LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Year 2 Report: Protein Function Prediction Platform

C. E. Zhou

May 2, 2012

LLNL-TR-554191

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

REPORT DOCUMENTATION PAGE	*Form Approved* **OMB No. 0704-0188**

1. REPORT DATE *(DD-MM-YYYY)* 05-01-2012	**2. REPORT TYPE** progress report	**3. DATES COVERED** *(From - To)* 1 April 2011 - 31 March 2012

4. TITLE AND SUBTITLE Year 2 Final Report - Protein Function Prediction Platform (v.0.5)	**5a. CONTRACT NUMBER** PE0603384BP
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Carol L. Ecale Zhou	**5d. PROJECT NUMBER**
	5e. TASK NUMBER 0009
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Lawrence Livermore National Laboratory 7000 East Avenue Livermore, CA 94550	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) DoD Chemical and Biological Defense Program (CBDP) Diagnostics and Disease Surveillance Joint Science and Technology Office (JSTO) Ft. Belvoir, VA	**10. SPONSOR/MONITOR'S ACRONYM(S)** CBDT, JSTO
	11. SPONSORING/MONITORING AGENCY REPORT NUMBER

12. DISTRIBUTION AVAILABILITY STATEMENT

Distribution Statement F: Further dissemination authorized only as directed by DTRA/JSTO or higher DoD authority. Requests for this document shall be referred to DTRA/JSTO-CBI.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Upon completion of our second year of development in a 3-year development cycle, we have completed a prototype protein structure-function annotation and function prediction system: Protein Function Prediction (PFP) platform (v.0.5). We have met our milestones for Years 1 and 2 and are positioned to continue development in completion of our original statement of work, or a reasonable modification thereof, in service to DTRA Programs involved in diagnostics and medical countermeasures research and development.

15. SUBJECT TERMS

software integration, data integration, protein function prediction, functional annotation, structure modeling, mechanistic modeling, systems biology, metabolic pathway modeling, genomics, diagnostics, therapeutics, biomarkers, medical countermeasures, bioinformatics, informatics

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT** SAR	**18. NUMBER OF PAGES** 31	**19a. NAME OF RESPONSIBLE PERSON** Carol L. Ecale Zhou
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U			**19b. TELEPONE NUMBER (*Include area code)*** 925-422-2117

Year 2 Final Report – Protein Function Prediction Platform

Lawrence Livermore National Laboratory

By Carol L. Ecale Zhou (zhou4@llnl.gov; 925-422-2117)

Contributions from: Eithon Cadag, Jerome Nilmeier, Jennifer Sirp, Amy Huang, Felice Lightstone, Patrik D'haeseleer, and Adam Zemla

Submitted to: DTRA Management

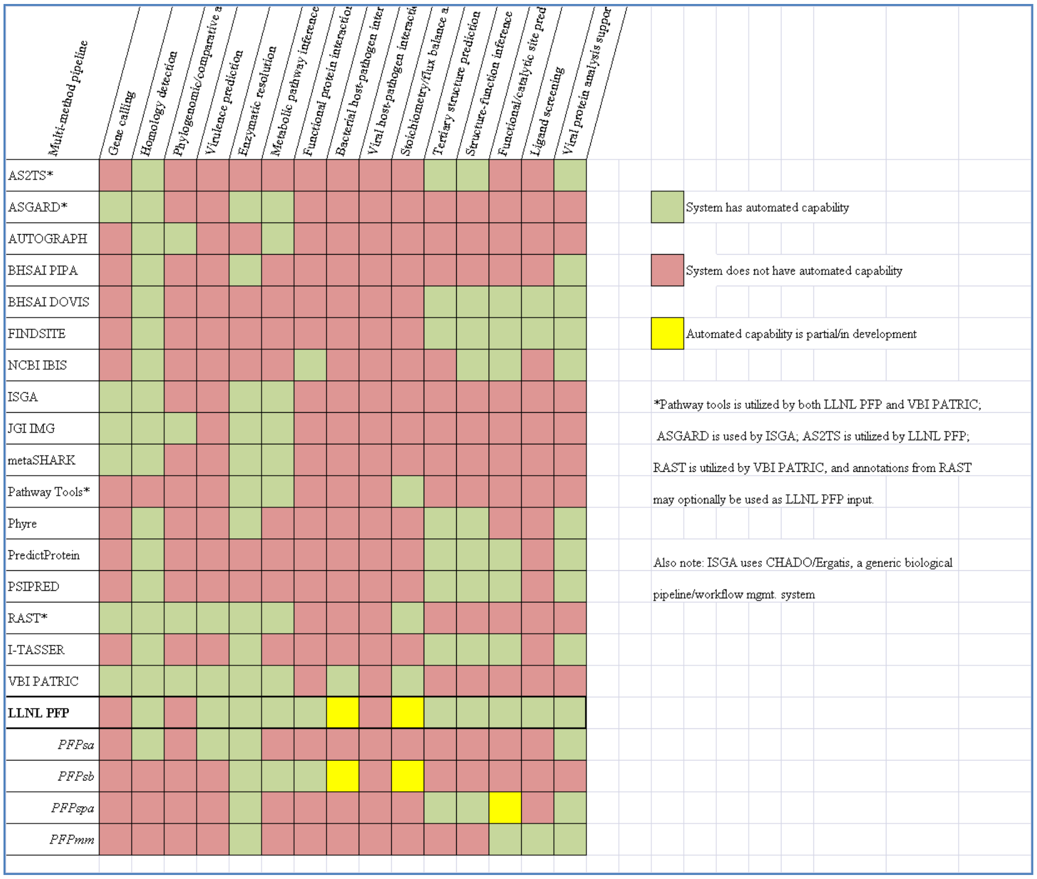
Date: 1 May 2012

Executive Summary

Upon completion of our second year of development in a 3-year development cycle, we have completed a prototype protein structure-function annotation and function prediction system: Protein Function Prediction (PFP) platform (v.0.5). We have met our milestones for Years 1 and 2 and are positioned to continue development in completion of our original statement of work, or a reasonable modification thereof, in service to DTRA Programs involved in diagnostics and medical countermeasures research and development.

Introduction

The Protein Function Prediction (PFP) platform is a multi-scale computational modeling system for protein structure-function annotation and function prediction. As of this writing, PFP is the only existing fully automated, high-throughput, multi-scale modeling, whole-proteome annotation platform, and represents a significant advance in the field of genome annotation (Fig. 1). PFP modules perform protein functional annotations at the sequence, systems biology, protein structure, and atomistic levels of biological complexity (Fig. 2). Because these approaches provide orthogonal means of characterizing proteins and suggesting protein function, PFP processing maximizes the protein functional information that can currently be gained by computational means. Comprehensive annotation of pathogen genomes is essential for bio-defense applications in pathogen characterization, threat assessment, and medical countermeasure design and development in that it can short-cut the time and effort required to select and characterize protein biomarkers.



Multi-method pipeline	Gene calling	Homology detection	Phylogenomic/comparative a	Virulence prediction	Enzymatic resolution	Metabolic pathway inference	Functional protein interactio	Bacterial host-pathogen inter	Viral host-pathogen interacti	Stoichiometry/flux balance a	Tertiary structure prediction	Structure-function inference	Functional/catalytic site pred	Ligand screening	Viral protein analysis suppor
AS2TS*	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟥	🟥	🟩
ASGARD*	🟩	🟩	🟥	🟥	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
AUTOGRAPH	🟥	🟩	🟩	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
BHSAI PIPA	🟥	🟩	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩
BHSAI DOVIS	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟩
FINDSITE	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟩
NCBI IBIS	🟥	🟩	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟩	🟩	🟥	🟩
ISGA	🟩	🟩	🟥	🟥	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
JGI IMG	🟩	🟩	🟩	🟥	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
metaSHARK	🟩	🟩	🟥	🟥	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
Pathway Tools*	🟥	🟥	🟥	🟥	🟩	🟩	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥
Phyre	🟥	🟩	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟥	🟥	🟩
PredictProtein	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟩
PSIPRED	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟩
RAST*	🟩	🟩	🟩	🟩	🟩	🟩	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥
I-TASSER	🟥	🟩	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟩
VBI PATRIC	🟩	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟥	🟩	🟥	🟥	🟥	🟥	🟥
LLNL PFP	🟥	🟩	🟥	🟩	🟩	🟩	🟩	🟨	🟥	🟨	🟩	🟩	🟩	🟩	🟩
PFPsa	🟥	🟩	🟥	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩
PFPsb	🟥	🟥	🟥	🟥	🟩	🟩	🟩	🟨	🟥	🟨	🟥	🟥	🟥	🟥	🟥
PFPspa	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟨	🟥	🟩
PFPmm	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟩

🟩 System has automated capability

🟥 System does not have automated capability

🟨 Automated capability is partial/in development

*Pathway tools is utilized by both LLNL PFP and VBI PATRIC; ASGARD is used by ISGA; AS2TS is utilized by LLNL PFP; RAST is utilized by VBI PATRIC, and annotations from RAST may optionally be used as LLNL PFP input.

Also note: ISGA uses CHADO/Ergatis, a generic biological pipeline/workflow mgmt. system

Fig. 1. Comparison of Protein Function Prediction (PFP) platform to like capabilities. PFP is currently the most comprehensive genome annotation system.

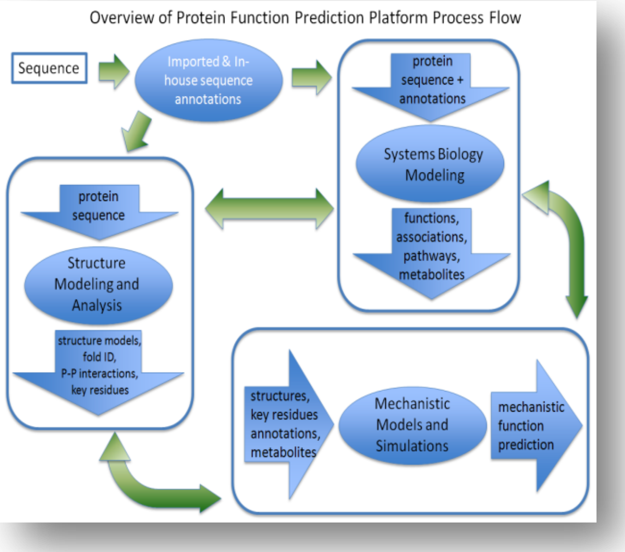

Overview of Protein Function Prediction Platform Process Flow
Sequence
Imported & In-house sequence annotations
protein sequence + annotations
Systems Biology Modeling
functions, associations, pathways, metabolites
protein sequence
Structure Modeling and Analysis
structure models, fold ID, P-P interactions, key residues
structures, key residues annotations, metabolites
Mechanistic Models and Simulations
mechanistic function prediction

Fig. 2. Overview of Protein Function Prediciton (PFP) platform process flow. PFP is a multi-scale whole-proteome structure-function annotation platform combining results from four software modules: sequence annotation, systems biology, structure prediction and analysis, and mechanistic modeling.

PFP was originally funded by DTRA's TMT Program for a 3-year development cycle. Our charter was to fill an informatics gap between genomic sequencing and computer-aided drug design, and to support TMT performers in identifying and characterizing potential drug targets. Design and development of the PFP platform has been a highly non-trivial endeavor. PFP comprises a mix of open-source codes, licensed codes, and original codes developed at LLNL based on novel algorithms; processing is supported by integration of more than 20 external data sets (Fig. 3). In this document we summarize our progress in PFP development at the end of Year 2, and propose additional features for which we respectfully request FY13 funding for development of a full-featured PFP platform in support of biomarker discovery and characterization.

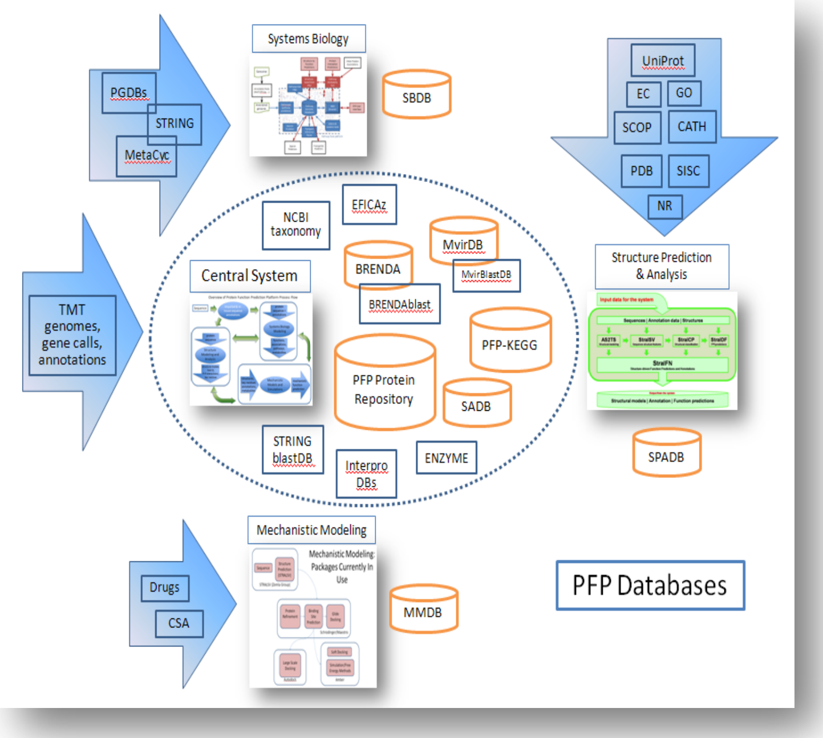


Fig. 3. PFP databases. Squares = flat-file databases; drums = relational databases; blue arrows = external data sources imported to PFP and managed by PFPmain (central system) or the respective modules; blue oval in center encloses data sources managed by PFPmain.

This report summarizes technical progress in the design and development of the PFP platform, and contains the following sections:

- Overview of the PFP System Integration Module – p.4
- Overview of the Sequence Annotation Module – p.5
- Overview of the Systems Biology Module – p.5

For detailed descriptions of PFP software system requirements and design specifications, please refer to the Software Requirements Specification (SRS; v.2.0) and Software Design Document (SDD; v.1.0) documents previously submitted to DTRA management.

Overview of the PFP System Integration Module (PFPmain)

The Integration module (PFPmain) is responsible for the execution of system components (software modules) and for management of system data flows. Upon job creation, PFPmain constructs a module workflow and calls each module, in turn, passing it the data it needs to perform its function. Once a module has been initiated, PFPmain monitors the progress of the module, and when the module has finished executing, it collects the results, loads the data into a central database, and calls the next module, until the workflow is complete.

Year 2 status: Summary of PFPmain

Job Submission: PFPmain provides a prototype user interface (UI) that allows users to submit, monitor, and view job results. User input data is uploaded through the UI. Required data for analysis of bacterial genomes includes: genome sequence and an annotation file with gene calls. For all other taxa, or for sets of proteins with or without taxonomic association, the minimal user input comprises a fasta file of protein sequences.

Job Processing: Three of the four module pipelines are managed by PFPmain in a fully automated fashion (Table I). Communications between PFPmain and each subordinate module are achieved by means of a client-server paradigm, which facilitates processing in a distributed processing environment and accommodates modules that run on different hardware configurations and different operating systems. Currently, the mechanistic modeling module (MM) is running on a high performance computing (HPC) system that cannot be accessed directly (programmatically) by PFPmain due to LLNL security constraints; thus, data packages between PFPmain and MM are transferred manually to and from this module.

The PFPmain process flow first uploads the user input and inserts the proteins into the central database, then creates a new job with a default execution flow. This job is then processed by the job manager, which carries out the following process for each of the modules: prepare the input data by selecting out data from the database and packaging it for the module, call the module and pass it the necessary input data and run parameters, check the status of the module's processing, download the results when they are ready, load the results into the central database. Each protein is tracked through the PFP platform by a unique database identifier assigned by PFPmain.

Job Analysis: Once analysis is complete, users view preliminary results via the UI. At this time only a few simple user interface pages have been constructed.

Table 1. Job processing status for PFP system modules.

Distributed Module	Send and receive data to and from pfpi	Check module status from pfpi	Send cancel request to the module from pfpi	Data processing, loading, and validation	Data selection and packaging
Sequence Annotation	Complete	Complete	Complete	Complete	Complete
Systems Biology	Complete	Complete	Complete	Complete	Complete
Structural Prediction Analysis	Complete	Complete	Hardware dependent	Complete	Complete
Mechanistic Modeling	Hardware dependent	Hardware dependent	Hardware dependent	Complete	Complete

Overview of the Sequence Annotation Module (SA)

The Sequence Annotation (SA) module runs at the beginning of PFP processing in the default workflow. SA currently comprises several protein functional annotation tools, which were selected or created based on their support of the down-stream systems biology and structure-based analyses, or based on their relevance to the goals of medical countermeasures research. These tools include EFICAz (enzyme prediction), KEGG (database of metabolic proteins) Blast, Interproscan (functional motifs), Brenda (database of known enzymes) Blast, Virulence (LLNL MvirDB) Blast, SignalP (signal peptide prediction), and BEOracle (immune epitope prediction). SA functions as a stand-alone pipeline and is also fully integrated into the PFP platform; the user can send a job request either through the PFPmain portal, or directly through the SA website. PFPmain accesses SA by means of an HTTP URL call. SA processing is fully automated and runs at whole-proteome scale. SA manages the underlying annotation tool processing using a multi-threaded approach, with a configurable default number of process threads. SA performs 'start a job', 'stop a job', 'check job status', and 'get result'. When a job completes, SA returns the results to PFPmain in a pre-defined XML format, which is subsequently parsed by PFPmain and loaded to the central database. If desired, the raw output from the annotation tools can be manually retrieved from the SA web interface.

Overview of the PFP Systems Biology Module (SB)

The function of a protein is not merely determined by its behavior at the atomistic level, but also by its wider role within the organism. The higher-level context of a protein within the organism (e.g., adjacent genes, pathway structure, regulons, protein interactions, protein localization) provides vital clues to its function that could not be achieved by studying the protein in isolation. The Systems Biology (SB) module enables rapid, genome-scale analysis of sequenced bacteria, integration of functional clues from a variety of sequence- and structure-based analyses, and prioritization of proteins and pathways for further investigation (Fig. 4).

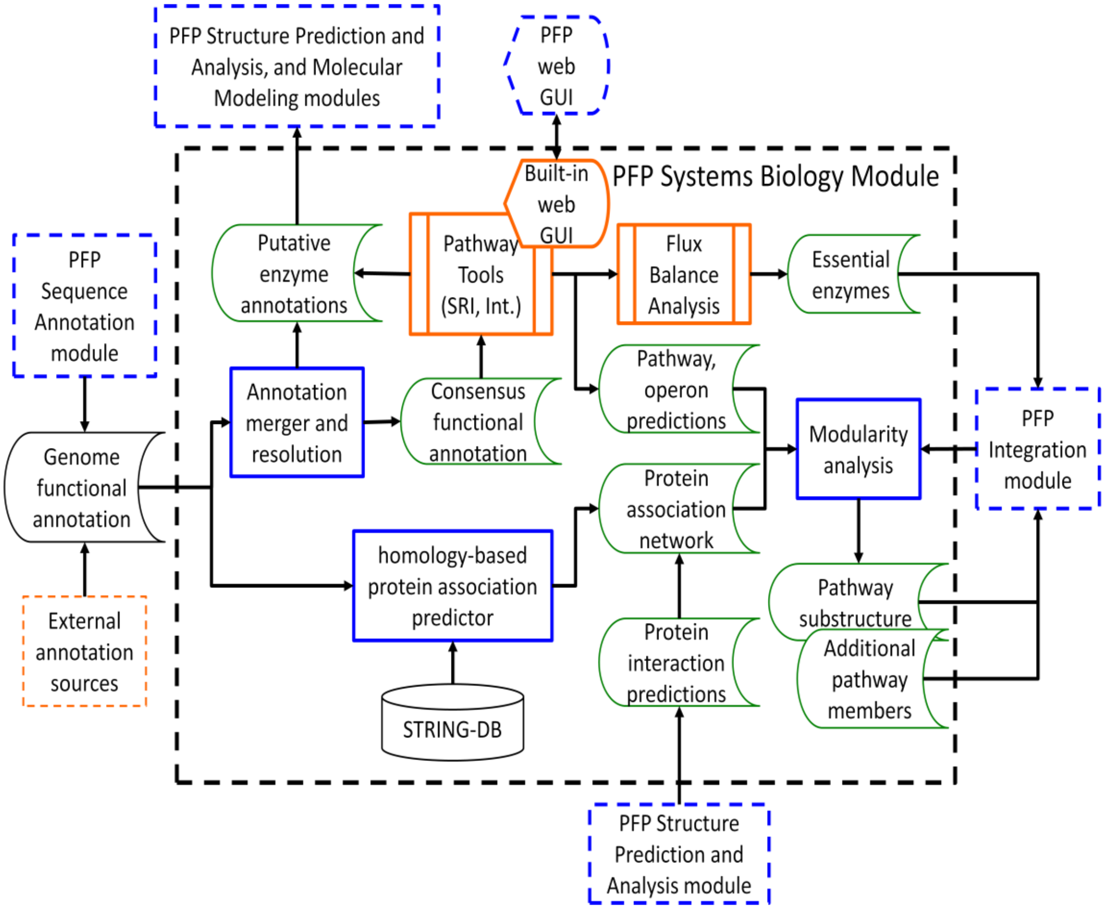

PFP Structure Prediction and Analysis, and Molecular Modeling modules
PFP web GUI
Built-in web GUI
PFP Systems Biology Module
PFP Sequence Annotation module
Putative enzyme annotations
Pathway Tools (SRI, Int.)
Flux Balance Analysis
Essential enzymes
Pathway, operon predictions
Genome functional annotation
Annotation merger and resolution
Consensus functional annotation
Modularity analysis
PFP Integration module
Protein association network
homology-based protein association predictor
Pathway substructure
Additional pathway members
External annotation sources
Protein interaction predictions
STRING-DB
PFP Structure Prediction and Analysis module

Fig. 4. Schematic of the process and data flows of the Systems Biology (SB) module. The elements outlined in orange are part of the existing Pathway Tools platform. The elements outlined in blue are new codes developed by LLNL researchers and developers, along with all codes necessary to integrate them.

SB was constructed using the SRI International Pathway Tools (PT) platform as a basis to which PFP developers added a novel modularity tool for functional clustering of interacting proteins. SB functions as a stand-alone pipeline, and PT processing is also fully integrated into the PFP platform; the user can send a job request either through the PFPmain portal, or directly through the SB web portal. PFPmain accesses SB by means of an HTTP URL call. Pathway genome databases (PGDBs) generated by PT can be explored interactively within SB using the PT UIs, and further data products can be derived automatically from within the PT system. SB processing is fully automated and runs at whole-genome scale for bacterial genomes. The PT package includes a standalone web server that provides a well-developed user interface for browsing and querying pathway genome databases, including visualization of the complete metabolic network, an omics viewer, cross-species comparisons, complex database queries, separate webpages for each gene, protein, compound, and pathway (with different levels of detail).

Year 2 Status: Summary of SB pipeline processing

SB processes genomic sequence and annotation data (RAST or Genbank) to infer metabolic pathway predictions and enzymatic functions related to the metabolism of a bacterium. This is done by processing through a local instance of the Pathway Tools (PT) platform, followed by further analyses through a Modularity Tool (MT) that was designed and developed by LLNL researchers. In addition to the input data provided by the user, PFPmain provides SB with annotation data produced by SA. SB combines the SA and user-provided annotation data sets, resolves and documents disagreements in enzymatic annotations, updates out-of-date controlled vocabulary terms (thereby improving initial draft annotations based on user guidance or expert-derived defaults), and re-formats the resulting merged data for loading to PT. Upon completion of PT processing, results are automatically loaded into an SRI (Ocelot Lisp-object) database. SB then queries the PT results by means of programmed Lisp objects, and subsequently loads the results into a relational database within the SB module proper. Data are then queried out and formatted for input into the Modularity Tool (MT). The MT predicts protein associations using rapid homology-based methods, driven by a curated database of known associations (STRING). This information is extracted and combined with pathway and operon predictions to construct an annotated, multi-layered network view of the bacterium's interactome. Functional subunits of the combined networks arising from modularity analysis are generated, which can aid in filling gaps in metabolic pathways and can help elucidate function of poorly annotated proteins. Finally, functional information, along with reliability metrics, are derived from the PT and MT analyses, and results are formatted in a pre-defined XML format for return to PFPmain.

SB Components

The Pathway Tools core engine: Our main pathway inference engine and systems biology query and exploration system is based on SRI International's Pathway Tools (PT) software. The PT system is integrated within the SB module, allowing fast and automated metabolic modeling of an annotated microbial genome. A built-in pathway hole-filling capability can be used to generate additional enzyme function assignments that are missing in the genome annotation. To interact with results, we are integrating PT's extensive Web-accessible UI into the larger PFP UI; to date, a few of the web links to SB have been incorporated into the PFP UI.

Annotation merge tool: We have recently completed a merge and resolution tool that can take genome-wide enzyme annotations from a range of different sources and tools, including EFICAz, RAST, EC numbers, and InterPro results (generated by SA); derive a user customizable, weighted consensus annotation from all of these; and produce output that can be fed directly into PT for metabolic inference. This approach to annotation resolution also produces a series of logs, including incomplete or low-quality enzyme annotations that are not included in the metabolic network model, but that may provide useful protein function hypotheses testable by further structure-based analyses.

Modularity and pathway membership analysis: Each protein may be associated with other proteins in a variety of different ways, including membership in metabolic pathway regulatory networks or operons, involvement in protein interactions and similar functional categories. By mapping each of these different types of functional association evidence to a common standard (e.g., membership in the same KEGG metabolic pathway), it is possible to derive an integrated genome-wide gene association network. Our modularity analysis tool augments metabolic pathway inference predictions by including non-metabolic interactions (including many virulence factors), and discovers clusters in this network that may correspond to specific pathways, protein complexes, and functional systems. Starting from a set of proteins of interest, our modularity tool may be used to discover the underlying substructure of the set (e.g., distinct sub-pathways of a metabolic pathway). Conversely, the modularity tool may also be used to expand a set, discovering additional members of a pathway. The pathway modularity tool lays part of the groundwork for a structure-based pathway hole-filling tool (to be developed)

Flux Balance Analysis: We have also been working to include Flux Balance Analysis (FBA) modeling to the SB module. FBA can be used to predict enzymes that are essential to the growth of a bacterium; such enzymes might therefore make useful drug targets. FBA can be used to predict secreted pathogen-specific metabolites that could serve as biomarkers of its activity. We have developed a tool that allows us to rapidly develop draft FBA models using genome annotations from the KEGG database. In addition, SRI International has recently released an FBA toolkit for use in conjunction with Pathway Tools, and we are working directly with the developers to integrate this capability into PFP's SB module. Although FBA is not currently incorporated into PFP, the central database was designed to accommodate FBA results, and the independent FBA development is sufficiently mature to begin incorporation of this module should there be sponsor interest and funding to support the work.

Overview of the PFP Structure Prediction and Analysis (SPA) Module

Protein structure can be highly informative in terms of elucidating protein function. Unfortunately, high cost and technical difficulties often preclude the ability to obtain experimental structures for many proteins. Although the Structure Genomics Initiative has greatly increased the number and diversity of protein structures in the PDB, there remains a need for protein structure modeling. Indeed, even experimentally solved structures do not tell a complete story; it is essential to perform a variety of computational analyses to derive functional information from a protein structure. LLNL computer scientists have developed a suite of codes that perform quantitative analyses on protein structures, including high-quality models, to identify function in multiple structural dimensions.

PFP's Structure Prediction and Analysis (SPA) module itself comprises a multi-scale modeling system, which begins with structure modeling and follows with structure-fragment-based homolog identification, sequence variability analysis, and domain-based clustering. These analyses provide structure-function annotations at the residue/sequence, structure-motif, domain, and whole-protein levels. (A future release of PFP may include codes for detecting protein-protein interactions and modeling quaternary structure.) These codes are incorporated into a fully automated analysis pipline, which functions in stand-alone mode, yet is also fully integrated into the PFP platform; the user can send a job request either through the PFPmain portal, or directly through the SPA website. PFPmain accesses SPA by means of an HTTP URL call. SPA processing runs at whole-proteome scale and is configurable, based on the user's requirements for speed vs. accuracy.

A schematic of the SPA module is shown in Fig. 5. Once complete, SPA will comprise five sub-modules: Model Builder, StralSV, StralCP, StralDF (in development), and StralFN (in development).

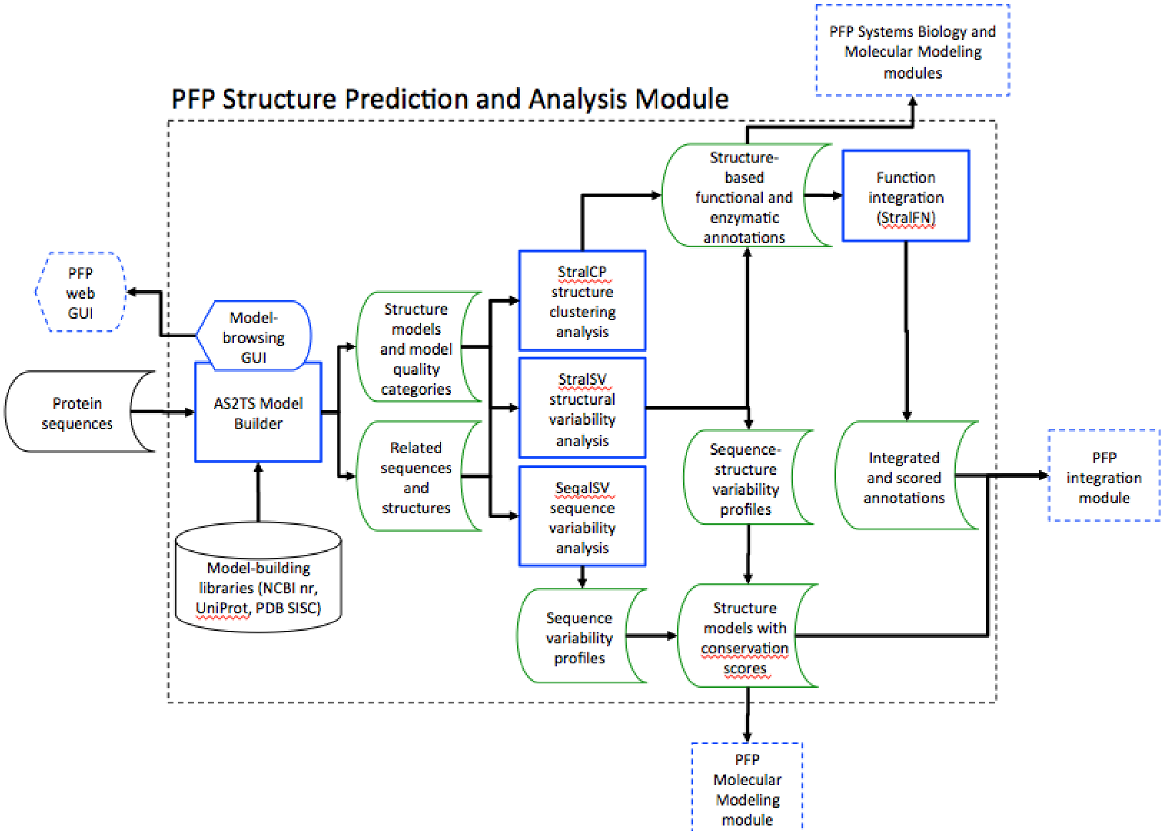

PFP Systems Biology and Molecular Modeling modules
PFP Structure Prediction and Analysis Module
Structure-based functional and enzymatic annotations
Function integration (StralFN)
PFP web GUI
Model-browsing GUI
StralCP structure clustering analysis
Structure models and model quality categories
Protein sequences
AS2TS Model Builder
StralSV structural variability analysis
Related sequences and structures
Sequence-structure variability profiles
Integrated and scored annotations
PFP integration module
SeqalSV sequence variability analysis
Model-building libraries (NCBI nr, UniProt, PDB SISC)
Sequence variability profiles
Structure models with conservation scores
PFP Molecular Modeling module

Fig. 5. Schematic of the Structure Prediction and Analysis (SPA) module. Blue solid boxes denote processes or algorithms that have already been developed and are in place; blue dotted boxes are processes that are situated outside SPA, but are part of the PFP system; green boxes indicate important information, and their arrows direction of flow.

Year 2 Status: Summary of SPA pipeline processing

The SPA pipeline workflow currently consists of a sequential execution of three sub-modules: AS2TS Model Builder high-throughput protein structure homology modeling, StralSV structure-based sequence variability analysis, and StralCP structure-based protein clustering. Currently, input to SPA comprises a fasta file containing protein sequences.

SPA generates protein structure models for whole proteomes by means of LLNL's structure modeling code, AS2TS (Zemla et al. 2005). Draft models can be generated for entire proteomes within hours (affording rapid turnaround of preliminary results), while more detailed, experiment-quality models can be generated for a whole proteome in about one week. "Best" models are generated based on several quality metrics (e.g., sequence identity to primary template, e-value, coverage at N- or C- terminus, secondary structure similarity), and those that pass a pre-defined quality filter are forwarded for further processing.

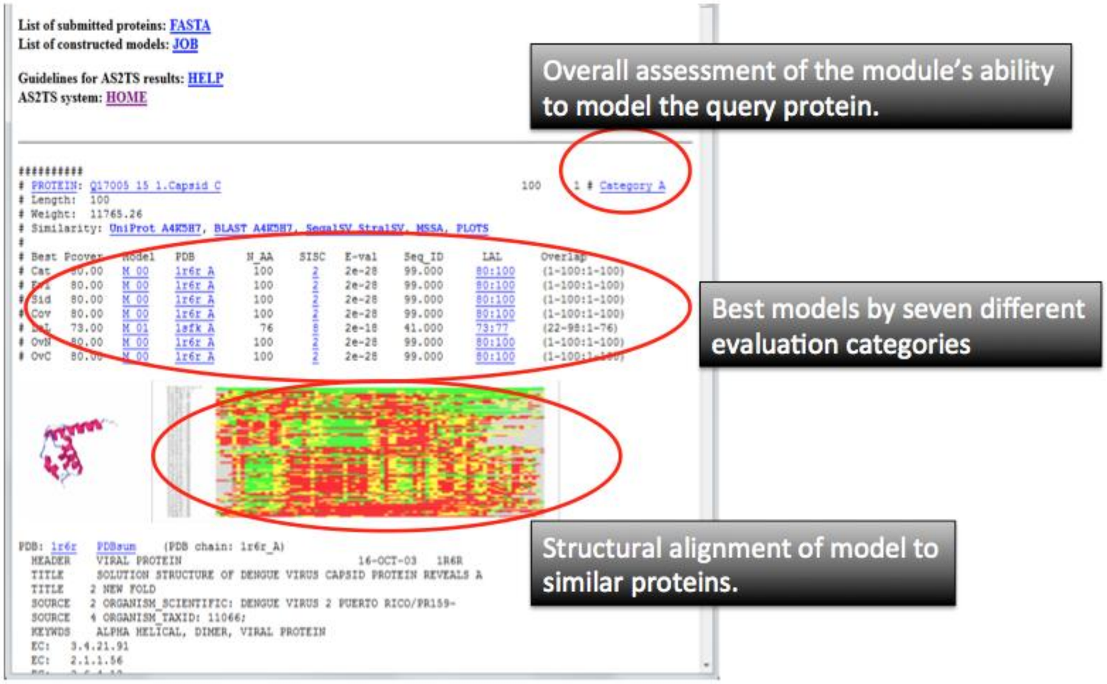
The StralSV code identifies structurally similar fragments using an algorithm that is sensitive and captures structural matches even for rare structure conformations. This is accomplished by searching structure libraries (typically PDB or user-defined) for homologous structure fragments that have tight local alignments that fit within a larger structure context (Zemla et al. 2011). The StralSV algorithm functions in "linear" and "spherical" modes, whereby structure fragments comprise contiguous residues and residues that locate within a given spatial radius, respectively. StralSV analysis yields position-specific sequence variability profiles for the protein of interest. An addendum to StralSV is a new code, SeqalSV, which is a sequence-based alignment code that combines sequences of related proteins with data from StralSV to provide an enriched data set for scoring sequence variability for individual residue positions. Currently StralSV is executed only in "linear" mode for automated processing within PFP.

The structure clustering code, StralCP (Zemla et al 2007), is used in SPA to predict function by transference of functional annotations to the protein of interest from co-clustered structure templates. This analysis enables an approach to annotation transfer at a finer level of granularity than mere association with SCOP domains, and facilitates localization of active-site residues by means of structure-based sequence alignment, when active site residues are identified in one or more co-clustered templates. Currently active-site residues are not extracted from structure alignments in an automated fashion.

Output from the above described analyses are packaged into two tarballs for transfer back to PFPmain, comprising a results file written in a pre-defined XML format and a directory of PDB-formatted model files. PFPmain retrieves, unpacks, parses, and loads the SPA results data to the central database; model files are stored on disk with pointers, only, stored in the database

SPA Components

AS2TS Model Builder: Model Builder implements AS2TS structure prediction in high-throughput. The raw output from Model Builder, viewable through the SPA model-browsing UI, provides voluminous structure and function information to the user. A sample interface to Model Builder is shown in Fig. 6.



List of submitted proteins: FASTA
List of constructed models: JOB

Guidelines for AS2TS results: HELP
AS2TS system: HOME

```
##########
# PROTEIN: Q17005 15 1.Capsid C                                          100      1 # Category A
# Length:  100
# Weight:  11765.26
# Similarity: UniProt A4K5H7, BLAST A4K5H7, SeqalSV StralSV, MSSA, PLOTS
#
# Best Pcover  Model   PDB      N_AA   SISC   E-val    Seq_ID    LAL      Overlap
# Cat   80.00  M 00   1r6r A    100     2     2e-28    99.000   80:100   (1-100:1-100)
# Evl   80.00  M 00   1r6r A    100     2     2e-28    99.000   80:100   (1-100:1-100)
# Sid   80.00  M 00   1r6r A    100     2     2e-28    99.000   80:100   (1-100:1-100)
# Cov   80.00  M 00   1r6r A    100     2     2e-28    99.000   80:100   (1-100:1-100)
# LaL   73.00  M 01   1sfk A     76     8     2e-18    41.000   73:77    (22-98:1-76)
# OvN   80.00  M 00   1r6r A    100     2     2e-28    99.000   80:100   (1-100:1-100)
# OvC   80.00  M 00   1r6r A    100     2     2e-28    99.000   80:100   (1-100:1-100)
```

```
PDB: 1r6r   PDBsum    (PDB chain: 1r6r_A)
  HEADER    VIRAL PROTEIN                             16-OCT-03   1R6R
  TITLE     SOLUTION STRUCTURE OF DENGUE VIRUS CAPSID PROTEIN REVEALS A
  TITLE    2 NEW FOLD
  SOURCE   2 ORGANISM_SCIENTIFIC: DENGUE VIRUS 2 PUERTO RICO/PR159-
  SOURCE   4 ORGANISM_TAXID: 11066;
  KEYWDS    ALPHA HELICAL, DIMER, VIRAL PROTEIN
  EC:   3.4.21.91
  EC:   2.1.1.56
```

Overall assessment of the module's ability to model the query protein.
Best models by seven different evaluation categories
Structural alignment of model to similar proteins.

Fig. 6. Sample UI generated by SPA. Salient results are indicated by red ovals and labels at right: model grade ('A', 'B', 'C1', 'C2', or 'C3'), list of "best" models (depending on various criteria), and structure alignment of the "top" model to structurally similar proteins.

Structure-sequence variation with StralSV: To assign enzymatic and functional annotations using structural comparisons of known structures against an AS2TS-generated model, we employ the StralSV submodule. StralSV is capable of assigning protein function annotation from structural (as opposed to sequence) homology even in cases where overall fold conformations between two structures are dissimilar. StralSV output includes a sequence-structure profile from the comparisons, and we use this data, and data from SeqalSV (below) to measure the level of structural conservation of various regions (spans) along the structure or structure model. This information can be particularly important for mechanistic modeling experiments, as highly conserved regions may be indicative of important binding sites; conversely, less conserved regions may comprise distinguishing structural features of a novel protein. Sequence variability statistics are written into each structure model forwarded from SPA by PFPmain to the Mechanistic Modeling (MM) module.

Structural clustering and annotation with StralCP: Additional structure comparisons are performed using the StralCP algorithm, which, like StralSV, is used to generate functional and enzymatic protein annotations. However, whereas StralSV relies on a fragment-based search for substructure homology, StralCP compares whole protein chains, thus providing a complementary aspect of protein annotation. Annotations from StralCP are detected by cluster, and clusters under which the model (or its template) claims membership are evidence of shared function between members of the cluster.

Sequence variation with SeqalSV: Similar to StralSV, we use the recently developed SeqalSV algorithm to measure sequence variability and conservation at a sequence-based level. Sequence-based variation is combined with structure-based variation from StralSV and reported as part of the model files created by AS2TS and exploited by MM for prioritizing analyses in specific regions of a protein.

Overview of the Mechanistic Modeling (MM) Module

The goal of mechanistic modeling is to predict the function of a protein from its chemistry. The mechanistic modeling (MM) module addresses certain protein details (such as catalytic function and the knowledge about which metabolite binds to a specific protein) to predict function. Fig. 7 illustrates the high-level MM workflow and the types of analyses performed in the module. Input to MM comprises high-quality protein structure models arising from SPA. MM uses a suite of modeling algorithms and inputs structural and function information to predict biochemical function; MM combines an understanding of the biochemistry, chemical and biological informatics, and physical chemistry of a given protein. The features of a protein both in terms of its physical properties and its structural similarity to proteins with known function are examined. MM results are then packaged in a pre-defined XML format and returned to PFPmain, which records the results of the module in the PFP central database.

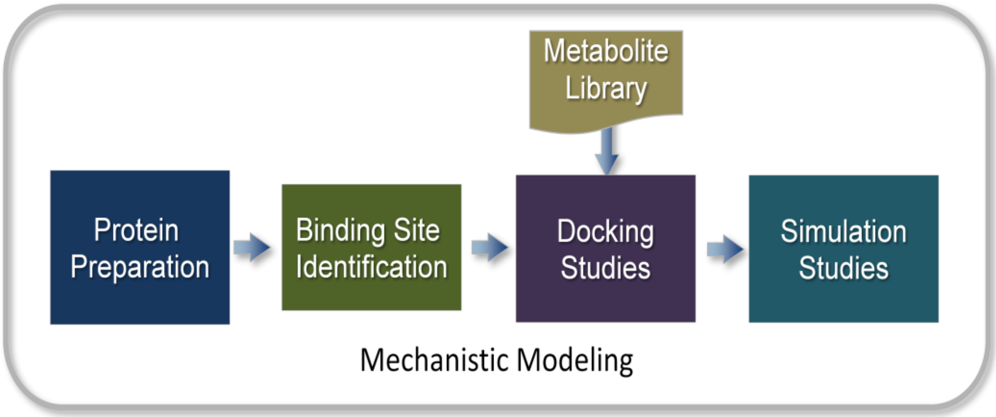


Fig. 7. Simplified MM Workflow. Input to the module is a protein structure of unknown function, and output is a prediction of protein function, along with a list of potential substrates and refined protein structures. Each box represents one or more codes (LLNL original, open-source, and licensed) applied in the given analysis.

The Design Concept

Function predictions generated by MM processing include enzyme classification (EC) numbers, the names of substrates and products (metabolites) associated with a protein, and structural representations of the binding interactions between relevant metabolites and the protein. Our novel approach to protein function prediction relies heavily on the biophysical properties of both the protein and the metabolites of interest. The workflow depicted in Fig. 7 is constructed using LLNL custom software to manage data flow between each software component and differential processing of proteins through the pipeline.

MM Components

Protein Preparation: The starting point of any structural study begins with a standard cadre of preparatory steps. Regardless of the quality of a protein structure model provided for MM processing—indeed, even if a crystal structure is provided, each still needs to be prepared for MM processing. First we determine that the structure is complete with no missing side-chains or loop regions (some missing residues at the N- and C- termini are usually acceptable), and assign hydrogen positions, with appropriate pH considerations. We also determine whether hetero-atomic and/or ion coordinates are needed for a complete structural description, and which of those are necessary to be included in molecular docking (described below). These preparatory steps are fully automated in the current version of the MM module. Considerable effort has been invested at the interface between the Structure Modeling group and the Mechanistic Modeling group to assure that data flows seamlessly through this preparatory step, and this has proven to be a robust procedure in the context of the present workflow. The output of this module is a protein that can be read by virtually any molecular modeling package. We currently use many of the default settings of the Schrodinger software package to accomplish this task.

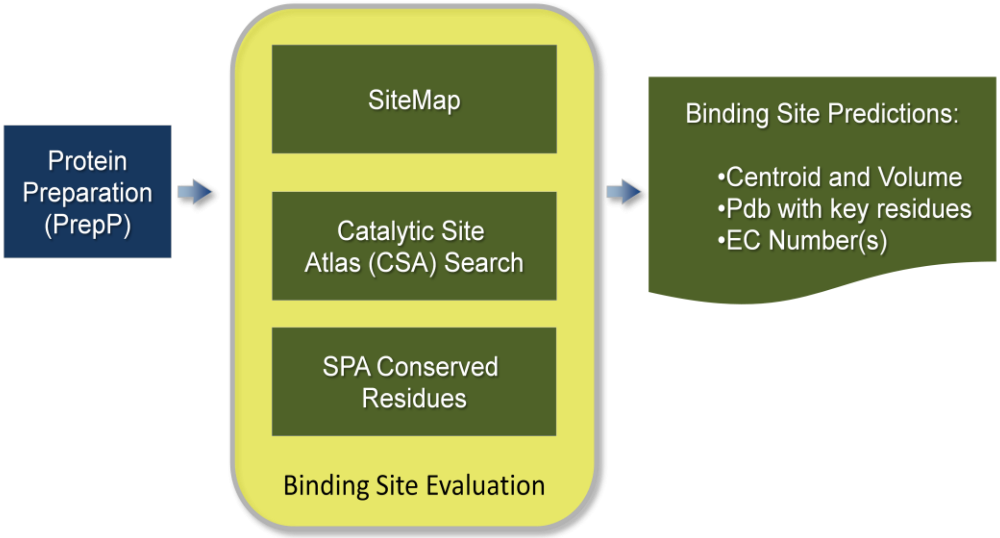


Fig. 8. Binding Site Identification within MM. This component comprises existing software packages (SiteMap) as well as novel approaches coded by LLNL researchers (CSA Search, SPA Conserved Residues) combined into a component workflow.

Binding Site Identification: The binding site identification steps are intended to provide as much knowledge about the protein binding site as possible prior to performing the computationally intensive down-stream tasks. This component of the MM module incorporates many new in-house methodological and algorithmic developments. Two key elements of binding site characterization are a) the location and size of the functional binding site, and b) the location and function of the catalytic residues. Item a) is accomplished using the SiteMap (Schrodinger package; Halgren 2007; Halgren 2009), which identifies the best binding site in a protein automatically by studying the biophysical properties and geometry of the protein and all possible positions that a binding site may occupy. This approach has been shown to be very reliable at identifying binding sites and is a completely automated procedure, which is essential for large scale studies. Item b) relies on annotated structural information about catalytic sites. Currently, this data comes from the Catalytic Site Atlas (Porter 2004). We have developed in-house code to rapidly screen databases of binding sites to identify those which are most similar to the protein of interest. A number of metrics, including the proximity to the SiteMap prediction, are employed to further confirm the hypothetical binding site prediction. The resulting predictions include an EC number and putative catalytic residues. Since an EC number is associated with a known substrate, we can use this knowledge to inform our docking predictions (described in the next section).

Molecular Docking: Molecular docking takes a prepared protein and a library of chemical compounds as inputs, and determines which compounds fit best in the predicted binding site. The current docking procedure uses a library of metabolites, under the hypothesis that the best fitting molecules will be the true metabolic substrates. Because a substrate is associated with an EC number, the substrate itself provides useful annotation information. More importantly, detailed information about the biophysics of the interaction of ligand and protein (specifically, in the coordinates and energetics of the metabolite poses) are obtained. This provides insight into interactions that are unavailable through any other means, and as a result is often considered the workhorse of any computational effort. Due to the complexity of these calculations, well-validated and widely accepted software tools are required.

Software platforms currently used for MM docking are the Schrodinger Glide (Friesner 2004; Halgren 2004) and Autodock (Goodsell 1996) programs, which run on high performance platforms through the DOVIS (Jiang 2008) software package. Glide is a commercial package with documented performance, and the Autodock/Dovis suite is open source, with well established protocols for high performance computing. Our strategy is to use Autodock for large-scale calculations of the full metabolite library, while using Glide for more refined and targeted calculations with library subsets.

Simulation Studies: While molecular docking studies are the mainstay of a protein-ligand interaction study, they are an approximate approach to calculating and understanding binding. A better binding affinity calculation will account for the motion of a protein in an explicit solvation environment and for fluctuations between the protein and ligand, which may affect stability in unforeseen ways. Simulation studies allow detailed questions to be answered in a rigorous (yet automated) way. Since a simulation requires substantially more compute resources than does a docking study, a manually selected subset of docking outputs are subjected to simulations. From among numerous available software packages, we selected Amber (Case 2008), and NAMD (Phillips 2005) because they are scalable to thousands of processors.

Algorithmic and Methodological Advances Achieved in Years 1 and 2 of MM Development

Catalytic Site Atlas (CSA) Search Program: The Catalytic Site Atlas (CSA) Search program uses the CSA database of the Thornton group (Cambridge, UK). A new code was developed at LLNL for this purpose. The method comprises a novel search algorithm, which has been extensively tested and validated. The code scales linearly, and has been adapted to run on multiple processors on

Livermore Computing (LC) resources. This method has shown significant promise in automatically characterizing binding sites, and a publication describing the methodology is currently in preparation. We have found that this procedure is very useful for identifying gene mis-annotations, and an additional publication reporting this study is expected to follow. As the utility of this method is established through peer review, an expanded curation of the CSA will allow for more complete studies, and additional publications resulting from this line of research are expected.

Novel Docking and Rescoring Strategies: We are testing additional docking methodologies that may leverage the large-scale resources available through LC. We are nearly finished with the incorporation of the Autodock/Dovis docking suite into the MM pipeline, and will likely implement a related open source program (Vina[11]) in order to generate comparative studies.

Much work is being done on both the pre- and post-processing sides of the docking protocols to best interpret the results. For pre-processing, we will try to develop additional fitness metrics based on our acquired knowledge of enzymatic function in order to predict docking performance prior to running full-scale docking procedures. On the post-processing side, we are exploring the utility of various rescoring procedures available through the Amber molecular modeling package, including the widely used MMGBSA procedure and the LMOD flexible docking procedure. Additionally, we intend to incorporate knowledge about the known binding modes to evaluate docking protocol fitness on subsets of chemical libraries.

Summary of Current Open-source and Licensed Software Packages Used in MM:

- Schrodinger (Protein Preparation, SiteMap, Glide)
- Autodock/Dovis (Docking)
- Amber(Docking and rescoring)
- Desmond, Amber, NAMD (Molecular Dynamics)

Most of the above packages are open source, or (in the case of Amber) require a one-time purchase fee. The Schrodinger package, however, uses token-based licensing, whereby each process uses a number of tokens during runtime. This limits the number of processes that can be run simultaneously to the number of tokens that are purchased.

Hardware and Memory Requirements for MM Processing: There are no special RAM requirements for many of the MM processes, as they are usually designed to run easily on commodity CPUs. A typical requirement of about 2G of RAM per CPU should be sufficient for most processes described. Protein preparation has nominal requirements; the current procedure runs in minutes on a commodity CPU. However, it is a token-limited license and therefore highly parallel processing is cost prohibitive. Binding site characterization currently takes about 2 hours on 16 CPUs running on LC. Many algorithmic and coding advances should reduce this time. About 0.5G of storage per protein is required. Docking studies are limited by available Glide software token availability, which will far outweigh memory or CPU requirements. Autodock currently takes about 6 hours per protein for a compound library of approximately 30,000 metabolites. These wall-clock times are estimated from typical runs using 192 CPUs on LC (hera cluster). This is readily scalable to larger numbers of processors per protein, thus decreasing wall-clock time linearly. Simulation studies can include a variety of molecular mechanics approaches up to and including a fully flexible solvated complex simulation. For software packages such as Amber, many calculations of intermediate complexity can be carried out, resulting in memory and CPU requirements somewhere between a docking study and a full simulation. For a fully solvated protein-ligand complex, a single simulation typically requires a few days on 100 CPUs and storage exceeding 5G. In general, an allotment (excluding simulation data at present) of 0.5G to 1G per protein should provide adequate storage. For a 4,000 protein dataset, this is roughly 2 to 4 Tb. We could dedicate resources to optimizing storage and reduce this requirement to 1 Tb if necessary.

Database Dependencies and Curation Issues: Our chemical library of metabolites is derived from the Kegg database and uses the Kegg identifier naming convention. The full database, after preparation using Schrodinger software, contains approximately 37,000 metabolite entries. We also rely on the database of information provided by Kegg to map substrate and product identifiers to EC numbers. We currently use the Catalytic Site Atlas, which provides a list of PDB identifiers, catalytic residues, and associated EC numbers. The remainder of the database, which is comprised primarily of distance matrix files, is generated from this table and the protein files downloaded from the Protein Databank.

Hardware

In anticipation of serving DTRA performers in the area of medical countermeasures, we purchased new hardware to replace a small development machine and aging shared systems that are currently housing PFP v.0.5. This hardware consists of two "cluster master" nodes (one for development, one for production), each consisting of dual Intel Xeon X5690 6-core 3.46GHz processors with 12MB cache, 6.4GT/s QPI; eight Silicon Mechanics "cluster" nodes (1 for development and 7 for production), each consisting of a Rackform iServe R331.v2, dual Intel Xeon X5690 6-core with 3.46GHz processors, 12MB cache, 6.4GT/s QPI, 96GB DDR3-1333 ECC registered DIMMS, and a 1TB drive (6Gb/s); a database server, consisting of a Rackform iServ R346.v.2.1 with dual Intel Xeon X5660 6-core 2.8GHz processors, 12MB cache, 6.4GT/s QPI, 24GB DDR3-1333 ECC registered DIMMS, mirrored 500GB drives for the operating system, and 10TB RAID6 storage for the database; an enclosure RAID expansion for external storage to be connected to the database server, consisting of 26TB RAID6 storage to be used for shared data that will be NFS mounted on clusters. Once set up, this hardware configuration is intended to be used for development and deployment of PFP. In addition, LLNL's LC systems can eventually also be used for very high-throughput processing.

Future Directions and Request for Continued Funding

Our original mission goals under the TMT Program were to provide bioinformatics support to performers involved in design and development of therapeutic reagents. In this regard, we designed and partially built the PFP platform to be not only state-of-the-art with respect to bioinformatics analysis, but also modular, configurable, and expandable, to suit the needs of an evolving Program. Our Year 3 development was slated to complete platform development and to create user tools in service to the Program. When TMT performers were transitioned to DTRA's Diagnostics Division, we proposed to tailor our Year 3 development for identification and characterization of biomarkers for diagnostics. We maintain that this is a suitable outcome of PFP development, and we propose the following tasks, which we believe would enable a bioinformatics resource that would benefit multiple DTRA programs in medical countermeasures:

1. Port existing modules to new hardware (see above) to increase scale and performance of processing, and to allow for full automation to and from all the PFP subordinate modules. We have developed the prototype system (PFP v.0.5) on older, smaller hardware, and the full-scale capabilities of PFP will be realized only when the system is installed on new hardware that we purchased for service to DTRA medical countermeasure programs.

2. Perform thorough testing of all current PFP capabilities, including process control, modules, components, data flows, and UIs. Perform further testing of capabilities to be added in Year 3 development (see below).

3. Implement configurable and iterative workflow capabilities and differential processing of proteins through the PFP workflow, and improve user job control and system error recovery and reporting.

4. Design and implement automated updaters for PFPmain and all subordinate modules.

5. Complete integration of SB and SPA user interfaces with those of PFPmain; some of these links have been incorporated, but a more comprehensive, seamless integration between PFPmain and the UIs of the subordinate modules needs to be achieved.

6. Expand biomarker discovery capabilities within the SA, SB, and SPA modules, and develop post-processing logic within PFPmain to assist users in selecting biomarkers for further workup. SA: incorporate additional immunology tools (e.g., T- and additional B-cell epitope prediction codes; immune data sets; antigen identification algorithm currently under development via other funding). Design and develop post-processing codes for evaluation of antigenic characteristics, including structural overlay (SPA) on sequence-based predictions. SB: incorporate FBA modeling into PFP, and create post-processing logic for automated selection of potential therapeutic biomarkers based on identification of putatively essential genes; develop *in silico* default growth parameters, scale automated FBA modeling and analysis, link with SB module; incorporate alternative FBA modeling tools (i.e., SRI, Kegg, SEED) for automated comparison and scoring of likely biomarker candidates.

7. Integrate annotations being generated from all Structure Prediction and Analysis algorithms (StralSV, SeqalSV, StralCP and StralDF) to produce a scored and ranked consensus-based annotation for proteins of interest.

8. Achieve higher resolution modeling of protein-ligand interactions. Once a docking study has been run and a set of candidate substrates is identified, it is often the case that more detailed modeling will be needed to fully predict the binding properties. Current capabilities allow for molecular dynamics studies, but not yet on a scale that provides genome level utility. We propose to perform the research and development to scale this predictive capability.

9. Expand the SB module to include human metabolism and virus-host interaction data. Fully exploit virulence information (MvirDB) for identification of virulence-related biomarkers in systems biology context.

References

Case, D., et al., AMBER 10. University of California, San Francisco, 2008. 32.

Goodsell, D.S., G.M. Morris, and A.J. Olson. 1996. Automated docking of flexible ligands: applications of AutoDock. *Journal of Molecular Recognition* 9(1): p. 1-5.

Friesner, R.A., et al. 2004. Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. *Journal of medicinal chemistry* 47(7): p. 1739-1749.

Halgren, T.A., et al. 2004. Glide: a new approach for rapid, accurate docking and scoring. 2. Enrichment factors in database screening. *Journal of medicinal chemistry* 47(7): p. 1750-1759.

Halgren, T. 2007. New Method for Fast and Accurate Binding-site Identification and Analysis. *Chemical biology & drug design* 69(2): p. 146-148.

Halgren, T.A. 2009. Identifying and characterizing binding sites and assessing druggability. *Journal of chemical information and modeling* 49(2): p. 377-389.

Jiang, X., et al. 2008. DOVIS 2.0: an efficient and easy to use parallel virtual screening tool based on AutoDock 4.0. *Chem Cent Journal* 2(18).

Phillips, J.C., et al. 2005.Scalable molecular dynamics with NAMD. *Journal of Computational Chemistry* 26(16): p. 1781-1802.

Porter, C.T., G.J. Bartlett, and J.M. Thornton. 2004. The Catalytic Site Atlas: a resource of catalytic sites and residues identified in enzymes using structural data. *Nucleic acids research* 32(suppl 1): p. D129-D133.

Trott, O. and A.J. Olson. 2010. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry* 31(2): p. 455-461.

Zemla, A., et al. 2005. AS2TS system for protein structure modeling and analysis. *Nucleic acids research* 33(suppl 2): p. W111-W115.

Zemla A, DM Lang, T Kostova, R Andino, and CL Ecale Zhou. 2011. StralSV: assessment of sequence variability within similar 3D structures and application to polio RNA-dependent RNA polymerase. *BMC Bioinformatics* 12:226. Doi:10.1186/1471-2105-12-226.

Zemla, A, J Smith, M Lam, B Kirkpatrick, M Wagner, T Slezak, B Geisbrecht and CE Zhou. 2007. STRALCP: structure alignment-based clustering of proteins. *Nucleic Acids Research* 35, doi:10.1093/nar/gkm1049.