Published online 19 August 2010 *Nucleic Acids Research, 2010, Vol. 38, No. 20* ***e186***
doi:10.1093/nar/gkq726

Capturing, sharing and analysing biophysical data from protein engineering and protein characterization studies

Damien Farrell, Fergal O'Meara, Michael Johnston, John Bradley, Chresten R. Søndergaard, Nikolaj Georgi, Helen Webb, Barbara Mary Tynan-Connolly, Una Bjarnadottir, Tommy Carstensen and Jens Erik Nielsen*

Centre for Synthesis and Chemical Biology, School of Biomolecular and Biomedical Science, UCD Conway Institute, University College Dublin, Belfield, Dublin 4, Ireland

Received June 9, 2010; Revised July 15, 2010; Accepted July 30, 2010

ABSTRACT

Large amounts of data are being generated annually on the connection between the sequence, structure and function of proteins using site-directed mutagenesis, protein design and directed evolution techniques. These data provide the fundamental building blocks for our understanding of protein function, molecular biology and living organisms in general. However, much experimental data are never deposited in databases and is thus 'lost' in journal publications or in PhD theses. At the same time theoretical scientists are in need of large amounts of experimental data for benchmarking and calibrating novel predictive algorithms, and theoretical progress is therefore often hampered by the lack of suitable data to validate or disprove a theoretical assumption. We present PEAT (Protein Engineering Analysis Tool), an application that integrates data deposition, storage and analysis for researchers carrying out protein engineering projects or biophysical characterization of proteins. PEAT contains modules for DNA sequence manipulation, primer design, fitting of biophysical characterization data (enzyme kinetics, circular dichroism spectroscopy, NMR titration data, etc.), and facilitates sharing of experimental data and analyses for a typical university-based research group. PEAT is freely available to academic researchers at http://enzyme.ucd.ie/PEAT.

INTRODUCTION

The average protein is a polymer consisting of at least 100 amino acid residues, which fold into a dynamic 3D structure stabilized by thousands of atomic interactions. The amino acid sequence of many proteins has been optimized to produce a specific 3D structure able to perform highly specialized functions that are needed for the survival of the host organism. A significant fraction of biological and biotechnological research efforts are aimed at dissecting, optimizing or modulating the functions of proteins by changing their amino acid sequence using protein engineering techniques such as site-directed mutagenesis (1), directed evolution techniques (2) and computer-aided protein design (3).

Protein engineering techniques have found a large number of uses ranging from routine tasks such as the identification functional residues and investigations of the roles of individual amino acid residues, to more spectacular applications such as determining the folding nucleus of proteins (4), optimizing enzymatic performance (5), designing novel proteins and re-designing proteins to give them novel or altered functional properties (6,7). Much of this success stems from the development of excellent computer algorithms for the analysis and optimization of protein 3D structures, and especially in the areas of protein stability (8), protein folding (9,10) and protein interactions (11–13) have algorithms reached a high level of sophistication and agreement with experiments. In these areas, progress has been boosted by the availability of large, high-quality experimental datasets of from databases such as the PDB (14), ProTherm (15), BindingDB (16), BIND (17), BRENDA (18) and PPD (19). Development of algorithms for predicting other features such as catalytic turnover rates, protein solubility, protein pH-stability profiles, protein NMR titration curves and protein dynamics has been hampered by the lack of large experimental datasets in a digital form for benchmarking structural bioinformatics algorithms. Indeed, even the existing databases mentioned above are relatively scarcely populated considering the amount of data

*To whom correspondence should be addressed. Tel: +353 1 716 6724; Fax: +353 1 716 6898; Email: jens.nielsen@ucd.ie generated on proteins and protein characteristics. There are, for example, almost no digitized datasets available on the effects of point mutations on enzymatic activity despite more than 25 years of enzyme engineering and countless numbers of published research papers.

There is no lack of biological database solutions for storing experimental data as shown by the table of contents in the annual database issue of *Nucleic Acids Research*, and large amounts of data are produced and published every year by the ever-expanding community of biological research groups working on proteins and enzyme engineering. The scarcity of digitized experimental data on mutant enzymes, is thus neither due to lack of databases or lack of data, but instead due to a low benefit/cost ratio associated with database deposition. There is no requirement to deposit data before publishing research papers as it is the case for protein structures, submission instructions and protocols are often hard to find or non-existent, and there are no immediate benefits to the research group other than the questionable pleasure of being able to find one's data on the internet.

Although, recent years have seen large improvements in the data capture efficiency for genomics projects and system biology efforts, the main data generating entity in modern biological research remains the single-PI research laboratory. In many laboratories, data capture and database deposition is seen as a largely unrewarding process, and information on the effects of point mutations on protein function are often kept in spreadsheets, in laboratory notebooks or simply recorded in publications. The graduation of PhD students and the departure of post-doctoral workers and technicians often results in the loss of data, with the consequence that experiments sometimes have to be repeated and knowledge regenerated. Although a Lab Information Management System (LIMS) (20–26) could solve many of these problems, the cost, administration overhead and lack of short-term benefits often prevent PIs from implementing a LIMS solution. In addition most LIMS are rigidly designed to be used in assay laboratories and provide few, if any, analysis tools.

Typically primers are designed and sequences verified using DNA manipulation programs, gel pictures are obtained using digital cameras, kinetic and stability data are obtained using software connected to spectrophotometers and calorimeters, and finally a nonlinear fitting program is used to analyze kinetic and stability data. Information is thus available in digital form in all stages of a typical protein engineering project, and data capture should therefore be readily achievable. Our object is to integrate the data deposition steps with the analysis tools in one application.

Integrating data analysis and data capture

There has been progress in recent years toward open standards and protocols for interchange of biological data, such as the use of the XML format (http://www.w3.org/TR/xml) for easy data interchange. Such standard formats are being used by, for example, the Collaborative Computing Project for NMR (CCPN) (27) and the Structural Genomics Knowledgebase (SGKB) (28). These projects are designed to provide standard schema or data models for sharing across groups. At the same time, software tools that do significant analysis almost always need to reform the data into internal data structures. This internal format should be as convenient for analysis as possible as long as import/export tools are able to get the data in and out of the software for others to utilize.

In terms of end user tools, the BioBuilder project (29), has demonstrated the use of purely open source tools to build a web application platform for creating custom biological databases. Web-based LIMS, such as 'HalX' (22) have obvious advantages, as they require no installation and are easy to maintain. These web-based solutions, however, usually consist primarily of data submission forms with varying degrees of complexity and totally lack the flexibility of a desktop application. Sesame (30) moves beyond a simple LIMS and is designed for moving data around inside large proteomics projects. We have taken an approach with the emphasis on storage and analysis in a single end user application, specifically aimed at protein engineering and biophysical characterization of proteins.

Our solution for protein characterization and protein/enzyme engineering projects is to offer researchers an application that contains functions for import, secure storing and analyzing of experimental data, with an object database back-end. The object database means we do not have to worry about translation in and out of external database schema. We instead provide export tools to convert data to remove standard formats when required.

By integrating data analysis with data capture, we hope to incentivize researchers to store their data electronically and thus facilitate database deposition. We have constructed Protein Engineering Analysis Tool (PEAT) to be used at almost every stage of a typical protein engineering experiment. PEAT thus contains functions for designing point mutations, constructing PCR primers, keeping track of freeze stocks, capturing primary experimental data and analyzing the effects of point mutations in the context of the protein 3D structure. PEAT furthermore enables multi-user data sharing, a change history providing simple transaction rollback and provides a rudimentary web-interface for publishing of data online.

MATERIALS AND METHODS

Software architecture

PEAT is written in the python (http://www.python.org) programming language. The graphical user interface (GUI) for the client application uses the Tk/Tcl toolkit. Database functionality is provided by the Zope Object Database (ZODB) (http://docs.zope.org/zodb/zodbguide/index.html) which is a mature object database (31) originally designed to be used with the Zope web framework. The application client can be used to open single-user local databases or with multi-user functionality using remote storages. In the stand-alone regime all data is stored locally in a project file and can be modified and saved without a network connection. PEAT is available on Linux, Mac OS X and Windows operating systems. Several open source python libraries are utilized inside PEAT: 'numpy' for numerical processing, 'ZODB3′ for the database and 'matplotlib' for plotting.

Data format and storage

Much of the internal data management within PEAT is based around the Python dictionary data structure (32). Python dictionaries are basically collections of objects indexed by arbitrary keys. The ZODB persists the data structures directly to disk (what is called a 'storage' in the ZODB terminology) either locally or remotely. The storages are 'pluggable'; i.e. the database should behave the same way no matter what type of storage is used. PEAT can currently utilize three types of storage:

- FileStorage: Saves the database to a local file on disk.
- ClientStorage (ZEO): Basic network storage that sends data to a TCP/IP server.
- RelStorage (default): Enables the backing store to be a relational database system (RDBMS) such as MySQL.

The two latter remote stores allow multi-user access. RelStorage (http://pypi.python.org/pypi/RelStorage) was chosen as the default remote backend for PEAT since it allows for better authentication and security features than ZEO.

Binary files (such as images, pdf, zip or other media files) can be stored in the database by the use of 'blobs'. The files are not added to the database directly, but stored in the file system. PEAT includes a column type 'file' specifically designed for this. It is also possible to add files from inside the Labbook. For remote Databases using the ZEO backend any files you add will be available to other users connecting to the DB. The RelStorage implementation is currently more limited in that the files need to be on a shared file system, or users will not be able to see each others files.

Multiuser data sharing

A primary challenge in multi-user data sharing is avoiding conflicts between concurrent users of the same data. When there is more than one person working on a project, at some point two people will simultaneously alter the same record/field without being aware of the other users change. In the database world, this is known as 'concurrency handling' and is usually implemented by a form of transaction locking. Only one user at a time can update that field in the database record for example. The other solution is to allow users to simultaneously access the field, and provide warnings where conflicts arise. This is sometimes called 'optimistic concurrency control'.

The ZODB uses the latter approach, i.e. it's mode of operation is 'non-locking'. That means that if two users are accessing the same record item at the same time, nothing prohibits both of them from editing that item. Whilst saving their changes, users will be notified if conflicts have arisen because someone else has changed the same record/field item in the same protein since their last update. If there is a conflict, users are required to manually intercede and resolve the conflict in the context of the particular data. PEAT provides the user with the choice of reverting their changes and then accepting or overwriting the conflicting changes to the records/fields in question.

Conflicts are minimized if users save and refresh their view often. Users can optionally provide comments annotating their change. If users wish to check if other workers have made changes related to a particular record, they can check the 'change log'. This gives comments, name of user and time/date details. This method generally works quite well, assuming a reasonable degree of organization between and within groups collaborating on the same data concurrently.

At any time a user can undo any changes since their last save. In addition previously committed transactions may be rolled back. However this undo feature should not be relied upon too heavily since subsequent changes to the same records may prevent undo-ing past a certain revision.

Access control/authentication

The RelStorage implementation utilizes the MySQL authentication mechanisms. Therefore users can be added and given access to individual databases, with read/write or read-only privileges. The system allows for encrypted transmission of data with SSL additional security, which MySQL supports.

Database size and memory usage

There is no hard upper limit on database size in principle—except the size of the filesystem and RAM (for keeping the object index). Single filestorages of 20 GB and more with ZODB are common practice. Loading a large number of records, such as during a search, will use more memory. The more memory available the better, but the application keeps a limit on the number of objects in memory to prevent swapping. This can be managed on a per database basis by adjusting the object cache to a value appropriate to your memory and database structure. For example, a database might have small record sizes, but a very large number of records, in this case you can set the object cache high (~1000). We have tested PEAT with databases of >40 000 records without problems.

Since the ZODB retains a history of previous transactions, a large amount of redundant information may end up being stored. To avoid this, the databases can be periodically 'packed' to remove this redundant information from before a certain time. The transaction history before this time is lost.

Server Setup

RelStorage is a python library that utilizes MySQL as the data store, giving the security and reliability of a mature RDBMS. Therefore setup of your own server is essentially no different to setting up a MySQL server. We recommend using 'phpMyAdmin' for web-based administration of the server, making it easy to add databases, users and so on. Setup is described in more detail in our help page at http://enzyme.ucd.ie/main/index.php/PEAT_Server_Setup. Readers interested in trying PEAT, but who do not wish to host their own dataset, can contact the authors and we will gladly host the project for you.

Availability and requirements

Project name: PEAT.
Project home page: http://enzyme.ucd.ie/main/index.php/PEAT_DB.
Operating system(s): Unix, Windows, OS X.
Programming language: Python.
License: free for academic use, commercial license for industrial users.

RESULTS

PEAT is a workbench application designed to aid researchers in analyzing their data quicker and in facilitating efficient record-keeping to speed up productivity. In the following we describe three typical everyday activities encountered in almost any protein engineering laboratory, and describe how PEAT will contribute to effectivizing the process. We continue to provide a more detailed description of PEAT and its components, and finally we provide a brief outlook on the task of data management in modern biology.

Rationalizing protein engineering results

Protein engineering and protein optimization projects are guided by monitoring the mutation-induced change in several protein biophysical characteristics such as enzymatic activity, thermal stability, binding affinities and the pH dependence of these properties. Observing the changes in such characteristics for several mutant proteins allows the scientist to obtain an intuitive understanding of how a certain protein/enzyme works, and further mutations are constructed based on this intuitive understanding until a mutant protein with the desired characteristics are obtained. Such a process relies on quick access to different types of primary experimental data in an easy interpretable way, and every protein engineer familiar with a situation where sheets of paper with enzyme kinetics data, circular dichroism (CD) scans and differential scanning calorimetry (DSC) melting curves are scattered across a table while the project team tries to make sense of the experimental data using a multicolored 3D protein structure on a computer screen. In essence protein engineering is all about understanding the connection between changes in sequence, changes in structure and changes in biophysical characteristics and the better and faster we are at analyzing the experimental data and concluding on the inner workings of the target protein, the better we will be at engineering novel proteins with altered characteristics.

Figure 1 shows how PEAT handles the data analysis in a protein engineering project elegantly; a specially designed module is used to perform visual comparisons across mutants. A click on a mutant protein highlights the mutated residues and displays the modeled structure (comparative modeling is performed on the fly) in Yasara (http://www.yasara.org), while the desired mutant protein biophysical characteristics are shown with the characteristics of the wild type. Calculated changes in ΔG_{fold}, ΔG_{bind} and select pKa values can be shown provided in a table next to the experimental characteristics provided that PEAT_SA (M. Johnston *et al.*, in preparation) has been installed and set up. A plugin for PEAT_SA is designed to submit and retrieve jobs from the server and integrate the results into PEAT. More detailed information on this plugin and its usage is available on the PEAT website.

Fitting experimental data

Recording experimental data with a high fidelity is a task that often takes months for inexperienced researchers to master. Choosing the correct equation for fitting the data and assessing the error inherent in the fitted parameters can take years, and is a skill that escapes many due to the complicated statistical tests associated with the fitting procedure. PEAT assists in this task by including a large number of equations for fitting protein biophysical characteristics, by providing an automatic f-test-based selection of the correct fitting model, and by linking derived data (e.g. enzymatic reaction rates in a Michaelis–Menten plot) to the primary experimental data (product concentration versus time) and by propagating experimental errors throughout the fitting process.

Error assessment on individual data sets can be performed by the click of a button, and is implemented by randomly varying individual data points within a specified range and refitting the equation. The resulting average values of the fitted parameters are reported along with their standard deviations.

Managing research projects

Research projects in modern laboratories typically involve many mutant proteins, several data-generating experimental techniques, students and postdocs plus one or more collaborators, often off-site or abroad. Keeping track of progress in such a project is challenging because of the need to share information with all project members, to enforce deadlines and to make project decisions based on all available information. PEAT makes it possible for all project members to deposit their primary experimental data in an online project database. Information such as expression status, DNA sequences, gel pictures, relevant publications, kinetic assays etc. can be uploaded and shared with the project team thus allowing the lead PI to efficiently monitor the progress of the project. A further advantage is that experimental data can be fitted directly in the PEAT module Ekin, thus allowing both data and the physical interpretation of these to be communicated efficiently. The deposition of data in the online project furthermore ensures that data is preserved when team members graduate or leave.

PEAT components

The three usage scenarios above demonstrate the general purpose value of PEAT. In the following we provide a more in-depth description of the individual modules of

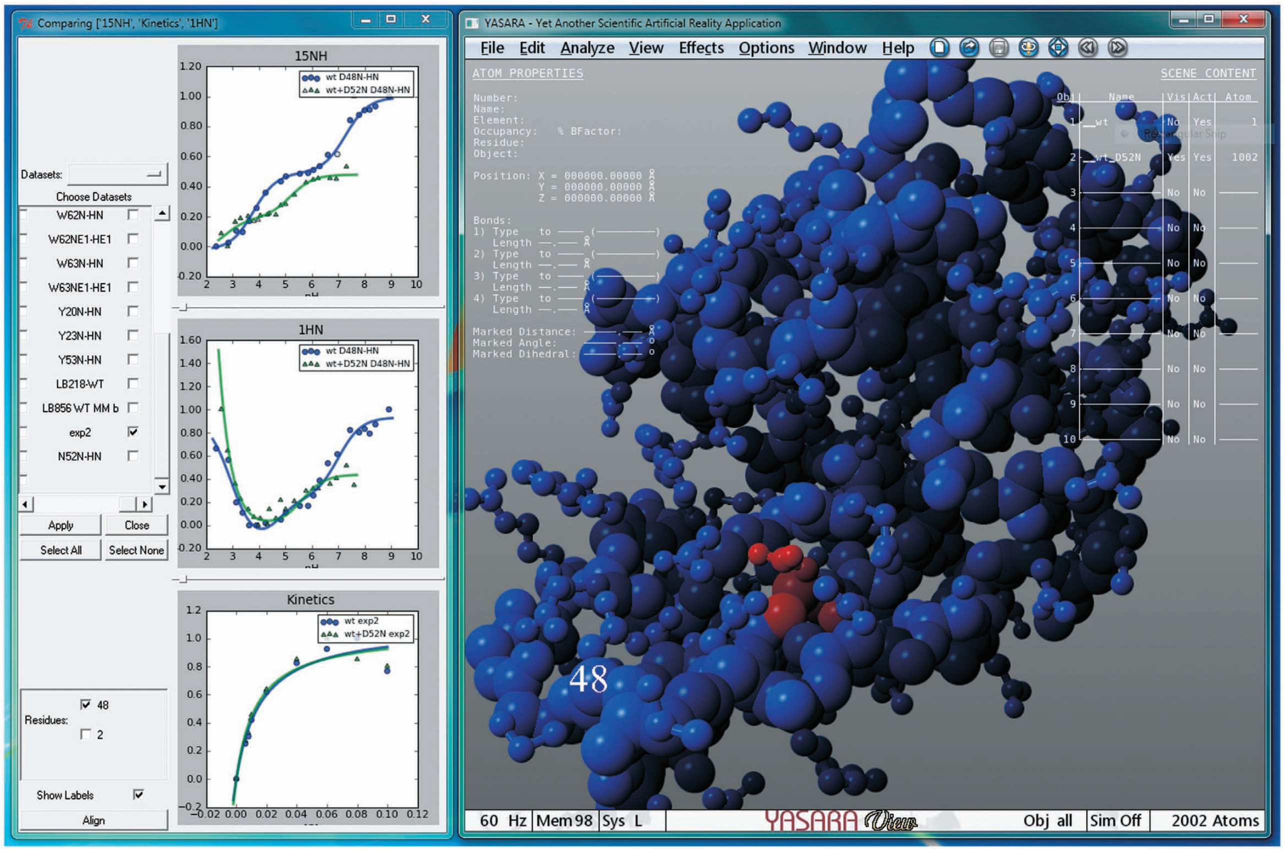


Figure 1. PEAT may be used to visualize changes in structure alongside changes in various biophysical characteristics. The figure shows the structure of HEWL and its D52N mutant (mutated residue marked in red). Plots of three experimental properties are displayed alongside: 15N and 1HN NMR titration curves for the ASP48 residue and kinetics data. In each case curves for wild type and mutant are overlayed. Plots can be changed interactively to show all other available datasets; if the structure is displayed, data labels are interpreted as residue numbers and marked for easy identification.

PEAT along with a description of the more technical aspects of the program.

User Interface and functionality

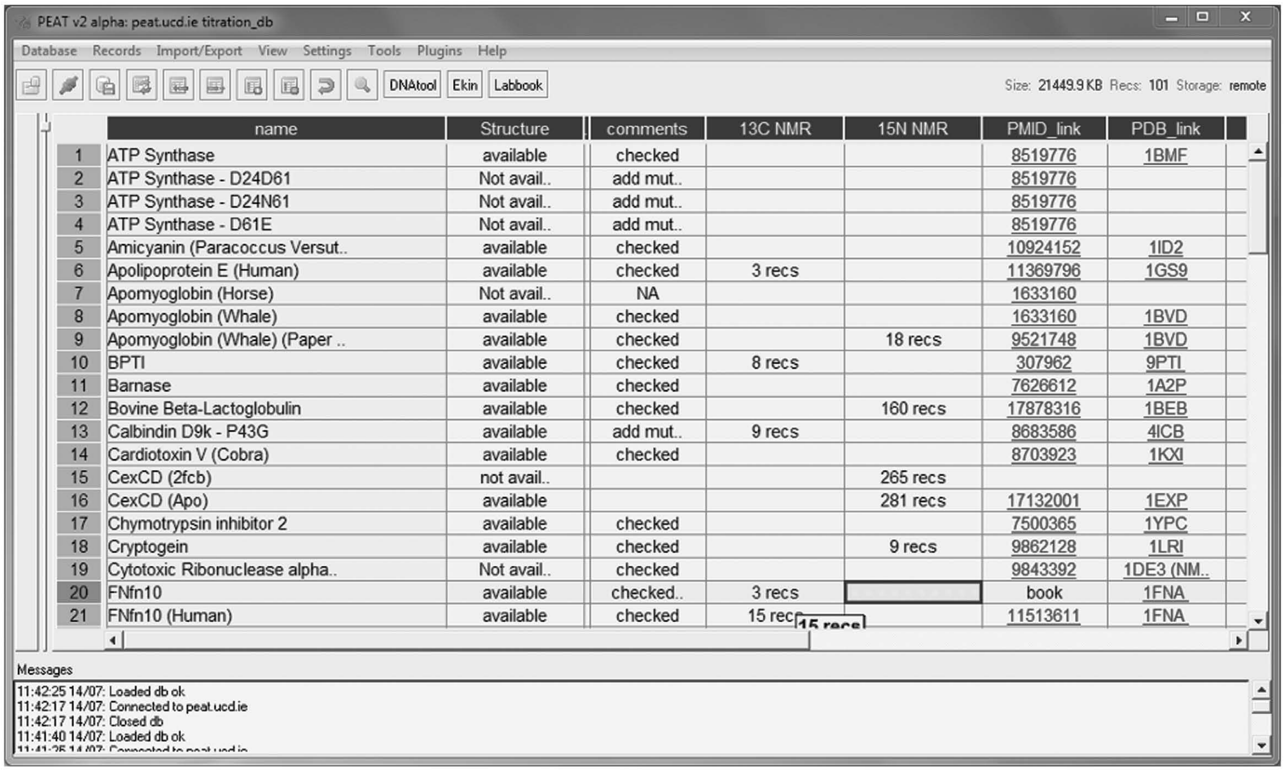
The main screen in PEAT displays a spreadsheet view where each row represents a single protein, while the information on measured biophysical characteristics are kept in individual columns, the protein sequence and 3D-structural information is also accessible from this table (Figure 2). Experimental data can added to the spreadsheet by direct import, import from within a sub-module or manual text entry into the cells. The predefined data types currently supported by PEAT are: Ekin format experimental data, arbitrary binary files (images, word documents, PDF files etc.), hyperlinks, simple tables (using the Labbook application), and a field type for number/text values. For the predefined experimental data types PEAT provides the 'Ekin' module that allows the user to fit primary experimental data (titration curves, kinetics data, CD denaturation curves) to a theoretical model and extract physico-chemical quantities (K_{cat}, K_m, pKa values, etc.) of interest. Once the column for the particular experimental data type has been created the data can be entered for each protein. Clicking on a cell given in a given column provides the functions specific to its data type.

The various PEAT modules are oriented at addressing specific tasks encountered during a typical protein engineering project. 'DNAtool' is used for primer design and for confirming DNA sequencing results. The 'Labbook' is geared to handling tabulated data and other simple record-keeping tasks. The fitting of raw data from biophysical characterization experiments is carried out with 'Ekin'. Figure 3 gives an overview of the PEAT components.

Functional modules

PEAT consists of a main database window and four sub-modules: DNAtool, Labbook, Ekin and a module for performing graphical comparisons across records. In addition functionality can be added in the form of plugins. The main database window functions mainly as an advanced spreadsheet that allows for storage and access to the sub-modules. The main functionality in PEAT thus lies in the sub-modules that we describe subsequently.

DNAtool. 'DNAtool' displays the nucleotide sequence alongside with the protein's amino acid sequence and


PEAT v2 alpha: peat.ucd.ie titration_db
Database Records Import/Export View Settings Tools Plugins Help
DNAtool Ekin Labbook
Size: 21449.9 KB Recs: 101 Storage: remote

	name	Structure	comments	13C NMR	15N NMR	PMID_link	PDB_link
1	ATP Synthase	available	checked			8519776	1BMF
2	ATP Synthase - D24D61	Not avail..	add mut..			8519776	
3	ATP Synthase - D24N61	Not avail..	add mut..			8519776	
4	ATP Synthase - D61E	Not avail..	add mut..			8519776	
5	Amicyanin (Paracoccus Versut..	available	checked			10924152	1ID2
6	Apolipoprotein E (Human)	available	checked	3 recs		11369796	1GS9
7	Apomyoglobin (Horse)	Not avail..	NA			1633160	
8	Apomyoglobin (Whale)	available	checked			1633160	1BVD
9	Apomyoglobin (Whale) (Paper ..	available	checked		18 recs	9521748	1BVD
10	BPTI	available	checked	8 recs		307962	9PTI
11	Barnase	available	checked			7626612	1A2P
12	Bovine Beta-Lactoglobulin	available	checked		160 recs	17878316	1BEB
13	Calbindin D9k - P43G	available	add mut..	9 recs		8683586	4ICB
14	Cardiotoxin V (Cobra)	available	checked			8703923	1KXI
15	CexCD (2fcb)	not avail..			265 recs		
16	CexCD (Apo)	available			281 recs	17132001	1EXP
17	Chymotrypsin inhibitor 2	available	checked			7500365	1YPC
18	Cryptogein	available	checked		9 recs	9862128	1LRI
19	Cytotoxic Ribonuclease alpha..	Not avail..	checked			9843392	1DE3 (NM..
20	FNfn10	available	checked..	3 recs		book	1FNA
21	FNfn10 (Human)	available	checked	15 recs		11513611	1FNA

```
Messages
11:42:25 14/07: Loaded db ok
11:42:17 14/07: Connected to peat.ucd.ie
11:42:17 14/07: Closed db
11:41:40 14/07: Loaded db ok
```

Figure 2. The PEAT main window is organized much like a spreadsheet with rows holding the data for a single protein. Each column is of a specific data type and right-clicking on the cell will provide the options specific to that field type. For typical protein studies each row/record is a protein, having a sequence, structure or other data associated it with it. The screenshot presented here is a view of the contents of Titration_DB (39).

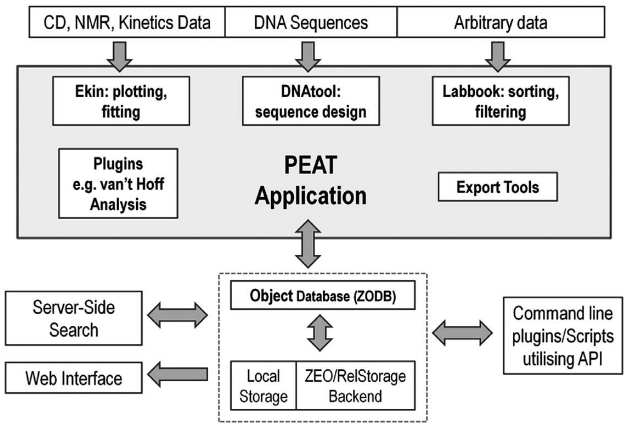


Figure 3. How the various components relate to each other in PEAT. The 'backend' database/server components inside the dashed box are not directly seen by the end-user and can in principle be replaced by another data storage/sharing solution without substantially affecting the other components. Plugins can be flexibly developed with knowledge of python. Developing a set of export tools would in principle be necessary for data sharing with standard databases.

allows for manipulation of the DNA sequence by applying primers to the nucleic acid sequence. DNA sequences can be import from files in the PIR, FASTA, GenBank and BSML formats.

Both normal and mutagenic PCR primers can be designed automatically in 'DNATool', and the binding site on the DNA strand is calculated on-the-fly, and presented with primer T_m, hairpin and primer–dimer scores.

Designing mutagenic PCR primers with DNAtool. The application allows the quick and easy design of mutations in any nucleotide sequence. Mutagenic primers are optimized for number of mismatches, melting temperature, hairpin score and primer-dimer score. Optional silent mutations can be generated that insert or remove unique restriction sites. All primers generated are stored in a project-specific primer database that can be recalled for subsequent viewing. Any primer can be aligned with a DNA sequence by a simple click and all primers can be manipulated by hand, to optimize length, melting temperature, or restriction site changes.

DNAtool furthermore allows for fast analysis of sequences of mutant clones. A newly sequenced mutant gene can be uploaded and aligned with the wild-type sequence and the mutagenic primer simultaneously as illustrated in Figure 4.

The nucleotide sequence can be searched for specific motifs, and all restriction digest sites can be displayed in a table, which can be sorted alphabetically by restriction enzyme name, or by number of cuts that appear in the sequence.

Labbook. The 'Labbook' is a general tool for storing information that typically is stored in spreadsheets. In a protein engineering context such information typically consists of the location of freeze stocks, expression batch yields and status, purification procedures, gel pictures, laboratory stocks of chemicals, ordering status, etc. Labbook records can be filtered, sorted and searched. Data can be imported into Labbook from text files or entered manually. As in the main PEAT table, external binary files (such as images) can be imported and annotated inside a Labbook table. The Labbook allows multiple sheets to be created in every instance of the application, and individual cells in the Labbook can be Labbook tables themselves.

Ekin. In PEAT, experimental data is recorded by the 'Ekin' module (Figure 5), which contains the following predefined data modes: general, protein stability, Michaelis–Menten kinetics, pKa system, pH activity profile, pH stability profile, and NMR titration. Although any type experimental data can be loaded into Ekin, the predefined modes are tailored to a specific type of experiment to allow specific functionality and model selections to be presented to the user. In all cases the Levenberg–Marquart algorithm (33) is used for non-linear curve fitting. New fitting models can currently only be added by changes to the code, but we intend to provide a user-end framework for designing new models in the near future

All experimental data must be associated with a specification of the experimental conditions in order to be useful. PEAT implements limited checks for the presence of experimental conditions before allowing the data to be saved to the database. Each of the predefined data types contains experimental 'meta data' and reference information specific to it (such as temperature, concentration, pH, task status, user and time stamp).

Ekin contains the following pre-defined modes:

Protein stability. Imported protein stability measurements of thermal denaturation are routinely recorded by CD spectroscopy, DSC or measurements of residual enzymatic activity. Ekin provides appropriate fitting models for the standard equations for deriving the required parameters (i.e. melting temperature, T_m). Also present is a more specialized plugin for performing van't Hoff analysis and extraction of enthalpy (ΔH_{vh}) values using a selection of methods from the literature (34–36).

Full wavelength CD scans can furthermore be viewed (ellipticity versus wavelength) although we currently do not provide tools for assessing the α-helical or β-sheet content of proteins.

Michaelis–Menten kinetics. Reaction kinetics measurements recording the reaction rate versus concentration can be fitted to the Michaelis–Menten model to determine K_m and V_{max}. Competitive and uncompetitive inhibition models are also available.

NMR titration. Single residue pKa values of titratable groups or binding constants can be determined from NMR recorded chemical shifts of carbon or nitrogen atoms in the titratable group. NMR data can be supplied to PEAT in form of a .csv text file containing the chemical shifts or imported from data saved in the format of the NMR peak assignment program SPARKY (http://www.cgl.ucsf.edu/home/sparky/). An additional plugin for analysis on pH Titration curves is available and will be integrated into PEAT. This will add the ability to export the data for use in external databases, such as BioMagResBank (BMRB) (37).

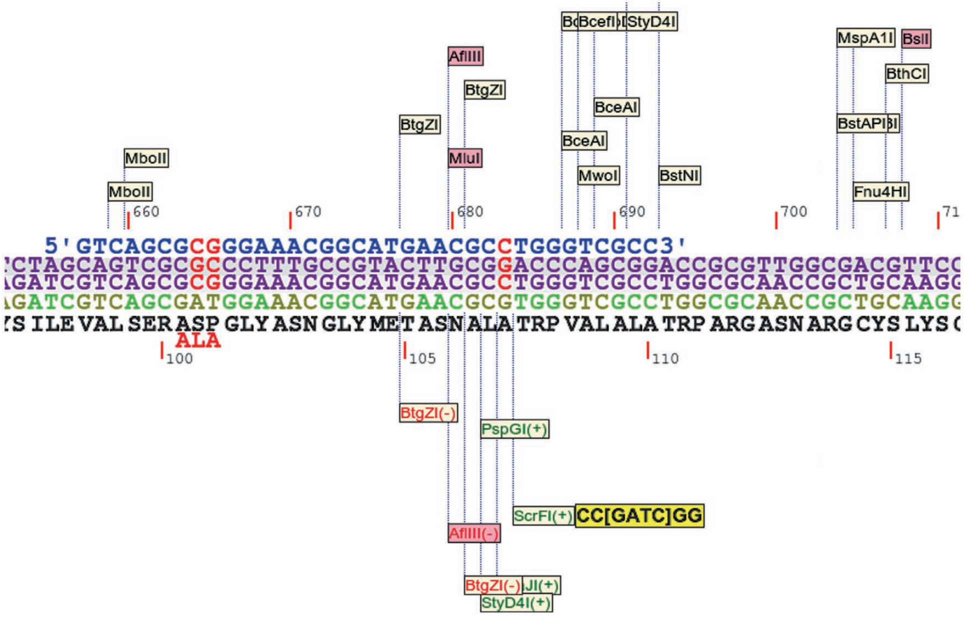
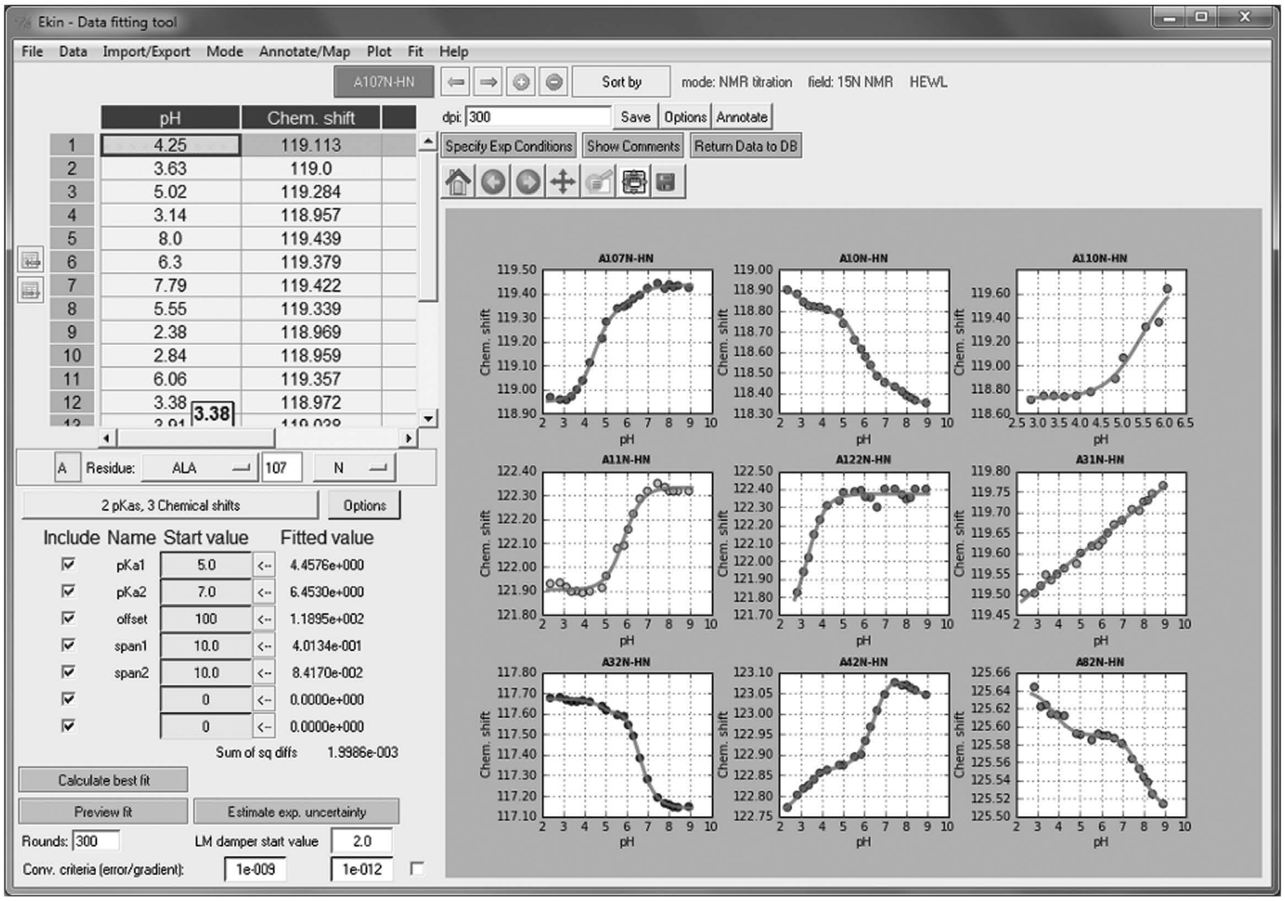

MboII
MboII
BtgZI
AflIII
BtgZI
MluI
BceAI
MwoI
BceAI
BstNI
StyD4I
MspA1I
BslI
BthCI
BstAPI
Fnu4HI
660
670
680
690
700
5'GTCAGCGCGGGAAACGGCATGAACGCCTGGGTCGCC3'
CTAGCAGTCGCGCCCTTTGCCGTACTTGCGGACCCAGCGGACCGCGTTGGCGACGTTCC
GATCGTCAGCGCGGGAAACGGCATGAACGCCTGGGTCGCCTGGCGCAACCGCTGCAAGG
GATCGTCAGCGATGGAAACGGCATGAACGCGTGGGTCGCCTGGCGCAACCGCTGCAAGG
S ILE VAL SER ASP GLY ASN GLY MET ASN ALA TRP VAL ALA TRP ARG ASN ARG CYS LYS
ALA
100
105
110
115
BtgZI(-)
PspGI(+)
ScrFI(+)
CC[GATC]GG
AflIII(-)
BtgZI(-)
JI(+)
StyD4I(+)

Figure 4. Sequencing with DNAtool. The wild-type DNA sequence is shown in green, with the corresponding amino acid sequence in black underneath. The mutant DNA sequence is shown in purple and the mutagenic primer in blue. Changes in the nucleotide sequence are highlighted in red (primer and mutant DNA sequence) and the corresponding positions in the wild-type DNA sequence are shown in brown. The wild-type restriction digest pattern is shown above the primer, and changes in restriction digest patterns when incorporating the mutagenic primer are highlighted below the sequence, with '+' or '–' indicating the addition or removal of a restriction site.



	pH	Chem. shift
1	4.25	119.113
2	3.63	119.0
3	5.02	119.284
4	3.14	118.957
5	8.0	119.439
6	6.3	119.379
7	7.79	119.422
8	5.55	119.339
9	2.38	118.969
10	2.84	118.959
11	6.06	119.357
12	3.38	118.972

A Residue: ALA 107 N

2 pKas, 3 Chemical shifts Options

Include	Name	Start value	Fitted value
✓	pKa1	5.0	4.4576e+000
✓	pKa2	7.0	6.4530e+000
✓	offset	100	1.1895e+002
✓	span1	10.0	4.0134e-001
✓	span2	10.0	8.4170e-002
✓		0	0.0000e+000
✓		0	0.0000e+000

Sum of sq diffs 1.9986e-003

Rounds: 300 LM damper start value 2.0

Conv. criteria (error/gradient): 1e-009 1e-012

Figure 5. Comparison of multiple NMR titration datasets in Ekin. On the left side are data entry and fitting panels. On the right multiple selected datasets can be compared/overlayed. Fitting curves are interactively updated in the plot window.

pH activity profiles. The pH dependence of enzymatic activity can be recorded in Ekin and fitted to Bell shaped curves for 2 and 3 ionizable groups. From the fit to experimental data estimates of pKa and K_{cat} and their statistical uncertainties are computed.

Searching and sharing PEAT data

Collaboration on a single project requires multi-user data sharing, so that many users can interact with the database simultaneously. Our solution utilizes the python Zope Object Database (ZODB). An administrator simply sets up a ZODB database and gives access privileges to individuals (see Materials and methods section for technical details).

Data deposited in PEAT can be shared with collaborators and comparisons can be performed on-the-fly. The dataset can be searched by doing filtering on all simple data fields (number or text), very much like a standard database query, with filters joined by logical operators. More complex data types, such as ekin, require specialized handlers which we are developing (see http://enzyme.ucd.ie/main/index.php/PEAT_Advanced_Search). The search is currently performed on the client side and will be slower for large databases over remote connections. In the near future we will supply a web script that can do server side searches and return results to a remote client much more quickly.

Applications of PEAT in daily work

A typical scenario for which PEAT is well suited is in directed evolution (DE) experiments (38). DE works by generating large sets of diverse mutants from a parent sequence and those with improved characteristics (e.g. higher stability or activity) are identified using a screening or selection step. Improved mutants are used as parent sequences for the next round and so on. PEAT is ideally suited for storing mutant sequences, the results of multiple assays used for screening and keeping track of community information in the Labbook.

Figure 6 illustrates a simplified workflow for a typical protein engineering project.

Case studies

We have used PEAT in our own work with several projects, and these are described briefly here. Some additional functionality in these projects was carried out with specifically written plugins. This requires knowledge of python and the PEAT API, but illustrates the flexibility of the concept to 'add value' specific to a project. Further information and a tutorial on developing plugins is

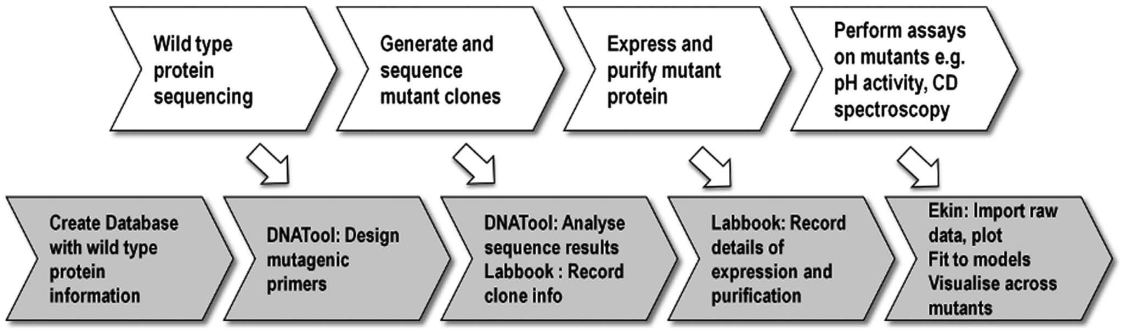

Wild type protein sequencing
Generate and sequence mutant clones
Express and purify mutant protein
Perform assays on mutants e.g. pH activity, CD spectroscopy
Create Database with wild type protein information
DNATool: Design mutagenic primers
DNATool: Analyse sequence results Labbook : Record clone info
Labbook: Record details of expression and purification
Ekin: Import raw data, plot Fit to models Visualise across mutants

Figure 6. A typical workflow illustrating how one might complement experimental work with PEAT. A protein is sequenced and mutagenic primers are designed in DNATool. The mutant clones are created and sequenced and sequences are checked in DNATool. Progress information on each clone is kept in the Labbook and is updated as a clone proceeds through various stages towards expression. The data collected from the various biophysical measurements and kinetic assays is imported into Ekin and fit to yield the desired parameters. Data for multiple mutants can then be compared within the application. Multiple users in parallel doing different assays on the same protein/mutants or handling different mutants would all share the same database.

available at http://enzyme.ucd.ie/main/index.php/PEAT_Plugins

NMR titration database. We have used PEAT to construct Titration_DB (39), a dataset providing statistics on primary experimental NMR pH titration data. The primary module utilized was 'Ekin' for data fitting and plotting. Raw data was imported into Ekin, globally fit using an algorithm to estimate the best fitting models (single or multiple pKa values), and stored incrementally as the work progressed. Also stored is the PDB structure for each protein and experimental conditions if available from the literature sources. This dataset is available via a web interface (written with the PEAT libraries) which can be searched and resultant curves dynamically generated.

Read-only access to this dataset is also available from the PEAT client by using these connection settings:

hostname: http://peat.ucd.ie; port: 8080; project: titration_db, password: 123

Handling of large datasets with the PEAT API. This project is an attempt to streamline data capture and analysis of large amounts of data. The dataset consists of multiple sets of kinetics and temperature assays for approximately 170 enzyme mutants, dumped from the spectrophotometer plate reader device into text files. For each clone we store and fit: time-course absorbance data ($\sim$90 time points) $\times$ eight substrate concentration values $\times$ eight pH points $\times$ three replicates, temperature data for five temperatures $\times$ 170 $\times$ three replicates. There are a total of $\sim$58 000 individual x–y datasets stored in this database.

One objective of this particular study was to create a processing pipeline to improve the previous method which use spreadsheet macros. A specially written command line plugin imports the raw data, extracts velocities, creates a Michaelis–Menten curve and extracts K_{cat} values. These steps being repeated for multiple pH values. The results are all saved to the database for viewing and further analysis in the client application. A full presentation of the scientific conclusions of the above study is outside the scope of the present article, but will be relayed in subsequent communications (D. Farrell *et al.*, in preparation).

Sample projects. We have several freely accessible sample databases available on our server. Up-to-date information is kept here: http://enzyme.ucd.ie/main/index.php/Sample_Projects. Users are invited to access these projects and try PEAT out.

The 'hewlsample' project is one example. This contains two records from our own dataset on Hen Egg White Lysozyme—the wild-type protein and a single mutant. For both records we store amide nitrogen and proton pH NMR titration curves for all residues, a number of pH activity measurements, kinetics data, the sequences in DNAtool with some primers and the wt PDB structure. Also stored is a sample labbook.

Details for access are:

hostname: http://peat.ucd.ie; port: 8080; project: hewlsample, user: guest, password: 123.

DISCUSSION

PEAT is a workbench for protein studies: a set of tools integrated into one application with a solid object database backend. The object database largely removes the problem of keeping track of external database schema and makes it easy to develop custom analysis plugins flexibly. Though there is basic functionality for laboratory management, PEAT is not simply a LIMS. Our most important goal is to satisfy the needs of a protein engineering laboratory, to prevent data loss and to ease the analysis of multiple disparate types of data. We have designed PEAT to provide the following advantages:

(i) Integrating data capture with analysis—using the same software for data deposition as for analysis/processing provides experimentalists with a motivation to systematically deposit their data. It is also relatively easy to develop new plugins for PEAT.
(ii) Coherent formats—consistency within and between groups in recording results in one format (e.g. all users depositing kinetics results into Ekin).

(iii) Storage of experimental conditions—PEAT forces researchers to deposit experimental conditions along with primary data.
(iv) Multi-user data sharing allows fast and flexible data sharing within or between research laboratories.
(v) Object database that allows fast development of plugin modules without reference to database schema.

PEAT is still under active development and has proved stable enough to be used with several of our own projects, and has ensured that data from graduated PhD students are still readily available for re-analysis.

Further areas for improvement include the inclusion of additional plugins to broaden the scope of available features, the development of bulk data-fitting algorithms for dealing with very large sets of biophysical characterization data and the addition of a variety of data exporting tools to facilitate deposition into standard databases such as BMRB.

In summary, we hope that the multiple protein biophysics-oriented features of PEAT combined with the readily available features for data sharing will encourage researchers to analyze and deposit their protein characterization data in online databases, and thus spur the creation of high-quality datasets on the connection between protein sequence, structure and function. Only by amassing large amounts of high-quality experimental data can we hope to fully understand complex characteristics such as protein electrostatics, dynamics and catalysis.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the discussions and bug reports from PEAT users.

FUNDING

Science Foundation Ireland PIYRA grant (04/YI1/M537), a project grant from the Irish Health Research Board (RP/2004/140); a CSCB (HEA) grant and a Science Foundation Ireland Research Frontiers Programme Award (08/RFP/BIC1140) to J.N.; UCD Ad Astra scholarship to J.B. Funding for open access charge: Research grant, UCD internal funding.

Conflict of interest statement. The software presented will be sought marketed to for-profit organizations by University College Dublin. D.F. and J.E.N. are recipients of inventor royalties from potential sales.

REFERENCES

1. Winter,G., Fersht,A.R., Wilkinson,A.J., Zoller,M. and Smith,M. (1982) Redesigning enzyme structure by site-directed mutagenesis: tyrosyl tRNA synthetase and ATP binding. *Nature*, **299**, 756–758.
2. Farinas,E.T., Bulter,T. and Arnold,F.H. (2001) Directed enzyme evolution. *Curr. Opin. Biotechnol.*, **12**, 545–551.
3. Jiang,L., Althoff,E.A., Clemente,F.R., Doyle,L., Rothlisberger,D., Zanghellini,A., Gallaher,J.L., Betker,J.L., Tanaka,F., Barbas,C.F. 3rd *et al.* (2008) De novo computational design of retro-aldol enzymes. *Science*, **319**, 1387–1391.
4. Fersht,A.R., Matouschek,A. and Serrano,L. (1992) The folding of an enzyme. I. Theory of protein engineering analysis of stability and pathway of protein folding. *J. Mol. Biol.*, **224**, 771–782.
5. Cherry,J.R., Lamsa,M.H., Schneider,P., Vind,J., Svendsen,A., Jones,A. and Pedersen,A.H. (1999) Directed evolution of a fungal peroxidase. *Nat. Biotechnol.*, **17**, 379–384.
6. Bolon,D.N., Voigt,C.A. and Mayo,S.L. (2002) De novo design of biocatalysts. *Curr. Opin. Chem. Biol.*, **6**, 125–129.
7. Yang,W., Jones,L.M., Isley,L., Ye,Y., Lee,H.W., Wilkins,A., Liu,Z.R., Hellinga,H.W., Malchow,R., Ghazi,M. *et al.* (2003) Rational design of a calcium-binding protein. *J. Am. Chem. Soc.*, **125**, 6165–6171.
8. Malakauskas,S.M. and Mayo,S.L. (1998) Design, structure and stability of a hyperthermophilic protein variant. *Nat. Struct. Biol.*, **5**, 470–475.
9. Alm,E. and Baker,D. (1999) Prediction of protein-folding mechanisms from free-energy landscapes derived from native structures. *Proc. Natl Acad. Sci. USA*, **96**, 11305–11310.
10. Galzitskaya,O.V. and Finkelstein,A.V. (1999) A theoretical search for folding/unfolding nuclei in three-dimensional protein structures. *Proc. Natl Acad. Sci. USA*, **96**, 11299–11304.
11. van der Sloot,A.M., Tur,V., Szegezdi,E., Mullally,M.M., Cool,R.H., Samali,A., Serrano,L. and Quax,W.J. (2006) Designed tumor necrosis factor-related apoptosis-inducing ligand variants initiating apoptosis exclusively via the DR5 receptor. *Proc. Natl Acad. Sci. USA*, **103**, 8634–8639.
12. Aloy,P. and Russell,R.B. (2002) Interrogating protein interaction networks through structural biology. *Proc. Natl Acad. Sci. USA*, **99**, 5896–5901.
13. Joachimiak,L.A., Kortemme,T., Stoddard,B.L. and Baker,D. (2006) Computational design of a new hydrogen bond network and at least a 300-fold specificity switch at a protein-protein interface. *J. Mol. Biol.*, **361**, 195–208.
14. Berman,H., Henrick,K. and Nakamura,H. (2003) Announcing the worldwide Protein Data Bank. *Nat. Struct. Biol.*, **10**, 980.
15. Gromiha,M.M., Uedaira,H., An,J., Selvaraj,S., Prabakaran,P. and Sarai,A. (2002) ProTherm, thermodynamic database for proteins and mutants: developments in version 3.0. *Nucleic Acids Res.*, **30**, 301–302.
16. Liu,T., Lin,Y., Wen,X., Jorissen,R.N. and Gilson,M.K. (2007) BindingDB: a web-accessible database of experimentally determined protein-ligand binding affinities. *Nucleic Acids Res.*, **35**, D198–D201.
17. Bader,G.D., Donaldson,I., Wolting,C., Ouellette,B.F., Pawson,T. and Hogue,C.W. (2001) BIND–the biomolecular interaction network database. *Nucleic Acids Res.*, **29**, 242–245.
18. Barthelmes,J., Ebeling,C., Chang,A., Schomburg,I. and Schomburg,D. (2007) BRENDA, AMENDA and FRENDA: the enzyme information system in 2007. *Nucleic Acids Res.*, **35**, D511–D514.
19. Toseland,C.P., McSparron,H., Davies,M.N. and Flower,D.R. (2006) PPD v1.0 – an integrated, web-accessible database of experimentally determined protein pKa values. *Nucleic Acids Res.*, **34**, D199–D203.
20. Quo,C.F., Wu,B. and Wang,M.D. (2005) Development of a laboratory information system for cancer collaboration projects. *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, **3**, 2859–2862.
21. Monnier,S., Cox,D.G., Albion,T. and Canzian,F. (2005) T.I.M.S: TaqMan Information Management System, tools to organize data flow in a genotyping laboratory. *BMC Bioinformatics*, **6**, 246.
22. Prilusky,J., Oueillet,E., Ulryck,N., Pajon,A., Bernauer,J., Krimm,I., Quevillon-Cheruel,S., Leulliot,N., Graille,M., Liger,D. *et al.* (2005) HalX: an open-source LIMS (Laboratory Information Management System) for small- to large-scale laboratories. *Acta Crystallogr. D Biol. Crystallogr.*, **61**, 671–678.
23. Tchuvatkina,O., Shimoni,L., Ochs,M.F. and Moloshok,T. (2006) Proteomics LIMS: a caBIG project, year 1. *AMIA Annu. Symp. Proc.*, 1116.
24. Li,H., Gennari,J.H. and Brinkley,J.F. (2006) Model driven laboratory information management systems. *AMIA Annu. Symp. Proc.*, **2006**, 484–488.

25. Baran,M.C., Moseley,H.N., Aramini,J.M., Bayro,M.J., Monleon,D., Locke,J.Y. and Montelione,G.T. (2006) SPINS: a laboratory information management system for organizing and archiving intermediate and final results from NMR protein structure determinations. *Proteins*, **62**, 843–851.
26. Viksna,J., Celms,E., Opmanis,M., Podnieks,K., Rucevskis,P., Zarins,A., Barrett,A., Neogi,S.G., Krestyaninova,M., McCarthy,M.I. *et al.* (2007) PASSIM – an open source software system for managing information in biomedical studies. *BMC Bioinformatics*, **8**, 52.
27. Fogh,R., Ionides,J., Ulrich,E., Boucher,W., Vranken,W., Linge,J.P., Habeck,M., Rieping,W., Bhat,T.N., Westbrook,J. *et al.* (2002) The CCPN project: an interim report on a data model for the NMR community. *Nat. Struct. Biol.*, **9**, 416–418.
28. Berman,H.M., Westbrook,J.D., Gabanyi,M.J., Tao,W., Shah,R., Kouranov,A., Schwede,T., Arnold,K., Kiefer,F., Bordoli,L. *et al.* (2009) The protein structure initiative structural genomics knowledgebase. *Nucleic Acids Res.*, **37**, D365–D368.
29. Navarro,J.D., Talreja,N., Peri,S., Vrushabendra,B.M., Rashmi,B.P., Padma,N., Surendranath,V., Jonnalagadda,C.K., Kousthub,P.S., Deshpande,N. *et al.* (2004) BioBuilder as a database development and functional annotation platform for proteins. *BMC Bioinformatics*, **5**, 43.
30. Zolnai,Z., Lee,P.T., Li,J., Chapman,M.R., Newman,C.S., Phillips,G.N. Jr., Rayment,I., Ulrich,E.L., Volkman,B.F. and Markley,J.L. (2003) Project management system for structural and functional proteomics: Sesame. *J. Struct. Funct. Genomics*, **4**, 11–23.
31. Atkinson,M., DeWitt,D., Maier,D., Francois,B., Dittrich,K. and Zdonik,S. (1992) *Building an object-oriented database system: the story of 02.* Morgan Kaufmann Publishers Inc., pp. 1–20.
32. Pilgrim M., Ravenscroft A. and Books24x7 Inc. (2004), Dive into Python. Apress, Berkeley, Calif.
33. Levenberg,K. (1944) Method for the solution of certain non-linear problems in least squares. *The Quar. Appl. Mathematics*, **2**, 164–168.
34. Allen,D.L. and Pielak,G.J. (1998) Baseline length and automated fitting of denaturation data. *Protein Sci.*, **7**, 1262–1263.
35. Kamal,J.K., Nazeerunnisa,M. and Behere,D.V. (2002) Thermal unfolding of soybean peroxidase. Appropriate high denaturant concentrations induce cooperativity allowing the correct measurement of thermodynamic parameters. *J. Biol. Chem.*, **277**, 40717–40721.
36. John,D.M. and Weeks,K.M. (2000) van't Hoff enthalpies without baselines. *Protein Sci.*, **9**, 1416–1419.
37. Ulrich,E.L., Akutsu,H., Doreleijers,J.F., Harano,Y., Ioannidis,Y.E., Lin,J., Livny,M., Mading,S., Maziuk,D., Miller,Z. *et al.* (2008) BioMagResBank. *Nucleic Acids Res.*, **36**, D402–D408.
38. Bloom,J.D. and Arnold,F.H. (2009) In the light of directed evolution: pathways of adaptive protein evolution. *Proc. Natl Acad. Sci. USA*, **106(Suppl. 1)**, 9995–10000.
39. Farrell,D., Miranda,E.S., Webb,H., Georgi,N., Crowley,P.B., McIntosh,L.P. and Nielsen,J.E. (2009) Titration_DB: Storage and analysis of NMR-monitored protein pH titration curves. *Proteins*, **78**, 843–857.