Your article (01002003) from "Proteins: Structure, Function, and Genetics" is available for download
=====
RE: Your article (01002003) from "Proteins: Structure, Function, and Genetics" is available for download

Proteins: Structure, Function, and Genetics Published by John Wiley & Sons, Inc.

Dear Sir or Madam,

PDF page proofs for your article are ready for review.

Please refer to this URL address

http://rapidproof.cadmus.com/RapidProof/retrieval/index.jsp

Login: your e-mail address
Password: ----

The site contains 1 file. You will need to have Adobe Acrobat Reader software to read these files. This is free software and is available for user downloading at
http://www.adobe.com/products/acrobat/readstep.html.

This file contains:

Author Instructions Checklist
Adobe Acrobat Users - NOTES tool sheet
Reprint Order form
Copyright Transfer Agreement
Return fax form
A copy of your page proofs for your article

After printing the PDF file, please read the page proofs carefully and:

1) indicate changes or corrections in the margin of the page proofs;
2) answer all queries (footnotes A,B,C, etc.) on the last page of the PDF proof;
3) proofread any tables and equations carefully;
4) check that any Greek, especially "mu", has translated correctly.

Within 48 hours, please return the following to the address given below:

1) original PDF set of page proofs,
2) Reprint Order form,
3) Return fax form

Return to:

Marc Nadeau
Journals Editorial/Production
John Wiley & Sons, Inc.
111 River Street
Hoboken, NJ 07030

U.S.A.

Your article will be published online via our EarlyView service within a few days of correction receipt. Your prompt attention to and return of page proofs is crucial to faster publication of your work. If you experience technical problems, please contact Doug Frank (e-mail: frankd@cadmus.com, phone: 800-238-3814 (X615).

If you have any questions regarding your article, please contact me. PLEASE ALWAYS INCLUDE YOUR ARTICLE NO. (01002003) WITH ALL CORRESPONDENCE.

Sincerely,

Marc Nadeau
Associate Production Editor
John Wiley & Sons, Inc.
E-mail: mnadeau@wiley.com
Tel: 201.748.6716
Fax: 201.748.6052

WILEY
Publishers Since 1807

111 RIVER STREET, HOBOKEN, NJ 07030

*****IMMEDIATE RESPONSE REQUIRED*****

Your article will be published online via Wiley's EarlyView® service (www.interscience.wiley.com) shortly after receipt of corrections. EarlyView® is Wiley's online publication of individual articles in full text HTML and/or pdf format before release of the compiled print issue of the journal. Articles posted online in EarlyView® are peer-reviewed, copyedited, author corrected, and fully citable via the article DOI (for further information, visit www.doi.org). EarlyView® means you benefit from the best of two worlds--fast online availability as well as traditional, issue-based archiving.

Please follow these instructions to avoid delay of publication.

☐ **READ PROOFS CAREFULLY**

- This will be your only chance to review these proofs**. Please note that once your corrected article is posted online, it is considered legally published, and cannot be removed from the Web site for further corrections.**
- Please note that the volume and page numbers shown on the proofs are for position only.

☐ **ANSWER ALL QUERIES ON PROOFS** (Queries for you to answer are attached as the last page of your proof.)

- Mark all corrections directly on the proofs. Note that excessive author alterations may ultimately result in delay of publication and extra costs may be charged to you.

☐ **CHECK FIGURES AND TABLES CAREFULLY** (Color figure proofs will be sent under separate cover.)

- Check size, numbering, and orientation of figures.
- All images in the PDF are downsampled (reduced to lower resolution and file size) to facilitate Internet delivery. ----- These images will appear at higher resolution and sharpness in the printed article.
- Review figure legends to ensure that they are complete.
- Check all tables. Review layout, title, and footnotes.

☐ **COMPLETE REPRINT ORDER FORM**

- Fill out the attached reprint order form. It is important to return the form even if you are not ordering reprints. You may, if you wish, pay for the reprints with a credit card. Reprints will be mailed only after your article appears in print. This is the most opportune time to order reprints. If you wait until after your article comes off press, the reprints will be considerably more expensive.

RETURN ☐**PROOFS**
☐**REPRINT ORDER FORM**
☐**CTA (If you have not already signed one)**

RETURN IMMEDIATELY AS YOUR ARTICLE WILL BE POSTED ONLINE SHORTLY AFTER RECEIPT; FAX PROOFS TO 201-748-6052

QUESTIONS?

Marc Nadeau, Associate Production Editor
John Wiley & Sons, Inc.
111 River Street
Hoboken, NJ 07030
Phone: 201-748-6716
E-mail: mnadeau@wiley.com
Please refer to journal acronym and article production number

Softproofing for advanced Adobe Acrobat Users - NOTES tool

NOTE: ACROBAT READER FROM THE INTERNET DOES NOT CONTAIN THE NOTES TOOL USED IN THIS PROCEDURE.

Acrobat annotation tools can be very useful for indicating changes to the PDF proof of your article. By using Acrobat annotation tools, a full digital pathway can be maintained for your page proofs.

The NOTES annotation tool can be used with either Adobe Acrobat 3.0x or Adobe Acrobat 4.0. Other annotation tools are also available in Acrobat 4.0, but this instruction sheet will concentrate on how to use the NOTES tool. Acrobat Reader, the free Internet download software from Adobe, DOES NOT contain the NOTES tool. In order to softproof using the NOTES tool you must have the full software suite Adobe Acrobat Exchange 3.0x or Adobe Acrobat 4.0 installed on your computer.

Steps for Softproofing using Adobe Acrobat NOTES tool:

1. Open the PDF page proof of your article using either Adobe Acrobat Exchange 3.0x or Adobe Acrobat 4.0. Proof your article on-screen or print a copy for markup of changes.

2. Go to File/Preferences/Annotations (in Acrobat 4.0) or File/Preferences/Notes (in Acrobat 3.0) and enter your name into the "default user" or "author" field. Also, set the font size at 9 or 10 point.

3. When you have decided on the corrections to your article, select the NOTES tool from the Acrobat toolbox and click in the margin next to the text to be changed.

4. Enter your corrections into the NOTES text box window. Be sure to clearly indicate where the correction is to be placed and what text it will effect. If necessary to avoid confusion, you can use your TEXT SELECTION tool to copy the text to be corrected and paste it into the NOTES text box window. At this point, you can type the corrections directly into the NOTES text box window. **DO NOT correct the text by typing directly on the PDF page.**

5. Go through your entire article using the NOTES tool as described in Step 4.

6. When you have completed the corrections to your article, go to File/Export/Annotations (in Acrobat 4.0) or File/Export/Notes (in Acrobat 3.0). Save your NOTES file to a place on your harddrive where you can easily locate it. **Name your NOTES file with the article number assigned to your article in the original softproofing e-mail message.**

7. **When closing your article PDF be sure NOT to save changes to original file.**

8. To make changes to a NOTES file you have exported, simply re-open the original PDF proof file, go to File/Import/Notes and import the NOTES file you saved. Make changes and re-export NOTES file keeping the same file name.

9. When complete, attach your NOTES file to a reply e-mail message. Be sure to include your name, the date, and the title of the journal your article will be printed in.

John Wiley & Sons, Inc.

Publishers Since 1807

REPRINT BILLING DEPARTMENT • 111 RIVER STREET • HOBOKEN, NJ 07030
PHONE: (201) 748-8789; FAX: (201) 748-6326
E-MAIL: reprints @ wiley.com

PREPUBLICATION REPRINT ORDER FORM

Please complete this form even if you are not ordering reprints. This form **MUST** be returned with your corrected proofs and original manuscript. Your reprints will be shipped approximately 4 weeks after publication. Reprints ordered after printing are substantially more expensive.

JOURNAL: ***PROTEINS: Structure, Function, and Genetics*** VOLUME_____ ISSUE_____

TITLE OF MANUSCRIPT_____

MS. NO._____ NO. OF PAGES_____ AUTHOR(S)_____

REPRINTS 8 1/4 X 11

No. of Pages	100 Reprints	200 Reprints	300 Reprints	400 Reprints	500 Reprints
	$	$	$	$	$
1-4	336	501	694	890	1,052
5-8	469	703	987	1,251	1,477
9-12	594	923	1,234	1,565	1,850
13-16	714	1,156	1,527	1,901	2,273
17-20	794	1,340	1,775	2,212	2,648
21-24	911	1,529	2,031	2,536	3,037
25-28	1,004	1,707	2,267	2,828	3,388
29-32	1,108	1,894	2,515	3,135	3,755
33-36	1,219	2,092	2,773	3,456	4,143
37-40	1,329	2,290	3,033	3,776	4,528

****** REPRINTS ARE ONLY AVAILABLE IN LOTS OF 100. IF YOU WISH TO ORDER MORE THAN 500 REPRINTS, PLEASE CONTACT OUR REPRINTS DEPARTMENT AT (201)748-8789 FOR A PRICE QUOTE.

❑ Please send me _____ reprints of the above article at..... $_____

Please add appropriate State and Local Tax {Tax Exempt No._____} $_____
Please add 5% Postage and Handling..... $_____
TOTAL AMOUNT OF ORDER** ..... $_____

***International orders must be paid in U.S. currency and drawn on a U.S. bank*

Please check one: ❑ Check enclosed ❑ Bill me ❑ Credit Card
If credit card order, charge to: ❑ American Express ❑ Visa ❑ MasterCard ❑ Discover
Credit Card No._____ Signature_____ Exp. Date_____

Bill To:

Name_____
Address/Institution_____
_____
_____
_____
Purchase Order No. _____ Phone _____ Fax _____
E-mail: _____

Ship To:

Name_____
Address/Institution_____
_____
_____
_____

PROTEINS: Structure, Function, and Genetics 53:000–000 (2003)

Combining Local Structure, Fold Recognition, and New Fold Methods for Protein Structure Prediction

AQ: 1

Kevin Karplus,* Rachel Karchin, Jenny Draper, Jonathan Casper, Yael Mandel-Gutfreund, Mark Diekhans, and Richard Hughey

AQ: 2 *Computer Engineering Department, University of California, Santa Cruz, California*

ABSTRACT **This article presents an overview of the SAM-T02 method for protein fold recognition and the UNDERTAKER program for ab initio predictions. The SAM-T02 server is an automatic method that uses two-track hidden Markov models (HMMS) to find and align template proteins from PDB to the target protein. The two-track HMMS use an amino acid alphabet and one of several different local structure alphabets. The UNDERTAKER program is a new fragment-packing program that can use short or long fragments and alignments to create protein conformations. The HMMS and fold recognition alignments from the SAM-T02 method were used to generate the fragment and alignment libraries used by UNDERTAKER. We present results on a few selected targets for which this combined method worked particularly well: T0129, T0181, T0135, T0130, and T0139. Proteins 2003;53:000–000.**

Key words: SAM-T02; UNDERTAKER programs amino acid alphabet; fragment-packing program

INTRODUCTION

In previous CASP experiments, our team has concentrated on fold-recognition using hidden Markov models (HMMS) with fairly good results.[1–3] We have also had some success using standard neural net methods to predict secondary structure,[4] as measured by the EVA project.[5] In 2000, we started incorporating secondary structure prediction in our fold recognition method for CASP4.[3]

We entered two automatic servers in CASP5 and CAFASP3: the old SAM-T99 server and a new server, SAM-T02. SAM-T02 incorporated the most important of the fold recognition improvements we had made in CASP4: multitrack HMMs, in which each match node contains emission probabilities of predicted local structure information, in addition to amino acid emission probabilities.[3,6] The two-track HMM is illustrated in Figure 1. F1

Because both servers used the same protein sequence database and the same templates from PDB,[8] any improvement in performance could be attributed to im provements in the method rather than in the underlying databases. The multitrack HMMs we used in CASP4 relied on a helix-strand-coil description of secondary structure, whereas those used in CASP5 use a variety of local structure descriptions.

For our human-assisted entry to CASP5, we added a new fragment-packing program, UNDERTAKER, to our tool set. The program was added to make predictions in the new fold and difficult fold recognition areas, where we previously had had no success. The same method was used for all targets, independent of the degree of similarity to any targets that we found.

According to the CASP5 assessors, our group had good results in the new fold category and the analogous fold recognition category, so the new fragment-packing program, UNDERTAKER, is the main focus of this article.

MATERIALS AND METHODS

Our overall structure prediction method can be conveniently divided into several parts:

- finding similar sequences with iterative search (using SAM-T2K)
- predicting local structure properties with neural nets
- finding possible fold recognition templates using two-track HMMS (the SAM-T02 method)
- making alignments to the templates;
- building a specific fragment library for the target (with FRAGFINDER)
- packing fragments and fold recognition alignments to make a 3D structure (with UNDERTAKER).

The overall structure of the process is outlined in Figure 2. F2

The iterative search method was exactly the same as the SAM-T2K method used in CASP4—the only change was in the size of the NR database searched.[9]

We predicted local structure with the same neural net software as in CASP4, but with newly trained nets and different local structure alphabets. For CASP4, we used the standard EHL alphabet that is assessed in CASP. For CASP5, we used four local structure alphabets: EBGHSTL based on DSSP labeling,[10] EBGHTL based on STRIDE labeling,[11] an extension to DSSP (STR) that divided the

Grant sponsor: National Science Foundation; Grant numbers: DBT-9808007 and EIA-9905322; Department of Energy; Grant number: DE-FG0395-99ER62849; Grant sponsor: National Physical Sciences Consortium graduate fellowship.

*Correspondence to: Kevin Karplus, Computer Engineering Department, University of California, Santa Cruz, CA 95064. E-mail: karplus@soe.ucsc.edu

Received 3 March 2003; Accepted 3 April 2003

Published online 00 Month 2003 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/prot.10540

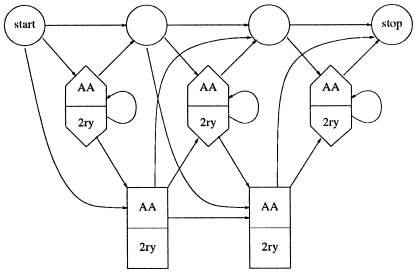

Fig. 1. A multitrack HMM has multiple emission tables in each letter-generating (match or insert) state but is otherwise similar to the standard profile HMMs used in the SAM package.[7] The multitrack HMMs model the amino acids and local structure as conditionally independent, conditioned on the state of the model.

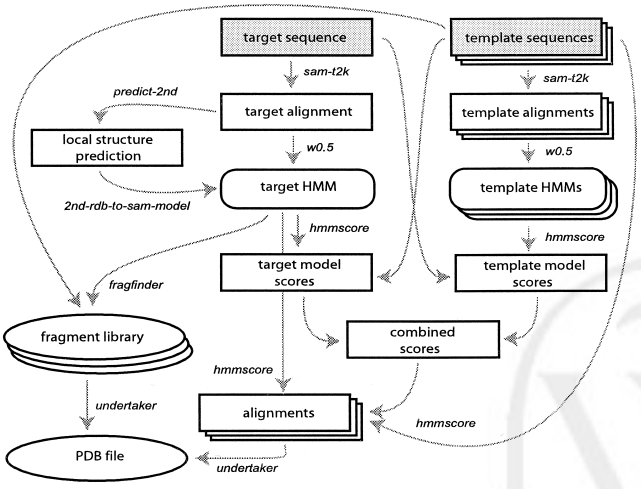

Fig. 2. The SAM-T02 prediction process consists of several parts: building a multiple alignment for the target sequence using the SAM script target2k, predicting secondary structure with predict-2nd, building HMMs for the target sequence with the SAM script w0.5, scoring the template library against the target HMMs using the SAM program hmmscore, scoring the target sequence against the template HMMs with hmmscore, combining scores to select templates, choosing target-template alignments, generating fragments with the SAM program fragfinder, and doing fragment packing with UNDERTAKER.

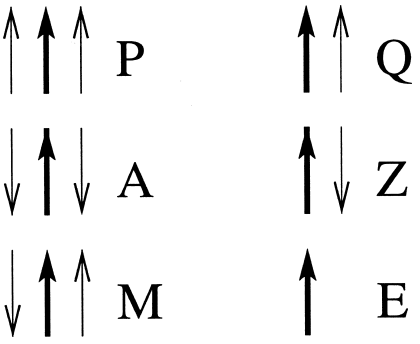

Fig. 3. Six letters in the STR alphabet, which expand on the DSSP "E" or strand state. The strand of the residue being assigned is indicated with a bold arrow. In a β-sheet, this strand is either surrounded by two parallel partners "P," two anti-parallel partners "A," or one anti-parallel and one parallel partner "M." Edge strands (that have only one β-strand partner) have either a parallel partner "Q" or an anti-parallel partner "Z." Finally, we retain the "E" label for strand residues to which DSSP assigns no partners (generally beta bulges).

F3 β-strands into six classes (see Fig. 3),[6] and an 11-letter alphabet (alpha11) based on the torsion angle formed by four successive C_α atoms.[6] We also provided a reduction to the three-letter EHL alphabet for CASP assessment but did not use this reduced alphabet for any other purpose.

As in CASP4, we used multitrack HMMs for target models and amino acid-only HMMs for template models. Because we now had five different target models (amino acid-only and four different two-track HMMs corresponding to the four local structure alphabets), we did a weighted combination of scores for the template-based and target-based searches. Weights were chosen arbitrarily, based on our assumptions about how well the methods would work. The combined weights were used to select templates.

The SAM-T02 automatic server did not include the alpha11 torsion-angle predictions and HMMs but was otherwise the same as the template selection step in our hand predictions. The automatic server made no attempt to produce 3D structures but returned alignments to the templates based on a single two-track target HMM. The alignment HMM used our new STR alphabet as the local structure track, because that method had performed best in our alignment tests.[6] Because our template library is highly redundant, the server attempted to remove duplication and report five distinct predictions. In some cases, we managed to get decent predictions for multiple domains as different models (e.g., for T0184 with T0184_1 as models 1 and 2 and T0184_2 as models 3 and 4), even though the server did not explicitly consider domains.

Our SAM-T02-human method did not use a single alignment to the template. Instead, we generated about 25 alignments for each template, using the different target- and template-based HMMs and different alignment options (global vs local, Viterbi vs posterior decoding). In addition to fold recognition alignments, we also used FRAGFINDER, a new tool in our SAM tool suite, to find six fragments of length 9 for each position in the sequence. The fragments were found with a two-track HMM that uses the STR local structure alphabet, taking the best six gapless matches in the template library for each position.

Although we have not yet optimized the FRAGFINDER method nor done extensive testing, we expect that the use of two-track HMMs for finding fragments will be a big help when the local structure prediction is accurate and will be comparable to other fragment-finding techniques when the local structure prediction is weak. The HMMs will cause serious problems when the local structure prediction is confident, but wrong.

The alignments and specific fragment library were given to UNDERTAKER, along with a generic fragment library containing all one-, two-, three-, and four-residue fragments indexed by their amino acid strings from a training set of 448 monomeric protein chains. UNDERTAKER used the alignments to get an initial conformation and then applied many rounds of a genetic algorithm with randomly applied conformation-change operations to minimize a cost function.

The following subsections describe some of the internals of UNDERTAKER.

tapraid5/7e-protein/7e-protein/7e1503/7e1455-03g | sweigarm | S=7 | 8/25/03 | 7:59 | Art: 01002003

Conformation Representation

Selection of a conformation representation and data structure is critical to effective fragment packing, because it affects the computation time, the possible conformation change operators, and the possible cost functions. UNDERTAKER represents protein conformations as the 3D coordinates for all heavy atoms (not hydrogens). Using a full 3D representation for all heavy atoms, rather than a more compact one such as ϕ-ψ angles or side-chain centroids, slows down conformation generation slightly but allows much more flexibility in defining cost functions. One decision we plan to revisit is whether to include explicit hydrogens—having explicit hydrogens would make hydrogen bond scoring simpler but would increase the size of the conformational space, because torsion angles for the NH_3 and OH groups would then need to be set. We could optimize the torsion angles after determining that an H-bond was desired, but this does not seem to offer much advantage over the current implicit hydrogens.

UNDERTAKER does not require the backbone to be contiguous but does allow breaks between residues. This allows us to represent directly the multiple-segment information we get from fold recognition alignments, bringing fold recognition and new fold techniques into a unified framework. For homology modeling with UNDERTAKER, unlike Rosetta,[12] we do not pick a single alignment and freeze the backbone for the core residues, but we allow many alignments to be sampled and parts of different ones to be combined. For distant target-template relationships, mixing several alignments can help find the correct parts, but for closer relationships, choosing a single alignment that is most likely to be good avoids adding noise to the search.

Our poor performance on comparative modeling targets (relative to our automatic server) is probably due in large part to not freezing the core.

Allowing broken backbones introduces a problem that is not present in programs (like Rosetta) that use a frozen core or contiguous backbone: What is the relationship between unconnected parts of the backbone? What moves when a backbone fragment is replaced?

To solve this problem in UNDERTAKER, we represent the protein as a tree with segments as leaves, where each segment is a contiguous piece of the protein with properly formed peptide bonds. When we do fragment replacement within a segment, the transformation does not propagate across the gap between segments. To preserve 3D relationships between segments, we add edges between segments, called *tertiary edges*, which indicate which pairs of atoms are thought of as holding the segments together. These are usually chosen to be the closest pair of atoms in the two segments, such as a disulfide bond or Van der Waals contact.

Any two residues in the protein chain are connected by a unique path through the tree. Removing a peptide bond or a tertiary edge breaks the tree into two trees each of which can be rigidly transformed, maintaining the structure within that subpart of the protein, without requiring a contiguous backbone or a frozen core. The rigid transformations may result from any of several conformation-changing operations, described in the next section.

Note that a subtree may consist of segments that are widely separated along the protein chain, as would be necessary for holding together a domain while another inserted domain changes shape.

Conformation-Changing Operators

We have implemented several conformation-changing operations in UNDERTAKER, beginning with the fragment replacement operation introduced by Simons and Baker.[13] Fragments to use for replacement came from three sources: very short ones (1–4 residues) from a generic fragment library, which must match exactly on all residues, medium-length ones (9–12 residues) found by FRAGFINDER, and variable-length ones that come from fold recognition alignments. We also used an operator for replacing two fragments simultaneously to allow for hinge-like motions of part of the conformation, although there is currently no constraint that the conformation change actually be hingelike.

In addition to fragment replacement, UNDERTAKER has alignment replacement, which replaces several segments at once, keeping them in the same physical relationships as they have in the template they are copied from. This operator allows us to import complete fold recognition results into our fragment-packing optimization.

UNDERTAKER includes a number of operators that attempt to improve some part of our cost function—reducing breaks, forming or improving disulfide bonds reducing clashes, reducing the cost of user-specified constraints, and so forth. Many of these operators work by trying a small number of potential fragment replacements and computing for each the effect that it would have on only part of the cost function, selecting the fragment replacement that appears to make the most improvement.

UNDERTAKER also has operators for repositioning subtrees. It can either jiggle them a small amount or try to find the optimal placement for them, given the constraints and peptide bonds on the segments in the tree. There are various ways of splitting the tree into subtrees, which move larger or smaller sections of the protein. On a smaller scale, the method also has operators for changing the rotamers of the residues without changing the backbone, to improve packing or reduce clashes.

Because UNDERTAKER uses a genetic algorithm for the stochastic search, the method also includes crossover operators that combine parts of two conformations to get a new one.

Stochastic Search

As mentioned above, UNDERTAKER uses a genetic algorithm to search conformational space. We start from a set of conformations (random conformations based on fold recognition alignments or from previous runs of UNDERTAKER) and randomly apply operators to generate new conformations. New conformations that score well are added to the pool for the next generation, and poorly scoring older conformations are eliminated. To make sure

that the pool mixes rapidly, we keep no more than 40% of the conformations from the previous generation.

We keep track of the success rate for each operator (how often it results in a conformation being kept in the pool) and adjust the probability of applying the operators based on their success. The adaptation scheme we are currently using is rather crude and sometimes gets stuck applying only one or two of the operators, if it has initial success with them.

We use the results of several runs of the genetic algorithm to seed the pool for another run, often getting noticeable reduction in cost from applying crossover operators to conformations from different runs.

Cost Function

Substantial effort was put into making the cost function in UNDERTAKER easy to modify and extend, because it was quite clear that much future work would be put into different scoring functions. Much less effort has been put into making a good first version of the cost function. For example, our cost function does not yet include a hydrogen-bonding term, but such a term is essential for forming β-sheets. For close fold recognition targets and for α-helical proteins, the compactness of the right structure usually held it together in the subsequent optimization of the cost, but for more distant folds and new folds, β-sheets often came apart during optimization, even if they were present in the initial conformation. We often had to add desired hydrogen bonds as manual constraints in the cost function.

The cost function can be defined at run time as a linear combination of any subset of a large number of different basic cost functions, and the basic cost functions themselves can be parameterized at run time. New basic cost functions are very easily added to the code, and they add no computational cost unless they are specifically requested in the linear combination specified at runtime. Currently, we have >24 basic cost functions, and there are several more that we believe we should implement and test.

One of UNDERTAKER's most important cost functions, indeed the one that gives the method its name, is the burial function. This is a parameterized function that counts the number of atoms within a given radius for each residue and scores the sphere based on the probability of seeing that number of atoms. The sphere is referred to as a spot, and the number of atoms whose centers are within the sphere as the burial of the spot. The parameter files for a burial function include a specification of where the center of the sphere is relative to the residue, the size of the sphere, and the smoothed probability distribution of burial for each residue type.

The UNDERTAKER program includes functionality for optimizing the spot locations. We define dry spots as those for which burial has been maximized, wet spots as those for which burial has been minimized, and generic spots whose location does not depend on the type of residue. For generic spots, we maximize the mutual information between the burial and the residue identity.

UNDERTAKER also has basic cost functions that can accept the predicted probabilities over a local structure alphabet for a target and score the conformation using them (currently working only for the ALPHA11 torsion-angle alphabet).

One important basic score function accepts user-specified distance constraints on pairs of atoms and tries to satisfy these constraints while generating conformations. These constraints can come either from educated guesses by the user of the program or from experimental data (such as NMR experiments or cross-linking experiments). The use of constraints turned out to be essential for our CASP5 predictions.

RESULTS AND DISCUSSION

Because our human-assisted prediction method began with essentially the same fold recognition process that was used by our SAM-T02 automatic server, it is instructive to look at the differences in performance between the two. For the comparative modeling targets, the server did better (according to the GDT score) on 76% of the targets—our use of UNDERTAKER without freezing the core resulted in an overall loss of model quality for the closer homologs. For the easier fold recognition targets (classes CM/FR and FR(H)), the server did better on about a third of the targets, and the additional input, either by the UNDERTAKER program or by hand, made improvements on the remaining two thirds. For the difficult targets [FR(A), NF/FR, and NF], the hand-assisted UNDERTAKER program did better than the automatic server on about 84% of the targets.

We looked at the results of the automatic servers registered with CAFASP and often included the models generated by Robetta (the automatic server produced by the Baker group using Rosetta [14] as possible conformations in the initial pool for our genetic algorithm.

On most of the new fold targets, we did not come up with anything resembling a correct structure. This is not surprising, given the crude nature of our cost function and the amount of handwork necessary to get vaguely protein-like conformations. We discuss only the rarer successes in this article.

We did reasonably well for the new fold targets T0129 and T0181. Also an FR(A) target did well (T0135), and a CM/FR target was popular with most of the speakers at CASP5 (T0130). One target that was withdrawn from CASP5, T0139, deserves some comment. Each of these targets is discussed below.

New Fold: T0129

Target T0129 was the first target to be released, so we had plenty of time to look at it and to adjust the scoring function of UNDERTAKER to produce more protein-like conformations. Our secondary structure predictor gave strong predictions for seven helices and for an extended piece. It turned out that our secondary-structure prediction was reasonably accurate (Q3 = 82%), which helped in assembling the protein.

The first three helices of the N-terminal domain were usually packed by the fragment assembly quite consis-

tently, but we had difficulty with the C-terminal domain. We mentally partitioned the target into two domains, but we mistakenly grouped helix 4 in the second domain instead of the first, which resulted in mispacking both domains. The program was not informed of our domain division, but we selected 10 hand-created constraints: 4 to try to keep helices 4–7 straight and 6 to try to form an up-down bundle of the four helices plus the extended piece from F80 to G85. We would have done better not to constrain the N-terminal end of helix 4.

Our best model was model 3, whether the domains were considered separately or together. This model was one where we liked the (incorrect) packing of helix 4 with helices 6 and 7, but we did not like the way that helix 5 was messed up. Although we had included conformations provided by the Robetta server in some of our optimizations, they had not been included in the optimizations leading to models 2 and 3.

New Fold: T0181

Our model 2 for target T0181 had the N-terminus basically right, but we had trouble getting the third strand of the sheet (which we had correctly predicted as a strand) to join the sheet, probably because of the large number of residues between the second and third strands. We tried adding constraints by hand to position the third strand, but we could not simultaneously form the sheet and keep the backbone contiguous. Because of the bad break in the backbone, we never submitted any of our models that had the complete sheet—these might very well have been better than what we did submit.

We had some weak fold recognition results for T0181, but because we still have not seen the correct structure, it is difficult to decide what went right and wrong.

Fold Recognition (Analogous): T0135

We submitted only one model for T0135, which we obtained through a combination of fold recognition and new fold techniques. Our fold recognition method by itself had the correct fold in third place in its list of hits, but the E-value of 9.6 gave us no confidence in the result, and there were several other folds that scored essentially as well. We had no way of choosing the correct fold using just our fold recognition methods.

When UNDERTAKER was run with no hand-added constraints, the sheet was not assembled. To assemble it, we tried to find topologies that were consistent with our predictions that strand 1 would be an antiparallel or mixed middle strand, strand 2 would be an antiparallel edge strand, strand 3 would be a parallel strand, and strand 4 would be a mixed middle strand (using a neural net with our extended STR alphabet). We also wanted strands 1 and 2 to be oriented the same way, because we had predicted a single helix between them. We did not find any topologies that met all our predictions, so we experimented with adding constraints for various topologies. The most promising one was a 4132 antiparallel sheet. We obtained a model that looked roughly like a protein to us, so we submitted it to the VAST web server[15] to see if any existing proteins had a similar structure. We got excellent alignments to proteins with a ferredoxin-like fold, probably because our library of fragments and fold recognition alignments contained templates with this fold.

We edited VAST's structural alignments to add more fold recognition alignments for this fold to UNDERTAKER's collection. Several runs of UNDERTAKER, both with and without constraints on the sheets, resulted in models with different flaws. We superimposed the models and did cut-and-paste editing to put together a model with better features, which we then reoptimized. We fiddled with hand-added constraints and cut-and-paste editing, to try to close gaps and pack the helices against the sheet. The final run did not use the packing constraints but did include constraints corresponding to the hydrogen bonds of the predicted sheet, because our score function still does not include a hydrogen-bonding term.

For target T0135, the new fold methods allowed us to recognize and align a fold that was just a little too remote for our fold recognition methods alone to manage. Our success on this target is exactly what we were hoping for by combining methods, but several other targets in the FR(A) category were not nearly as successful.

Comparative Modeling/Fold Recognition: T0130

Target T0130 is one that almost all the presenters at CASP5 felt obligated to present—indeed, one could almost have selected the speakers for CASP5 just based on their performance on this target.

Recognizing the nucleotidyltransferase fold was easy (almost all the fold recognition servers got it), but getting a good alignment was harder. Most of the servers (including both of ours) did not have the third aspartic acid of the catalytic triad (D46, D48, D79)—there was excellent sequence conservation up to residue 1, but a hairpin had to be deleted from the templates to get the third strand reasonably aligned.

We added constraints by hand to keep this triad properly spaced (based on the triads in 1bpyA, 1fa0A, and 1fa0B). These constraints managed to get most of the fold for us, but we incorrectly predicted a helix for the final strand. UNDERTAKER consistently unwound the helix, but we did not think to question the rather weak predictions of the neural net on this segment and try to attach it as a strand. Instead, we kept adding constraints to try to force the incorrectly predicted helix to form and to pack against the sheet.

Withdrawn: T0139

Target T0139 had a picture of its structure published just a week before the CASP5 deadline.[6] We found the picture about 24 h before the CASP5 deadline. We tried estimating constraints from the picture and adding these constraints to the UNDERTAKER score function. There were a number of problems creating these constraints (unlabeled atoms, mislabeled residues, and distances that were difficult to guess). We ended up adding about 40 rather loose distance constraints. Just adding these noisy constraints was not enough to get a good solution—helix 4

ended up on the wrong side of the cluster of helices 1, 2, and 3. We ended up moving the helix by hand to the other side and reoptimizing, because our move set seemed unwilling to unfold the conformation enough to change which side the helix was on, and we did not have enough time to start over from a random configuration. This reoptimization resulted in a roughly correct structure, so we did some further optimization without the constraints from the article. This reoptimization did not make many changes (our model 1 submission included the distance constraints, and our model 2 submission did not).

In short, we got a good model for target T0139 (4.86 ϕÅ for all CA atoms) by adding about 40 correct but noisy distance constraints and knowledge of the chirality of the helix bundle. This was, of course, cheating, so we informed the organizers that target T0139 should be removed from the CASP5 evaluation. Much more information could have been extracted from the article—Alexey Murzin managed to get a 3.84 Å CA-RMSD model using the same article. We were encouraged to see how little extra information was needed to go from a rather bad model to quite a good one, because one of our hopes is that the UNDERTAKER program will be useful for aiding structure determination from data sets that would normally be insufficient or of too low quality for the purpose.

CONCLUSION

The CASP5 experiment this year let us test both our new use of local structure alphabets in fold recognition (comparing the SAM-T02 server to the older SAM-T99 server) and our new fragment-packing method.

Almost universally, the SAM-T02 server made better predictions than the older SAM-T99 server, showing that the use of predicted local structure is valuable in fold recognition.

Hand-assisted fragment packing did substantially better than the fold recognition server on the more difficult targets, but worse on the easiest (comparative modeling) targets. This loss of performance is almost certainly due to having a large number of alignments to various templates, with no information given to UNDERTAKER about the scores of the alignments. UNDERTAKER's crude cost function was not able to pick out the best template and alignment reliably from the set it was presented with, and the fragment packing often resulted in some movement of the core residues.

Our future work will concentrate on improving the cost function in UNDERTAKER, adding new conformation-change operators, and providing a way to preserve good conformations from fold recognition without having to freeze the core.

ACKNOWLEDGMENTS

We thank David Haussler and Anders Krogh for starting the hidden Markov model and Dirichlet mixture work at UCSC, because these approaches were instrumental to our success. We also thank Christian Barrett and Spencer Tu Basu, who imple mented earlier versions of our prediction server, and who made other contributions to the techniques. We began work on T0129 and T0130 while Kevin Karplus was on sabbatical in David Baker's laboratory and conversations with members of that laboratory were fruitful in guiding our initial work on these targets.

REFERENCES

1. Karplus K, Sjölander K, Barrett C, Cline M, Haussler D, Hughey R, Holm L, Sander C. Predicting protein structure using hidden Markov models. Proteins 1997;Suppl 1:134–139.
2. Karplus K, Barrett C, Cline M, Diekhans M, Grate L, Hughey R. Predicting protein structure using only sequence information. Proteins, 1999; Suppl 3:121–125.
3. Karplus K, Karchin R, Barrett C, Tu S, Cline M, Diekhans M, Grate L, Casper J, Hughey R. What is the value added by human intervention in protein structure prediction? Proteins 2001;45:86–91.
4. Karplus K, Barrett C, Hughey R, Hidden Markov models for detecting remote protein homologies. Bioinformatics 1998;14:846–856.
5. Eyrich VA, Marti-Renom MA, Przybylski D, Madhusudhan MS, Fiser A, Pazos F, Valencia A, Sali A Rost B. EVA: continuous automatic evaluation of protein structure prediction servers. Bioinformatics December 2001;17:1242–1243.
6. Karchin R, Cline M, Mandel-Gutfreund Y, Karplus K. Hidden Markov models that use predicted local structure for fold recognition: alphabets of backbone geometry. Proteins June 2003;51:504–514.
7. Hughey R, Karplus K, Krogh A. SAM: sequence alignment and modeling software system, version 3. Technical Report UCSC-CRL-99-11, University of California, Santa Cruz, Computer Engineering, UC Santa Cruz, CA 95064, October 1999. Available from http://www.soe.ucsc.edu/research/ compbio/sam.html.
8. Bernstein FC, Koetzle TF, Williams GJ, Meyer EE, Brice MD, Rodgers JR, Rennard O, Shimanouchi T, Tasumi M. The Protein Data Bank: a computer-based archival file for macromolecular structures. J Mol Biol 1977;112:535–542.
9. NR (All non-redundant GenBank CDS translations+ PDB+SwissProt+PIR+PRF Database) Distributed on the Internet via anonymous FTP from ftp://ftp.ncbi.nlm.nih.gov/blast/db. Information on NR is available at http://www.ncbi.nlm.nih.gov/ BLAST/blast-databases.html.
10. Kabsch W, Sander C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers December 1983;22(12):2577–2637.
11. Frishman D, Argos P. Knowledge-based protein secondary structure assignment. Proteins 1995;23:566–579.
12. Bonneau R, Tsai J, Ruczinski I, Chivian D, Rohl C, Strauss CEM, Baker D. Rosetta in CASP4: progress in ab initio protein structure prediction. Proteins 2001;45:119–126.
13. Simons KT, Bonneau R, Ruczinski I, Baker D. Ab initio protein structure prediction of CASP III targets using ROSETTA. Proteins 1999;Suppl 3:171–176.
14. Simons KT, Kooperberg C, Huang C, Baker D. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. J Mol Biol. 1997;268:209–225.
15. Gilbrat J, Made T, Bryant S. Surprising similarities in structure comparison. Curr Opin Struct Biol 1996;6:377–85.
16. Fukushima K, Kikuchi J, Koshiba S, Kigawa T, Kuroda Y, Yokoyama S. Solution structure of the DFF-C domain of DFF45/ICAD: a structural basis for the regulation of apoptotic DNA fragmentation. J Mol Biol August 9, 2002;321:317–327.

tapraid5/7e-protein/7e-protein/7e1503/7e1455-03g | sweigarm | S=7 | 8/25/03 | 7:59 | 4/Color Figure(s): | Art: 01002003

AQ1: RRH OK ad added?

AQ2: OK as added?