UltraScan Solution Modeler: Integrated Hydrodynamic Parameter and Small Angle Scattering Computation and Fitting Tools

Emre Brookes*, Raminderjeet Singh†, Marlon Pierce†,
Suresh Marru†, Borries Demeler* and Mattia Rocco‡

*University of Texas Health Science Center at San Antonio
San Antonio, Texas
{emre,demeler}@biochem.uthscsa.edu

†Pervasive Technology Institute
Indiana University
Bloomington, IN, USA
{ramifnu,smarru,marpierc}@iu.edu

‡Biopolimeri e Proteomica
IRCCS AOU San Martino-IST
Genova, Italy
mattia.rocco@istge.it

ABSTRACT

UltraScan Solution Modeler (US-SOMO) processes atomic and lower-resolution bead model representations of biological and other macromolecules to compute various hydrodynamic parameters, such as the sedimentation and diffusion coefficients, relaxation times and intrinsic viscosity, and small angle scattering curves, that contribute to our understanding of molecular structure in solution. Knowledge of biological macromolecules' structure aids researchers in understanding their function as a path to disease prevention and therapeutics for conditions such as cancer, thrombosis, Alzheimer's disease and others. US-SOMO provides a convergence of experimental, computational, and modeling techniques, in which detailed molecular structure and properties are determined from data obtained in a range of experimental techniques that, by themselves, give incomplete information. Our goal in this work is to develop the infrastructure and user interfaces that will enable a wide range of scientists to carry out complicated experimental data analysis techniques on XSEDE. Our user community predominantly consists of biophysics and structural biology researchers. A recent search on PubMed reports 9,205 papers in the decade referencing the techniques we support. We believe our software will provide these researchers a convenient and unique framework to refine structures, thus advancing their research.

The computed hydrodynamic parameters and scattering curves are screened against experimental data, effectively pruning potential structures into equivalence classes. Experimental methods may include analytical ultracentrifugation, dynamic light scattering, small angle X-ray and neutron scattering, NMR, fluorescence spectroscopy, and others. One source of macromolecular models is X-ray crystallography. However, the conformation in solution may not match that observed in the crystal form. Using computational techniques, an initial fixed model can be expanded into a search space utilizing high temperature molecular dynamic approaches or stochastic methods such as Brownian dynamics. The number of structures produced can vary greatly, ranging from hundreds to tens of thousands or more. This introduces a number of cyberinfrastructure challenges. Computing hydrodynamic parameters and small angle scattering curves can be computationally intensive for each structure, and therefore cluster compute resources are essential for timely results. Input and output data sizes can vary greatly from less than 1 MB to 2 GB or more. Although the parallelization is trivial, along with data size variability there is a large range of compute sizes, ranging from one to potentially thousands of cores with compute time of minutes to hours.

In addition to the distributed computing infrastructure challenges, an important concern was how to allow a user to conveniently submit, monitor and retrieve results from within the C++/Qt GUI application while maintaining a method for authentication, approval and registered publication usage throttling. Middleware supporting these design goals has been integrated into the application with assistance from the Open Gateway Computing Environments (OGCE) collaboration team. The approach was tested on various XSEDE clusters and local compute resources. This paper reviews current US-SOMO functionality and implementation with a focus on the newly deployed cluster integration.

Categories and Subject Descriptors

A.0 [**General**] Conference Proceedings. J.3 [**Applications**] Life and medical sciences - *Biology and genetics*

General Terms

Algorithms, Design, Security.

Keywords

Bead modeling, hydrodynamics, analytical ultracentrifugation, small angle scattering, structural biology, Apache open source community, Apache Rave, Apache Airavata, Open Gateway computing Environment

1. INTRODUCTION

1.1 Purpose

Understanding the functions of individual and functional collections of biological macromolecules is fundamental to the prevention and treatment of diseases. Identifying a key molecule in a disease pathway enables the possibility of the development of molecules (inhibitors) which can bind and thus potentially block the pathway. For example, the inhibition of the enzyme tyrosine kinase by the drug Imatinib in the treatment of chronic myelogenous leukemia is a recent paradigmatic success story [1,2]. A first step to understanding the function of a biomacromolecule is to know its structure. Towards this end, various experimental methods provide structural information of varying accuracy and precision. It may be possible to grow a crystal and use x-ray crystallography to determine a complete structure. The x-ray determined structure is representative of the structure in the specific crystal, but it could undergo subtle or relatively large conformational changes in solution. There, a biological macromolecule is a dynamic system that is continuously moving in a variety of modes, assuming a variety of conformations dependent on the environmental conditions. The

structures obtained from crystals thus should be validated against other sources of experimental data. The alternative method of nuclear magnetic resonance (NMR) can instead provide structures in solution, but the data are usually collected at quite high macromolecular concentrations and often in non native conditions, such at low pH. Furthermore, deriving structures with NMR is currently limited to macromolecules <50 kDa in size. Finally, in the absence of high-resolution data, it is sometimes possible to obtain a homologous structure from the sequence of a protein, usually determined from genomic data or by Edman degradation/mass spectrometry. However, the homologous structure needs to be validated against further experimental data.

The overarching goal of our software is to provide an extensible general framework for generating collections of candidate structures from an initial structure or structures, modeling candidate structures under various experimental methods and conditions, and subsequently globally fitting and screening candidate structure's models against sets of experimental data.

Excepting the "extensible general framework", which is in planning, it is possible to perform the steps of our goal in the current software release for a defined set of experimental methods and parameters.

1.2 UltraScan and SOMO

UltraScan [3,4] was originally developed by B.D. in 1989 as a package for the analysis and management of analytical ultracentrifugation (AUC) experiments. The current software is a GUI application written in C++ utilizing Qt [5]. The code is multi-platform, with binaries available for Linux, OSX and Windows. Source is available via a wiki integrated subversion repository. The current user base includes approximately 700 registered individual biophysical and biomedical researchers and 56 registered laboratories world-wide.

In 2006, the 2D spectrum analysis [6,7,8] and a genetic algorithm method for parsimonious regularization [9,10] were added to UltraScan-II. These MPI [11] based parallel methods require high performance computing infrastructure. Initially installed on UTHSCSA's bioinformatics core facility with a home-brew queuing and gateway solution, the usage demands of the software expanded to Texas-wide resources via HiPCaT [12], facilities at the Texas Advanced Computer Center [13] and eventually on to TeraGrid as a Science Gateway, relying on Globus WS-GRAM 4 [14]. A TeraGrid ASTA and subsequent National Science Foundation OCI grant provided support to transition the gateway to use the Open Gateway Computing Environment's GFAC [15] component, as a replacement to the now obsolete WS-GRAM 4. In 2010-2012, the code was completely rewritten with a focus on clean code and released as UltraScan-III [16]. This includes a new gateway infrastructure integrating GFAC by design, as opposed to the ad-hoc layering of GFAC offered in the previous version.

Table 1: Brief history of UltraScan

Year	Event	Platform	Code	Grid utilization	Gateway
1989	UltraScan	MS-DOS	C		
1996	UltraScan-II	MS-DOS → Linux	C++, Qt 2		
2004	UltraScan-II ver. 9	+ Mac X11 + Windows	Qt 2 → Qt 3		
2006	+ 2DSA		+ MPI	Local cluster	Home Brew (perl and php)
2007	+ GA			+ TACC	
2008		+ Max OSX		+ TeraGrid	
2009	+ US-SOMO				
2010					Home Brew → OGCE/GFAC
2011	UltraScan-III		Qt 3 → Qt 4	TeraGrid → XSEDE	
2012	+ US-SOMO / Cluster				

In 2009, E.B. and B.D. began collaboration with M.R. to integrate a bead modeling software, Solution Modeler (SOMO, developed by M.R. and O. Byron at the Glasgow University, UK) [17] into UltraScan as US-SOMO [18,19,20]. US-SOMO computes hydrodynamic parameters from structural representations of macromolecules, a natural fit into UltraScan, which computes hydrodynamic parameters from experimental data. The combination of these methods enables one to screen hydrodynamic parameters computed from structure against those derived from experimental data. In 2010, E.B. began integrating small angle x-ray scattering (SAXS) [21] and small angle neutron scattering capabilities (SANS) [22], collectively (SAS), into US-SOMO. SAS methods provide additional structural information about molecules in solution. With SAS tools, SAS experiments can be modeled from structure, enabling another method for screening structures against experimental data. As a method to expand the space of possible structures and to model the local motions of molecules in solution, discrete molecular dynamics (DMD) [23,24] capabilities were added. Also newly added is the ZENO [25] method for hydrodynamic computations from macromolecular models. These additional methods drove the implementation of, at first, efficient methods for computing parameters from large numbers of structures within the application, named "batch", to the recently implemented "cluster" methods, which packages jobs, submits them to cluster resources, monitors job status, retrieves packaged results, and extracts their contents. An historical summary of UltraScan is shown in Table 1 and the organization of the software is shown in Figure 1.

Based upon the experience with the UltraScan gateway evolution it was decided to integrate the cluster facilities directly within the application. In contrast, within the standard UltraScan gateway, the user must synchronize their application data with a database, switch from the application to a web browser to submit and monitor the job, and finally retrieve the results back to the application. Integrating the cluster facilities within the application simplifies matters and creates a seamless experience for the users which increases their productivity and simplifies training. The opposite choice of pushing the entire application to a browser based interface would require a fundamental rewrite of the application code, and it is not clear to the authors whether or not sufficient capabilities currently exist to cover the current application's functionality, particularly the advanced plotting and 3D molecular viewing methods, although Jmol [26] may be a promising candidate for the latter. A third option of bringing the browser inside the application is being investigated, with the potential of enabling a smooth pathway to a browser based interface. The QtWebKit [27] and Qt5's [28] JavaScript support offer possibilities towards this end. Starting with a relatively basic, yet powerful, set of functions, the US-SOMO program has grown considerably and is now a major component of the UltraScan code.

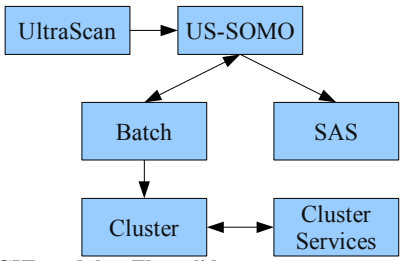

Figure 1: GUI modules. The solid arrows are user navigation paths. The US-SOMO builds bead models and computes hydrodynamic parameters. SAS provides small angle scattering computations. Batch selects input files and processing options and can process locally or forward to Cluster. Batch and Cluster are described in section 3.

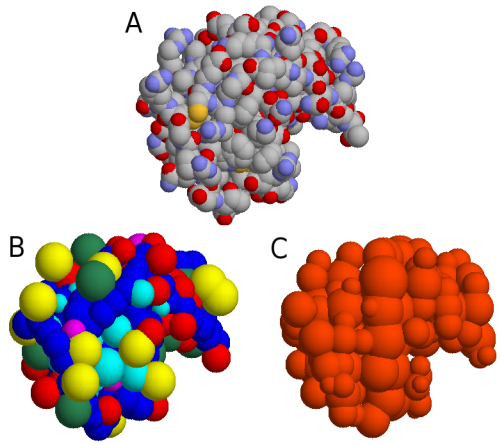

Figure 2: Three representations of molecular structure. A) PDB (1HEL) atomic structure with each sphere representing an atom. B) a SOMO bead model of 1HEL with 2 spheres (beads) representing each residue. C) An AtoB grid model of 1HEL with each sphere (bead) representing an collection of atoms based upon their presence within a box in space. No radial reduction was performed when generating the beads.

1.3 Atomic Structures and Bead Models

A biological macromolecule's structure consists of collections of atoms positioned in space. In the strictest sense, a molecule consists of covalently bound atoms. Generally, biological macromolecules consist of multiple "molecules" known as chains, as well as bound ions such as calcium and molecules such as water. Chains are generally composed of sequences of "residues". For example, in proteins, the residues are the amino-acid polymers, and in RNA/DNA, the residues are ribo- and deoxyribo-nucleotides. The standard for electronic representation of a biological macromolecule's structure is the Protein Data Bank (PDB) [29] format. This is a text file format that contains chain, residue, atom type and coordinate information grouped into models. A coarser grained approach to the structure of molecules is the bead model. In this case, multiple atoms are represented by an individual bead (see Figure 2). Bead modeling has multiple purposes. Bead models may be used as best representation of experimental data when the atomic structure is unknown, for instance starting from electron micrographs or SAXS/SANS-derived envelopes. Bead models can also be used as computational efficiency tools to create a lower-resolution representation of the structure when atomic detail is not required.

1.4 Discrete Molecular Dynamics

Discrete molecular dynamics (DMD) [23,24] is an implementation of molecular dynamics by N. Dokholyan and collaborators. Traditional molecular dynamics (MD) takes an atomic structure and simulates it under physically realistic conditions. DMD modifies this by discretizing the potential function. In the limit of increasing discretization step count, DMD is equivalent to MD. In practice, discretizing the potential function allows much faster simulations with some loss in accuracy. This loss is of little concern when the purpose is to take an initial structure and expand it into a search space of candidate structures.

1.5 Experimental Background

Each experimental method provides specific information with varying degrees of accuracy and precision. They could range from single variable parameters such as the molecular weight, radius of gyration, intrinsic viscosity, partial specific volume, relaxation time, sedimentation and diffusion coefficients, and frictional ratio, to more detailed structural information such as bead models and complete atomic-resolution structures. We will briefly describe below a few of the techniques currently relevant to US-SOMO and the parameters that can be determined.

1.5.1 Analytical Ultracentrifugation

In analytical ultracentrifugation (AUC) [30], a sample in solution is placed in a sector-shaped cylindrical cell with top and bottom transparent windows. The cell is vertically placed in the AUC rotor and spun at speeds up to 60,000 revolutions per minute. The instrument provides an optical path for imaging the cell during the experiment, detecting the concentration distribution of the sample. Images are taken periodically, providing a time series of radial concentration profiles. This can be recorded by any one of three different optical systems: UV/visible absorbance, Rayleigh interference and fluorescence emission detection, allowing the investigator to exploit a range of chemical properties from different types of samples. The two primary physical processes affecting the sample are sedimentation towards the exterior ("bottom") of the cell and random diffusion. There are two main modes for running an AUC experiment, one is known as the velocity experiment, when the sample is tracked during its movement from a uniform radial concentration through zonal depletion, until it is all concentrated at the bottom of the cell. The other is an equilibrium experiment, where the sample is spun at relatively low speed until an equilibrium state between sedimentation and diffusion is attained and examined. From the analysis of AUC velocity experiments it is possible to determine the sedimentation and diffusion coefficients. From the analysis of AUC equilibrium experiments it is possible to determine the molecular weigh of the sample, which depends also on its partial specific volume. The analysis of AUC velocity and equilibrium experiments is fully supported by the UltraScan package.

1.5.2 Small Angle Scattering

In SAS experiments, a monochromatic beam targets a sample in solution and a scattering pattern is observed. In SAXS experiments the beam is x-ray light; in SANS experiments the beam is composed of neutrons. X-rays are scattered by electron clouds and neutrons are scattered by collisions with nuclei. A two dimensional detector records the scattering pattern. In solution SAS experiments, the particles are randomly oriented and the two dimensional image is radially integrated to form a one dimensional curve, known as the scattering intensity curve. The curve is generally reported in scattering intensity vs. units of momentum transfer, q, and the result is the scattering profile, I(q) vs. q curve. In post processing of the data, an inverse Fourier transform is applied to produce the real space radial distribution function, P(r) vs. r. The radial distribution function has a geometric interpretation as a histogram of the distances between pairs of scattering centers (electron clouds or nuclei). From the produced scattering curves, the radius of gyration and molecular weight can be determined. Advanced methods are available which can compute bead model representations of the structure based upon the scattering curves [31,32].

1.6 Hydrodynamic Computations

Bead modeling methods were developed starting in the late 1960s by Bloomfield and collaborators (see [33] and references therein). In these methods, a macromolecule is represented by a collection of *n* beads of a certain radius, appropriately positioned in space. While the frictional force exerted by each bead on the solvent is straightforward to compute from the Stokes-Einstein

relation, the motion of each bead creates an additional internal velocity field in the solvent. A "hydrodynamic interaction" tensor taking this perturbation into account has been developed for non-overlapping beads of different sizes [34]. The frictional properties of the ensemble are then calculated by solving a system of N linear equations with 3*N unknowns. The 3*N unknowns are the components of the frictional force vector for each bead.

US-SOMO initially contained only a bead modeling utility that was originally developed by the Rocco and Byron labs [17]. The original code was mainly written by B. Spotorno, G. Tassara, N. Rai and M. Nöllmann. The bead modeling utility in SOMO is based on a reduced representation of a biomacromolecule, starting from its atomic coordinates (PDB format), as a set of non-overlapping beads of different radii, from which the hydrodynamic properties can be calculated using the Garcia de la Torre-Bloomfield (GTB) rigid-body approach. The reduced representation is afforded by grouping together atoms and substituting them with a bead of the same volume, appropriately positioned. Importantly, the volume of the water of hydration theoretically bound to each group of atoms can then be added to each bead. The overlaps between the beads are then removed in sequential steps but preserving as much as possible the original surface envelope of the bead model. The method has been fully validated and reported in the literature [17,18,19]. Among the main advantages of this method over other methods such as shell-modeling [35] and grid-based procedures, like AtoB [36], are a better treatment of the hydration and the preservation of a direct correspondence between beads and original residues. For instance, the latter feature could be used to include flexibility effects into the computations using Brownian dynamics [37]. Furthermore, by identifying and excluding from the hydrodynamic computations beads that are buried and thus not in contact with the solvent, a large span in the size of the structures that can be analyzed with this method without loss of precision is obtained: currently, structures from 5 KDa to 250 KDa and above have been successfully studied (e.g. [38,39,40]). We have improved the original AtoB grid method, included within US-SOMO, by adding the theoretical hydration, accessible surface area screening, and a better preservation of the original surface. Alternatively, the recently added ZENO method can be used to calculate hydrodynamics based on an approximate analogy between hydrodynamic and electrostatic properties [25,41,42].

2. SERIAL METHODS

2.1 Discrete Molecular Dynamics

A discrete molecular dynamics simulation can be run by loading a structure and selecting to run DMD on the main US-SOMO window. A panel will appear where appropriate run parameters can be set, such as the duration, temperature and the number of "snapshots" of the simulation requested. Each "snapshot" is a PDB of the structure at a time during the simulation. The DMD source code is not publicly available, and it is only installed on the cluster resources. Although DMD is quite efficient, simulations can be time consuming and therefore, DMD simulations are primarily run on our 238 core cluster "Alamo".

2.2 Hydrodynamic Computations

The initial function of SOMO was to perform hydrodynamic calculations on a single PDB or bead model file. To compute hydrodynamic parameters from an atomic structure, the user simply loads the structure from a PDB format file. A molecular viewer showing the structure is automatically initialized with the structure. There are currently two methods to compute bead models from the atomic structure. The SOMO methods, in which a structure is converted into a bead model representation via a direct residue correspondence using a customizable lookup table, and the AtoB method, in which generic beads are produced based upon the geometric positioning of atoms in the structure (see Figure 2). Each method is highly configurable, but with well established default values. The user selects one of these methods to produce the bead model. Multi-model PDB files will produce multiple bead models. The user may have a prepared bead model from a previous computation within US-SOMO or from some external program. These can also be directly loaded.

The user then computes the hydrodynamic parameters. This initializes the hydrodynamic computations in either the default GTB or the ZENO method based on current settings. The complete results are written to a text file and selected computed parameters are available for direct viewing within the application. A comma separated format (CSV) for the output is also available allowing the user to select from a current set of 49 computed/preset parameters. The CSV file can be loaded in any standard spreadsheet program.

Computationally, generating a bead model with the AtoB method is proportional to the volume of the structure. Generating a bead model with SOMO, requiring the lookup and assignment of residue to bead correspondence, is O(number of atoms). Subsequent to the generation of bead models, both methods

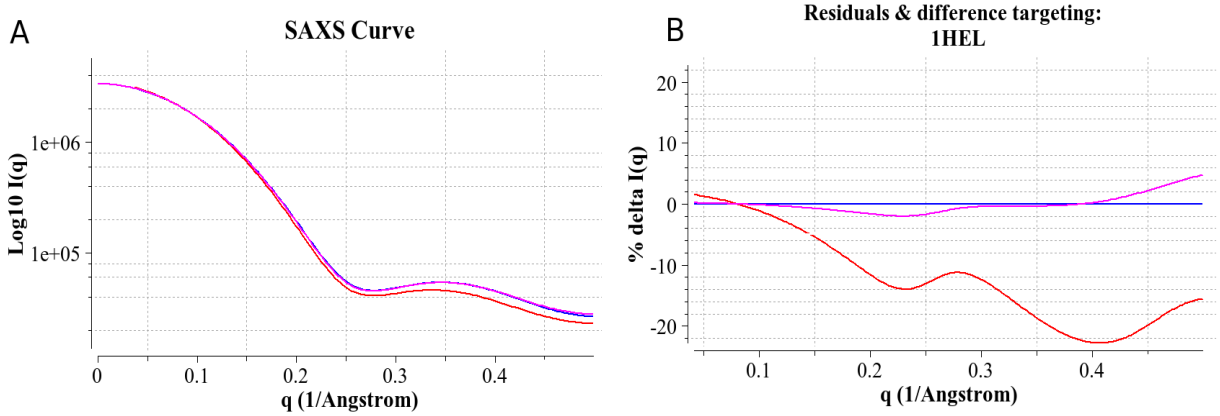

Figure 3: SAXS curves computations and their differences. The computation was performed on PDB 1HEL. Four computations were performed, but only three are visible due to superimposition of the full Debye method with the Hybrid method. Red is FoXS. Magenta is CRYSOL. Dark blue is Hybrid. All methods were computed with no hydration to the structure. A) The scattering intensity curves. B) The percentage differences vs. full Debye computation.

require reduction of overlapping radii and which is O(number-of-beads^2). Computation times for a single structure are typically in the range of less then one minute to tens of minutes for a larger structure.

The hydrodynamics calculation is generally the limiting step. In the GTB method, the tensor equations require a Cholesky decomposition which is O(number-of-beads^3). The ZENO method, although being O(number-of-beads), can be much slower due to the large number of random walks required to accurately compute the parameters. Computation times for the hydrodynamics calculation are typically from minutes to hours.

2.3 Small Angle Scattering Computations

As with the computation of hydrodynamic parameters, small angle scattering requires loading of an atomic structure or bead model. The user then enters the SAXS functions where a choice among several computation methods is available. The methods for scattering curve computation range from a full Debye calculation, a Hybrid method, a Fast method and external methods FoXS [43,44] and CRYSOL [45]. The full Debye calculation is the most accurate, but can take 30 minutes to several hours to compute, and requires an explicit representation of the water of hydration of the macromolecule. The external FoXS method and its internal Fast implementation are significantly faster, but at the cost of accuracy. The external CRYSOL method is based upon a spherical harmonic approximation and is intermediate in accuracy between the full Debye and Fast methods. The scattering curves and their differences are shown in Figure 3. Radial distribution curve computations are also available for both SAXS and SANS.

2.4 Fitting Methods

Results from hydrodynamic computations can be compared to experimental values within the model classifier utility of US-SOMO. Multiple CSV files containing saved parameters can be easily loaded. Relevant hydrodynamic experimental values can be entered. Ranking of models can be performed in three basic ways. The first method is by manual rank of parameters, using an absolute or a percentage difference. In this method the highest ranking parameter's difference becomes the primary sort key, the next highest ranking parameter's difference the secondary sort key, etc. The second method ranks by the weighted sum of absolute or percentage difference. In this way, multiple parameters contribute to the overall fit, and parameters from experiments with higher confidence can be given a greater weight. The third method groups by equivalence classes. In this method parameter ranges are individually partitioned and the distance of a particular parameter is measured by the distance in partitions from the parameter value's partition to the experimental value's partition. Fitting multiple parameters simply sums up the partition's distances and the results are ranked in ascending distance. The results from any of the three fitting methods are written to a CSV format file with the additional fitting columns and experimental values appended. An internal CSV viewer is also included.

Small angle scattering results are compared directly with a target experimental scattering intensity curve or the processed real space curve. A CSV file can be loaded containing multiple model curves and a target curve selected. The user can choose comparisons of either best fit or non-negative least squares to rank the candidate structure's curves.

Hydrodynamic computations and small angle scattering data can be computed for each candidate structure and fit against experimental data. Simultaneous fitting of hydrodynamic and small angle scattering will be supported in a future release.

3. PARALLEL METHODS

3.1 Batch Module

The batch module of US-SOMO allows the user to select a large number of initial structures for computation of bead models, hydrodynamic parameters, and small angle scattering curves. The processing occurs on the users' workstation. It is a convenient method for the user and researchers have reported processing as many as 10,000 structures through the batch module [46]. Hydrodynamic computations on large numbers of even relatively small structures can take days to finish. Results for both hydrodynamic parameters and small angle scattering curves can be combined into CSV files for subsequent fitting.

Although useful, the batch module is insufficient for the following reasons. Computing hydrodynamic parameters and SAXS curves for large structures can take hours. DMD software is only available for 64 bit Linux systems and can not be distributed to the end users. For example, M.R. is investigating a large (253 kDalton) protein. Computing a single 200 point full Debye SAXS curve on his workstation takes ~3 hours. DMD runs on this protein produced 100 structures. Computing these SAXS curves for all 100 structures would take ~12.5 days. Typical run times for a range of proteins are shown in Table 2.

Table 2: Serial run times. Times were computed on a Intel® Core™ i3 running at 2.53 GHz. US-SOMO defaults were used. SAS Debye curves were computed on 500 grid points, except for CRYSOL which is limited to 256 grid points.

PDB	8RAT	1HCO	1GZX	1ADO	4BLC
Molecular Weight	13.6 kDa	32.3 kDa	64.6 kDa	157.3 kDa	232.6 kDa
PDB Screening	0.1 s	0.3 s	0.7 s	1.1 s	1.8 s
SOMO Bead Model	1.7 s	4.0 s	8.9 s	25.4 s	43.4 s
AtoB Bead Model	1.3 s	1.9 s	3.9 s	18.4 s	41.5 s
GTB Hydro-dynamics	0.8 s	2.1 s	8.8 s	96.9 s	147.4 s
Fast Debye	0.8 s	1.8 s	3.4 s	9.7 s	13.1 s
CRYSOL*	13.2 s	15.7 s	20.3 s	38.7 s	51.5 s
Hybrid Debye	2.5 s	8.0 s	20.8 s	133.1 s	280.8 s
Full Debye	25.1 s	166.7 s	683.9 s	4,720.3 s	10,642.1 s

3.2 Cluster Module

The long computing times observed when large numbers of structures are processed with the batch module motivated the development of the cluster module, enabling the user to package a processing request and to submit it to parallel resources. The design is quite simple, allowing the user to select files and methods within the batch module and then to enter the cluster module. Within the cluster module, a package is created. The user then selects the created package(s) and the target computational resource and submits the job. The job can be monitored in a status window that supports canceling the job. When completed, the job results are retrieved to the user's workstation.

The parallelization currently supported is trivial, as each processor handles the computations based upon one or more structure and requires no interprocess communication. Although some of the processing methods have been threaded in the GUI application, for cluster processing the number of independent computations is large enough to dispense with individual job parallelization. Typical jobs consist of

computations upon conformational variants of one structure, implying similar run times for each computation. This provides submissions a wall time speedup approximately equal to the number of processors allocated. Deviations from this ideal include the time overheads of queuing, staging and retrieval, the minor overhead of GFAC service time, and processor idling due to early termination of individual sub-jobs. The package itself is a single tar file containing a collection of from a dozen to a thousand or more of gzipped tar files, one for each computation. Within the individual gzipped tar files is a text control file along with any job specific input files which typically range from one to ten in number. Files common to multiple jobs are placed in a common gzipped tar file included in the package to reduce overall size. The size becomes quite a constraint when processing large numbers of PDB files. Often these are quite similar with identical atoms, simply placed at different spatial coordinates. A differential compression mechanism would likely offer significant compression over gzip, but this has not yet been implemented.

When the job begins to run, the MPI controlling program expands the tar archive, and divides the total number of jobs among the available processors. An advancement on this method would be for the worker processes to pull from a queue of unprocessed jobs, providing better load balancing. This is not currently a major issue, as the jobs in a package are usually similar in computational time, due to the restrictions setup in the packaging GUI. When all the jobs are completed, the controlling process collects the results and repackages them into a single gzipped tar archive file, which is retrieved by the GUI application.

Initial testing of the job mechanism was restricted to two trusted test users running in a Linux environment. Staging of the job package and recovery of the results were performed by a system call to "scp" with private keys installed and copying directly to the scratch space of the target resource. Submission of the job was done via an HTTP/REST call. This is conveniently handled by Qt's QHTTP class. Although this methodology was functional for testing, it was not acceptable to distribute publicly due primarily to security concerns and secondarily to the reliance on an external program (scp), which would have to be packaged for Windows users or integrated into the code base. In this methodology, once a user is registered via HTTP and authorized to submit jobs by the administrator, the staging is done via ftp to a user-specific general staging location automatically setup on a virtual server (currently an IU Gateway Hosting Virtual Machine) utilizing Qt's QFTP class. The file has to be subsequently re-transferred to the target resource's staging area and this is handled by the services outside of the GUI application's concern. The second transfer of the package is somewhat wasteful, as the packages can be large. Discussions were had about mounting the appropriate scratch space of the target resource directly to the virtual ftp server, but this idea was rejected. Similarly, after the job has completed the services must return the job output to the virtual ftp server for user access.

3.3 Gateway Middleware Services

The UltraScan-II and UltraScan-III gateways as described in Section 1.2 are running in production using Open Gateway Computing Environment (OGCE) software [47]. In this work, we reuse the generic REST APIs for job management and file transfers and extend the functionality to include US-SOMO applications. The software enhancements are contributed back to the core software which will benefit UltraScan and other gateways supported by OGCE software. The OGCE software is now developed in an open community process by the Apache Software Foundation as Apache Airavata. The Apache open community model encourages contributions, and that users become vested stake holders in the software. Along with Apache Airavata, the US-SOMO project is built upon various other open source software and integrates them together to provide a uniform API. The underlying open source software Apache Rave, Apache FTP and Airavata's GFAC software are described further in this section. More developer specific implementation details are provided on the UltraScan wiki [48].

3.3.1 Software Components:

Apache Rave: Rave is a lightweight extendible web and social mash-up engine to host, serve and aggregate gadgets [49]. Rave is designed using the Spring framework [50,51]. The Spring Framework itself provides a configurable environment to customize components based on project needs which are utilized by Rave. Customization of security management to provide single sign-on to multiple clients is a key feature for this project [52]. This project utilizes Apache Rave's use of Maven overlays (discussed in detail for science gateways [52]). We found some design flaws while extending the user model object in Rave and contributed the solution back to the Rave community.

Apache FTP: For facilitating user uploads in US-SOMO, Apache FTP software is used which provides a portable FTP server engine solution based on open protocols. Apache FTP provides a plug-in for the Spring framework, which was utilized to share the user model with Apache Rave. The integration was assisted by the fact that all database queries for the user model are in configuration files.

Airavata GFAC: The Airavata GFAC [53] toolkit exposes network accessible services for wrapped command line applications. These resulting services provide programmable APIs to facilitate integration with portal or desktop applications. The GFAC software is used for job management in the UltraScan gateway. The US-SOMO project builds on this foundation and extends it providing authentication tokens for validation and authorization.

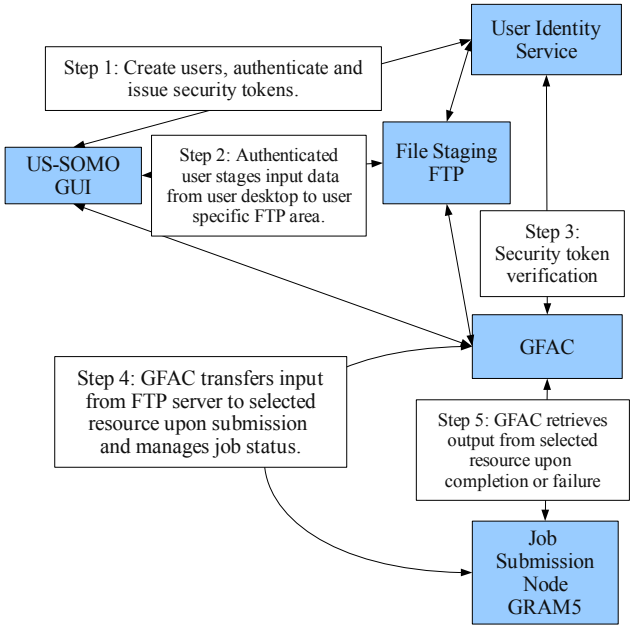

Figure 4: High level gateway architecture of US-SOMO. The interactions of Steps 1 through 4 are described in Section 3.3.2

3.3.2 Gateway Architecture:

As illustrated in the Figure 4 the US-SOMO gateway is implemented by integrating the US-SOMO GUI and OGCE software. Its multi-step process is defined in US-SOMO's GUI allowing the user to run jobs on compute resources without knowledge of these details. Each step is discussed below to provide details of each service.

1. **User Identity:** The gateway identity software provides the following functionality: 1) Creation of a new user without requirement for any authentication. 2) Administrator is notified of the user request through email or other messaging protocols. 3) Administrators approve the user account before any user activity. 4) User accounts are unique. 5) Users are able to provide profile information including email for system notifications. 6) Methods to update and delete user profiles. To meet the above requirements, we extended Rave's user model and user services objects to add project specific data attributes such as FTP home directory, allowed connection, etc. User "creation", "get", "update", "delete" REST services were developed and an HTTP/s client with basic authentication support has been implemented for these services. Services have support for XML and JSON input/output formats. The user creation request is configured in Spring security to bypass authentication.
2. **File transfer:** The gateway needs features for users to upload data from user's workstation to compute resources and download the results without installing any custom software. We explored options like Globus Online [54] and commercial services like Dropbox [55]. To keep the usage barrier low, a user friendly option was developed. In the future, more community software will be explored. As the data size can grow to 2GB or more, an FTP server was chosen for file staging. We use the Apache FTP server's Spring framework plugin utilizing the same data model developed by Apache Rave. This provides uniformity in user credentials. The user home folder and access control is defined as part of the user management APIs. The administration interface can be used to enable/disable users and optimize FTP connection parameters.
3. **Token Validation:** The token service facilitates the gateway requirements by: 1) Securing APIs for file transfer and job management. 2) Providing proper access control. 3) Enabling the application to present identity created using user management APIs for authorization. 4) Issueing short-lived user token to handle the user session. 5) Securing user input/output data from other users. The US-SOMO GUI uses the same credentials for file transfer and job submissions. To submit a new job submission, the application must first make a call to the "authenticate" service to get a security token. The new token is valid for 30 minutes by default, but is configurable as a request property. The application is able to use the same token for multiple service requests based on validity. An existing token is invalidated on issuance of a new token or on expiration. The administrator is also allowed to invalidate the tokens. The job submission service validates the token with the authentication validation service and responds accordingly.
4. **Job Management:** The core gateway job requirements include: 1) User input data location to be passed in the job submit request along with other job parameters such as compute location, processes count, maximum wall time, etc. 2) Monitor job status of submitted jobs. 3) Cancel running jobs. 4) Resubmit failed jobs. 5) User credentials to be shared between multiple user requests. We have developed a secure service to "submit", "status", "cancel" and "resubmit" the jobs. The US-SOMO application sends a security token as part of the request header, and services validate the token using request filters. The application creates a unique experiment id per request. This experiment id is used as a key for other operations. Service operations are described in detail on the UltraScan wiki [48].

4. FUTURE

Development efforts are ongoing. The file transfer mechanism may benefit from other community developments. During job submission, the user is required to select a compute resource. This could be simplified by an automatic selection based upon system availability and other factors such as expected queuing time, potentially fed from an online service provided by GFAC. Certain types and sizes of jobs may be automatically directed to specific resources. For example, we currently recommend targeting long running DMD jobs to our local cluster which has no run time limitations. Timeouts and other job failures are, as of this writing, handled somewhat ungracefully. Development efforts are underway to flush out intermediate results to the output package on availability and provide detailed status for incomplete and failed sub-jobs. We will pass the job time limit to the job and utilize Qt's timer mechanism to flush out available results before being terminated. We envision a post-processing segment where failed sub-jobs are repackaged for submission, which could be done either at the GUI level, allowing the user some discretion, or automated at the GFAC level. Along with coarse grain job monitoring, application level monitoring will be added to the gateway messaging framework. UDP status update messages sent from the application's compute resource job will be consolidated through Airavata middleware. This information will provide users with real-time monitoring and application steering capabilities. US-SOMO's batch and cluster module provides a framework for additional methods and techniques. We are currently testing a new MPI based analysis method for shape reconstruction from SAS curves already integrated into the cluster facilities of our development tree along with support for multiple other shape reconstruction methodologies.

5. CONCLUSIONS

US-SOMO is a comprehensive package for the calculation of hydrodynamic parameters and small angle scattering curves for biological macromolecules. Initial structures can be expanded into a space of candidate structures using discrete molecular dynamics. Candidate structures can be processed and screened against experimental data for multiple experimental methods. The cluster module provide the user with the ability to process large numbers of structures on remote compute resources. The work funded through the Open Gateway Computing Environment Collaboration directly integrates these capabilities into the US-SOMO GUI application. Researchers will benefit from the advanced capabilities present in US-SOMO.

6. ACKNOWLEDGMENTS

This work was supported by NIH grant K25GM090154 to EB, NSF grant OCI-1032742 to MP, NSF grant TG-MCB070040N to BD, and NIH grant RR-022200 to BD. We thank Shahani M Weerawarana, visiting research scientist at Indiana University, for editorial assistance.

7. REFERENCES

[1] Miller, B.A. 2009. Imatinib and its successors: how modern chemistry has changed drug development. *Curr Pharm Des* 15:120-133.

[2] Goldman, J.M. 2010. Chronic myeloid leukemia: an historical perspective *Semin Hematol*. 47:302-311.

[3] Demeler, B. 2005. UltraScan: a comprehensive data analysis software package for analytical ultracentrifugation experiments. *Modern AUC: Techniques and Methods*. Scott, D.J. et al., Eds. Royal Society of Chemistry 210-9

[4] UltraScan. http:/www.ultrascan.uthscsa.edu

[5] Qt. http://qt.nokia.com

[6] Brookes, E., Boppana, R.V., and Demeler, B. 2006. Computing large sparse multivariate optimization problems with an application in biophysics. *Proceed. SC2006*. ACM.

[7] Brookes, E., Cao, W., Demeler, B. 2009. A two-dimensional spectrum analysis for sedimentation velocity experiments of mixtures with heterogeneity in molecular weight and shape. *Eur Biophys J*

[8] Brookes, E. and Demeler B. 2010. Performance optimization of large non-negatively constrained least squares problems with an application in biophysics. *ACM TG10.* N.Y.

[9] Brookes, E. and Demeler, B. 2006. Genetic algorithm optimization for obtaining accurate molecular weight distribution from sedimentation velocity experiments. *AUC VIII, Progr Colloid Polym Sci* 131:78-82. Springer.

[10] Brookes, E. and Demeler, B. 2007. Parsimonious regularization using genetic algorithms applied to the analysis of analytical ultracentrifugation experiments. *Proceedings GECCO 07.* ACM.

[11] Message passing interface standard. http://www.mcs.anl.gov/research/projects/mpi/

[12] High perf. comp. across texas. http://www.hipcat.net

[13] Texas advanced comp. center. http://www.tacc.utexas.edu

[14] Globus ws-gram 4. http://www.globus.org/toolkit/docs/4.0/execution/wsgram/

[15] The generic service toolkit. http://www.extreme.indiana.edu/gfac/

[16] Pierce, M., S. Marru, et al. 2010. Open grid computing environments: advanced gateway support activities. *Proceedings of the TG10 Conference,* ACM: 16:11—16:19.

[17] Rai, N, et al., M. SOMO (SOlution MOdeler): Difference between x-ray and nmr-derived bead models suggest a role for side chain flexibility in protein hydrodynamics. *Structure* 13, 723-734, 2005

[18] Brookes, E., Demeler., B, and Rocco, M. 2010. The implementation of somo in the ultrascan analytical data analysis suite: enhanced capabilities allow the reliable hydrodynamic modeling of virtually any kind of biomacromolecule. *Eur Biophys J*

[19] Brookes, E., Demeler, B., Rosano, C., and Rocco, M. 2010. Developments in the us-somo bead modeling software: new features in the direct residue-to-bead method, improved grid routines, and influence of accessible surface area screening, *Macromol Biosci* 10:746-753

[20] Brookes, E. US-SOMO. http://somo.uthscsa.edu

[21] Glatter, O. Kratky, O. 1982. *Small angle x-ray scattering*. 1982. Academic Press., London, ISBN-0-12-286280-5

[22] Roe, R.J., 2000. *Methods of x-ray and neutron scattering in science.* Oxford University Press, New York.

[23] Dokholyan, N.V.,Buldyrev, S.V., Stanley, H.E., and Shaknovich, E.I. 1998. Discrete molecular dynamic studies of the folding of a protein-like model. *Folding & Design* 3:577-587

[24] Ding F, Dokholyan NV. 2006. Emergence of protein fold families through rational design. *Public Library of Science Comput Biol* 2(7):e85

[25] Mansfield et al., 2001. Intrinsic viscosity and the electric polarizability of arbitrarily shaped objects, *Phys Rev E* 64:61401-16

[26] Jmol. http://www.jmol.org/

[27] Qt port of webkit. http://trac.webkit.org/wiki/QtWebKit

[28] Qt5. http://qt-project.org/wiki/Qt_5.0

[29] The protein data bank. http://www.rcsb.org

[30] van Holde, K. E. 1985 *Phys. Biochem.*, 2nd Ed. Prentice Hall.

[31] Svergun, D.I. 1999. Restoring low resolution structure of biological macromolecules from solution scattering using simulated annealing. *Biophys J* 2879-86.

[32] Svergun, D.I., Petoukhov, M.V., and Koch, M.H.J. 2001. Determination of domain structure of proteins from X-ray solution scattering. *Biophys J*, 80, 2946-2953

[33] Garcia de la Torre, J., Bloomfield, V.A. 1981. Hydrodynamic properties of complex, rigid, biological macromolecules: theory and application. *Q Rev Biophys* 14:81-139.

[34] Garcia de la Torre, J, Bloomfield, V.A. 1977. Hydrodynamic properties of macromolecular complexes. *Biopol* 16:1765-78

[35] Ortega, A., Amoros, D., Garcia de la Torre, J. 2011. Prediction of hydrodynamic and other solution properties of rigid proteins from atomic- and residue-level models. *Biophys J* 101, 892-898

[36] Byron, O. 1997. Construction of hydrodynamic bead models from high-resolution X-ray crystallographic or nuclear magnetic resonance data. *Biophys J* 72, 408-415.

[37] Garcia de la Torre, J. et al. 2009. Simuflex: algorithms and tools for simulation of the conformation and dynamics of flexible molecules and nanoparticles in dilute solution. *J Chem Theor Comput* 5, 2606-2618.

[38] Moeller, A, et. al 2012. Nucleotide-dependent conformational changes in the n-ethylmaleimide sensitive factor (nsf) and their potential role in snare complex disassembly. *J Struct Bio* 177:335-43

[39] Nishio, M., et al. 2010. Structural basis for the cooperative interplay between the two causative gene products of combined factor v and factor viii deficiency. *Proc Natl Acad Sci* USA 107 (9) 4034-4039

[40] Rosano, C, and Rocco, M. 2010. Solution properties of full-length integrin αIIbβ3 refined models suggest environment-dependent induction of alternative bent/extended resting states. *FEBS J* 277:3190-3202

[41] Douglas et al. 1994. Hydrodynamic friction and the capacitance of arb. shaped objects. *Phys. Rev. E* 49:5319-31

[42] Zeno. http://www.stevens.edu/zeno

[43] Schneidman-Duhovny, D., Hammel, M., and Sali, A. 2010. Foxs: a web server for rapid comp. and fitting of saxs profiles. *Nucleic Acids Res* 38 Suppl:W540-4.

[44] FoXS webserver. http://modbase.compbio.ucsf.edu/foxs/about.html

[45] Svergun, D.I., Barberato, C. and Koch, M.H.J. 1995. Crysol - a program to evaluate x-ray solution scattering of biological macromolecules from atomic coordinates *J Appl Cryst* 28, 768-73.

[46] Jowitt, Tom, Scott, David. *Separate pers. comm*.

[47] Pierce, M. et al. 2009. Open grid computing environments.

[48] US-SOMO OGCE Bridge Clients Information. http://wiki.bcf.uthscsa.edu/ultrascan/wiki/OGCEIntegration

[49] Rave Identity service. https://ogce.svn.sourceforge.net/svnroot/ogce/rave-extensions/rave-id-extension.

[50] Spring Security. http://static.springsource.org/spring-security/site/index.html

[51] Spring Framework. http://www.springsource.org/

[52] Pierce, M. E., Singh, R., et al. 2011. Open community development for science gateways with apache rave. *Proceedings of the 2011 ACM workshop on Gateway computing environments.* 29-36.

[53] Marru, S., Gunathilake, L., et al. 2011. Apache airavata: a framework for distributed applications and computational workflows. *Proceedings of the 2011 ACM workshop on Gateway computing environments.* 21-28.

[54] Globus Online. http://www.globusonline.org

[55] Dropbox. http://www.dropbox.com