Methods in Molecular Biology 1529

Springer Protocols

Ilan Samish *Editor*

Computational Protein Design

Humana Press

METHODS IN MOLECULAR BIOLOGY

Series Editor
John M. Walker
School of Life and Medical Sciences
University of Hertfordshire
Hatfield, Hertfordshire, AL10 9AB, UK

For further volumes:
http://www.springer.com/series/7651

Computational Protein Design

Edited by

Ilan Samish

Department of Plants and Environmental Sciences
Weizmann Institute of Science, Rehovot, Israel

Department of Biotechnology Engineering
Braude Academic College of Engineering, Karmiel, Israel

Amai Proteins Ltd., Ashdod, Israel

Humana Press

Editor
Ilan Samish
Department of Plants and Environmental Sciences
Weizmann Institute of Science, Rehovot, Israel

Department of Biotechnology Engineering
Braude Academic College of Engineering, Karmiel, Israel

Amai Proteins Ltd., Ashdod, Israel

ISSN 1064-3745 ISSN 1940-6029 (electronic)
Methods in Molecular Biology
ISBN 978-1-4939-6635-6 ISBN 978-1-4939-6637-0 (eBook)
DOI 10.1007/978-1-4939-6637-0

Library of Congress Control Number: 2016959982

Printed on acid-free paper

This Humana Press imprint is published by Springer Nature
The registered company is Springer Science+Business Media LLC
The registered company address is: 233 Spring Street, New York, NY 10013, U.S.A.

Contents

Contributors

Rudresh Acharya • *School of Biological Sciences, National Institute of Science Education and Research, Odisha, India; Homi Bhabha National Institute, Training School Complex, Anushakti Nagar, Mumbai, India*

Carlos Aleman • *Departament d'Enginyeria Química, E. T. S. d'Enginyeria Industrial de Barcelona, Universitat Politècnica de Catalunya, Barcelona, Spain; Center for Research in Nano-Engineering, Universitat Politècnica de Catalunya, Barcelona, Spain*

David Allouche • *Unité de Mathématiques et Informatique Appliquées de Toulouse, Castanet Tolosan, France*

Amy C. Anderson • *Department of Pharmaceutical Sciences, University of Connecticut, Storrs, CT, USA*

Isabelle André • *INSA, UPS, INP, Université de Toulouse, Toulouse, France; Laboratoire d'Ingénierie des Systèmes Biologiques et des Procédés, INRA, UMR792, Toulouse, France; CNRS, UMR5504, Toulouse, France*

Naureen Aslam • *Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, MI, USA*

Chris Bailey-Kellogg • *Department of Computer Science, Dartmouth College, Dartmouth, USA*

Sophie Barbe • *INSA, UPS, INP, Université de Toulouse, Toulouse, France; Laboratoire d'Ingénierie des Systèmes Biologiques et des Procédés, INRA, UMR792, Toulouse, France; CNRS, UMR5504, Toulouse, France*

Arménio J. M. Barbosa • *UCIBIO, REQUIMTE, Departamento de Química, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Caparica, Portugal*

Ricardo J. F. Branco • *UCIBIO, REQUIMTE, Departamento de Química, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Caparica, Portugal*

Jeffrey R. Brender • *Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, MI, USA*

Henrique F. Carvalho • *UCIBIO, REQUIMTE, Departamento de Química, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Caparica, Portugal; Instituto de Tecnologia Química e Biológica António Xavier, Universidade Nova de Lisboa, Av. da República, Oeiras, Portugal*

Matthieu Chartier • *Department of Biochemistry, Faculty of Medicine and Health Sciences, University of Sherbrooke, Sherbrooke, QC, Canada*

Quan Chen • *School of Life Sciences, Hefei National Laboratory for Physical Sciences at the Microscales, and Collaborative Innovation Center of Chemistry for Life Sciences, University of Science and Technology of China, Hefei, Anhui, China*

Roberto A. Chica • *Department of Chemistry and Biomolecular Sciences, University of Ottawa, Ottawa, ON, Canada*

Yoonjoo Choi • *Department of Computer Science, Dartmouth College, Dartmouth, USA*

James A. Davey • *Department of Chemistry and Biomolecular Sciences, University of Ottawa, Ottawa, ON, Canada*

Charlotte M Deane • *Department of Statistics, University of Oxford, Oxford, UK*

Nikolay V. Dokholyan • *Department of Biochemistry and Biophysics, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA*
Bruce R. Donald • *Department of Computer Science, Duke University, Durham, NC, USA; Department of Biochemistry, Duke University, Durham, NC, USA; Department of Chemistry, Duke University, Durham, NC, USA*
James Dunbar • *Department of Statistics, University of Oxford, Oxford, UK*
Juan Fernández-Recio • *Joint BSC-CRG-IRB Research Program in Computational Biology, Barcelona Supercomputing Center, Barcelona, Spain*
Vincent Frappier • *Department of Biochemistry, Massachusetts Institute of Technology, Cambridge, MA; Faculty of Medicine and Health Sciences, University of Sherbrooke, Sherbrooke, QC, Canada*
Pablo Gainza • *Department of Computer Science, Duke University, Durham, NC, USA*
Wenxun Gan • *Research Center of Basic Medical Sciences and Cancer Institute and Hospital, Tianjin Medical University, Tianjin, China*
Ivelin Georgiev • *Department of Computer Science, Duke University, Durham, NC, USA; Vaccine Research Center, National Institute of Allergy and Infectious Diseases, Bethesda, MD, USA*
Fabian Glaser • *Bioinformatics Knowledge Unit, The Lorry I. Lokey Interdisciplinary Center for Life Sciences and Engineering, Technion, Israel*
Karl E. Griswold • *Thayer School of Engineering, Dartmouth, USA*
Nurit Haspel • *Department of Computer Science, The University of Massachusetts Boston, Boston, MA, USA*
Wenkang Huang • *Department of Pathophysiology, Key Laboratory of Cell Differentiation and Apoptosis of Chinese Ministry of Education, Shanghai JiaoTong University School of Medicine (SJTU-SM), Shanghai, China*
Olga Iranzo • *Aix Marseille Université, Centrale Marseille, CNRS, iSm2 UMR 7313, Marseille, France*
Armando J. De Jesus • *Department of Chemistry and Biochemistry, University of Colorado, Boulder, CO, USA; The BioFrontiers Institute, University of Colorado, Boulder, CO, USA*
Sagar D. Khare • *Department of Chemistry and Chemical Biology, Center for Integrative Proteomics Research, Rutgers, The State University of New Jersey, Piscataway, NJ, USA*
Daisuke Kihara • *Department of Biological Sciences, Purdue University, West Lafayette, IN, USA; Department of Computer Science, Purdue University, West Lafayette, IN, USA*
Patrice Koehl • *Department of Computer Science and Genome Center, University of California, Davis, CA, USA*
Konrad Krawczyk • *Department of Statistics, University of Oxford, Oxford, UK*
David La • *Department of Biochemistry, University of Washington, Seattle, WA, USA*
Jie Li • *Computational and Systems Biology Group, Genome Institute of Singapore, Agency for Science, Technology and Research, Singapore, Singapore*
Haiyan Liu • *School of Life Sciences, Hefei National Laboratory for Physical Sciences at the Microscales, University of Science and Technology of China, Hefei, Anhui, China; Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei, Anhui, China*
Anna Lowegard • *Program in Computational Biology and Bioinformatics, Duke University, Durham, NC, USA*
Buyong Ma • *Basic Science Program, Leidos Biomedical Research, Inc, Frederick, MD, USA; Cancer and Inflammation Program, National Cancer Institute, Frederick, MD, USA*
David D. Mowrey • *Department of Biochemistry and Biophysics, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA*

Rafael Najmanovich • *Department of Pharmacology and Physiology, Faculty of Medicine, University of Montreal, Montreal, QC, Canada*

Ruth Nussinov • *Department of Human Genetics, Sackler School of Medicine, Sackler Inst. of Molecular Medicine and Molecular Medicine, Tel Aviv University, Tel Aviv, Israel; Basic Science Program, Cancer and Inflammation Program, Leidos Biomedical Research, Inc., National Cancer Institute, Frederick, MD, USA*

Adegoke Ojewole • *Program in Computational Biology and Bioinformatics, Duke University, Durham, NC, USA*

Chiara Pallara • *Joint BSC-CRG-IRB Research Program in Computational Biology, Barcelona Supercomputing Center, Barcelona, Spain*

Stephanie M. Reeve • *Department of Pharmaceutical Sciences, University of Connecticut, Storrs, CT, USA*

Miguel Romero-Durana • *Joint BSC-CRG-IRB Research Program in Computational Biology, Barcelona Supercomputing Center, Barcelona, Spain*

Ana C. A. Roque • *UCIBIO, REQUIMTE, Departamento de Química, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Caparica, Portugal*

Ilan Samish • *Department of Plants and Environmental Sciences, Weizmann Institute of Science, Rehovot, Israel; Department of Biotechnology Engineering, Braude Academic College of Engineering, Karmiel, Israel; Amai Proteins Ltd., Ashdod, Israel; Dept of Biotechnology Engineering, Braude Academic College of Engineering, Karmiel, Israel; Amai Proteins Ltd., Ashdod, Israel*

Thomas Schiex • *Unité de Mathématiques et Informatique Appliquées de Toulouse, UR 875, INRA, Castanet Tolosan, France*

Dina Schneidman • *Department of Bioengineering and Therapeutic Sciences, University of California, San Francisco, CA, USA*

David Shultis • *Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, MI, USA*

David Simoncini • *Structural Bioinformatics Team, Division of Structural and Synthetic Biology, Center for Life Science Technologies, RIKEN, Yokohama, Kanagawa, Japan; MIAT, UR-875, INRA, Castanet Tolosan, France*

Jeremy R. H. Tame • *Drug Design Laboratory, Graduate School of Medical Life Science, Yokohama City University, Yokohama, Kanagawa, Japan*

Christine E. Tinberg • *Department of Biochemistry, University of Washington, Seattle, WA, USA*

Seydou Traoré • *INSA, UPS, INP, Université de Toulouse, Toulouse, France; INRA, UMR792, Université de Toulouse, Toulouse, France; Laboratoire d'Ingénierie des Systèmes Biologiques et des Procédés, CNRS, UMR5504, Toulouse, France*

Barsa Tripathy • *School of Biological Sciences, National Institute of Science Education and Research, Odisha, India; Homi Bhabha National Institute, Training School Complex, Anushakti Nagar, Mumbai, India*

Deeptak Verma • *Department of Computer Science, Dartmouth College, Dartmouth, USA*

Arnout R. D. Voet • *Structural Bioinformatics Team, Division of Structural and Synthetic Biology, Center for Life Science Technologies, RIKEN, Yokohama, Kanagawa, Japan*

Qing Wei • *Department of Computer Science, Purdue University, West Lafayette, IN, USA*

Peng Xiong • *School of Life Sciences, Hefei National Laboratory for Physical Sciences at the Microscales, and Collaborative Innovation Center of Chemistry for Life Sciences, University of Science and Technology of China, Hefei, Anhui, China*

HANG YIN • *Department of Chemistry and Biochemistry, The BioFrontiers Institute, University of Colorado, Boulder, CO, USA*
DAVID ZANUY • *Departament d'Enginyeria Química, E. T. S. d'Enginyeria Industrial de Barcelona, Universitat Politècnica de Catalunya, Barcelona, Spain*
JIANYANG ZENG • *Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, P. R. China*
JIAN ZHANG • *Department of Pathophysiology, Key Laboratory of Cell Differentiation and Apoptosis of Chinese Ministry of Education, School of Medicine (SJTU-SM), Shanghai JiaoTong University, Shanghai, P. R. China*
KAM Y. J. ZHANG • *Structural Bioinformatics Team, Division of Structural and Synthetic Biology, Center for Life Science Technologies, RIKEN, Yokohama, Kanagawa, Japan*
NING ZHANG • *Research Center of Basic Medical Sciences and Cancer Institute and Hospital, Tianjin Medical University, Tianjin, China*
YANG ZHANG • *Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, MI, USA; Department of Biological Chemistry, University of Michigan, Ann Arbor, MI, USA*
JIE ZHENG • *Department of Chemical and Biomolecular Engineering, The University of Akron, Akron, OH, USA*
YICHAO ZHOU • *Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, P. R. China*
CHENG ZHU • *Department of Biochemistry and Biophysics, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA*

Chapter 11

Computational Protein Design Through Grafting and Stabilization

Cheng Zhu, David D. Mowrey, and Nikolay V. Dokholyan

Abstract

Computational grafting of target residues onto existing protein scaffolds is a powerful method for the design of proteins with novel function. In the grafting method side chain mutations are introduced into a preexisting protein scaffold to recreate a target functional motif. The success of this approach relies on two primary criteria: (1) the availability of compatible structural scaffolds, and (2) the introduction of mutations that do not affect the protein structure or stability. To identify compatible structural motifs we use the Erebus webserver, to search the protein data bank (PDB) for user-defined structural scaffolds. To identify potential design mutations we use the Eris webserver, which accurately predicts changes in protein stability resulting from mutations. Mutations that increase the protein stability are more likely to maintain the protein structure and therefore produce the desired function. Together these tools provide effective methods for identifying existing templates and guiding further design experiments. The software tools for scaffold searching and design are available at http://dokhlab.org.

Key words Scaffold search, Refinement, Stabilization, Mutation, Free energy, Protein design

1 Introduction

The goal of the protein design field is to engineer proteins with novel function with implications for developing new enzymes, biosensors, and therapeutics [1–4]. One approach to protein design is grafting, which has been successful in the developing of novel inhibitors [2, 5], biomarkers [6], enzymes [7], and antigens [8]. In the grafting approach one identifies an existing structural scaffold that can host a specific motif and replaces residues to match a desired active site or binding motif. This approach relies on the availability of potential scaffolds into which design mutations can be introduced. To this end, several methods have been developed to identify these scaffolds including Multigraft Match [9], GRAFTER [10], and MaDCaT [11]. These approaches rely on matching of backbone atoms, Cα-Cβ vectors, or Cα distance maps to identify potential scaffolds. In our approach we use the

Ilan Samish (ed.), *Computational Protein Design*, Methods in Molecular Biology, vol. 1529,
DOI 10.1007/978-1-4939-6637-0_11,

Erebus webserver, which allows greater user flexibility in matching user-specified atom types and residues, while maintaining high speed and accuracy [12] .

Having obtained a viable scaffold onto which target residues can be introduced, a further challenge is to determine a priori whether desired mutations will distort the original protein scaffold or even completely destabilize the protein structure. To improve thermodynamic stabilities of designed proteins, we use the protein stability prediction software Eris [13]. Eris has been shown to effectively predict effects of mutations, even for the more challenging case of small to large mutations, without the need for computationally intensive molecular dynamics simulations.

The overall workflow for our protein design protocol is composed of three major steps. In the first step we identify protein scaffolds for redesign using the Erebus. In the second step we submit the structure to the Chiron webserver for pre-relaxation prior to introducing mutations. In the final step we use the Eris webserver to identify potential redesign sequences. The theory and protocol for each of these methods are outlined in the following sections.

1.1 Identify Protein Scaffolds for Redesign

The identification of promising candidates for redesign makes use of the Erebus substructure search [12]. Provided with a query structure Erebus scans the protein data bank (PDB) for matching structural scaffolds [14]. Atom pairs of target structures in the structural database matching the name, residues, and distances in the query structure are collected to create the candidate substructures. The best substructures are selected based on their final weights representing the agreement between query and target structures. The final weights are calculated according to the following equation:

$$W = \left(\prod_{i=1}^{N} W_i\right)^{1/N} \prod_{j=1}^{M} (1 - w_j)$$

where the final weight (W) is the geometric mean of the weights (W_i) for each of the N atom pairs. The product is multiplied by an additional penalty (w_j) for each of the M missing atoms. Individual weights (W_i) for each atom pair are calculated according to the formula:

$$W_i = \mathrm{e}^{-\frac{\left(\Delta q_i - \Delta t_j\right)^2}{\sigma^2}}$$

Where the term $(\Delta q_i - \Delta t_j)$ represents the difference in distances between atom pairs i and j in the query (q) and target (t) structures, and σ^2 is a user-defined precision parameter.

1.2 Protein Preparation

As steric clashes are common structural artifacts observed in homology models and low-resolution crystal structures [15, 16], we first relax structures using Chiron before proceeding to Eris. In this method, a clash is defined as any atomic overlap resulting in energy greater than 0.3 kcal/mol (0.5 k_BT). Structural relaxation in Chiron is achieved by performing a series of short (~10 ps) discrete molecular dynamics simulations [17, 18], using a high heat exchange rate (5 fs) between the protein and the thermal bath. The exchange rate is used to effectively quench large atomic velocities resulting from large van der Waals clashes, which could result in broken bonds. To prevent large structural distortions, we also constrain backbone and Cβ atoms with a harmonic potential. The algorithm alternates between high temperature (0.7 ε/k_B, ~350 K) and low temperature (0.5 ε/k_BT, ~250 K) simulations until the overall clash score is less than 0.02 kcal/mol/contact. Pre-relaxing the structures in this way significantly improves sidechain packing of the protein core, which improves the accuracy of $\Delta\Delta G$ evaluations.

1.3 Protein Redesign

Protein redesign of preexisting scaffolds is accomplished using Eris [13]. Eris introduces a mutation or set of mutations into a protein structure and calculates free energies for both mutant (ΔG_{MUT}) and native (ΔG_{NAT}) structures. For free energy calculations rapid side-chain repacking and backbone relaxation are performed around the mutation site(s) using a Monte Carlo algorithm. The free energies are the result of a weighted sum of van der Waals forces, solvation, hydrogen bonding, and backbone-dependent statistical energies derived from the Medusa force field [19]. The final prediction of protein stability induced by mutations is expressed as the $\Delta\Delta G$ ($\Delta G_{MUT} - \Delta G_{NAT}$). Weighting parameters for free energy calculations were independently trained on 34 high-resolution X-ray protein structures and tested on a large dataset of 595 mutants where we found significant agreement between predicted and measured $\Delta\Delta G$ values ($R^2 = 0.75$) [13]. Furthermore, Eris can model the backbone flexibility, which is crucial for $\Delta\Delta G$ estimation of small-to-large mutations [20].

2 Software Requirements

The webservers of Erebus, Chiron, and Eris (ddg module) are freely available on our group page (http://dokhlab.org). The current version of Eris (ddg, scan and design module) also supports Linux/Unix-like platforms with the C and C++ compilers installed. It has been tested on Linux, Microsoft Windows (with the Linux port Cygwin), and Mac OS X. Our methods usually require a molecular viewer for preparation of crystal structures and analysis of results. For these purposes PyMol (http://pymol.org), an Open Source molecular viewer available on Windows, Mac OS X, and Linux, is used [21].

3 Methods

3.1 Erebus: Structure Search and Grafting

Erebus is a protein substructure search server (http://Erebus.dokhlab.org). It searches the entire PDB database for a match to a user-defined substructure, which can be any atoms from the backbone (N, Cα, O) or functional sites (see the following example). Erebus reads ATOM and HETATM records for atom coordinates in the PDB format and will only match atoms in a target substructure with atom names exactly matching those in the query structure [3, 4]. This feature is useful for identifying proteins that have the same catalytic sites, bind to similar small molecules or have the same backbone structures.

As an example, we used the copper-binding site in Cu/Zn superoxide dismutase (SOD1) to prepare the query structure and to find similar metal binding sites in PDB.

1. Create a file containing the atoms for the substructure query in PDB format. In our example these atoms are the Nδ or Nε atoms of H46, H48, H63, and H120 forming the copper-binding site in SOD1 (an example of the file is below). We saved this file as Cu_His.pdb:

ATOM 2 ND1 HIS A 46 11.519 -11.568 8.749 1.00 9.92 N

ATOM 4 NE2 HIS A 48 13.185 -15.110 8.862 1.00 11.95 N

ATOM 6 NE2 HIS A 63 10.975 -13.745 10.609 1.00 8.89 N

ATOM 8 NE2 HIS A 120 12.452 -13.024 6.807 1.00 2.00 N

END

2. Upload the query PDB file to the to Erebus server (Fig. 1).
3. After uploading the query PDB structure the user will have the option to adjust parameters for each atom in the search. Adjustable parameters include:

   'Residue Name': Under the 'Residue Name' column the user can specify the particular residue to match via a dropdown menu or may specify ANY to return matches from any residue.

   'Matching precision' or σ: This is a user-defined precision parameter (*see* Subheading 1.1). Smaller values for σ result in smaller deviations between the query and the targets.

   'Minimum weight' or *W*: This parameter measures how well the query and target structure match (*see* Subheading 1.1). Values for *W* range from 0 to 1, where smaller *W* means a worse match (the RMSD is bigger) and $W = 1$ is an identical match. The user can define the minimum acceptable *W*.

4. For this example Erebus finds over 100 matching structures in the PDB. These structures are ranked based on their RMSD to the query structure (Fig. 2). For each match the user can

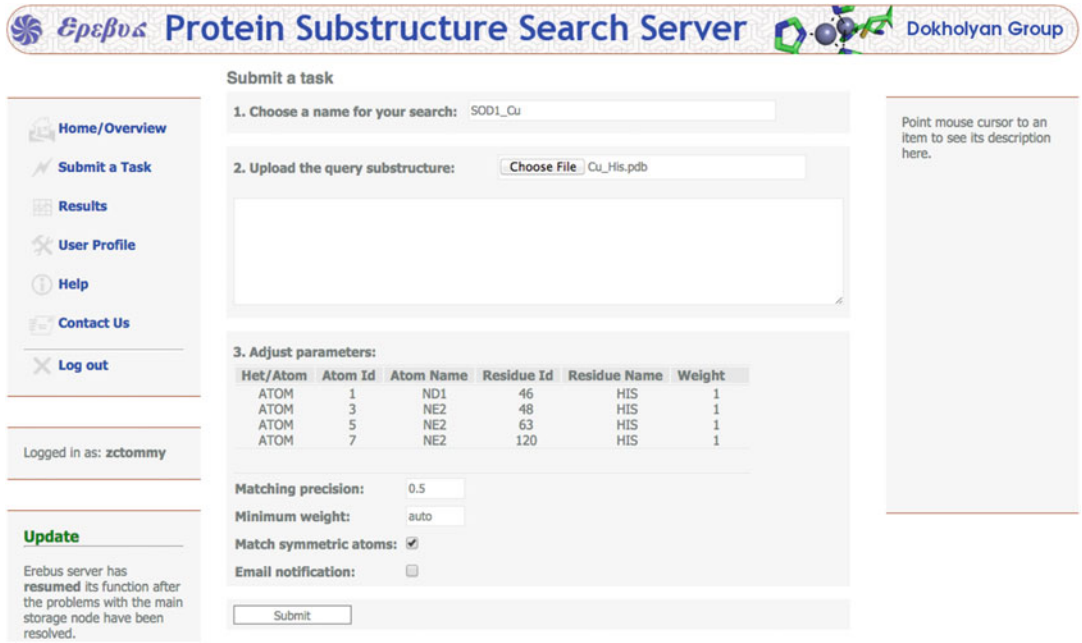

Fig. 1 The Erebus Web interface. Users may either upload a query PDB file or paste query coordinates into the field below in PDB format. Clicking a residue name under the 'Residue Name' column brings up a menu allowing the user to specify the particular residue to match. The 'ANY' selection allows for matches to any residue

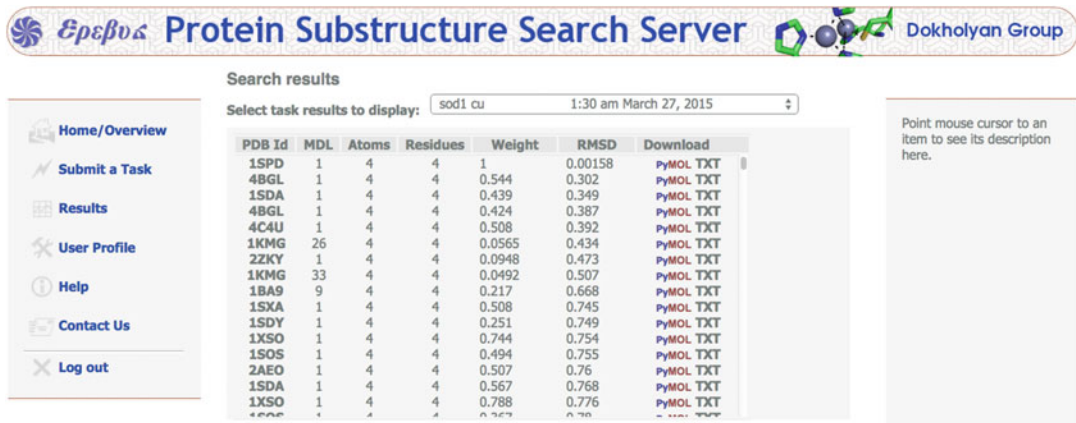

Fig. 2 Results from the Erebus scaffold search. The results are sorted by weight and root mean square deviation (RMSD) to the query structure

download a summary text file (TXT) and a structural model (PyMOL). The structural model file (.pml) is used to visualize the match between query and target structures (Fig. 3).

Erebus identified several protein families containing copper-binding site, including superoxide dismutase, laccase, and multi-copper oxidase. Several iron and zinc binding sites were also found.

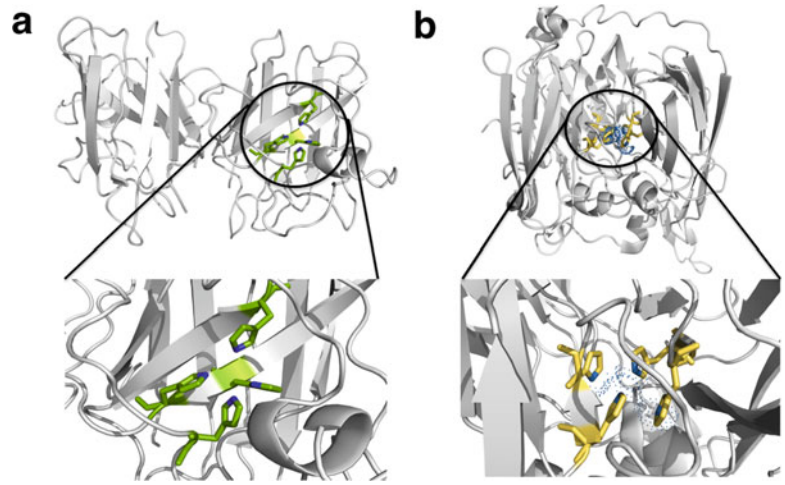

Fig. 3 Comparison of the copper binding site in the query structure (**a**) and one target structure (**b**). The query is based on the crystal structure of SOD1 (1SPD) and the target is a laccase (2XUW). The copper-interacting atoms (Nδ or Nε of histidine) are shown in *blue color*

3.2 Chiron: Preparation of Input Structures: Minimizes Steric Clashes Before Redesign

Chiron estimates the quality of a structure with respect to clashes and minimizes clashes using a series of short discrete molecular dynamics simulations [17, 18]. For homology models and low-resolution crystal structures in which structural artifacts often exist, Chiron is a useful refinement tool.

The Chiron webserver is freely available at http://chiron.dokhlab.org. After logging into the server, the user will be directed to the task submission page by clicking 'Submit Task' in the left panel. The user provides a structural model in PDB format or a PDB ID. During relaxation all backbone and Cβ atoms are constrained. To constrain all side-chain atoms not involved in steric clashes check the 'Constrain Sidechain' box (Fig. 4).

Results of the calculation can be accessed from the 'Home/Overview' page. Chiron outputs a refined structural model ({JOBID}.pdb) and the record of clashes ({JOBID}.py).

As an example, we submitted the crystal structure of Cu/Zn superoxide dismutase (SOD1, PDB ID: 1SPD) to the Chiron server. The resulting files are 12489.pdb and 12489.py. The following steps can be applied to visualize the clashes before and after the refinement:

1. Open 12489.pdb with PyMOL.
2. In the PyMOL command prompt, type 'run {Path}/ {JOBID}.py' and enter, omitting the quotation marks. The path to the .py should be indicated. In the provided example the command is 'run ~/user/12489.py'

The script generates two structural models: i-12489 is the structure of SOD1 before refinement and f-12489 is the structure

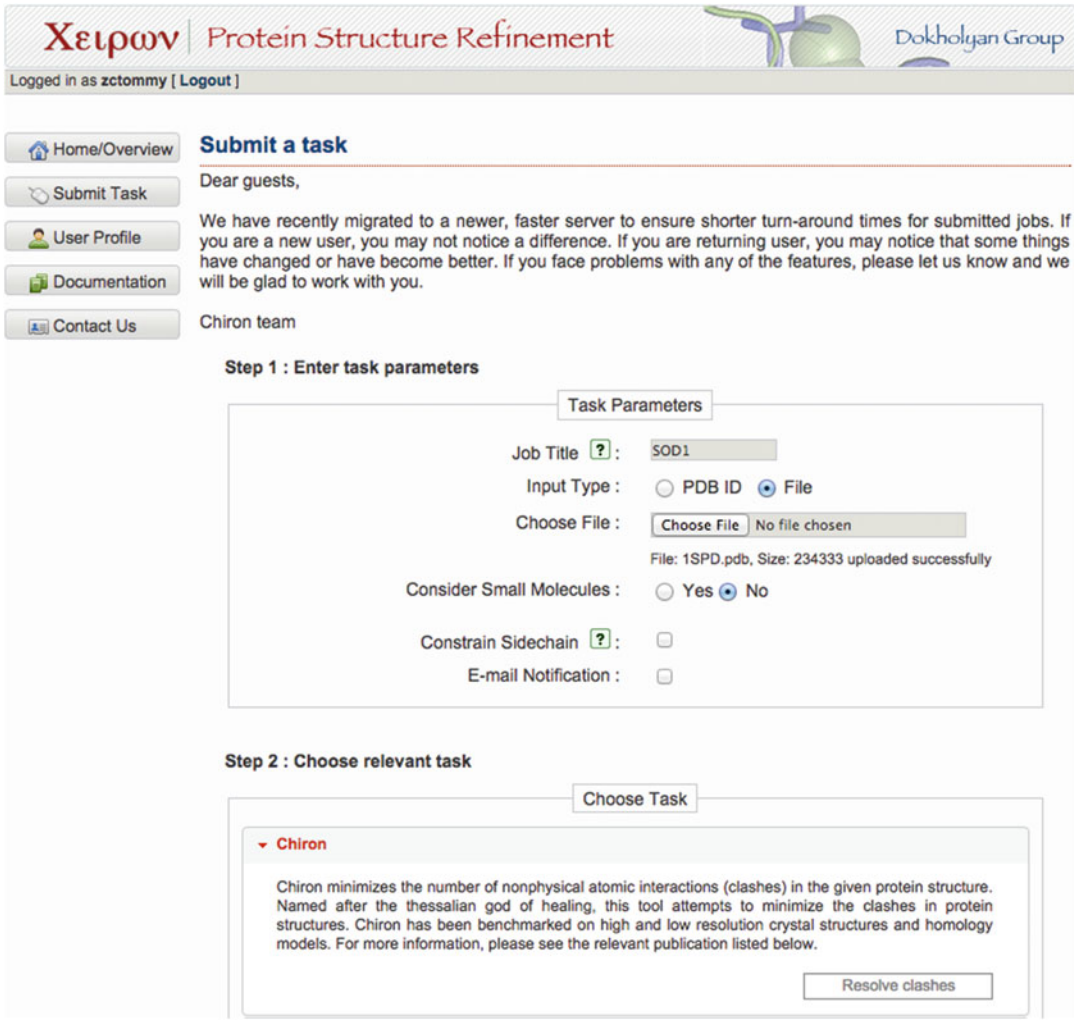

Fig. 4 Image of the Chiron Web interface

after refinement. The steric clashes are represented as color-coded (rainbow spectrum) cylinders of different radii. The clashes with higher repulsion energy are denoted as cylinders of larger radii. For SOD1, Chiron refinement successfully reduced the number of clashes and eliminated all major clashes (Fig. 5).

3.3 Eris: Identify Mutations That Stabilize a Protein Scaffold

Eris has three modules: ddg, scan, and design. The 'ddg' module exhaustively calculates the ΔΔG for individual mutations. The 'scan' module is used to rapidly search for stabilizing single mutations at a specified site. The 'design' function identifies the lowest-energy sequence for a given backbone, which can be a complete protein structure or a user-defined region in the whole structure. In the following section we elaborate the methods and procedures for each module.

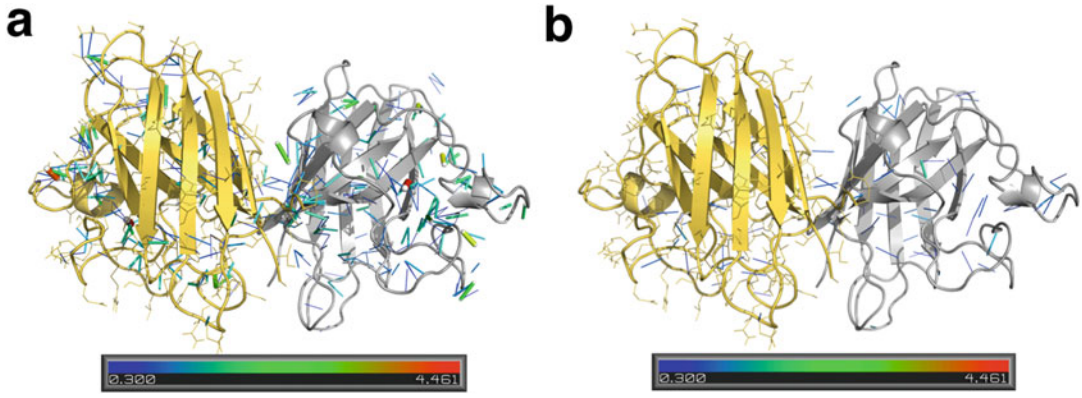

Fig. 5 Resolution of clashes using the Chiron webserver. Structures of SOD1 (PDB ID: 1SPD) are shown (**a**) before and (**b**) after refinement by Chiron. Colored cylinders connect atoms involved in clashes. The color and thickness of the cylinder denote the energy associated with the clash. Comparing panel **a** to panel **b** demonstrates that Chiron greatly reduces the number of clashes, and eliminates the large steric clashes

Table 1
Eris usage flags

Flag	Usage
-w	Specify the working directory
-j	Specify the 'JOBID'
-f	Flexible backbone (Not valid in 'scan' mode)
-r	Provide random seeds in Monte Carlo simulation
-m	Specify the mutation(s) for $\Delta\Delta G$ evaluations in 'ddg' mode
-s	Specify the site(s) in 'scan' mode
-d	Specify the path to a 'DesignTable' in 'design' mode
-h	Print the help, then exit
-v	Verbose output

3.3.1 Installation and Preparation for Input Files

The Eris package is available at http://moleculesinaction.com. Installation instructions are provided with the package (*see* **Note 1**).

The input file for all three modules should be in the PDB format. Currently, the Eris server will only read the first chain of a PDB file (*see* **Note 2**). Eris also renumbers the index of the first residue as 1. The common flags in Eris for command line usage are listed in Table 1.

3.3.2 ddg: An Estimation of ΔΔG for Given Mutations

For a single mutation or multiple substitutions, Eris-ddg repacks the sidechain 20 times using simulated annealing and computes stabilities by averaging all the conformations for both wild type and

mutant. The stability change, ΔΔG, is computed as the difference between the average stabilities of mutant and native structure ($\Delta G_{MUT} - \Delta G_{NAT}$).

As an example, we calculate the stability change induced by Y36E mutation in protein kinase B (PDB ID: 1UNP).

1. Submit the job:

   $ eris ddg –m {MUTATION} –w {DIRECTORY} –j {JOBID} INPUT.pdb

   MUTATION is a comma-delimited string of all mutations to be performed in a particular eris run in the format {native residue}{residue number}{target residue}. For example to change an alanine dipeptide to a serine dipeptide the input would be A1S, A2S.
   In our case we used the following command line and the results were kept in a folder called 'Eris_ddg':

   $ ~/Eris/eris ddg -m Y36E -w ~/Eris_ddg -j 1UNP_Y36E ~/Eris_ddg /1UNP.pdb

2. Analyzing the results:

   A summary of the results can be found in ~/Eris_ddg/eris_ddg.txt, which looks like:

   1UNP_Y36E ddg ~/Eris_ddg/1UNP.pdb Y36E 9.95
   9.95 7.11 0.11 2.84 0 0.81 0.05 0.40 0.46 -4.62 3.37

   The job (JOBID = 1UNP_Y36E) is the 'ddg' calculation for input structural model of 1UNP.pdb. The mutation (Y36E) is the substitution of Tyrosine at position 36 to Glutamic acid. The total ΔΔG equals to 9.95, i.e., destabilizing. In the second line the total stability change and its decomposition are illustrated (*see* **Note 3**).

   Structure files of both native and mutant proteins for each of the 20 rounds of calculation are stored in PDB format in the folder ~/Eris_ddg/1UNP_Y36E.

3.3.3 Scan: Search for Stabilizing Mutations

In the scan module a native amino acid at a specified site is substituted to all other 19 types of amino acids and only the stabilizing substitution ($\Delta\Delta G < 0$) are kept. If positions are not explicitly specified using '-s', then all the residues are scanned.

As an example, we used Eris-scan to find stabilizing mutations at position 37 in protein kinase B (PDB ID: 1UNP).

1. Submit the job:

   $ eris scan –s {SITE} –w {DIRECTORY} –j {JOBID} INPUT.pdb
   SITE is a comma-delimited string of integers specifying the residue positions to be scanned. Residue positions are determined from their order in the PDB file and Eris-scan

renumbers the index of the first residue as 0 ('-sN' means the scan is performed on the N+1 site).

In our case we used the following command line and the results were kept in a folder called 'Eris_scan':

$ ~/Eris/eris scan -s 36 -w ~/Eris_scan -j 1UNP ~/Eris_scan /1UNP.pdb

2. Analyzing the results:

The stabilizing mutations were listed in 'Eris_scan/1UNP/output/ddgStabilizing.dat', which looks like:

K37L -3.04264 -0.0944399 -0.0642849 -1.14021 0 -0.132146...
K37V -2.11104 0.0127475 -0.064736 -1.33434 0 0.0623585...
K37Q -0.111982 -0.878894 0.05282570.354524 0 0.0364916...
K37N -0.817968 0.106138 -0.10984 -0.245153 0 0.0981676...
...

Each line in the ddgStabilizing.dat specifies the stabilizing single mutations (*see* **Note 4**). In the same line, the numbers starting from the second column correspond to the total stability change and its decomposition.

The atomic structures of repacked conformations are stored in the same folder. The calculation results for all 19 substitutions were stored in ddgAll.dat.

3.3.4 Design: Find the Optimal Amino Acid Sequence for the Given Protein Backbone

In the 'design' module, users specify the protein segments to optimize and which subset of amino acids can be used to substitute the original one (polar, hydrophobic or user-defined subsets). The search can be performed using either a fixed backbone protocol (C, O, CA, and N positions fixed during design), or a flexible backbone protocol (allowing small adjustments of the backbone atoms to minimize energy). Eris-design then searches the lowest-energy sequence that satisfies the constraints listed in the design table.

Before submitting the 'design' job, the user should prepare a design table in .txt format. The design table consists of two columns: the first column (Index) specifies the mutation sites and the second column (Keyword) defines the subset of amino acids for substitution.

Values in the index column can be a single integer (m), a set of integers separated by commas (m,n,..), a range defined as m − n (m < n), or a mixture such as (a−b,c,d,e−f). The "DEFAULT" keyword can be used to represent all residues that have not been explicitly specified.

The keyword column takes as an argument one of a list of predefined flags. The flags and definitions are listed in Table 2.

For clarity an example design table for protein kinase B (PDB ID: 1UNP) is shown below:

Table 2
Keywords and definitions for Eris design table

Flag	
ALLAA	All available amino acids and the corresponding rotamers
NATAA	Fixed with native amino acid, but with all its available rotamers
NAROT	Fix the amino acid in its native rotamer but with sub-rotamer allowed
FIXNR	Fix the amino acid in its native rotamer without sub-rotamer motion
POLAR[a]	Polar amino acids and their rotamers
HYDPH[b]	Hydrophobic amino acids
AROMA[c]	Aromatic amino acids
PIKAA	User selected amino acids represented by single letter

[a]POLAR includes: SER, THR, GLN, GLU, ASN, ASP, LYS, ARG, HIS
[b]HYDPH includes: GLY, ALA, MET, VAL, LEU, ILE, PHE, TYR, TRP, PRO, CYS
[c]AROMA includes: PHE, TYR, TRP, HIS

#Index Keyword
DEFAULT NATAA
4-10 ALLAA
102, 112 PIKAA STYWL
50,69-76 HYDPH QEND

This design table will perform all-amino substitutions for residues 4 through 10, substitute residues 102 and 112 with Ser, Thr, Tyr, Trp, and Leu. And substitute residues 50 and 69 through 76 with hydrophobic amino acids. The QEND flag serves to denote the end of the design table

1. Submit the job:

   $ eris design –d {DesignTable} –w {DIRECTORY} –j {JOBID} STUCTURE.pdb

   In our case we used the following command line and the results were kept in a folder called 'Eris_design' (*see* **Note 5**):

   $ ~/Eris/eris design -d ~/DesignTable.txt -w ~/Eris_design -j 1UNP ~/Eris_design/1UNP.pdb

2. Analyzing the results
   The output of Eris-design is a PDB file of redesigned structural model and its free energy. In this module 20 rounds of Monte

Carlo simulations are performed. The results are given as design.run [00-19] and kept in Eris_design/1UNP/design. In each ".run" file, the first 20 lines record the temperature, Monte Carlo acceptance rate, total energy and its decompositions. The following lines are in PDB format so that the user can open it with PyMOL.

For example, design.run00 looks like:

```
0 10 0.763302 222.253 -409.689 371.847 319.997...
1 6.35799 0.70095 120.507 -402.49 196.314 296.97...
...
19 0.101625 0.0279097 -184.007 -464.69 8.64136 316.316...
ATOM 1 N ASP A 1 31.522 1.268 -6.333 1.00 0.00 N
ATOM 2 CA ASP A 1 30.972 2.648 -6.220 1.00 0.00 C
...
ATOM 1245 2HH2 ARG A 119 22.086 -5.596 -26.587 1.00 0.00 H
TER
```

3.3.5 An Online Server for 'ddg' Calculations

A Web-based Eris server for $\Delta\Delta G$ estimation is freely accessible online (http://eris.dokhlab.org). The users can follow the simple procedures listed below to submit their own task after registration:

1. Use the 'Submit a Task' bar on the left to submit the protein structure file. It can be a PDB ID or your own .pdb file. Eris only recognizes the first chain by reading the 'TER' line in a .pdb file.
2. Click on any residue site and choose the amino acids you want to substitute (Fig. 6).
3. Choose 'Fixed Backbone' or 'Flexible Backbone' and choose whether you want to include a pre-relaxation of backbone structure. Pre-relaxation remarkably improves the prediction accuracy when a high-resolution protein structure is not available. Alternatively the users can use Chiron to minimize steric clashes in the input structure (*see* **Note 6**).

4 Notes

1. After installation, typing "eris" without any command line arguments will display the brief help information. Typing "eris −h" will bring more detailed instructions.
2. The Eris webserver only reads the first chain of the provided PDB file. To modify the protein chains, the user can apply the 'alter chain' and 'alter resi' command in PyMOL for this modification (http://www.pymolwiki.org).

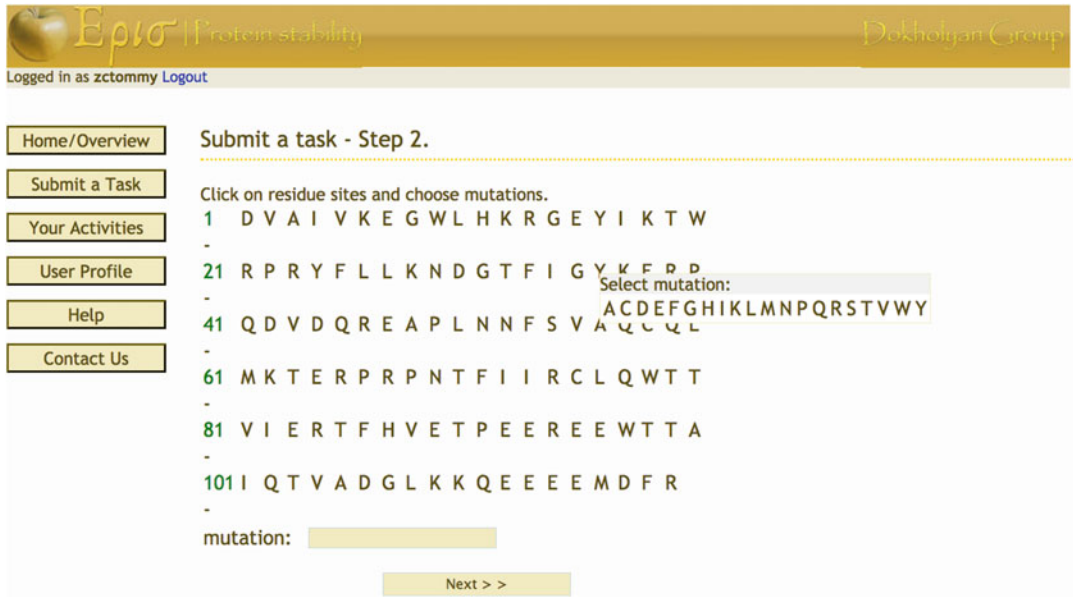

Fig. 6 Image of the Eris Web interface. Clicking on a residue results in a pop-up window allowing the user to select the target residue

3. The energy decomposition of total $\Delta\Delta G$ is composed of ten values. From left to right they are van der Waals attraction, van der Waals repulsion, solvation, backbone hydrogen bonds, backbone–side chain hydrogen bonds, side chain–side chain hydrogen bonds, backbone dependent statistical energy for amino acid, backbone-dependent statistical energy of the rotamer, reference energy of the unfolded states, and the correction for reference energy, respectively [19].
4. Eris-scan does not support flexible-backbone protocol and the Monte Carlo simulation is performed only once. Due to the limitations of the fixed-backbone method and sampling inefficiency, atomic clashes may not be resolved during structure minimization. These clashes can be identified by checking the van der Waals repulsion energy terms in the results. We find that the predictions are relatively more accurate for buried residues than exposed residues.
5. Eris uses a Monte Carlo algorithm for identifying changes in $\Delta\Delta G$. As such there is a certain amount of stochasticity in the results. We suggest that the user run the calculation multiple times to check for convergence in the distribution of energies. Performing multiple rounds of Eris calculation can be performed using the following bash script.

   CUR_DIR=‘pwd‘

   for i in {1..N}; do

```
~/eris ddg -m Y36E -w $CUR_DIR -j 1UNP_Y36E_${i} -r
    $RANDOM $CUR_DIR/1UNP.pdb
done
```

6. If there is difficulty with formatting or running the calculations it is suggested that the user submits the structural model to Chiron first. The server will both reformat the file to be compatible with Eris and fix clashes that could produce problems with the Eris calculations.

Acknowledgment

This work was supported by National Institutes of Health Awards R01GM080742 and R01AI102732.

References

1. Mandell DJ, Kortemme T (2009) Computer-aided design of functional protein interactions. Nat Chem Biol 5(11):797–807. doi:10.1038/nchembio.251
2. Martin L, Stricher F, Misse D, Sironi F, Pugniere M, Barthe P, Prado-Gotor R, Freulon I, Magne X, Roumestand C, Menez A, Lusso P, Veas F, Vita C (2003) Rational design of a CD4 mimic that inhibits HIV-1 entry and exposes cryptic neutralization epitopes. Nat Biotechnol 21(1):71–76. doi:10.1038/nbt768
3. Tinberg CE, Khare SD, Dou J, Doyle L, Nelson JW, Schena A, Jankowski W, Kalodimos CG, Johnsson K, Stoddard BL, Baker D (2013) Computational design of ligand-binding proteins with high affinity and selectivity. Nature 501(7466):212–216. doi:10.1038/nature12443
4. Schulze H, Vorlova S, Villatte F, Bachmann TT, Schmid RD (2003) Design of acetylcholinesterases for biosensor applications. Biosens Bioelectron 18(2-3):201–209
5. Sia SK, Kim PS (2003) Protein grafting of an HIV-1-inhibiting epitope. Proc Natl Acad Sci U S A 100(17):9756–9761. doi:10.1073/pnas.1733910100
6. Hao J, Serohijos AW, Newton G, Tassone G, Wang Z, Sgroi DC, Dokholyan NV, Basilion JP (2008) Identification and rational redesign of peptide ligands to CRIP1, a novel biomarker for cancers. PLoS Comput Biol 4(8), e1000138. doi:10.1371/journal.pcbi.1000138
7. Rothlisberger D, Khersonsky O, Wollacott AM, Jiang L, DeChancie J, Betker J, Gallaher JL, Althoff EA, Zanghellini A, Dym O, Albeck S, Houk KN, Tawfik DS, Baker D (2008) Kemp elimination catalysts by computational enzyme design. Nature 453(7192):190–195. doi:10.1038/nature06879
8. Drakopoulou E, Zinn-Justin S, Guenneugues M, Gilqin B, Menez A, Vita C (1996) Changing the structural context of a functional beta-hairpin. Synthesis and characterization of a chimera containing the curaremimetic loop of a snake toxin in the scorpion alpha/beta scaffold. J Biol Chem 271(20):11979–11987
9. Azoitei ML, Correia BE, Ban YE, Carrico C, Kalyuzhniy O, Chen L, Schroeter A, Huang PS, McLellan JS, Kwong PD, Baker D, Strong RK, Schief WR (2011) Computation-guided backbone grafting of a discontinuous motif onto a protein scaffold. Science 334(6054):373–376. doi:10.1126/science.1209368
10. Hearst DP, Cohen FE (1994) GRAFTER: a computational aid for the design of novel proteins. Protein Eng 7(12):1411–1421
11. Zhang J, Grigoryan G (2013) Mining tertiary structural motifs for assessment of designability. Methods Enzymol 523:21–40. doi:10.1016/B978-0-12-394292-0.00002-3
12. Shirvanyants D, Alexandrova AN, Dokholyan NV (2011) Rigid substructure search. Bioinformatics 27(9):1327–1329. doi:10.1093/bioinformatics/btr129
13. Yin S, Ding F, Dokholyan NV (2007) Eris: an automated estimator of protein stability. Nat Methods 4(6):466–467. doi:10.1038/nmeth0607-466
14. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE (2000) The Protein Data Bank. Nucleic Acids Res 28(1):235–242

15. Hooft RW, Vriend G, Sander C, Abola EE (1996) Errors in protein structures. Nature 381(6580):272. doi:10.1038/381272a0
16. Ramachandran S, Kota P, Ding F, Dokholyan NV (2011) Automated minimization of steric clashes in protein structures. Proteins 79 (1):261–270. doi:10.1002/prot.22879
17. Ding F, Tsao D, Nie H, Dokholyan NV (2008) Ab initio folding of proteins with all-atom discrete molecular dynamics. Structure 16 (7):1010–1018. doi:10.1016/j.str.2008.03.013
18. Dokholyan NV, Buldyrev SV, Stanley HE, Shakhnovich EI (1998) Discrete molecular dynamics studies of the folding of a protein-like model. Fold Des 3(6):577–587. doi:10.1016/S1359-0278(98)00072-8
19. Ding F, Dokholyan NV (2006) Emergence of protein fold families through rational design. PLoS Comput Biol 2(7), e85. doi:10.1371/journal.pcbi.0020085
20. Yin S, Ding F, Dokholyan NV (2007) Modeling backbone flexibility improves protein stability estimation. Structure 15(12):1567–1576. doi:10.1016/j.str.2007.09.024
21. The PyMOL Molecular Graphics System, Version 1.7.4 Schrödinger, LLC