W532–W535 *Nucleic Acids Research, 2004, Vol. 32, Web Server issue*
DOI: 10.1093/nar/gkh357

Wurst: a protein threading server with a structural scoring function, sequence profiles and optimized substitution matrices

Andrew E. Torda*, James B. Procter and Thomas Huber[1]

University of Hamburg, Zentrum für Bioinformatik, Bundesstrasse 43, D-20146 Hamburg, Germany and
[1]Departments of Mathematics and Biochemistry, University of Queensland, Brisbane, QLD 4072, Australia

Received February 10, 2004; Revised and Accepted February 25, 2004

ABSTRACT

Wurst is a protein threading program with an emphasis on high quality sequence to structure alignments (http://www.zbh.uni-hamburg.de/wurst). Submitted sequences are aligned to each of about 3000 templates with a conventional dynamic programming algorithm, but using a score function with sophisticated structure and sequence terms. The structure terms are a log-odds probability of sequence to structure fragment compatibility, obtained from a Bayesian classification procedure. A simplex optimization was used to optimize the sequence-based terms for the goal of alignment and model quality and to balance the sequence and structural contributions against each other. Both sequence and structural terms operate with sequence profiles.

INTRODUCTION

Protein threading is justified by the observation that when a protein structure is solved, it is often similar to one that was previously known, even in the absence of detectable sequence homology. This is the main reason for optimism in the area of protein threading or, more generally, protein fold recognition. Given a sequence of interest, one should find the most appropriate template, calculate the sequence to template alignment and make the best possible initial model (1,2). Historically, threading was distinguished from pure sequence-based methods because it relied on a structural or force field-like score and might be able to find more remote similarities than sequence-based methods. More than ten years ago, there was some mixing of methods (3), but now the borders between techniques are even more blurred. Sequence comparison methods are now more sensitive (4–10) and many threading codes include terms from sequence similarity (11,12) or other prediction methods (13).

The Wurst program and server use protein threading, but with an optimized weighting of sequence and structural terms. Even the pure structure components take advantage of sequence profiles. Unlike most other packages, the contributions of the different components have come from a numerical optimization procedure geared to producing the best possible sequence to structure alignments as measured by the resulting models. This principle has been applied throughout, even to the selection of gap penalties and construction of substitution matrices.

OVERVIEW, INPUT/OUTPUT AND METHODS

The server accepts a sequence and builds a conservative sequence profile using psi-blast (4). This profile is aligned to just over 3000 single chain template structures from the protein data bank (PDB) (14) using the Smith and Waterman alignment algorithm (15) and a scoring scheme comprising of a combination of structure- and sequence-based terms. This gives every template an 'alignment score' for the sequence. At the same time, three-dimensional models are calculated from all alignments and evaluated using a more expensive quasi-energy function (16) and gap penalty cost based on distances within the model rather than simply the number of residues in a gap or insertion (17,18). The final score associated with each model is the combination of the alignment score, rescored model and gap penalties.

The models/alignments are sorted according to their final scores and a list of high-scoring alignments and models (in PDB format) is returned by email. The protein models contain side-chains only as far as the C^{β} atom and will have missing residues or gaps between residues when these occur in the alignment.

The server does let one choose the amount of output in the results, but it does not encourage freedom of choice in the calculation details. Gap penalties, substitution matrix and

*To whom correspondence should be addressed. Tel: +49 40 42838 7331; Fax: +49 40 42833 7332; Email: torda@zbh.uni-hamburg.de

coefficients for the various terms are fixed in a dictatorial manner since these terms have been optimized to fit each other.

Sequence to structure score function

The structure-based score function used by wurst allows alignments to be calculated with a Smith and Waterman algorithm (15), without the need for a double dynamic programming approach (3) or use of the frozen approximation (19–21). It could be seen as a statistical measure of the compatibility of a residue (and its immediate neighbours) with small fragments of structure. Unlike other fragment-based approaches, there is no assumption that fragments can be characterized by simple amino acid frequencies (22). Instead, wurst fragment library construction assumes that fragments can be described by a collection of intertwined or correlated sequence and structural tendencies. In practice, this is quite different from other fragment classifications. For example, two classes may be structurally similar but differ completely in their sequence properties. To build the structure score functions, every possible overlapping fragment of length nine was extracted from several hundred parameterization proteins. From each fragment, discrete descriptors (amino acid types) were collected alongside continuous descriptors for structural properties (φ, ψ angles, end to end C^{α} distance and a simple measure of solvent accessibility). A Bayesian classification procedure (23) was then used to find a near optimal mixture model describing the sequence–structure fragment data. In normal Bayesian style, one starts by noting that the probability $P(F_i \in c_j \,|\, F_i)$ that fragment is in class c_j is

$$P\left(F_i \in c_j \,|\, F_i\right) = \frac{P\left(F_i \,|\, F_i \in c_j\right) P\left(F_i \in c_j\right)}{P(F_i)}$$

$P(F_i)$ and $P(F_i \in c_j)$ are prior probabilities of the attribute vector (sequence and structure descriptors) of F_i and the class c_j, so they describe the degree of prior belief. For realistic data sets, the probabilities $P(F_i \in c_j \,|\, F_i)$ are impossible to compute, so approximations are used. Gaussian distributions were assumed for continuous descriptors and the sequence and structural descriptors of each fragment were taken as independently distributed. Then one can express $P(F_i \in c_j \,|\, F_i)$ in terms of products of distributions of observations within each class. The prior probability $P(F_i \in c_j)$ of a fragment being part of a class requires the class description itself and is obviously not initially available. Therefore, it was obtained by starting from initial estimates and optimization of the attribute probability distribution by expectation maximization (24). The last step is then to determine the total number of classes. We again apply Bayes' rule,

$$P(m \,|\, F) = \frac{P(m) P(F \,|\, m)}{P(F)}$$

where $P(m \,|\, F)$ is the posterior probability of class m given data F. In concrete terms, the form of $P(m)$ favours a small number of classes, thus minimizing the problems of over-fitting or fitting to noise. The classification used for the wurst score function reduced a set of more than 10^5 fragments to just over 400 classes.

Once constructed, the attribute distribution model is a direct estimate of sequence to structure compatibility. It can be directly cast as a score function for sequence to structure alignment which does not use the template sequence in any way. Finally, the implementation in the wurst server uses a sequence profile, generated by psi-blast (4). At each position, the score is calculated based on the fractional occupation of each residue type at each sequence position.

Optimization of sequence-based terms

As well as the structural terms, the wurst server uses a sequence similarity term, and this requires some amino acid substitution/compatibility matrix. Normally this reflects evolution and the rate at which amino acids mutate into each other (25), but for protein structure modelling, this is not exactly the property of interest. One simply wants the matrix that produces the alignments and consequent models which are geometrically closest to the correct answer (structure for the sequence) (26). In a parameterization process, the 210 values of the substitution matrix were treated as adjustable parameters as well as gap opening and widening costs. Alignment quality was then numerically optimized with respect to all these parameters. For this process, all that is required is a numerical measure of alignment quality and a method to optimize the parameters. This is not difficult to construct. Given a pair of proteins, A and B, of known structure, the sequence of A can be aligned to B and a model built. The best alignment is the one that produces the best model for A. Obviously, the sequence of B can be aligned to A and the quality of the model for B calculated. The final cost function is the average of model quality for a large set of proteins with at least some structural similarity.

The alignment quality was quantified using a previously introduced measure (27) which is close in spirit to the Q-value often quoted in the protein folding literature (28). Unlike the root mean square difference (RMSD) between Cartesian coordinates of model and correct structure, this measure is relatively insensitive to overall motions such as hinge bending in structures and, furthermore, avoids the problem that when a model is poor, it does not matter whether it is bad or very bad. Once the model quality is below some threshold, its contribution to a cost function should be near zero. In this implementation, the quality measure is based on the fraction of C^{α}–C^{α} contacts within a model that are within 4 Å of the correct value. In the uninteresting case of modelling a sequence to its own structure, this fraction is 1.0 for a perfect alignment. For remote homologues, no alignment will give a measure of 1.0, but in optimization terms, this does not matter. One simply wants the best alignments possible.

To optimize the substitution matrix and parameters used by the server, a set of 1544 pairs of proteins was selected with sequence identity ranging from less than 15% to 70%, wherein each member had structural similarity to the other with a root mean square difference of less than 5.0 Å for at least 50% of the sequence. Within each pair, the sequence of A was aligned to B and vice versa, giving more than 3000 alignments to be calculated. Next, each alignment calculation was repeated with 20% added to and then subtracted from the gap opening penalties. Although this could have the effect of building in tolerance to various parameter values, it is really an aid to removing susceptibility to noise. The final cost function came from summing over the different gap penalties and both permutations of alignments giving a set of 9264 = (6 × 1544)

alignments to be calculated at each optimization step. For most calculations described in the next section and for all calculations in the running server, a sequence profile was calculated using extremely conservative settings. A sequence was compared with the non-redundant database and sequences accepted with an expectation (e-) value of less than 10^{-10} for a maximum of three iterations. This profile was then expanded by no more than two iterations accepting homologues with an e-value less than 10^{-8}.

As described above, the procedure would generate an amino acid substitution matrix optimized for alignment quality as built by Qian and Goldstein (26). Although a similar calculation was done for testing, the wurst parameter set is quite different. Wurst alignments are calculated by constructing a score matrix $\mathbf{S}_{tot}$ for the Smith and Waterman (15) alignment where $\mathbf{S}_{tot} = \mathbf{S}_{struct} + w_{seq}\mathbf{S}_{seq}$. The structural/fragment score function described above is used to calculate $\mathbf{S}_{struct}$ with the use of sequence profiles. $\mathbf{S}_{seq}$ is calculated from sequence similarity and w_{seq} is a coefficient weighting the two terms. The final set of parameters was then gap opening and widening in sequence or profile, gap opening and widening in structure, w_{seq} and the 210 elements of a substitution matrix.

There are some consequences of this optimization procedure. All the parameters have been tuned to produce the best models possible, on average, over a range of difficulty. Because all the parameters have been simultaneously optimized, they form a self-consistent set. For example, the substitution matrix is not a general substitution matrix, but rather a numerical creation, fitted to the influence of the structural score function.

RESULTS

Ultimately, the results will be, and are currently being, judged by comparison with other approaches. The wurst server participates in two continuously running fold recognition assessment servers (29,30), hopefully highlighting its flaws and deficiencies or perhaps its success in recognizing remote homologues and the quality of alignments to these homologues. In the EVA assessment (30), it is also compared against modelling servers, to allow comparison with servers whose strength should be in producing relatively refined models.

Those longer-term assessments on new structures will be the most objective measure of quality. In the meantime, there are results that show why the server uses its current mixture of terms. Figure 1 shows the alignment quality, as measured by the cost function described in the previous section. It is a measure of model quality over the set of 1544 protein pairs. For each result, at least gap penalties were optimized to give the best results.

The poorest results are obtained with either simple sequence–sequence alignment using a blosum62 matrix (31) or with the fragment-based structural score function alone, as shown by the left-hand bars. The sequence–sequence alignment can be improved slightly by optimizing the substitution matrix (results not shown). More importantly, replacing one sequence by a sequence profile taken from psi-blast (4) has a marked effect both on the sequence term (labelled 'sequence + profile') and on the structural term (labelled 'bayes + profile'). In the case of sequence comparison, the improvement due to sequence profiles is entirely expected (32–35). With hindsight,

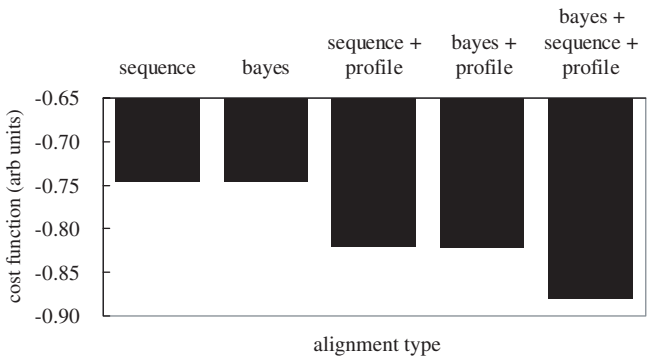

Figure 1. Alignment performance of sequence and structure terms. The cost is in arbitrary units, with 0.0 being worse than random and −1.0 being ideal. Sequence refers to a blosum62 matrix; bayes to the fragment-based, structure based term based on Bayesian statistics; sequence + profile to the use of sequence profiles with an optimized substitution matrix; bayes + profile to the structure-based term, but using sequence profiles rather than single amino acid types; bayes + sequence + profile refers to the combination of the preceding two terms.

it is not surprising that the structural term also benefits from using partial amino acid types. Technically, it is interesting that although the pure sequence and pure structure terms give similar results, they also contain some independent information.

The bar furthest to the right gives by far the best alignment results and is the method and parameter set used by the wurst server. It has the sequence–sequence term with an optimized substitution matrix added to the Bayesian statistics, fragment-based score term. All gap penalties and w_{seq} have come from the numerical optimization and both sequence and structure-based terms used sequence profiles/partial amino acid types.

DISCUSSION

The performance of the server is being judged objectively and very publicly (29,30), but given the description of the methods, one can speculate as to its strengths and weaknesses. Wurst has been built to produce the best possible alignments, but further improvements in the methodology are possible. For example, we are currently measuring the effect of replacing the sequence to profile terms with a profile to profile version with an appropriately optimized substitution matrix.

One may also note that a single substitution matrix is used, regardless of sequence identity. This is different from the series of Blosum matrices with different members built for differing degrees of amino acid similarity (31). It would certainly be possible to recalculate the wurst matrices using subsets of the protein pairs, but this would be at the cost of signal to noise ratio in the optimization calculation.

The description of the structure-based terms gives the impression that they are completely fixed. From the point of view of the wurst server, they are quite stable, but at the same time we are testing different structural descriptors. Fortunately, the kind of measurement given in Figure 1 is easy to repeat, so testing new terms is straightforward and the components of the wurst server will be replaced if there is numerical evidence they have been improved.

The only area where the wurst server asks for attention is in the final ranking of structures and in confidence measures.

Currently these are mostly based on the same measures used for the alignment calculation, but more complicated approaches have been used by other servers (36,37). Given the philosophy underlying the wurst server, this area will be the next target for an optimization and server update.

REFERENCES

1. Godzik,A. (2003) Protein fold recognition. In Bourne,P. E. and Weissig,H. (eds.), *Structural Bioinformatics*. Wiley, Hoboken, NJ, Vol. 44, pp. 525–546.
2. Torda,A.E. (1997) Perspectives in protein-fold recognition. *Curr. Opin. Struct. Biol.*, **7**, 200–205.
3. Jones,D.T., Taylor,W.R. and Thornton,J.M. (1992) A new approach to protein fold recognition. *Nature*, **358**, 86–89.
4. Altschul,S.F., Madden,T.L., Schaffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.
5. Eddy,S.R. (1996) Hidden Markov models. *Curr. Opin. Struct. Biol.*, **6**, 361–365.
6. Krogh,A., Brown,M., Mian,I.S., Sjölander,K. and Haussler,D. (1994) Hidden Markov models in computational biology: Applications to protein modeling. *J. Mol. Biol.*, **235**, 1501–1531.
7. Hughey,R. and Krogh,A. (1996) Hidden Markov models for sequence analysis: Extension and analysis of the basic method. *Comput. Appl. Biosci.*, **12**, 95–107.
8. Karplus,K., Barrett,C. and Hughey,R. (1998) Hidden Markov models for detecting remote protein homologies. *Bioinformatics*, **14**, 846–856.
9. Eddy,S.R. (1998) Profile hidden Markov models. *Bioinformatics*, **14**, 755–763.
10. Karplus,K., Sjolander,K., Barrett,C., Cline,M., Haussler,D., Hughey,R., Holm,L. and Sander,C. (1997) Predicting protein structure using hidden Markov models. *Proteins*, 134–139.
11. Panchenko,A.R., Marchler-Bauer,A. and Bryant,S.H. (2000) Combination of threading potentials and sequence profiles improves fold recognition. *J. Mol. Biol.*, **296**, 1319–1331.
12. Jones,D.T., Tress,M., Bryson,K. and Hadley,C. (1999) Successful recognition of protein folds using threading methods biased by sequence similarity and predicted secondary structure. *Proteins*, **S3**, 104–111.
13. Rost,B. (2001) Review: Protein secondary structure prediction continues to rise. *J. Struct. Biol.*, **134**, 204–218.
14. Berman,H.M., Westbrook,J., Feng,Z., Gilliland,G., Bhat,T.N., Weissig,H., Shindyalov,I.N. and Bourne,P.E. (2000) The Protein Data Bank. *Nucleic Acids Res*, **28**, 235–242.
15. Smith,T.F. and Waterman,M.S. (1981) Identification of common molecular subsequences. *J. Mol. Biol.*, **147**, 195–197.
16. Huber,T. and Torda,A.E. (1998) Protein fold recognition without boltzmann statistics or explicit physical basis. *Protein Sci.*, **7**, 142–149.
17. Huber,T. and Torda,A.E. (1999) Protein sequence threading, the alignment problem and a two step strategy. *J. Comput. Chem.*, **20**, 1455–1467.
18. Koretke,K.K., Luthey-Schulten,Z. and Wolynes,P.G. (1996) Self-consistently optimized statistical mechanical energy functions for sequence structure alignment. *Protein Sci.*, **5**, 1043–1059.
19. Wilmanns,M. and Eisenberg,D. (1995) Inverse protein folding by the residue pair preference profile method: estimating the correctness of alignments of structurally compatible sequences. *Protein Eng.*, **8**, 627–639.
20. 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.
21. Godzik,A., Kolinski,A. and Skolnick,J. (1992) Topology fingerprint approach to the inverse protein folding problem. *J. Mol. Biol.*, **227**, 227–238.
22. Simons,K.T., Kooperberg,C., Huang,E. and Baker,D. (1997) Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. *J. Mol. Biol.*, **268**, 209–225.
23. Cheeseman,P. and Stutz,J. (1995) Bayesian classification (AutoClass): Theory and results. In Fayyad,U., Piatetsky-Shapiro,G., Smyth,P. and Uthurusamy,R. (eds), *Advances in Knowledge Discovery and Data Mining*. The AAAI Press, Menlo Park, pp. 61–83.
24. Dempster,A. (1977) A maximum likelihood from incomplete data via the EM algorithm. *Royal J. Stat. Soc., B*, **39**, 1–38.
25. Dayhoff,M., Schwartz,R. and Orcutt,B. (1978) A model of evolutionary change is proteins, matrices for detecting distant relationships. In Dayhoff,M. (ed.), *Atlas of Protein Sequence and Structure*. National Biomedical Research Foundation, Washington DC, Vol. 5, pp. 345–358.
26. Qian,B. and Goldstein,R.A. (2002) Optimization of a new score function for the generation of accurate alignments. *Proteins*, **48**, 605–610.
27. Russell,A. and Torda,A.E. (2002) Protein sequence threading - averaging over structures. *Proteins*, **47**, 496–505.
28. Goldstein,R.A., Luthey-Schulten,Z.A. and Wolynes,P.G. (1992) Protein tertiary structure recognition using optimized Hamiltonians with local interactions. *Proc. Natl Acad. Sci. USA*, **89**, 9029–9033.
29. Rychlewski,L., Fischer,D. and Elofsson,A. (2003) LiveBench-6: large-scale automated evaluation of protein structure prediction servers. *Proteins*, **53**, 542–547.
30. Koh,I.Y.Y., Eyrich,V.A., Marti-Renom,M.A., Przybylski,D., Madhusudhan,M.S., Eswar,N., Grana,O., Pazos,F., Valencia,A., Sali,A. *et al.* (2003) EVA: evaluation of protein structure prediction servers. *Nucleic Acids Res.*, **31**, 3311–3315.
31. Henikoff,S. and Henikoff,J.G. (1992) Amino acid substitution matrices from protein blocks. *Proc. Natl Acad. Sci. USA*, **89**, 10915–10919.
32. Elofsson,A. (2002) A study on protein sequence alignment quality. *Proteins*, **46**, 330–339.
33. Capriotti,E., Fariselli,P., Rossi,I. and Casadio,R. (2004) A Shannon entropy-based filter detects high-quality profile-profile alignments in searches for remote homologues. *Proteins*, **54**, 351–360.
34. Jaroszewski,L., Rychlewski,L. and Godzik,A. (2000) Improving the quality of twilight-zone alignments. *Protein Sci.*, **9**, 1487–1496.
35. Yona,G. and Levitt,M. (2002) Within the twilight zone: a sensitive profile-profile comparison tool based on information theory. *J. Mol. Biol.*, **315**, 1257–1275.
36. Jones,D.T. (1999) GenTHREADER: An efficient and reliable protein fold recognition method for genomic sequences. *J. Mol. Biol.*, **287**, 797–815.
37. McGuffin,L.J. and Jones,D.T. (2003) Improvement of the GenTHREADER method for genomic fold recognition. *Bioinformatics*, **19**, 874–881.