Methods in Molecular Biology™

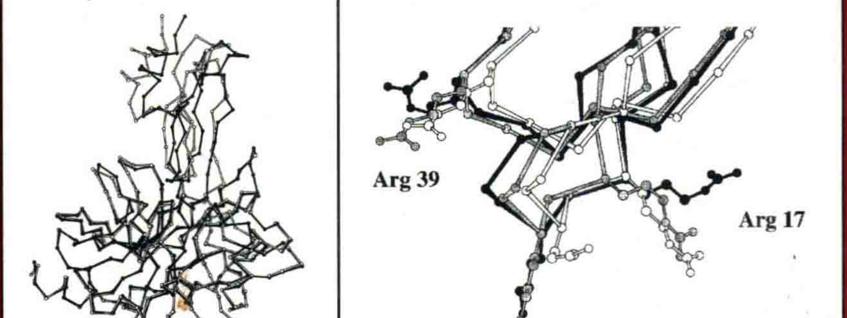
VOLUME 143

Protein Structure Prediction

Methods and Protocols

Edited by

David M. Webster



HUMANA PRESS

METHODS IN MOLECULAR BIOLOGY™

Protein Structure Prediction

Methods and Protocols

Edited by

David M. Webster

Southern Cross Molecular, Freshford, Bath, UK

Humana Press **Totowa, New Jersey**

©2000 Humana Press Inc.
999 Riverview Drive, Suite 208
Totowa, New Jersey 07512

All authored papers, comments, opinions, conclusions, or recommendations are those of the author(s), and do not necessarily reflect the views of the publisher.

This publication is printed on acid-free paper. ∞
ANSI Z39.48-1984 (American Standards Institute)
Permanence of Paper for Printed Library Materials.

Cover Design by Patricia F. Cleary
Cover Illustration: Figure 2 from Chapter 18, "Protein–Protein Docking: *Generation and Filtering of Complexes*" by Michael J. E. Sternberg, Henry A. Gabb, Richard M. Jackson, and Gidon Moont.

For additional copies, pricing for bulk purchases, and/or information about other Humana titles, contact Humana at the above address or at any of the following numbers: Tel.: 973-256-1699; Fax: 973-256-8341; E-mail: humana@humanapr.com or visit our Website: http://humanapress.com

Printed in the United States of America. 10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging in Publication Data

Protein Structure Prediction Methods and Protocols / edited by David M. Webster.
p.; cm.--(Methods in molecular biology; 143)
Includes bibliographical references and index.
ISBN 0-89603-637-5
1. Proteins--Structure--Mathematical models. 2. Amino acid sequence. I. Webster,
David M. II. Methods in molecular biology (Totowa, N.J.); v. 143
[DNLM: 1. Protein Conformation. 2. Models, Molecular. 3. Probability. QU 55 P9688 2000]
QP551 .P69769 2000
572'.633--dc21
DNLM/DLC 00-027596
For Library of Congress CIP

Protein Structure Prediction

METHODS IN MOLECULAR BIOLOGY™

John M. Walker, SERIES EDITOR

171. **Proteoglycan Protocols,** edited by *Renato V. Iozzo, 2001*
170. **DNA Arrays:** *Methods and Protocols,* edited by *Jang B. Rampal, 2001*
169. **Neurotrophin Protocols,** edited by *Robert A. Rush, 2001*
168. **Protein Structure, Stability, and Folding,** edited by *Kenneth P. Murphy, 2001*
167. **DNA Sequencing Protocols,** *Second Edition,* edited by *Colin A. Graham and Alison J. M. Hill, 2001*
166. **Immunotoxin Methods and Protocols,** edited by *Walter A. Hall, 2001*
165. **SV40 Protocols,** edited by *Leda Raptis, 2001*
164. **Kinesin Protocols,** edited by *Isabelle Vernos, 2001*
163. **Capillary Electrophoresis of Nucleic Acids, Volume 2:** *Practical Applications of Capillary Electrophoresis,* edited by *Keith R. Mitchelson and Jing Cheng, 2001*
162. **Capillary Electrophoresis of Nucleic Acids, Volume 1:** *The Capillary Electrophoresis System as an Analytical Tool,* edited by *Keith R. Mitchelson and Jing Cheng, 2001*
161. **Cytoskeleton Methods and Protocols,** edited by *Ray H. Gavin, 2001*
160. **Nuclease Methods and Protocols,** edited by *Catherine H. Schein, 2000*
159. **Amino Acid Analysis Protocols,** edited by *Catherine Cooper, Nicole Packer, and Keith Williams, 2000*
158. **Gene Knockoout Protocols,** edited by *Martin J. Tymms and Ismail Kola, 2000*
157. **Mycotoxin Protocols,** edited by *Mary W. Trucksess and Albert E. Pohland, 2000*
156. **Antigen Processing and Presentation Protocols,** edited by *Joyce C. Solheim, 2000*
155. **Adipose Tissue Protocols,** edited by *Gérard Ailhaud, 2000*
154. **Connexin Methods and Protocols,** edited by *Roberto Bruzzone and Christian Giaume, 2000*
153. **Neuropeptide Y Protocols,** edited by *Ambikaipakan Balasubramaniam, 2000*
152. **DNA Repair Protocols:** *Prokaryotic Systems,* edited by *Patrick Vaughan, 2000*
151. **Matrix Metalloproteinase Protocols,** edited by *Ian M. Clark, 2000*
150. **Complement Methods and Protocols,** edited by *B. Paul Morgan, 2000*
149. **The ELISA Guidebook,** edited by *John R. Crowther, 2000*
148. **DNA–Protein Interactions:** *Principles and Protocols* **(2nd ed.),** edited by *Tom Moss, 2000*
147. **Affinity Chromatography:** *Methods and Protocols,* edited by *Pascal Bailon, George K. Ehrlich, Wen-Jian Fung, and Wolfgang Berthold, 2000*
146. **Mass Spectrometry of Proteins and Peptides,** edited by *John R. Chapman, 2000*
145. **Bacterial Toxins:** ***Methods and Protocols,*** edited by *Otto Holst, 2000*
144. **Calpain Methods and Protocols,** edited by *John S. Elce, 2000*
143. **Protein Structure Prediction:** *Methods and Protocols,* edited by *David Webster, 2000*
142. **Transforming Growth Factor-Beta Protocols,** edited by *Philip H. Howe, 2000*
141. **Plant Hormone Protocols,** edited by *Gregory A. Tucker and Jeremy A. Roberts, 2000*
140. **Chaperonin Protocols,** edited by *Christine Schneider, 2000*
139. **Extracellular Matrix Protocols,** edited by *Charles Streuli and Michael Grant, 2000*
138. **Chemokine Protocols,** edited by *Amanda E. I. Proudfoot, Timothy N. C. Wells, and Christine Power, 2000*
137. **Developmental Biology Protocols, Volume III,** edited by *Rocky S. Tuan and Cecilia W. Lo, 2000*
136. **Developmental Biology Protocols, Volume II,** edited by *Rocky S. Tuan and Cecilia W. Lo, 2000*
135. **Developmental Biology Protocols, Volume I,** edited by *Rocky S. Tuan and Cecilia W. Lo, 2000*
134. **T Cell Protocols:** *Development and Activation,* edited by *Kelly P. Kearse, 2000*
133. **Gene Targeting Protocols,** edited by *Eric B. Kmiec, 2000*
132. **Bioinformatics Methods and Protocols,** edited by *Stephen Misener and Stephen A. Krawetz, 2000*
131. **Flavoprotein Protocols,** edited by *S. K. Chapman and G. A. Reid, 1999*
130. **Transcription Factor Protocols,** edited by *Martin J. Tymms, 2000*
129. **Integrin Protocols,** edited by *Anthony Howlett, 1999*
128. **NMDA Protocols,** edited by *Min Li, 1999*
127. **Molecular Methods in Developmental Biology:** Xenopus *and Zebrafish,* edited by *Matthew Guille, 1999*
126. **Adrenergic Receptor Protocols,** edited by *Curtis A. Machida, 2000*
125. **Glycoprotein Methods and Protocols:** *The Mucins,* edited by *Anthony P. Corfield, 2000*
124. **Protein Kinase Protocols,** edited by *Alastair D. Reith, 2000*
123. ***In Situ* Hybridization Protocols (2nd ed.),** edited by *Ian A. Darby, 2000*
122. **Confocal Microscopy Methods and Protocols,** edited by *Stephen W. Paddock, 1999*
121. **Natural Killer Cell Protocols:** *Cellular and Molecular Methods,* edited by *Kerry S. Campbell and Marco Colonna, 2000*
120. **Eicosanoid Protocols,** edited by *Elias A. Lianos, 1999*
119. **Chromatin Protocols,** edited by *Peter B. Becker, 1999*
118. **RNA–Protein Interaction Protocols,** edited by *Susan R. Haynes, 1999*
117. **Electron Microscopy Methods and Protocols,** edited by *M. A. Nasser Hajibagheri, 1999*
116. **Protein Lipidation Protocols,** edited by *Michael H. Gelb, 1999*
115. **Immunocytochemical Methods and Protocols (2nd ed.),** edited by *Lorette C. Javois, 1999*
114. **Calcium Signaling Protocols,** edited by *David G. Lambert, 1999*
113. **DNA Repair Protocols:** *Eukaryotic Systems,* edited by *Daryl S. Henderson, 1999*
112. **2-D Proteome Analysis Protocols,** edited by *Andrew J. Link, 1999*
111. **Plant Cell Culture Protocols,** edited by *Robert D. Hall, 1999*
110. **Lipoprotein Protocols,** edited by *Jose M. Ordovas, 1998*
109. **Lipase and Phospholipase Protocols,** edited by *Mark H. Doolittle and Karen Reue, 1999*
108. **Free Radical and Antioxidant Protocols,** edited by *Donald Armstrong, 1998*

Preface

The number of protein sequences grows each year, yet the number of structures deposited in the Protein Data Bank remains relatively small. The importance of protein structure prediction cannot be overemphasized, and this volume is a timely addition to the literature in this field.

Protein Structure Prediction: Methods and Protocols is a departure from the normal *Methods in Molecular Biology* series format. By its very nature, protein structure prediction demands that there be a greater mix of theoretical and practical aspects than is normally seen in this series. This book is aimed at both the novice and the experienced researcher who wish for detailed information in the field of protein structure prediction; a major intention here is to include important information that is needed in the day-to-day work of a research scientist, important information that is not always decipherable in scientific literature.

Protein Structure Prediction: Methods and Protocols covers the topic of protein structure prediction in an eclectic fashion, detailing aspects of prediction that range from sequence analysis (a starting point for many algorithms) to secondary and tertiary methods, on into the prediction of docked complexes (an essential point in order to fully understand biological function). As this volume progresses, the authors contribute their expert knowledge of protein structure prediction to many disciplines, such as the identification of motifs and domains, the comparative modeling of proteins, and *ab initio* approaches to protein loop, side chain, and protein prediction. Also covered is the development of suitable folding potentials, essential for many of the methods presented in this volume. One of the more difficult areas of protein structure prediction is that of predicting the structure of transmembrane proteins and finally an area of great importance to the pharmacological field is receptor site prediction.

With advances in computer technology and a greater understanding of the principles of protein folding, the accuracy and the usefulness of predicted structures is increasing. *Protein Structure Predicting: Methods and Protocols* outlines many of the most recent advances in protein structure prediction and provides an essential resource to both the nonexpert and expert in the field.

I would like to thank all of the authors for their contributions to this work.

David M. Webster

Contributors

Ewan Birney• *The Sanger Centre, Cambridge, UK*

Robert E. Bruccoleri• *Bristol-Myers Squibb, Pharmaceutical Research Institute, Princeton, NJ*

Leo Davison • *Laboratory of Molecular Biophysics, University of Oxford, Oxford, UK*

Marc De Maeyer • *Center for Transgene Technology and Gene Therapy, Flanders Interuniversity Institute for Biotechnology, Leuven, Belgium*

Johan Desmet • *Interdisciplinary Research Center, Kortrijk, Belgium*

Alexei V. Finkelstein • *Institute of Protein Research, Russian Academy of Sciences, Pushchino, Moscow Region, Russian Federation*

Henry A. Gabb • *Biomolecular Modelling Laboratory, Imperial Cancer Research Fund, London, UK*

Desmond G. Higgens • *Department of Biochemistry, University of College Cork, Cork, Ireland*

Enoch S. Huang • *Department of Molecular Cell Biology, Phizer Discover Technology Center, Cambridge, MA*

Richard M. Jackson • *Biomolecular Modelling Laboratory, Imperial Cancer Research Fund, London, UK*

Inge Jonassen • *Department of Informatics, University of Bergen, Bergen, Norway*

David T. Jones • *Department of Biological Sciences, University of Warwick, Conventry, UK*

Ignace Lasters • *Beagle bvla., Antwerpen, Belgium*

Gidon Moont • *Biomolecular Modelling Laboratory, Imperial Cancer Research Fund, London, UK*

Ruth Nussinov • *Laboratory of Experimental and Computational Biology, SAIC, Frederick, MD*

Britt H. Park • *Department of Structural Biology, Stanford University School of Medicine, Stanford, CA*

Chris P. Ponting • *MRC Functional Genetics Unit, Department of Human Anatomy and Genetics, University of Oxford, Oxford, UK*

Boris A. Reva • *Department of Molecular Biology, The Scripps Research Institute, La Jolla, CA*

Burkhard Rost • *Department of Biochemistry and Molecular Biophysics, Columbia University, New York, NY*

ROBERT B. RUSSELL • *Research and Development, Bioinfomatics Research Group, SmithKline Beecham Pharmaceuticals, Essex, UK*

ANDREJ ŠALI • *Laboratories of Molecular Biophysics, The Rockefeller University, New York, NY*

RAM SAMUDRALA • *Department of Structual Biology, Stanford University School of Medicine, Stanford, CA*

ROBERTO SÁNCHEZ • *Laboratories of Molecular Biophysics, The Rockefeller University, New York, NY*

CHRIS SANDER • *Millenium Bioinformatics, Boston, MA*

MARK S. P. SANSOM • *Laboratory of Molecular Biophysics, University of Oxford, Oxford, United Kingdom*

STEFFEN SCHULZE-KREMER • *Max-Planck Institute for Molecular Genetics, Berlin, Germany*

JEFFERY SKOLNICK • *Department of Molecular Biology, The Scripps Research Institute, La Jolla, CA*

JAN SPRIET • *Interdisciplinary Research Center, Kortrijk, Belgium*

MICHAEL J. E. STERNBERG • *Biomolecular Modelling Laboratory, Imperial Cancer Research Fund, London, UK*

WILLIAM R. TAYLOR • *Division of Mathematical Biology, National Institute for Medical Research, London, UK*

D. ERIC WALTERS • *Department of Biological Chemistry, Finch University of Health Sciences, The Chicago Medical School, North Chicago, IL*

DAVID M. WEBSTER • *Southern Cross Molecular, Freshford, Bath, UK*

HAIM J. WOLFSON • *Computer Science Department, School of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel*

Contents

1

Multiple Sequence Alignment

Desmond G. Higgins and William R. Taylor

1. Introduction

The alignment of protein sequences is the most powerful computational tool available to the molecular biologist. Where one sequence is of unknown structure and function, its alignment with another sequence that is well characterized in both structure and function immediately reveals the structure and function of the first sequence. This ideal transfer of information is, unfortunately, not always attained and can fail either because the two sequences are equally uncharacterized (although they might align quite well) or because the alignment is too poor to be trusted. Both these situations can be helped if the analysis is extended to incorporate more sequences. In the former case, the addition of further sequences can reveal portions of the protein that are important in structure and function (even if that structure or function is unknown), whereas in the latter, the revelation of conserved patterns can help add confidence in the alignment.

In this chapter, we describe two methods that can be used to produce multiple sequence alignments. Both are based on the simple heuristic that it is best to align the most similar sequences first and gradually combine these, in a hierarchic manner, into a multiple sequence alignment.

2. MULTAL

2.1. Outline of the Algorithm

The Program MULTAL was originally devised to deal with large numbers of protein sequences that are typically encountered in the analysis of large families (such as the immunogobulins or globins) or in sifting out the often extensive collections of sequences produced as the result of a search across the

From: *Methods in Molecular Biology*, vol. 143: *Protein Structure Prediction: Methods and Protocols*
Edited by: D. Webster © Humana Press Inc., Totowa, NJ

sequence databanks. These applications are the main topic considered in this section. Those who wish to use the program only as an alignment/editor for a small number of sequences would be best to seek out the program CAMELON <http://www.oxmol.co.uk/prods/camelon/> (which is an implementation of MULTAL by Oxford Molecular) or CLUSTAL (*see* **Subheading 3.**).

Where CLUSTAL takes a more rigorous phylogenetic approach to ordering of sequences prior to alignment, MULTAL uses a simple single-linked clustering iterated over several cycles. On each cycle, only sequences that have a pairwise similarity greater than a predefined cutoff (specified of each cycle) are aligned. If more than two sequences are mutually similar above the current cutoff score, then all are brought together in one step using a fast concatenation algorithm (*see* **ref.** ***1***). However, as this is only robust for closely related sequences, later cycles are restricted to pairwise combinations.

In each cycle, all subalignments and all single sequences are again compared with each other. Here the algorithm differs significantly from CLUSTAL, which adheres to the original guide tree and is more similar to the GCG program PILEUP (http://www.gcg.com/products/software.html) that developed out of a simpler approach (2). When aligning a sequence with an alignment or an alignment with an alignment, MULTAL calculates a pairwise sum over the similarity of each amino acid in one alignment with each amino acid in the other alignment. MULTAL retains this simple sum, whereas CLUSTAL provides a weighting scheme to down-weight the contribution from similar sequences. This feature was not provided in MULTAL, as the alternate approach (which is more practical with large numbers of sequences) is simply to remove one of a pair of similar sequences. A protocol for this is described as follows.

2.2. Strategies for Large Numbers of Sequences

MULTAL contains numerous methods to deal with large numbers of sequence (where large is considered to be hundreds or thousands of sequences). Although very valuable, this aspect can require understanding and careful treatment if the program is not to miss expected similarities. Generally, there is a trade-off between time spent and the chance of missing a relationship.

2.2.1. The Span Parameter

The greatest saving in time that can be made when dealing with a large number of sequences is to avoid the costly comparison of all against all (this is especially true for MULTAL, where this calculation is performed on each cycle). If the sequences were presented in an optimal order in which the most similar sequences were adjacent, then MULTAL would only need to consider adjacent sequences on each cycle — transforming a time dependency that was proportional to the square of the sequences into a time dependency that is linear in the number of sequences. As such an optimal order cannot easily be obtained, MULTAL considers the pairwise similarity over a number of adjacent sequences, specified by a parameter called the **span**, which can be varied from cycle to cycle, as can all the MULTAL parameters.

In general, the span starts small (comparing only local sequences) and expands from cycle to cycle. However, even if it remains fixed at a small number, there is still a good chance of obtaining a complete multiple alignment, because, as the cycles progress, the number of "sequences" (which now includes subalignments) decreases relative to the span so that by the final cycles, the number of subalignments plus unaligned sequences (referred to jointly as *blocks*) is less than the span and so all are eventually compared to all.

2.2.2. The Window Parameter

A related saving can be made at the level of the detailed calculation of the alignment. If the initial cycles are only aligning relatively similar sequences, then the size of relative insertion and deletion needed to obtain the optimal alignment can be expected to be relatively small. If restrictions are placed on the alignment path, then a calculation of time dependent on the product of the sequences becomes approximately linear in sequence length. The parameter that controls this is called the **window** and its value specifies a diagonal stripe (placed symmetrically) through the matrix (dot-plot) constructed from placing each sequence on the sides of a rectangle. As a safeguard, however, if the difference in sequence length is greater than the size of the window parameter value, then the sequences are not compared on that cycle. In general (as with the span parameter), the value of the window parameter should be increased through successive cycles.

2.2.3. Peptide Presort

The efficient operation of both the span and window parameters rely on having a well-ordered starting list of sequences. Often, sequences are fouund preordered in existing databanks or as the result of a previous alignment using MULTAL or some other program. (Both MULTAL and CLUSTAL record the resulting alignment to be used in this way.) However, if this is not avaliable, then MULTAL can (optionally) attempt to create it based on a rough measure of similarity based on an analysis of the peptide composition of each sequence — specifically, the number of common peptides between sequences. This can be calculated very quickly using a simple hash-table or as in the current versions of MULTAL, using a dynamic radix tree structure that can accommodate any peptide size. The size of peptide that is used for this analysis can be specified but, in general, less than three is too general and over four is too specific (too few common peptides are found in all but the most similar sequences). Originally, a tetrapeptide was used *(3)* and it was also shown *(4)* that a tripeptide measure can capture sequence similarity quite well down to roughly the level of 50% identity.

2.3. Alignment Parameters

As in all alignment methods, it is necessary to specify a measure of similarity between amino acids to provide an alignment score and, in addition, specify both a model and parameters for the penalty attached to relative insertions and deletions (gaps). As in other aspects of MULTAL, these aspects are kept very simple as it is the general philosophy of the approach that the important contribution to the alignment is the number and quality of the sequences (with respect to their phylogenetic distribution) that makes a good alignment and not the fine tuning of parameters. For example, if a good selection of sequences are obtained, then these effectively define their own local amino acid exchange matrix at every position.

2.3.1. Amino Acid Exchange Matrix

MULTAL allows two matrices to be used in each run and these can be combined in varying proportions on each cycle. Generally, the two matrices used are the identity matrix (in which amino acid identites score 10 and all else 0) and the PAM_{120} matrix *(5)*. These are stored in the files id.mat and md.mat but can be substituted for any other matrix, e.g., Dayhoff's PAM_{250} matrix, a BLOSUM matrix *(15)*, or even the JTT matrix *(4)*. Through the different cycles, the current matrix is a linear interpolation between the two given matrices, specified by the parameter matrix that gives the porportion (out of 10) that the matrix in md.mat contributes. For example, if matrix = 3, then (with the PAM_{120} matrix in md.mat), the values used in the alignment calculation are 30% of the PAM_{120} values augmented by 7 on the diagonal (being 70% of the values in the identity matrix in id.mat). The same overall effect might have been attained by using a series of PAM or BLOSUM matrices (as can be used in the CLUSTAL program), however, the fine specification of values makes little difference to the alignment and the use of an identity matrix produces values that are more familiar.

In the past, the matrix parameter was increased from cycle to cycle, with the expectation that later alignments would be composed of more distant sequences and should therefore have a matrix suited to their degree of divergence (e.g., the PAM_{250} matrix). However, although this is still true for isolated sequences that have not aligned, it does not apply to subalignments, as these have already effectively created their own individual amino acid exchange matrix at every position composed out of the sum of amino acid pairwise similarities. This effect combined with a "soft" matrix (one that scores general similarity) leads to too much flexibility in the match and tends to diminish the importance of highly conserved positions (of which there are often relatively few) and can lead to both misalignment and the false incorporation of sequences that do not belong in the family.

2.3.2. Gap Penalties

Adhering to the philosophy that the simplest alignment principles are sufficient, MULTAL has only one gap penalty that is paid once for a gap of any size — but not at the beginning or end of a sequence. This is justified in the context of the alignment of distant protein sequences by the expectation (1) that the locations where insertions can occur in the protein structure are generally on the surface and (2) that if a small insertion can be made, there are probably few constraints on this forming a linker out to a larger insertion that might even comprise a complete domain. As with the matrix parameter, the gap penalty can be varied over the cycles, but little justification has been seen for this and, generally a constant gap value in the range 20–30 is maintained over the full run.

Some later and more experimental versions of MULTAL embody more complex gap functions. These were designed to take account of the structural expectation that matches in a sequences alignment are correlated, often being found in runs (typical of a conserved secondary structure) ***(6,7)***, or having an overall distribution that cannot be adequately controlled by a penalty applied independently at each insertion point ***(8)***. These more subtle aspects have also been reviewed in a less technical volume ***(9)***.

2.4. When to Stop Aligning

Programs such as MULTAL or CLUSTAL (or any of their ilk) contain no inherent method to detect when two sequences (or subalignments) should not be aligned together. The various algorithms can produce an alignment even when the sequences are random. Rough guidelines, such as percentage sequence identity can be used, or statistics such as those employed in databank search methods. However, there are no adequate statistics that can be applied to the more complex situation of aligning alignments. Even the percentage identity is not a good guide as the pairwise similarity among sequences that can be reliably aligned using multiple sequence alignment methods extends far into what would be considered random were the two sequences to be extracted and assessed as a pair. These scores are also directly derived from the current matrix and gap penalty, which is also difficult to allow for.

Strategies, that can be employed with MULTAL are to allow the alignment to go to completion (one big family) but then to backtrack up the cycles (using careful visual assessment) until the point at which the subfamilies last seemed

Table 1
MULTAL Parameter Files for Alignment

Matrix	Gap	Span	Win.	Cutoff
5	20	3	30	700
5	20	5	40	600
5	20	7	50	500
5	20	9	60	400
5	20	9	70	300
5	20	9	80	250
5	20	9	90	200
5	20	9	100	150
5	20	9	100	150

The columns are, respectively, the *matrix* parameter (5 = 50% PAM_{120}), the *gap* penalty, the number of adjacent sequences considered (*span*), boundary (window) on alignment deviation (*win.*), and the score *cutoff*. Each line of parameters is used in successive cycles. (*See* and **ref. 3** for details.)

to be credible. This places considerable burden on the method used for "visual assessment" and in the absence of any structural or functional knowledge, this can only be judged by the conservation of groups that might be involved in structure or function. The former are generally interesting residues, such as arginine, aspartate, histidine, or any charged amino acid that might be capable of catalysis or binding. The residues of structural importance are generally hydrophobic, with glycine, proline, and cysteine often conserved because of their unique properties.

Visual assessment cannot be employed in automatic family compilation or where the user has little "feel" for the data. In this situation, it has been found (through accumulated experience) that with a matrix value of 3 and a gap penalty of 20–30, the recommended lower limit on the score cutoff is 150. At this level, in repeated trials, there are roughly as many family members that do not align as there are false alignments. A value of 200 or 250 would be recommended as a safer choice for those who have little or no feel for the quality of sequence alignments (*see* **Table 1** for an example of parameter file).

2.5. Sequence Selection with MULTAL

2.5.1. Sequence Criteria

Sequences can be selected using the program MULTAL as a prefilter to form subfamilies above a preset degree of similarity (details in **Tables 1** and **2**). From each subfamily, a representative sequence was chosen according to the weighting scheme that valued sequences with a respresentative length that did not contain any nonstandard amino acids. A measure *r* was calculated:

Table 2
MULTAL Parameter Files for Filtering

(A) Filter to 90%

Matrix	Gap	Span	Win.	Cutoff
0	20	1	1	990
0	20	2	1	980
0	20	4	2	960
0	20	8	3	940
0	20	10	4	920
0	20	10	5	900
0	20	10	5	900

(B) Filter to 80%

Matrix	Gap	Span	Win.	Cutoff
0	20	1	5	890
0	20	2	6	880
0	20	4	7	860
0	20	8	8	840
0	20	10	9	820
0	20	10	10	800
0	20	10	10	800

(C) Filter to 70%

Matrix	Gap	Span	Win.	Cutoff
0	20	1	10	790
0	20	2	12	780
0	20	4	14	760
0	20	8	16	740
0	20	10	18	720
0	20	10	20	700
0	20	10	20	700

The columns are, respectively, the *matrix* parameter (0 = identity), the *gap* penalty, the number of adjacent sequences considered (*span*), boundary (window) on alignment deviation (*win.*), and the score *cutoff*. Each line of parameters is used in successive cycles. (*See* above and **ref. *3*** for details.)

$$r = \log(d^2 + 1) + s \qquad (1)$$

where d is the difference in length of an individual sequence from the mean length of the subfamily in which it is aligned and s is the number of nonstandard amino acid symbols (included, B J O U X Z). To this basic score, penalties and bonus points were added as defined in **Table 3** and the sequence with the lowest score was selected.