ open_str param $param_file ]
set pnlty_scl 1000
set algn [ open_str t_align $coord $seq
$param ]
print_str $algn
```

When appropriate, sausage can use some experimental data. Disulfide bonds can contribute to scores in the final ranking and in simulated annealing calculations. Secondary structure information can be used during alignments and final scoring. This was originally intended to take advantage of secondary structure determination from NMR assignments (Ayers *et al.*, 1999b), but the implementation can also directly read the output from a secondary structure prediction server (Rost *et al.*, 1994).

It is difficult to objectively compare protein threading programs, but results from a recent comparison/blind test are publicly available (http://predictioncenter.llnl.gov/casp3/Casp3.html). Since then, several new force fields and methods have been implemented. Sausage is continually evolving and continues to serve both for experimentation and routine calculations.

Acknowledgement

We thank Fujitsu (Japan) for financial support of TH and generous computational time.

References

Ayers,D.J., Huber,T. and Torda,A.E. (1999a) Protein fold recognition score functions: unusual construction strategies. *Proteins*, **36**, 454–461.

Ayers,D.J., Gooley,P.R., Widmer-Cooper,A. and Torda,A.E. (1999b) Enhanced protein fold recognition using secondary structure information from NMR. *Protein Sci.*, **8**, 1127–1133.

Godzik,A., Kolinski,A. and Skolnick,J. (1992) Topology fingerprint approach to the inverse protein folding problem. *J. Mol. Biol.*, **227**, 227–238.

Gotoh,O. (1982) An improved algorithm for matching biological sequences. *J. Mol. Biol.*, **162**, 705–708.

Huber,T. and Torda,A.E. (1998) Protein fold recognition without Boltzmann statistics or explicit physical basis. *Protein Sci.*, **7**, 142–149.

Huber,T. and Torda,A.E. (1999) Protein sequence threading, the alignment problem and a two step strategy. *J. Comput. Chem.*, **20**, 1455–1467.

Jones,D.T., Taylor,W.R. and Thornton,J.M. (1992) A new approach to protein fold recognition. *Nature*, **358**, 86–89.

Lathrop,R.H. (1994) The protein threading problem with sequence amino acid interaction preferences is NP-complete. *Protein Eng.*, **7**, 1059–1068.

Needleman,S.B. and Wunsch,C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**, 443–453.

Rost,B., Sander,C. and Schneider,R. (1994) PHD—an automatic mail server for protein secondary structure prediction. *Comput. Appl. Biosci.*, **10**, 53–60.

Sippl,M.J. (1990) Calculation of conformational ensembles from potentials of mean force. An approach to the knowledge-based prediction of local structures in globular proteins. *J. Mol. Biol.*, **213**, 859–883.

Sippl,M.J. (1993) Boltzmann's principle, knowledge-based mean fields and protein folding. An approach to the computational determination of protein structures. *J. Comput. Aided Mol. Des.*, **7**, 473–501.

Sternberg,M.J. E. (1996) *Protein Structure Prediction: A Practical Approach*. IRL Press, Oxford.

Westhead,D.R. and Thornton,J.M. (1998) Protein structure prediction. *Curr. Opin. Biotech.*, **9**, 383–389.