Journal of Clinical Monitoring and Computing (2005) 19: 307–317
DOI: 10.1007/s10877-005-0676-z

TOWARDS DATA WAREHOUSING AND MINING OF PROTEIN UNFOLDING SIMULATION DATA*

Daniel Berrar,[1] *Frederic Stahl,*[2] *Candida Silva,*[3] *J. Rui Rodrigues,*[3] *Rui M.M. Brito,*[3] *and Werner Dubitzky*[4]

Berrar D, Stahl F, Silva C, Rodrigues JR, Brito RMM, Dubitzky W. Towards data warehousing and mining of protein unfolding simulation data.

J Clin Monit Comput 2005; 19: 307–317

ABSTRACT. Objectives. The prediction of protein structure and the precise understanding of protein folding and unfolding processes remains one of the greatest challenges in structural biology and bioinformatics. Computer simulations based on molecular dynamics (MD) are at the forefront of the effort to gain a deeper understanding of these complex processes. Currently, these MD simulations are usually on the order of tens of nanoseconds, generate a large amount of conformational data and are computationally expensive. More and more groups run such simulations and generate a myriad of data, which raises new challenges in managing and analyzing these data. Because the vast range of proteins researchers want to study and simulate, the computational effort needed to generate data, the large data volumes involved, and the different types of analyses scientists need to perform, it is desirable to provide a public repository allowing researchers to pool and share protein unfolding data. **Methods.** To adequately organize, manage, and analyze the data generated by unfolding simulation studies, we designed a data warehouse system that is embedded in a grid environment to facilitate the seamless sharing of available computer resources and thus enable many groups to share complex molecular dynamics simulations on a more regular basis. **Results.** To gain insight into the conformational fluctuations and stability of the monomeric forms of the amyloidogenic protein transthyretin (TTR), molecular dynamics unfolding simulations of the monomer of human TTR have been conducted. Trajectory data and meta-data of the wild-type (WT) protein and the highly amyloidogenic variant L55P-TTR represent the test case for the data warehouse. **Conclusions.** Web and grid services, especially pre-defined data mining services that can run on or 'near' the data repository of the data warehouse, are likely to play a pivotal role in the analysis of molecular dynamics unfolding data.

KEY WORDS. data warehousing, grid, molecular dynamics simulation, data mining, protein unfolding, transthyretin.

1. INTRODUCTION

Probing ever deeper into living nature and devising increasingly intricate artifacts, most endeavors in modern life science and biotechnology are driven by the generation and analysis of large and complex-structured data sets. Besides the ever-growing mountain of information, more and more life science investigations focus on holistic modeling and understanding of entire life *systems*. Systems life science or systems biology distinguishes itself from a more traditional approach to biological research by an attempt to view collections of interacting entities as a whole that is as a system, rather than on studying individual interactions in isolation. What exactly constitutes a system (collection

From the [1]School of Biomedical Sciences, University of Ulster, Coleraine, Northern Ireland Cromore Road, BT52 1SA, Coleraine, Northern Ireland, [2]School of Biomedical Sciences, University of Ulster, Coleraine, Northern Ireland, and Weihenstephan University of Applied Sciences, Freising, Germany, [3]Departamento de Química, Faculdade de Ciências e Tecnologia, and Centro de Neurociências de Coimbra, Universidade de Coimbra, Coimbra, Portugal, [4]School of Biomedical Sciences, University of Ulster, Coleraine, Northern Ireland

Received and accepted for publication June 30, 2005.

Address correspondence to D. Berrar, School of Biomedical Sciences, University of Ulster, Coleraine, Northern Ireland Cromore Road, BT52 1SA, Coleraine, Northern Ireland.
Email: dp.berrar@ulster.ac.uk

*Based on "Grid Warehousing of Molecular Dynamics Protein Unfolding Data", by Frederic Stahl, Daniel Berrar, Candida Silva, J. Rui Rodrigues, Rui M.M. Brito, and Werner Dubitzky, which appeared in Proceedings of the IEEE/ACM International Symposium on Cluster Computing and the Grid, Cardiff, UK, May 9–12, 2005.

of interacting entities) needs to be defined on a case-by-case basis. Commonly, a system refers to a complex multi-cellular ensemble, an intra-cellular network of chemical reactions, collections of interacting proteins, networks of mutually regulating genes, or control and signaling pathway networks. In this paper we deviate somewhat from the conventional notion of systems biology and chose to view individual proteins as systems, with their constituent amino acid residues as system components. In this view the collective and dynamic interactions of the protein's amino acid residues lead to the formation of the native structure of the protein. This leads to the well-known *folding problem*. Protein folding and unfolding has been at the center stage in molecular biophysics in recent years due to their involvement in issues so diverse as amyloid diseases and protein structure prediction.

As a result of genomic projects already concluded or under way worldwide, the amount of genome sequence data has recently been growing exponentially. The inability of experimental high-resolution techniques such as *X-ray crystallography* or *nuclear magnetic resonance* (NMR) to keep up with this explosion of new sequences and required resolution of corresponding three-dimensional protein structures stresses the need of reliable modeling methods for protein structure prediction. Although significant progress has been made in predicting protein structures among members of the same structural family, much remains to be done in structure prediction for proteins sharing low homology with known structures, or for new protein topologies. Some recent efforts using simplified *lattice models* have been successful in predicting kinetic aspects of the protein folding process, but require the prior knowledge of the native protein structure [1]. Another approach attempts to predict protein structure from basic physical-chemical interactions by using *molecular dynamics* (MD) simulations. In MD simulation studies of proteins, the time-dependent behavior of the molecular system is obtained by integrating Newton's equations of motion (classical mechanics) and the potential energy function (a.k.a. force fields). The result of the simulation is a time series of conformations; this is called a trajectory or the path followed by each atom in accordance with Newton's laws of motion. MD simulations of protein folding have been successfully performed for small proteins (less than 50 amino acid residues) and require considerable computer resources [2, 3]. Thus, approaches that somehow reduce the computational burden but are still based on basic physical-chemical interactions are particularly interesting to explore in the area of protein structure prediction. In addition, grid computing can play its role by facilitating the seamless sharing of available computer resources and thus enable many groups to perform complex MD simulations on a more regular basis. Several groups are already exploring this technology in this context [4–6].

The data generated in MD simulations is massive. For example, to record the coordinates of the Cα atoms of a protein with 127 residues over 8,000 time frames (representing 8 ns) for a single simulation run is 32 MB (storage in a flat ASCII text file). Given 5-15 heavy atoms per residue, the data volume capturing the coordinates of all involved atoms is in the order of 300 MB. This data volume requires adequate management by means of database technology. Sharing such data and facilitating their analysis among globally dispersed research groups is a considerable challenge. This work is part of a more comprehensive effort to create a repository for MD unfolding simulation data of biological macromolecules. Eventually, this repository should serve interested research groups worldwide by facilitating the pooling and analysis of related MD unfolding data. Another aspect of the overall project is to develop suitable data mining techniques to analyze large sets of unfolding data. Because different types of data mining require the data to be formatted or pre-processed in different ways, data warehousing is adopted as a methodology to implement the shared repository. Key for the data warehouse and the data mining is to implement some of the most common analysis methods as part of the repositories. This is needed because the expected data volumes will be too high to transfer large subsets of the full trajectory data to remote sites. This project explores grid as a technology to meet the requirements arising from the MD unfolding simulation data management, warehousing, and mining.

2. METHODS AND RESULTS

The goal of organizing and managing data on a shared, grid-enabled data warehouse is to gain insight into the structure, stability, and other properties of the underlying proteins. To achieve this goal the data will need to be analyzed by different data mining techniques. In the following subsections we describe the underlying experimental setup and the data mining objectives.

2.1. Experimental setup

In order to gain insight into the conformational fluctuations and stability of the monomeric forms of the amyloidogenic protein transthyretin (TTR), Rodrigues and Brito [7, 19] have conducted molecular dynamics (MD) unfolding simulations of the monomer of human TTR. The starting point for the simulations were the X-ray coordinates of the wild-type (WT) protein and the highly amyloidogenic variant L55P-TTR (with a Proline replacing a Leucine in position 55). Among the numerous pathogenic variants, L55P-TTR is the most amyloidogenic, and V30M-TTR is one of the most prevalent [8].

Water molecules and NaCl ions were added to the protein to produce a system of more than 45,000 atoms. All MD procedures were performed with the program NAMD [9]. Several 8 to 10 ns MD trajectories were run for each one of the WT- and L55P-TTR monomers. Protein unfolding was induced by high temperature, running the simulations at 500 K [3, 7, 19]. The simulations were performed using Charm27 as the force field with all atoms explicitly represented. The molecular systems were first equilibrated at 500 K with 50 ps of Langevin dynamics at constant pressure and with protein movements restrained, followed by 20 ps of free classical MD at constant volume. The production runs were started at this point and coordinates for the whole system were saved every 1 ps. The integration time step was 2 fs, which means that every 500 steps (2 fs × 500 = 1,000 fs = 1 ps) the data were recorded.

Hydrogen-heavy atom bond lengths were constrained with the SHAKE algorithm. The simulations were carried out using periodic boundary conditions. Long-range electrostatic interactions were computed at every step using the Particle Mesh Ewald method. The only difference in the setup of each run resides in the assignment of initial atomic velocities. Figure 1A shows a schematic view of the WT-TTR subunit; Figure 1B depicts the all-atom representation of the molecule in the simulation system.

2.2. Experimental results

From the data generated by the simulation process, two types of protein properties can be derived:

- *Local molecular properties*, referring to the 127 individual amino acids (residues), and
- *Global molecular properties*, referring to the conformational state of the protein.

2.2.1. Local molecular properties

Both mutant and WT proteins are comprised by 127 residues, consisting of 5 to 15 heavy atoms each. For each atom, the xyz-coordinates are recorded for tens of thousands of time frames representing tens of nanoseconds. To address some questions such as the general fold of the protein backbone, the coordinates of the Cα atoms may be sufficient. By comparing for example the trajectories of the Cα atoms of some residues in the WT and mutant proteins, we observe different behaviors. But local properties may also be studied using full atom representations. For example, analysis of the relative solvent accessible surface area (SASA) of individual residues reveals patterns of similar behavior among some of these residues, allowing the definition of classes or groups of residues [3]. Using data mining techniques, the objective is to discover more complex patterns of local properties across the simulation time frame.

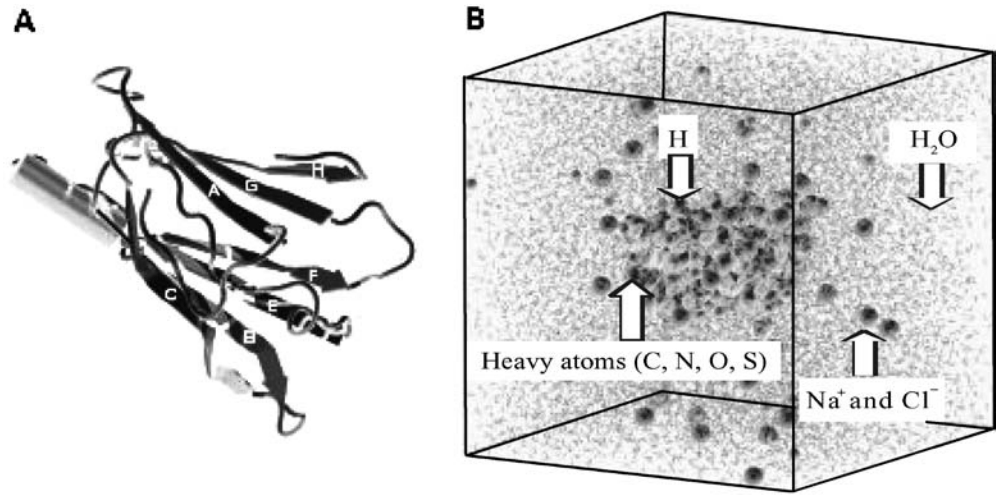


Fig. 1. (A) Schematic representation of the WT-TTR subunit showing the β-sandwich topology and the identification of the β-strands. (B) All-atom representation of the solvated L55P-TTR system used in the MD simulations, comprising a total of 44,394 atoms. Water molecules are displayed as grey V shapes, Na$^+$ and Cl$^-$ ions as grey spheres, protein heavy atoms (C, N, O, S) as black spheres and protein H atoms as white spheres.

2.2.2. Global molecular properties

A very important global property of the protein in the unfolding process is the *loss of native contacts* as a function of time. A native contact may be defined as an inter-atomic distance smaller than a threshold, for example 4.2 Å, between two non-adjacent amino acid residues observed in the protein's native state. The absolute number of native contacts is expected to decrease over time. By plotting the loss of native contacts over time we can observe different patterns between the WT and mutant proteins. Figure 2 shows the number of native contacts in (A) the WT and (B) the mutant as a function of time for two simulation runs (both at 500 K).

The root mean square deviation (RMSD) is the most commonly used similarity measure for protein structures and may be used as a measure for the progression of the unfolding process. Let $\mathbf{r}_{ai}$ and $\mathbf{r}_{bi}$ be the coordinate vector of atom i of structure a and structure b, and let n be the total number of atoms per structure. The root mean square deviation between the two structures is then defined as follows.

$$\mathrm{RMSD} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\mathbf{r}_{ai} - \mathbf{r}_{bi})^2} \tag{1}$$

Loosely speaking, the larger the RMSD, the more the protein structure deviates from its initial stage. By plotting various RMSD profiles, we observe, for example, that L55P-TTR shows slightly larger RMSD values than WT-TTR and, in general, deviates from the crystal structure earlier in the simulation. While the final RMSD value for several L55P-TTR runs are of comparable magnitude, the final RMSD values for WT-TTR show a greater diversity (see Figure 3A and B).

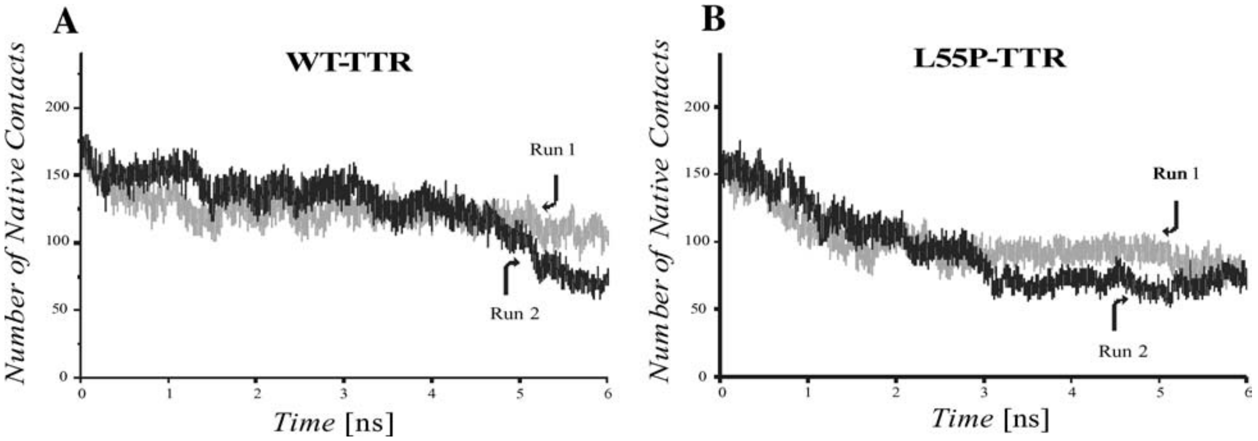


Fig. 2. Loss of native contacts as a function of time in (A) the WT and (B) the mutant (for two simulation runs each, at 500 K).

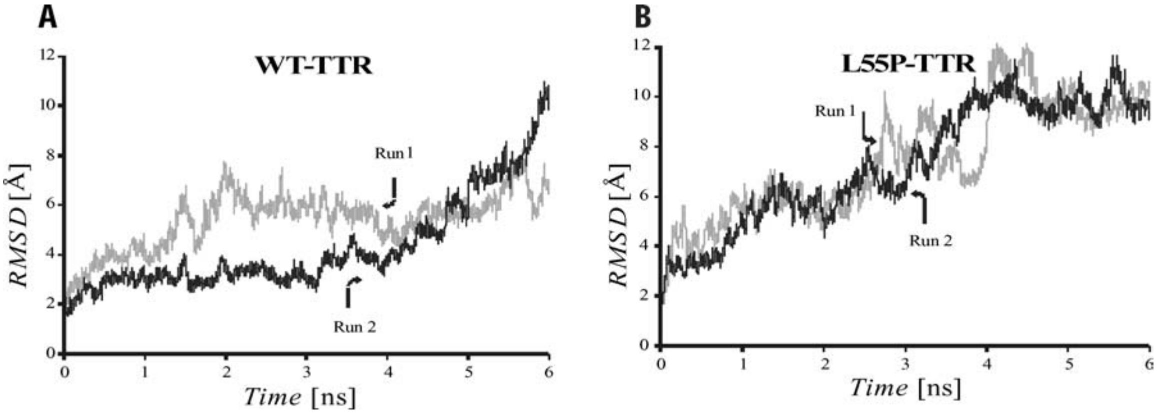


Fig. 3. Root mean square deviation from the initial structure as a function of time in (A) the WT and (B) the mutant (for two simulation runs each, at 500 K).

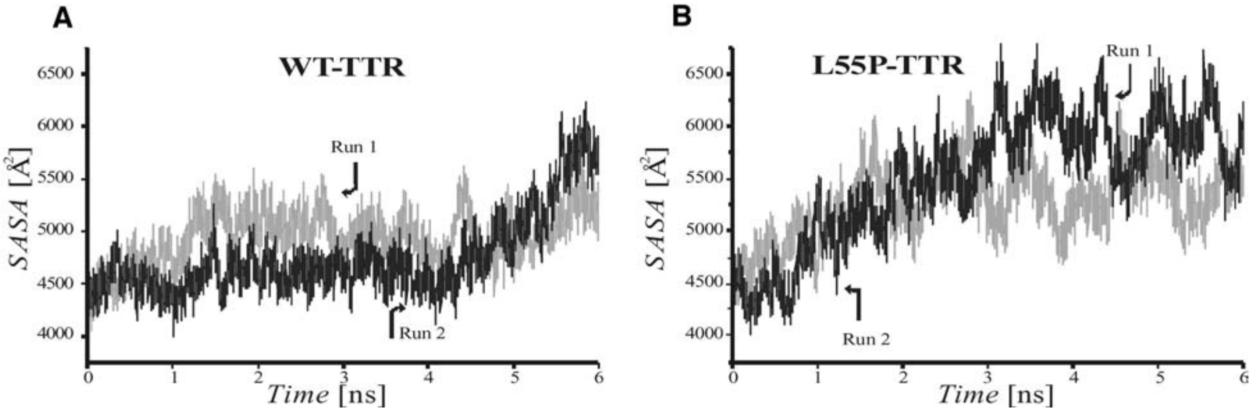


Fig. 4. Solvent accessible surface area of the non-polar residues as a function of time in (A) the WT and (B) the mutant (for two simulation runs each, at 500 K).

The solvent accessible surface area (SASA) is calculated as the protein surface accessible to a spherical probe of 1.4 Å in radius. The global SASA of the protein is expected to increase as protein unfolding progresses and less compact conformational states are reached. Additionally, local SASA (the relative solvent accessible surface area for each amino acid residue) is also a very interesting property to explore because it may reveal patterns of coordinated movements of different classes of amino acids). The global SASA may even be decomposed into the SASA of polar and non-polar residues. The following figure depicts the SASA of the non-polar residues as a function of time for the WT and the mutant protein for two simulation runs.

In the mutant, the SASA of the non-polar residues increases faster than the corresponding SASA in the WT. Loosely speaking, this implies that in the mutant, the unfolding process exposes more non-polar residues more quickly to the solvent than in the WT. In contrast, the SASA of the polar residues seems to be comparable in both the WT and the mutant.

2.3. Scientific questions and data mining tasks

As illustrated above and in more detail by Brito et al. [3], some of the data generated by MD unfolding simulations may be interpreted using simple visualization techniques. However, this method has severe limitations (a) once large data volumes need to be analyzed, and (b) in terms of producing quantitative results. Therefore, the aim is to develop data mining methods that are geared towards the requirements arising from MD unfolding simulations. Essentially, the analytical tasks that need to be tackled are (1) *pattern detection*, (2) *clustering*, and (3) *classification*. These are illustrated below on the basis of the thermally-induced protein unfolding simulations of transthyretin.

2.3.1. Pattern detection

A *local pattern* is an unexpected event in a single time series (e.g., the trajectory of a single residue). It may be defined as an event with a statistically significant 'surprise', expressed in terms of the reciprocal of the likelihood, for instance. For example, a sharp peak in a time series might be a local pattern.

A *global pattern* is a property that is characteristic for the time series of a number of simulation runs of the same protein. In the unfolding simulations mentioned above, several runs were carried out for the wild-type (WT) and mutant transthyretin. For example, the time series of the global SASA of the WT might exhibit a specific characteristic/property (e.g., upwards trend), which can be considered a global pattern of the time series of the WT. In contrast to the local pattern, the global pattern might not necessarily need to be statistically significant. The global pattern refers to a property that the time series of a protein share. For example, the relative slow upwards trend of the global SASA of the WT is observed in all time series of the WT, which hence can be considered a global pattern.

2.3.2. Clustering

Generally, 'unsupervised' grouping or *clustering* is a form of learning from observations that have not been pre-classified (by a supervisor). The goal is to determine *similarities* or associations among the set of observed entities, E, and *group* them into n groups or clusters, $C \subset E$, such that the entities in one group are similar to one another and dissimilar from entities in any other group. In contrast to classification (see below), clustering seeks a convenient and valid organization of data and not to establish rules for separating future data into categories [10].

Here it is interesting to investigate groups of local and global properties derived from the unfolding trajectories of a set of MD simulations. For example, it is possible to identify collections of residues 'moving' in an orchestrated way within a single simulation run. These may be residues that are close or far apart in the sequential structure of the protein. It is also of potential interest to detect outliers, i.e. residue trajectories within a single run that are completely different to those of most other trajectories. Similarly, groupings of trajectories of global properties across different experimental conditions (e.g., wild-type versus mutant, normal versus disease) may be of interest. Again, given large amounts of simulations, properties and conditions investigated, and the temporal nature of the data involved will make it necessary to employ automated methods for discovery of such groupings.

2.3.3. Classification

Classification is a form of learning from *pre-classified* examples or entities. Generally, given a collection of entities, E, and a set of pre-defined classes, C, *classification* refers to the process of approximating an unknown target function, ϕ: $E \times C \rightarrow \{true, false\}$ (describing how the entities ought to be classified) by means of an estimated function, f: $E \times C \rightarrow \{true, false\}$, called the *classifier*. If, for $x = (e, c) \in E \times C$, the function $\phi(x) = true$ then e is called a *positive example* of c, and if $\phi\ (x) = false$ then e is a negative example of c.

In this application, classification is useful, for example, to relate local and global protein unfolding trajectory features and characteristics among different mutant proteins and disease states (e.g. disease, no disease). Exploring the data in this way will eventually lead to insights into the critical features of the unfolding process and thus provide hints on the underlying biochemical mechanisms.

2.4. Data warehousing of molecular dynamics unfolding simulation data

Running protein unfolding simulations is computationally expensive and finding ways to enhance performance is a grid issue on its own [11]. It is expected that more and more groups will run such simulations and generate a huge amount of data, which raises new challenges in managing and analyzing these data [3]. Because the vast range of proteins researchers wish to simulate, the computational effort needed to generate data, the large data volumes and the different types of analyses scientists need to perform, it is desirable to provide (a) a public repository allowing researchers to pool and share protein unfolding data, (b) cached computations, in particular typical pre-processing of the data, (c) commonly used analyses as web or grid service. This leads to the notion of a *grid-enabled data warehouse for molecular dynamics protein unfolding simulation data*. Below we discuss the details of this approach.

2.4.1. Systems requirements

To address the information needs of the molecular dynamics community, the overall aim of this project is to create a public repository that will enable researchers around the globe to *share* and *analyze* data from protein unfolding simulation studies. This would allow the community to mutually create unfolding trajectories of a wide variety of proteins under a large range of experimental conditions (mutations, diseases, tissues, species, etc.) and compare and contrast these data. The two main functional requirements have two key implications.

- Sharing of the actual unfolding data means that users around the world can contribute (i.e. upload and store) unfolding data from their MD simulations to the shared repository or database and they can access, select and retrieve unfolding data from the repository. Such a shared repository would have immense advantages for the community.
- Analyzing protein unfolding data stored in the shared repository requires that users can apply a large set of different pre-processing and data analysis tasks (classification, clustering, time-series modeling, etc.) to the data needed to answer their scientific questions.

These two main requirements would be relatively easy to fulfill if the data volumes were small or moderate. In such a case, the system architecture would have (a) a central database as physical repository running on some publicly accessible server. Based on (b) a common data exchange format (e.g., an XML realization for the protein unfolding data) users would (c) upload their unfolding experiment data to the server, (d) select and download onto some local machine (or a dedicated analysis server, local or remote) subsets of the data for analysis, (e) pre-process the data on this machine, and (f) then run their favorite analytical method or algorithm on the prepared data.

However, because of the sheer size of the data involved this simple scheme is not practical as transfer times for retrieval of subsets of the repository would be too long for most practical investigations. Therefore, the systems architecture we propose is based on the following considerations (see Figure 5).

A *data warehouse* [12] is used to provide (a) the storage, access, and retrieval functionality for the 'raw' protein

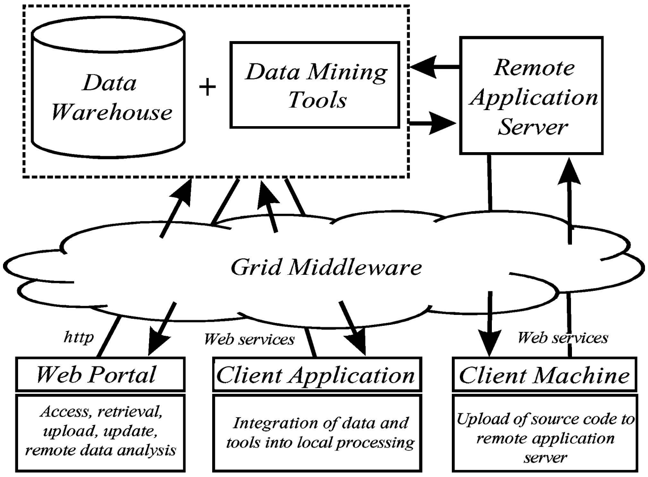

Data Warehouse
+
Data Mining Tools
Remote Application Server
Grid Middleware
http
Web services
Web services
Web Portal
Access, retrieval, upload, update, remote data analysis
Client Application
Integration of data and tools into local processing
Client Machine
Upload of source code to remote application server

Fig. 5. Grid-enabled MD simulation data warehouse.

unfolding simulations data, (b) many different 'views' of the same 'raw' data to facilitate fast access of this data, (c) many different pre-processed versions of the 'raw' data; these versions of the data would provide cleansed, normalized, standardized, summarized, and enriched 'views' to facilitate subsequent analyses, and (d) a meta-data facility allowing users to navigate and understand the content and structure of the data warehouse and its different 'views'. While being extremely redundant in terms of the data storage usage, the users will benefit enormously by being able to access pre-processed data that is much closer to their analysis needs and likely to be much smaller in volume. The advantage of locating typical analysis methods close to the data source is that many users will not need to download data to their local environments, but run those methods against the subsets of data they have selected. In the long run it is still likely that a large number of users will want to access and transfer considerable quantities of data to their local environments relatively frequently. This gives rise to an approach that provides the data part as *grid data services* and uses grid technology to address access latency and bandwidth consumption issues. Relevant grid technologies include *data replication services* [13], *caching* and *mapping* [14]. Currently, we do not envisage replication implementations but consider Open Grid Services Architecture – Data Access and Integration (OGSA-DAI) [15] tools as a middleware to assist with access and integration of the data warehouse via the grid. OGSA-DAI provides efficient grid-enabled middleware implementation of interfaces and services (also known as portTypes) to expose, access, and control physical data sources and sinks (currently relational and XML database management systems) and to represent client access points for these resources. OGSA-DAI's basic primitives, interfaces, and services may be used by higher-level services to provide greater transparency and facilitate easy integration within OGSA-compliant systems. Thus, it is possible to construct sophisticated higher-level services that allow data federation and distributed queries to take place within *virtual organizations*. Key motivations for the OGSA-DAI project include:

- *Controlled* exposure of physical data source to the grid;
- Support for accessing heterogeneous physical data resources through a common interface;
- Base services that allow higher-level data integration services to be constructed, for example, distributed query processing and data federation services;
- Leverage of emerging grid infrastructure for security, management, accounting, and so on;
- Standardization of data access interfaces as pursued by the Global Grid Forum's Database Access and Integration Services (DAIS) Working Group [16] (data description, access, factory, and management);
- Provision of a reference implementation of the DAIS specification.

A set of *data mining algorithms* is implemented on or 'near' the data warehouse server to facilitate commonly needed data analysis tasks such as classification, clustering, association and correlation tasks. Such a setup would minimize costly data transfers. This approach requires mechanisms that allow users to select subsets of the data in the data warehouse and associate them with the analysis algorithms and then to execute the algorithms. By providing grid or web interfaces for the data mining algorithms we will be able to provide these algorithms as grid or web services and thus be able to easily integrate them with the data grid/web services of the data warehouse and indeed with other grid/web services.

In a data-intensive application like this it would be highly economic to *ship computation to data* in the warehouse (i.e., implemented data analysis algorithms in the form of executable programs), run the analysis where the data resides (or at least close to the data on a dedicated compute server) to avoid transfer of large data volumes, and return the results to the user. However, while the grid provides mechanisms to (install and) execute programs on remote processors, many issues arise if the program to be executed interacts with data and database management systems at the remote location. In particular in science, the computations to be executed correspond to scientific models whose requirements for selecting, linking and processing data go far beyond what is currently permitted by database management systems. For example, such programs often need highly repetitive, iterative access to the same data, create large intermediate results, place high computational loads on processors, and remote executions typically need to be observed, controlled, monitored, and managed (sometimes highly interactively) by the user. These complex requirements are currently difficult to handle in a general way and are subject of ongoing grid research. Based on the data and analysis service, we will provide a *web portal* and *web client access* to the data warehouse and its analysis components.

With the above main requirements and considerations in mind, we can now state the requirement for this application in a more detailed fashion.

(i) Provide a public, globally accessible, shared repository for efficient storage and retrieval of MD unfolding simulation data and meta-data. Implement web/grid services facilitating above access. However, bulk download of large quantities of data will initially need to be restricted for performance reasons. Later versions of the repository foresee replication services.
(ii) Provide methods (implemented on or 'near' warehouse storage server) that automatically pre-process the 'incoming' data to provide cleansed, normalized and enriched data in multiple formats important for subsequent analysis (this, basically, provides 'cached' computations frequently needed for data mining – this is the main purpose of a data warehouse).
(iii) Provide methods (implemented on or 'near' warehouse storage server) that implement commonly needed data mining algorithms to be run on user-selected subsets of the warehouse. Provide these operations as web/grid services.
(iv) Provide user-friendly web portals and clients for web-enabled access to the grid/web services defined above.

The requirements related to storing and accessing data can be addressed by means of conventional database technology such as a relational database management system like Postgres or Oracle. The administrative aspects of user access, the management of computational resources, and the seamless integration of the data warehouse components into a possibly distributed analysis process are much more challenging issues. This project incorporates key concepts of grid technologies such as *Open Grid Service Architecture* (OGSA) into the basic design and rationale of a data warehouse.

Currently, we are focusing on the design of the actual MD unfolding simulation data warehouse. The design is subdivided into three parts:

(i) *Conceptual model.* The data warehouse's conceptual design describes on a high level the conceptual entities and their relationships. It is independent of the actual database model. The conceptual design is concerned with the identification of the entities and their relationships (a.k.a. the 'miniworld').
(ii) *Logical data model.* Defines the mapping of the conceptual model to a specific data or database model. Typically, entity-relationship modeling and object definition languages such as the Unified Modeling Language (UML) are used here [17]. The logical design specifies the entities, their attributes and attribute domains, and how the relationships between the entities are realized. The logical design depends on the actual database model; for example, if a relational database is chosen, then the relationships between the entities are realized via the referential key concept. Furthermore, the logical data model specifies the data tables, the meta-data tables, and dictionary tables.
(iii) *Physical data model.* Refers to the collection of files, indices, and other storage structures that make up the final database on a specific software and hardware platform.

2.4.2. Conceptual data model

We can distinguish between three levels of data:

(i) Folding/unfolding simulation trajectory data,
(ii) Simulation parameter data, and
(iii) Property analysis data.

The trajectory data have a spatial and a temporal component. Each molecule in a MD simulation consists of a number of residues, which are made up of a number of atoms. For each atom, the Cartesian coordinates and the velocities are captured over a series of time points. Adopting the notation of Murdock et al. [18], each MD simulation is accompanied by meta-data, describing the experimental parameter settings, information on the software used, who conducted the experiment, where, when, etc. Finally, property analysis data refer to derived data, such as RMSD, SASA and other molecular properties.

Figure 6 depicts the *entity-relationship model* of the involved conceptual entities and their relationships.

For example, an experimentator is conducting none, one, or many MD simulation experiments. One specific simulation, on the other hand, is carried out by exactly one experimentator. One specific MD experiment is carried out with a set of clearly defined parameter settings, and the same experimental settings may apply to multiple simulations.

2.4.3. Logical data model

The logical data model consists of a mapping (or realization) of the conceptual data model to a specific data model – in

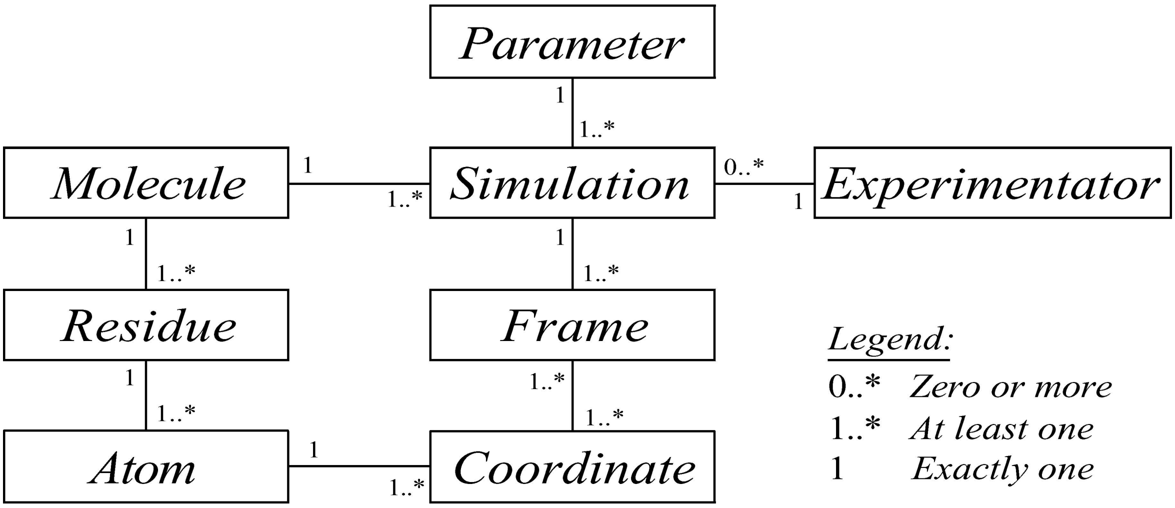


Fig. 6. The conceptual data model of the data warehouse.

this case the *relational data model* used in relational database management systems. In this phase of the design process, all relevant attributes of the entities are identified. Furthermore, the key integrity constraints, domain integrity constraints, and semantic integrity constraints are defined. To warrant the first constraints, surrogate primary keys are introduced for each relation. The consistency among the relations is maintained by the referential integrity constraint (foreign key concept). Domain integrity constraints specify the domains (value ranges) of attributes. For each attribute, the valid values have to be defined. For example, the domain of the attribute *X* of the entity *Coordinate* is *float*. Figure 7 shows the logical data model including the referential and domain integrity constraints (due to space limitation, the tables are not shown completely).

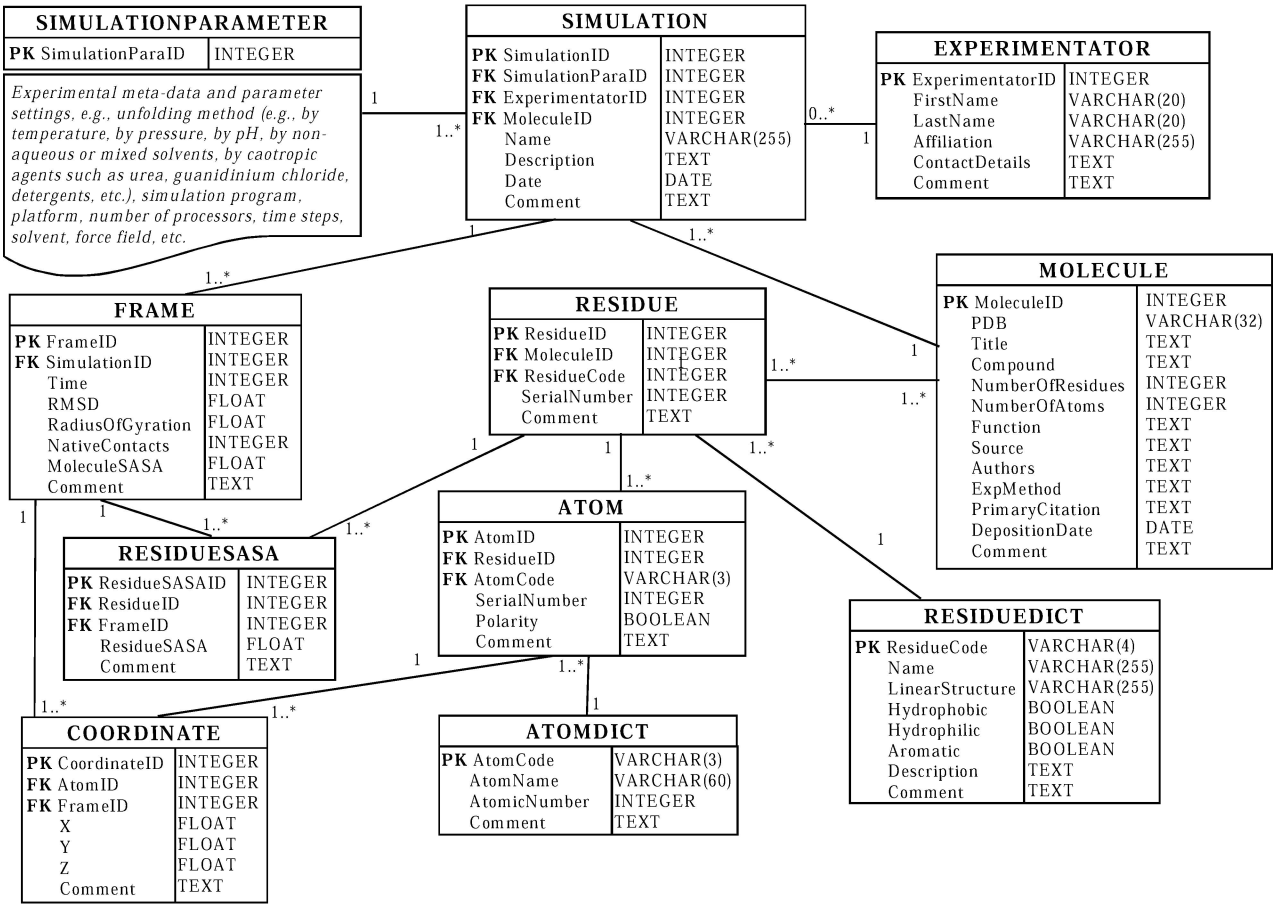
The first prototype of the data warehouse data model is implemented based on the PostgreSQL database management system. The database scheme reflects the entity-relationship as shown in Figure 7 and the logical data model as depicted in Figure 6. Semantic integrity is warranted by means of standard database concepts such as triggers, integrity constraints (e.g., uniqueness constraints on data records such as *ResidueSASAID*, *ResidueID*, and *FrameID* in the table RESIDUESASA), and domain integrity constraints.

For each simulation, there exist a number of clearly defined, unique experimental parameters such as details concerning the simulation program being used, the hardware platform on which the simulation program is executed, the number of processors, the solvent, the force field settings, etc. These data are stored in the table SIMULATIONMETA.

The central table in the data model is SIMULATION, bringing together the data and the meta-data. Each MD simulation experiment is carried out by an experimentator, whose details are managed in the table EXPERIMENTATOR. For each time frame, the relative SASA of the residues can be computed and stored in the table RESIDUESASA. This table is linked to the table RESIDUE, which contains an entry for each residue of the molecule. transthyretin, for example, consists of 127 residues, some of which comprising the same amino acid. Therefore, the table RESIDUE contains the attribute *SerialNumber* to identify the amino acids within the molecule. For example, the attribute *ResidueCode* may be PRO and the associated attribute *SerialNumber* may be 1. The table RESIDUE is linked to the dictionary table RESIDUEDICT, which contains details about the residues. For example, the residue identified by *ResidueCode* = PRO has the attribute *Name* = Proline.

For each frame, the *xyz*-coordinates of all atoms are stored in the table COORDINATE. This table is linked to the table ATOM, which in turn is linked to the table RESIDUE via the foreign key *ResidueID*. The table ATOMDICT is a dictionary table for the table ATOM. For example, one entry in the table ATOM may be {101, 9, C, 2}, referring to the second carbon atom in the residue with the identification number 9. The associated entry in the table ATOMDICT is {C, carbon, 6}.

From the trajectory data, various measures can be computed, describing the structural changes over time, e.g., the root mean square deviation from the initial molecular structure, the radius of gyration, the number of native contacts, and the global solvent accessible surface area of the molecule. These structural changes are stored in the data table PROPERTYANALYSIS.



SIMULATIONPARAMETER	
PK SimulationParaID	INTEGER

Experimental meta-data and parameter settings, e.g., unfolding method (e.g., by temperature, by pressure, by pH, by non-aqueous or mixed solvents, by caotropic agents such as urea, guanidinium chloride, detergents, etc.), simulation program, platform, number of processors, time steps, solvent, force field, etc.

SIMULATION	
PK SimulationID	INTEGER
FK SimulationParaID	INTEGER
FK ExperimentatorID	INTEGER
FK MoleculeID	INTEGER
Name	VARCHAR(255)
Description	TEXT
Date	DATE
Comment	TEXT

EXPERIMENTATOR	
PK ExperimentatorID	INTEGER
FirstName	VARCHAR(20)
LastName	VARCHAR(20)
Affiliation	VARCHAR(255)
ContactDetails	TEXT
Comment	TEXT

FRAME	
PK FrameID	INTEGER
FK SimulationID	INTEGER
Time	INTEGER
RMSD	FLOAT
RadiusOfGyration	FLOAT
NativeContacts	INTEGER
MoleculeSASA	FLOAT
Comment	TEXT

RESIDUE	
PK ResidueID	INTEGER
FK MoleculeID	INTEGER
FK ResidueCode	INTEGER
SerialNumber	INTEGER
Comment	TEXT

MOLECULE	
PK MoleculeID	INTEGER
PDB	VARCHAR(32)
Title	TEXT
Compound	TEXT
NumberOfResidues	INTEGER
NumberOfAtoms	INTEGER
Function	TEXT
Source	TEXT
Authors	TEXT
ExpMethod	TEXT
PrimaryCitation	TEXT
DepositionDate	DATE
Comment	TEXT

RESIDUESASA	
PK ResidueSASAID	INTEGER
FK ResidueID	INTEGER
FK FrameID	INTEGER
ResidueSASA	FLOAT
Comment	TEXT

ATOM	
PK AtomID	INTEGER
FK ResidueID	INTEGER
FK AtomCode	VARCHAR(3)
SerialNumber	INTEGER
Polarity	BOOLEAN
Comment	TEXT

RESIDUEDICT	
PK ResidueCode	VARCHAR(4)
Name	VARCHAR(255)
LinearStructure	VARCHAR(255)
Hydrophobic	BOOLEAN
Hydrophilic	BOOLEAN
Aromatic	BOOLEAN
Description	TEXT
Comment	TEXT

COORDINATE	
PK CoordinateID	INTEGER
FK AtomID	INTEGER
FK FrameID	INTEGER
X	FLOAT
Y	FLOAT
Z	FLOAT
Comment	TEXT

ATOMDICT	
PK AtomCode	VARCHAR(3)
AtomName	VARCHAR(60)
AtomicNumber	INTEGER
Comment	TEXT

Fig. 7. The logical data model of the data warehouse.

3. CONCLUSIONS

This paper presents ongoing work on the development of a public grid-enabled data warehouse for protein unfolding data. We outlined the issues involved – in particular, issues in providing the warehouse as a grid-enabled facility. Particular challenges include (a) the data volume, especially if users want to access large subsets of future versions of the warehouse, (b) the type of data analysis to be performed on unfolding simulation data (currently, the authors are exploring this aspect in a separate project), (c) the provision of grid-enabled data access and analysis. With regard to (c) we are currently investigating OGSA-DAI as a technology to provide data access and future data mining functionality as web/grid services (especially pre-defined data mining services that can run on or 'near' the data repository of the data warehouse). However, provision of mechanisms that facilitate the shipping of data mining computations or algorithms to the data warehouse is still an open issue, as the interactions of such algorithms with the data in a data warehouse are complex. The first version of the data model for the data warehouse has been implemented. Future work includes the completion of the data warehouse (including multiple 'views' of the data content relevant to the molecular dynamics community), implementation of pre-processing function and data analysis web/grid services, and access to the warehouse via grid-enabled web clients and portals.

REFERENCES

1. Leopold PE, Montal M, Onuchic JN. Protein folding funnels: A kinetic approach to the sequence-structure relationship. Proc Natl Acad Sci USA 1992; 89: 8721–8725.
2. Pande VS, Baker I, Chapman J, Elmer SP, Khaliq S, Larson SM, Rhee YM, Shirts MR, Snow CD, Sorin EJ, Zagrovic B. Atomistic protein folding simulations on the submillisecond time scale using worldwide distributed computing. Biopolymers 2003; 68: 91–109.
3. Brito RMM, Dubitzky W, Rodrigues JR. Protein folding and unfolding simulations: A new challenge for data mining. In

Dubitzky W (guest editor) OMICS: A Journal of Integrative Biology 2004; 8(2): 153–166.

4. Jeong KJ. MGrid: A Molecular Grid System. At http://imc.konkuk.ac.kr/~mgrid/
5. Feig M, Abdullah M, Johnsson L, Pettitt BM. Large scale distributed data repository: Design of a molecular dynamics trajectory database. Future Generation Computer Systems 1999; 16: 101–110.
6. Zhang Y, Peters MH, Li Y. Nonequilibrium multiple time scale dynamic simulation of receptor-ligand interactions in structured protein systems. Proteins, Structure, Function and Genetics 2003; 52: 339–348.
7. Rodrigues JR, Brito RMM. Amyloid formation by transthyretin: How much unfolding is required? Biophys J 2004; 86(1): 340a.
8. Brito RMM, Damas AM, Saraiva MJS. Amyloid Formation by transthyretin: From Protein Stability to Protein Aggregation. Current Medicinal Chemistry – Immun. Endoc. & Metab. Agents 2003; 3: 349–360.
9. Kale L, Skeel R, Bhandarkar M, Brunner R, Gursoy A, Krawetz N, Phillips J, Shinozaki A, Varadarajan K, Schulten K. NAMD2: Greater scalability for parallel molecular dynamics. J Comp Physics 1999; 151: 283–312.
10. Dubes AC, Jain AK. Algorithms for Clustering Data. New York: Prentice Hall, 1988.
11. Shea JE, Brooks CL 3rd. From folding theories to folding proteins: A review and assessment of simulation studies of protein folding and unfolding. Ann Rev Phys Chem 2001; 52: 499–535.
12. Moss L, Adelman A. Data warehousing methodology. J Data Warehousing 2000; 5: 23–31.
13. Lamehamedi H, Szymanski B, Shentu Z, Deelman E. Data replication strategies in Grid environments. In *Proc. 5th Intl. Conf. Algorithms and Architecture for Parallel Processing (ICA3PP'2002)*, Bejing, China, October 2002, IEEE Computer Science Press, Los Alamitos, CA, 2002, pp. 378–383.
14. Keene C. Achieve bottleneck-free grid applications: Learn how to eliminate data bottlenecks for grid computing. At http://www.javaworld.com/javaworld/jw-08-2004/jw-0802-grid.html, 2005.
15. The Open Grid Services Architecture – Data Access and Integration (OGSA-DAI) project and website at http://www.ogsadai.org.uk/, 2004.
16. DAIS, Database Access and Integration Services Working Group at http://forge.gridforum.org/projects/dais-wg/, 2004.
17. Gornik D. Entity relationship modeling with UML. At http://www-106.ibm.com/developerworks/rational/library/content/03July/2500/2785/2785_uml.pdf, 2004.
18. Murdock S, Tai K, Ng MH, Johnston S, Fangohr H, Wu B, Cox S, Essex J, Sansom M, BioSimGrid Tutorial, at http://www.biosimgrid.org/docs/2004/manuals/Tutorial/tutorialManual.pdf, 2004.
19. Rodrigues JR, Brito RMM. How important is the role of compact denatured states on amyloid formation by transthyretin? *In* "Amyloid and Amyloidosis", 2004: 323–325, Gilles Grateau, Robert A. Kyle and Martha Skinner eds., CRC Press.