UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Development of a generic, structural bioinformatics information management system and its application to variation in foot-and-mouth disease virus proteins

by

Tjaart Andries Petrus de Beer

Submitted in partial fulfilment of requirements for the degree Philosophiae Doctor (Bioinformatics)

in the Faculty of Natural and Agricultural Sciences

Bioinformatics and Computational Biology Unit

Department of Biochemistry

University of Pretoria

Pretoria

November 2008

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Declaration

I, Tjaart Andries Petrus de Beer, declare that the thesis/dissertation, which I hereby submit for the degree Philosophiae Doctor at the University of Pretoria, is my own work and not previously been submitted by me for a degree at this or any other tertiary institution.

SIGNATURE .... DATE ....

Acknowledgments

I want to thank the following people:

- My parents who supported me through all my studies.
- My supervisors for all their guidance, support and help during the last few years.
- All the various funding agencies who made it possible for me to study.
- All my friends and special people for their valuable support.
- My fellow students at the BCBU over the years.
- Irene, ti ringrazio per essermi stata accanto, sostenendomi sempre. Te ne sono davvero grato.

Summary

Structural biology forms the basis of all functions in an organism from how enzymes work to how a cell is assembled. *In silico* structural biology has been a rather isolated domain due to the perceived difficulty of working with the tools. This work focused on constructing a web-based Functional Genomics Information Management System (FunGIMS) that will provide biologists access to the most commonly used structural biology tools without the need to learn program or operating specific syntax. The system was designed using a Model-View-Controller architecture which is easy to maintain and expand. It is Python-based with various other technologies incorporated. The specific focus of this work was the Structural module which allows a user to work with protein structures. The database behind the system is based on a modified version of the Macromolecular Structure Database from the EBI. The Structural module provides functionality to explore protein structures at each level of complexity through an easy-to-use interface. The module also provides some analysis tools which allows the user to identify features on a protein sequence as well as to identify unknown protein sequences. Another vital functionality allows the users to build protein models. The user can choose between building models online or downloading a generated script. Similar script generation utilities are provided for mutation modelling and molecular dynamics. A search functionality was also provided which allows the user to search for a keyword in the database. The system was used on three examples in Foot-and-Mouth Disease Virus (FMDV). In the first case, several FMDV proteomes were reannotated and compared to elucidate any functional differences between them. The second case involved the modelling of two FMDV proteins involved in replication, 3C and 3D. Variation between the several different strains were mapped to the structures to understand how variation affects enzymes structure. The last example involved capsid protein stability differences between two subtypes. Models were built and molecular dynamics simulations were run to determine at which protein structure level stability was influenced by the differences between the subtypes. This work provides an important introductory tool for biologists to structural biology.

Contents

List of Abbreviations

Å	Angstrom
aa/AA	Amino Acid
A	Alanine
ANSI	American National Standards Institute
C	Cysteine
CHARMM	Chemistry at HARvard Macromolecular Mechanics
CG	Conjugate Gradient
D	Aspartic acid
DNA	Deoxyribonucleic Acid
E	Glutamic acid
EBI	European Bioinformatics Institute
EC	Enzyme Commission
Ec	*Escherichia coli*
EM	Electron Microscopy
EST	Expressed Sequence Tag
F	Phenylalanine
FMDV	Foot and Mouth Disease Virus
FuGE	Functional Genomics Experiment
FunGIMS	Functional Genomics Information Management System
G	Glycine
GB	Gigabytes
H	Histidine
HAV	Hepatitis A Virus
HRV	Human Rhino Virus

HS	Heparan Sulfate
HMM	Hidden Markov Model
I	Isoleucine
ISO	International Standards Organization
K	Kelvin
K	Lysine
kb	kilobases
kD	kilo Dalton
L	Leucine
M	Methionine
MB	Megabytes
MSD	Macromolecular Structure Database
MVC	Model View Controller
N	Asparagine
NCBI	National Center for Biotechnology Information
ns	nanosecond
P	Proline
PDB	Protein Data Bank
Pfam	Protein families database
ps	picosecond
Q	Glutamine
R	Arginine
S	Serine
SD	Steepest Descent
sid	System Identifier
SQL	Structure Query Language
T	Threonine
TMHMM	Trans-Membrane Hidden Markov Model
V	Valine
W	Tryptophane

XML	eXtensible Markup Language
Y	Tyrosine

List of Figures

List of Tables

Chapter 1

Introduction

Protein structure affects everything around us from how enzymes work, how cells are assembled to how diseases function and spread. Biologists can use this information to cure diseases, understand how enzymes work and improve the quality of life for people all over the world. This study will highlight the important role of structural bioinformatics in solving modern day problems facing biologists. One of the main reasons for biologists under utilizing structural bioinformatics tools, is the perceived, and sometimes inherent complexity and setup of the tools. This problem can be addressed by designing more intuitive systems for biologists to obtain structural biology results. The problem is not just making the tools easy to use but also the management of the generated data. Integrating the data management and analysis tools into one, easy-to-use package would greatly assist biologists in accelerating knowledge discovery in structural bioinformatics and hence in solving pressing problems.

A modern structural biology application can be broadly divided into two basic components. The actual analysis tool which is used to generate the results and a system to manage the data generated by this application. Each of these play an integral role in the end result. If the analysis tool is based on wrong or erroneous data, the results are wrong. If the data is incorrectly managed, the analysis tool which relies on the data will give false results. Each of these roles will be discussed in the next few sections.

A good example of the role structural bioinformatics can play in solving problems, is the threat of Foot-and-Mouth Disease Virus (FMDV) to livestock all over the world. This virus can cause massive economic losses and affect people from all walks of life. Local researchers have identified some areas which would help in understanding problems such as variation in the FMDV 3C protease and 3D RNA polymerase, full proteome variation between serotypes and protein function and structure differences between various FMDV serotype capsid proteins.

The ideal solution to FMDV would be a capsid-based vaccine, but local researchers have found that there are stability differences between FMDV serotypes. Identifying the structural effects of the differences found in each serotype, could help to improve vaccine design. The capsid proteins are also important in infection and thus understanding what influence the differences have on the structure will provide vital information in understanding FMDV infection. FMDV replication speed differences have been tracked to differences in the 3C and 3D proteins. Mapping the differences to a structure and investigating the effect these differences have on function, will allow for a better understanding of virus replication and which areas of the protein are more conserved. Full proteome variation analysis will help to identify regions and features which are important to the virus. Comparing serotype-specific characteristics to proteome variation, differences between the serotypes can be mapped. The variation can then be tracked to features such as secondary structure or post translational modifications.

This is a typical example of where a group would require access to structural biology and bioinformatics tools, yet lack the resources and knowledge on how to proceed. This study aims to address this issue by providing structural bioinformatics tools that can assist the researchers in answering structural biology questions. The results can provide answers as well as guide biologists in designing experiments to verify the results from the tools.

The following sections will address the issues that biologists and structural bioinformatics programmers face with regard to the massive amount of data produced in modern high-throughput biology. Topics such as biological data management, data storage and data access will be discussed together with how it influences biologists and programmers alike. Each section is by no means an exhaustive overview of a topic but a discussion of how it applies to biologists with structural biology challenges.

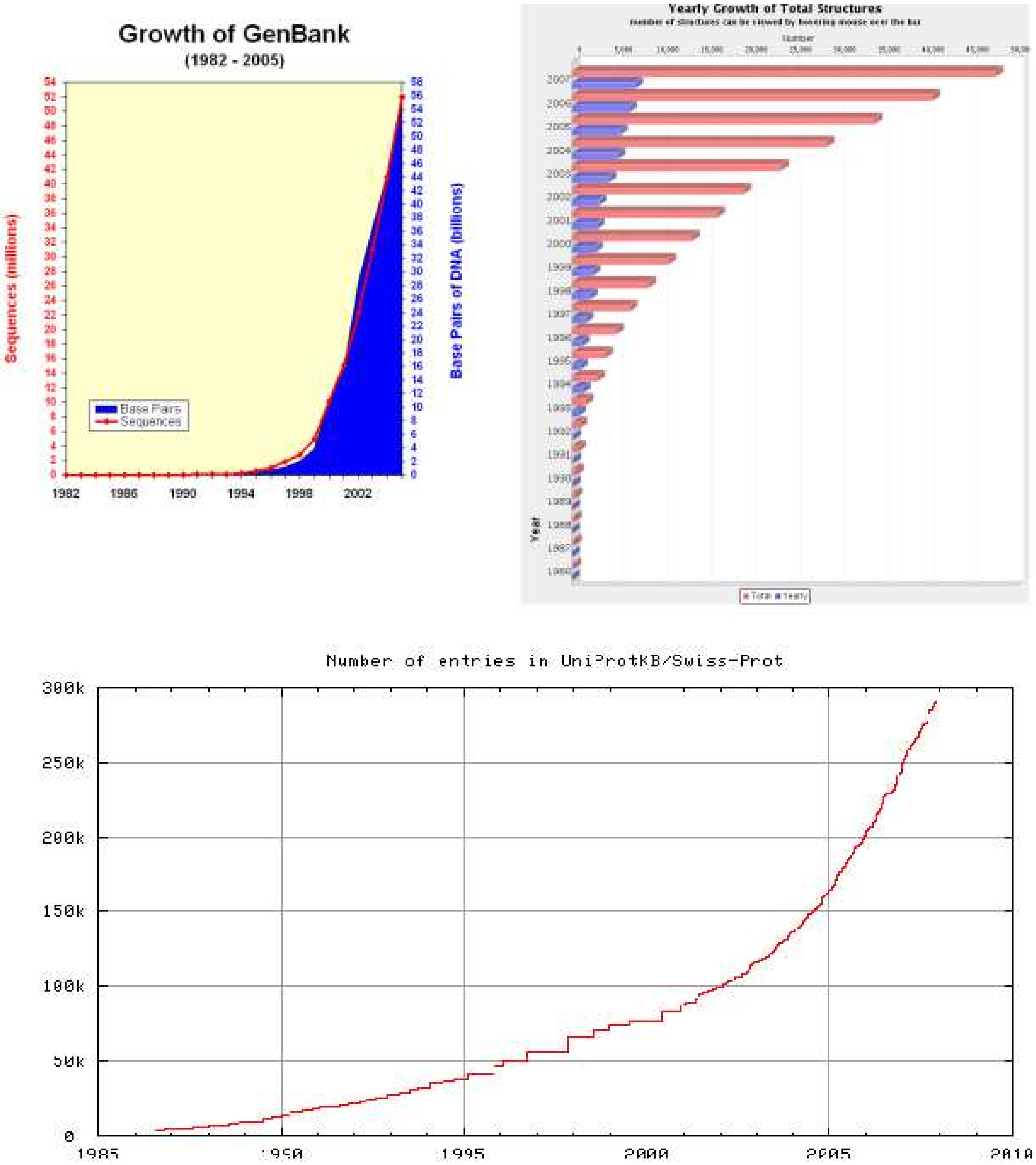


Figure 1.1: The exponential growth in data deposits as seen in GenBank, the PDB and SWISS-PROT (http://www.ncbi.nlm.nih.gov, http://www.pdb.org, http://expasy.ch).

1.1. Biological Data Management

Data production in modern biological sciences is growing at an exponential rate. This is due to high throughput methods (structure as well as sequence-based) and genome sequencing projects. Data banks such as GenBank (Benson *et al.*, 2006), the Protein Data Bank (PDB, Berman *et al.*, 2000) and Swiss-Prot (Gasteiger *et al.*, 2003) have all shown exponential growth in the last few years (Fig. 1.1). This exponential growth in data production has resulted in enormous datasets that need to be stored, curated and managed. Larger databases have overcome the problem of data management to a certain extent by forcing data depositors to conform to a certain format when depositing data. This allows for a more automated approach to data management. Some data banks have even gone further and are employing people to verify and cross check data before it is deposited. A good example of this is Swiss-Prot, which is a database dedicated to manual curation and storage of protein sequences. Before a protein sequence is accepted into Swiss-Prot, a human will verify the function and description of the protein by looking at various papers about the protein and comparing the data. If the function and description are deemed to be correct, it is included in Swiss-Prot. This type of data management is highly labour intensive and takes a long time for each protein sequence to be verified. Swiss-Prot also hosts another section of protein sequences called TrEMBL. TrEMBL is a computer translated version of cDNA sequences found in the EMBL database and thus contains very little annotation and may be of variable quality or hypothetical.

Not only is the storage of these datasets a problem but also the presentation of the data to the user in an effective way. The large data banks have improved during the last few years by presenting users with easy to use web-based interfaces to search the data. This allows the users to easily find and access the data located in a specific database. Larger service providers such as the PDB and SWISS-PROT, have taken it one step further by incorporating data from other sources as well when a user views a record. The PDB for example links out to Pubmed, Pubchem and to protein fold details at SCOP (Conte *et al.*, 2000) and CATH (Pearl *et al.*, 2005), while Swiss-Prot provides links to EMBL, PIR, UniGene, ModBase, InterPro and Pfam among others.

1.2. Data Storage

All data banks/databases rely on the storage and linking of data. Small amounts of data are easy to store and process with the processing power available today, but certain datasets are just too large e.g. the GenBank dataset in May 2008 was 66 GB and that of the PDB 6.5 GB of compressed text files (approximately 27 GB uncompressed). These large datasets require an efficient and fast way of storing and retrieving data. A good example is the Macromolecular Structure Database (MSD, Boutselakis *et al.*, 2003) from the European Bioinformatics Institute (EBI). Their approach was to parse out all the data from the PDB, correct it as far as possible using external analytical chemistry tools, enhance the data by extracting cross links between different data types and then storing it in a custom relational database. This has the drawback of increasing the dataset size when compared to the PDB dataset (27 GB uncompressed vs. 300 GB uncompressed for MSD). However the added advantage is that the relevance of the data is increased and by storing it in a relational database, it also increases the speed and efficiency by which the dataset can be queried by users.

There are two main types of general data storage: flat-file based or storage in a relational database such as Oracle or MySQL. Both have advantages and disadvantages (Table 1.1). Flat-files are defined as data being stored in a single file on disk with fields separated by delimiters. Relational databases are defined as databases which define relations between data sets using the Structured Query Language (SQL) to perform operations on the data, using a database management system.

SQL is a computer language that was designed to facilitate the management and retrieval of data as well as database access control and schema management. SQL has been standardized by American National Standards Institute (ANSI, http://www.ansi.org) and the International Organization for Standardization (ISO, http://www.iso.org). This was done to enable applications to be moved between different database systems without major code rewrites.

Another major problem in storing data is redundancy. A good example of this is GenBank. There is an enormous number of sequences which only differ by one or two bases

Table 1.1: Comparison between Flat-file data storage and Relational database data storage (Doyle, 2001).

	Relational database	Flat-file
Advantages	- Data entered only once - Files/tables are linked - Can handle complex search criteria	- Fast for storage of static information - Access speed limited by disk speed - Can be stored on shared file system
Disadvantages	- Usually hosted on one file server - Security need to be considered carefully - Direct users need additional training	- Difficult to search - Difficult to change/update data - No relations between different files

or amino acids. These sequences are usually Expressed Sequence Tags (ESTs) which were deposited. All of these ESTs are distributed with the full version of GenBank. The non-redundant version is distributed without these "duplicates". The PDB has a similar policy with regard to crystal structures. One protein sequence may have a few different conformations/structures depending on the crystallization conditions and ligands present. Some databases remove this redundancy to create a smaller, more manageable dataset, yet these redundant sequences contain a wealth of data that can also be utilized. Thus once again, there is a trade off between storing a smaller non-redundant dataset versus storing a larger, redundant dataset.

1.3. Data Models

All of the databases/data banks discussed in the previous sections store data in some way or another. Some of these systems such as MSD use a data model to store the data. A data model is a description of the organization of, and relationships between, data in a manner that reflects the information structure. This model is also usually used as a database structure.

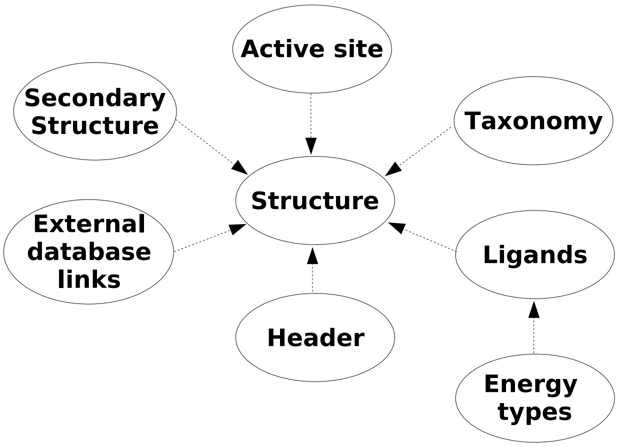


Figure 1.2: A high level overview of the data model of MSD (http://www.ebi.ac.uk/msd-srv/docs/dbdoc/). The main Structure entity is enhanced by linking it to other data types.

The data model used in MSD is based on the hierarchical structure of proteins and works in a top down manner. A structure serves as the main data object and other types of data such as active sites, ligands and taxonomy are added (Fig. 1.2). A structure entity is divided into many different sections (Fig. 1.3). Through a series of cross-links these different entities contain all the data about a structure. The MSD data model allows for various cross-links and external references to be incorporated into the model thus adding value to the pure structure data.

The Functional Genomics Experiment (FuGE) is an attempt to facilitate data standard convergence between the different high-throughput techniques used in biology (Jones *et al.*, 2006; Jones *et al.*, 2007). FuGE provides a foundation for the description of complete laboratory workflows and provides mechanisms for developing new data formats and for the integration of data between techniques. FuGE was designed so that different facets of a “’omics” experiment can be captured and stored. This includes data such as protocols, sample sources and results. Providing a common platform to store common data types would allow for data to be shared among different groups. This would, for example, allow a microarray study using MicroArray Gene Expression object (MAGE) data model to share a basic set of information with someone doing a study using the Proteomics Standards Initiative (PSI) data models. The FuGE model also allows for rich

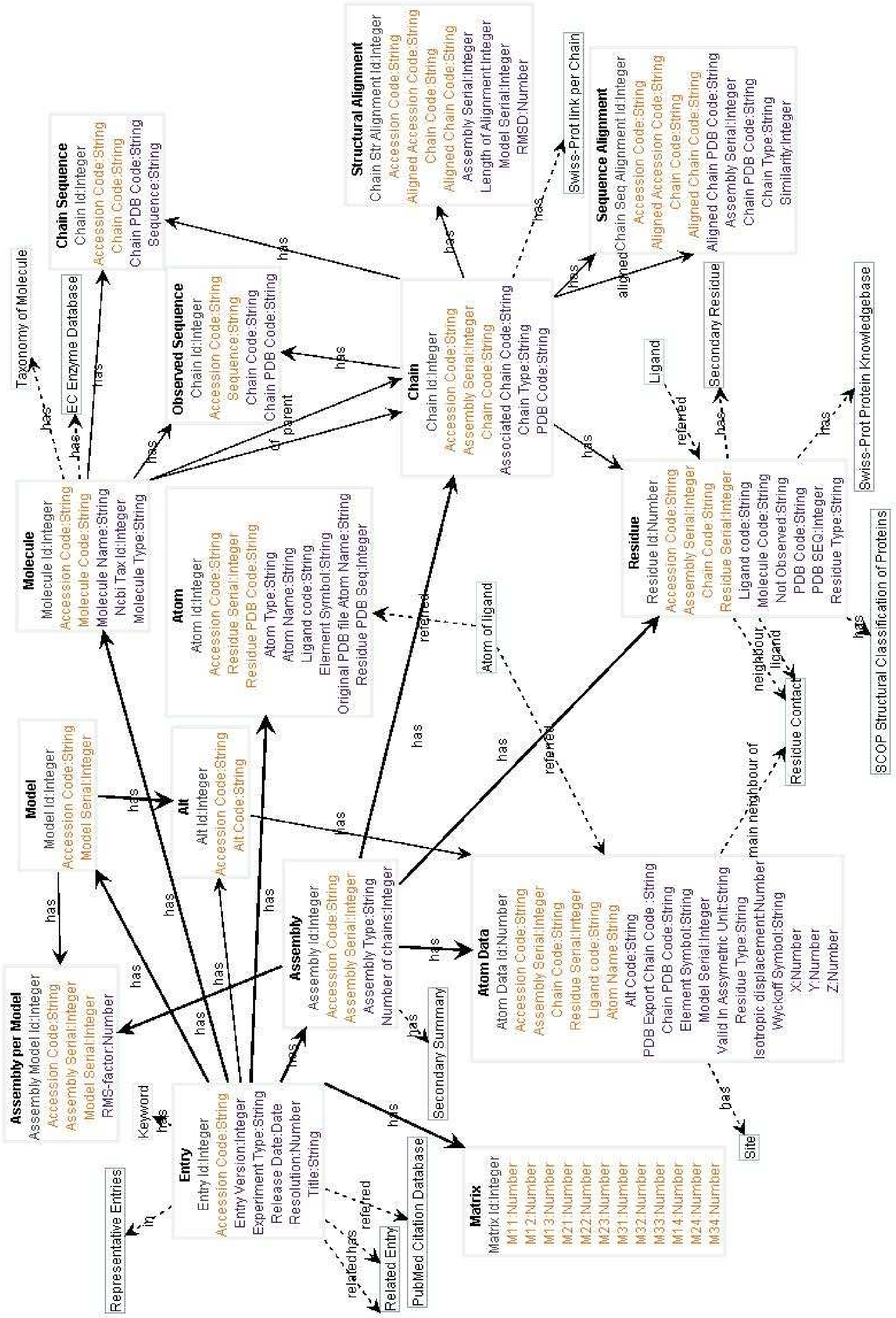


Figure 1.3: The different entities belonging to the Structure entity in MSD (http://www.ebi.ac.uk/msd-srv/docs/dbdoc). This data model is a representation of the data as well as a diagram of the actual database structure.

Table 1.2: The two categories of FuGE with the packages in each category (Jones *et al.*, 2007).

FuGE	Common	Audit
		Description
		Measurement
		Ontology
		Protocol
		Reference
	Bio	ConceptualMolecule
		Data
		Investigation
		Material

annotation of samples and because of the underlying standard model, it will allow data sharing between samples and methods.

FuGE has 10 different packages contained in two categories: Common and Bio (Jones *et al.*, 2007). The FuGE.Common class consists of Audit, Description, Measurement, Ontology, Protocol and Reference (Fig. 1.4). Audit provides security settings, Measurement provides slots for values and units, Ontology provides for external referencing vocabularies, Protocol provides a model for procedures and workflows and Reference provides links to external database references. Description allows free text annotations and descriptions for all objects and inherits directly from Describable. All objects in FuGE can be represented under the Common category. FuGE.Common has two base classes: Describable and Identifiable. All FuGE objects belong to either one of these classes.

Each of these classes are further separated to provide adequate methods to store protocols and samples. The Identifiable base class provides a unique identifier for each object in the system and Identifiable inherits from Describable. This provides each object in the FuGE system with a unique identifier which is linked to a free text description and security settings. Identifiable also provides a logical point from which to extend the FuGE system. FuGE.Bio contains ConceptualMolecule, Data, Investigation and Material. The ConceptualMolecule category provides classes for the storage of DNA, RNA and amino acid sequences but only in a limited way. In theory this can be extended to other molecules. Data provides a way to link to multidimensional experimental data using the subclass ExternalData. Investigation allows for overall experimental design

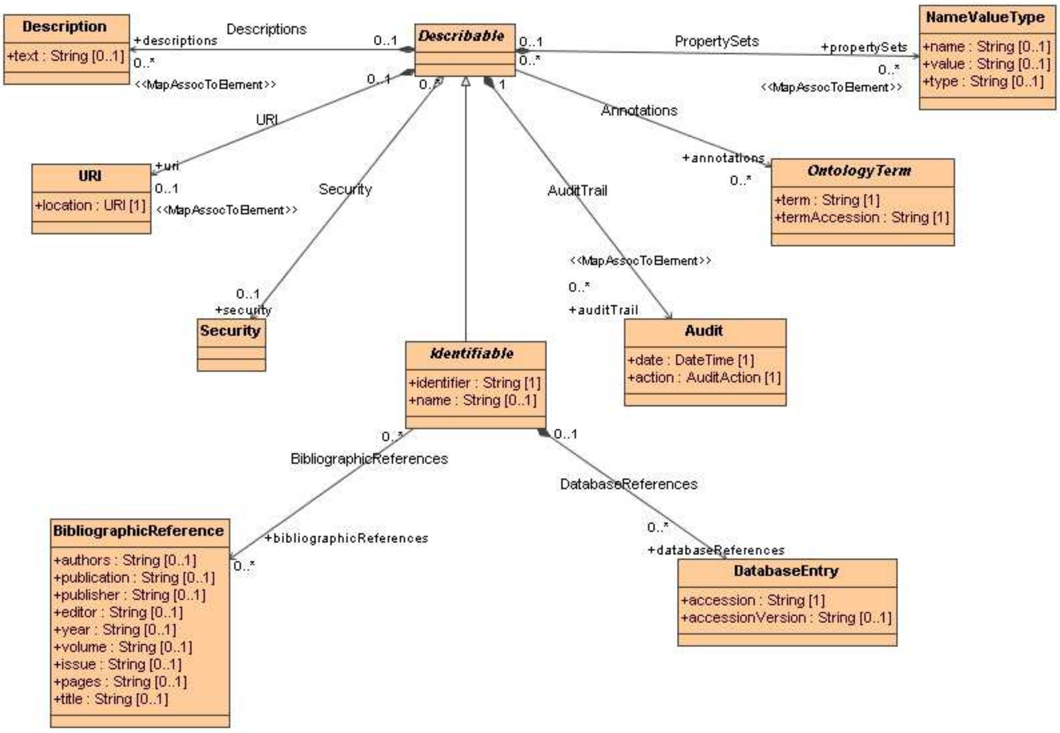


Figure 1.4: The classes of `FuGE.Common` (http://fuge.sourceforge.net). These classes allow for the storage of basic information about each sample.

storage as well as storage of experimental variables. `Material` caters for sample source identification using a controlled vocabulary.

Both MSD and FuGE are successful in storing specific data but most users use a range of data types. Whereas MSD stores all the structural data, it does not cater for storing analysis results nor does it store the methods used. FuGE stores laboratory procedures and protocols but it does not store extensive specific data such as sequences or structures (only basic storage is supported). This basic storage allows for a model that is very compatible between systems and also makes it easy to expand for a specific system. An ideal functional genomics system would store protocols, data and results in a data model compatible with models such as FuGE and MSD. The Functional Genomics Information Management System (FunGIMS) utilizes a data model which stores the most important parts of both FuGE and MSD in one data model without losing the integrity of each separate model yet provides an interface to both. Some parts of FuGE were not used as they represent experimental protocols and conditions and FunGIMS only caters for data storage and analysis.

1.4. Information Management Systems

While major databases host public data, laboratories often need to host their own data in a specialized way. Systems that host data in this way are usually referred to as a Laboratory Information Management System (LIMS). The main characteristic of a traditional LIMS is that it manages data and tracks samples through the system.

The last few years saw an explosion of LIMS, all specialized for dedicated tasks. For example a LIMS simply called LIMS was developed for tracking high throughput genetic sequencing and candidate mutant screening (Voegele *et al.*, 2007), CLIMS (Crystallography IMS) to organize the large amounts of data generated by crystallization experiments (Fulton *et al.*, 2004). PARPs was developed for managing liquid chromatography tandem mas spectrometry and the associated protein identification and data management (Droit *et al.*, 2007), PACLIMS for managing eukaryotic genome-wide mutational screens and the functional annotation thereof (Donofrio *et al.*, 2005), a 2-D gel electrophoresis LIMS was developed to deal with large-scale proteomic studies (Morisawa *et al.*, 2006) and MACSIMS for dealing with data mining from multiple sequence alignments (Thompson *et al.*, 2006). T.I.M.S is an example of a very specific LIMS designed for tracking genotyping data flow and analysis in a laboratory (Monnier *et al.*, 2005).

LIMS users are usually facilities or users who generate relatively large quantities of data in efforts such as large-scale sequencing or high throughput crystallographic studies. The large amount of data needs to be stored efficiently and analyzed in a consistent and effective way. This is one of the major advantages of LIMS but when it comes to detailed data analysis, it can also be a disadvantage. The trade off between being able to store and do basic analysis on a large amount of data and being able to do detailed analysis on a small set of data is one of the drawbacks of LIMS. Some systems like CLIMS can store a large amount of data but does not allow the user to do a detailed analysis of the structure. Other systems such as T.I.M.S. provides a very specific service for a subset

Table 1.3: Comparison between the technologies in currently available LIMS.

LIMS	**Main feature**	**Language**
LIMS	Automated high throughput mutation scanning	MySQL + Java
CLIMS	Crystallization procedure management	MySQL + Java Rich client
PARPs	Liquid Chromatography data management and analysis	Oracle + Perl
PACLIMS	Managing high throughput sequencing data and protocols	PostgreSQL + Perl
MACSIMS	Protein family alignment and data extraction	ANSI C
TIMS	Sample management and parsing of TaqMan data	Visual Basic

of data. LIMS can greatly enhance throughput in a lab as they allow for centralized storage and standardized analysis protocols. All data are treated and interpreted in the same way, providing a big advantage when doing analysis. It also allows users access to centralized analysis tools. All LIMS need to store data in some way. Most LIMS rely on the proven technology of relational databases with additional data stored as flat-files (Table 1.3).

One of the biggest advantages of LIMS is the ability to organize data. This is in sharp contrast to classical biology where results were written on paper in laboratory books and data stored on various CDs and DVDs. LIMS provides a way to store and search through data in an organized and systematic manner, thus increasing efficiency. The organization, analysis and data storage abilities of a LIMS will be illustrated in chapters 3-5 when various structural problems such as the capsid proteins in FMDV are investigated. Web-based systems provide an advantage to novice users venturing into structural bioinformatics, as web interfaces are experienced as a familiar environment, and preclude the need for the installation of local software, which sometimes has complicated dependencies. Available web-based systems for structural bioinformatics vary in terms of the level of analysis functionality available and the level of knowledge required for use. The Spice DAS client is an example of a system for viewing and performing basic exploration of a protein structure, starting with a PDB ID (Prlic *et al.*, 2005). Spice also provides a DAS-based annotation of especially structural properties of the protein being viewed. Web helper-based applications such as Cn3D also allow extensive visualization of protein features and structural alignments, together with the preparation of protein structure figures for reports and publication (Hogue, 1997). STRAP provides a Java web-start application to perform extensive multiple alignments and superimposition of protein structures, together with protein structure views and sequence-based analysis of structural features (Gille and Frömmel, 2001). Various other structural tools are also available depending on the needs of the user.

1.5. Common Structural Analysis Needs of Biologists

The Holy Grail of structural biology is the ability to predict the three dimensional structure of a protein given only the amino acid sequence. Although it sounds relatively easy, the solution to this problem is one of the most sought after in science. As protein structure is inherently linked to protein function, knowing the structure of a protein allows one to derive the function of that protein and change it. Once a structure can be predicted accurately from sequence, it allows the researcher to do *in silico* mutations and obtain a reliable result in a short space of time. This will not replace the need for experimental work, but provide assistance to guide experiments better. It will eliminate various problems encountered with proteins not expressing or not crystallizing. Protein structure can also help guide a researcher in designing more efficient and accurate experiments to address biological problems.

Biologists are familiar with working with DNA sequences or proteins *in vitro*. Often during this process, very little time is spent thinking about the protein in three dimensions. When keeping a three dimensional picture in mind, it gives a new perspective on the problem. If the protein structure is known or well studied it allows for much easier data retrieval, but when working on an unknown structure, the task of getting information about a protein can become rather daunting. By adding protein structural knowledge, they can guide or enhance experiments e.g. using a protein structure to identify possible sites for mutagenesis studies. However, accessing the protein data can sometimes be problematic.

Discussions with biologists have identified a few main problems which often prohibit them from utilizing protein analysis tools. Two main problems were cited, that of accessibility to programs and a lack of knowledge of new programs/databases. More and more programs are being released by authors on the Internet and thus the problem of accessibility will lessen with time. Biologists are generally comfortable with using a few general purpose programs or servers such as Excel, Word, NCBI Blast, the Genbank server and maybe one or two other specific programs or servers. Due to the nature of modern biology, these are the programs they use on a regular basis and they are not thus exposed to other servers and databases. Most of these programs are either web-based or are preinstalled on their computers, thereby leaving biologists with very little interaction regarding program installation and setup. This is in contrast to most open source structural programs which the user has to install by themselves. These mostly run on UNIX-based systems and requires a basic knowledge of the operating system and the program's syntax. Although these problems can be resolved relatively easily, they are seen as a major barrier to the more widespread usage of protein structure programs. In some cases this can be attributed to the perceived complexity of UNIX-type systems. Some authors of programs have realized this and started releasing their programs for the Windows and Apple Macintosh operating systems as well. Although this is a step in the right direction, it still does not solve the problem of setting up the program and the analysis. The ideal solution would be to have a system administrator who is capable in both Windows and UNIX environments, and who will assist the users with setting up these programs. Unfortunately, the responsibility usually falls on the researcher to install and manage the programs.

Another factor mentioned was the lack of knowledge of available databases or programs. This problem is two-fold. Firstly biologists should strive to read beyond their own field of interest, and not be hesitant to search for programs or servers. Secondly some programs or databases are simply not published in well known journals. The Nucleic Acids Research journal tries to address both of these problems with a yearly, open access issue of all the known, biological databases and servers but this does not cover any structural bioinformatics programs. A good approach, however, would be to have someone with a strong interest in structural bioinformatics, keep abreast of developments in the field. Even something as simple as subscribing to journal alerts, would be helpful. The best approach would be to have a person such as a postdoctoral student or technical staff member dedicated to looking for new programs, making them available and providing support for these programs. Such a person should ideally be aware of the different types of projects in a group, have a good biology background and be capable of installing and managing the application server as well as run the programs for users. He or she could also develop a website which allows for easy access to all of these tools to local researchers. Most biologists would just need an introduction to the program and a few basic guidelines to get started and continue on their own.

The problem with regards to program use and knowledge lies not only with the users thereof but also with the programmers. A program that has a good user interface with clearly defined functions and a good explanation of each step, is as valuable to the user as the person guiding them. The onus is on programmers to provide documentation, examples and a good interface for users, but unfortunately this is lacking in many programs. Another problem, which can be traced to the point-and-click method used in Windows, is the lack of understanding of file formats and the amount of information contained in a file. Many biologists are hesitant to explore inside files. A good example of this is a PDB file of a protein. Most users will simply load the protein in a visualization program and ignore the valuable information contained in the file header and comments. This problem can only be addressed by making users aware of the extra information and making them comfortable with exploring text files.

Biologists have an array of needs that can be resolved by using structural biology programs. When an unknown protein sequence is identified, most biologists just do a BLAST search in an effort to identify it. Although this usually yields results, there are many more tools that can be used to gain knowledge about a protein. By simply using the sequence, a biologist can identify whether the protein is a membrane protein or not, protein function may be derived from certain sequence patterns contained in the sequence and in some cases even cellular localization can be determined. This can be taken a few levels higher to a three dimensional view of the protein. From a similar protein structure, details such as active site residue conformations, residue interactions and sometimes even function, can be derived. If a similar protein structure is found, a homology model can be built which can guide the biologist in identifying active sites, important secondary structures and guiding site-directed mutagenesis experiments to confirm function. Analysis of the protein structure can also help in identifying surface areas involved in protein-protein interactions and identify flexible areas in proteins. These types of data can all be combined to give a far better understanding of the protein and the way it functions. It can also serve as a starting point for the researcher to investigate function or structure in more detail using molecular biology.

Some of the functionalities mentioned, are available on web servers around the world. Unfortunately, a lack of knowledge often prevented biologists from exploring the full range of programs available. This was one of the motivations behind this project, to provide services to local biologists in a centralized and locally available solution. If these services and programs can be provided and maintained locally, it would benefit researchers greatly.

1.6. The Functional Genomics Information Management System (FunGIMS)

The overall FunGIMS project was conceived when researchers from the Forestry and Agricultural Biotechnology Institute at the University of Pretoria, approached the Bioinformatics and Computational Biology Unit to provide them with bioinformatics support services related to the *Eucalyptus* genome sequencing project. They required a system which would allow them to store their sequences, annotate the data and do various types of analysis on the sequences, all in a local environment. From these requirements, FunGIMS was expanded to include different types of data such as protein structure and small molecule data.

The philosophy behind FunGIMS was based on allowing researchers access to various tools and data sources in an easy to use environment with extensive data management capabilities. Various problems related to data sources and tool access were identified by the researchers. These problems included the slow bandwidth in South Africa, the high costs associated with Internet use and the problem of storing data. The problem of data storage surfaced as one of the primary concerns. Researchers were used to sharing computers and thus stored data on CDs, laboratory books and memory sticks. This resulted in data being distributed in various places and formats. It also posed a problem to supervisors when they needed access to the data of students. A central repository where students can store and analyze data while still allowing supervisors access, would solve this problem to a large extent. The ability to store data was one of the primary factors considered during the design of FunGIMS. To prevent duplication of designing a way to store the data, it was decided to use FuGE as a starting point. As FunGIMS and FuGE had a similar goal with regard to storing data, it would be a great benefit to use this standardized way of storing data. It would also allow researchers the ability to share data between FunGIMS and FuGE compliant systems.

The slow and expensive bandwidth also affected the design of FunGIMS by forcing local repositories of all the major databases to be installed and used. Local repositories of all the major databases were set up. This allowed very fast, local access to these databases which allows for extensive integration between the different databases. All the services would also be hosted locally, thus providing fast access to data and results for the researchers. By keeping all the databases in one, central location, it made administration and updating of the databases far easier. A system administrator could automate the downloading of the database updates and keep all the databases up to date.

Another major goal of FunGIMS was integration between data types. Usually a database only provides one type of data with a few links to related data. Ideally, a system would provide a user with relevant links to other types of data e.g. when looking at a cDNA sequence, the system would provide the user with links to the protein sequence, protein structure (if present), literature references and possible small molecule interactions. This would allow the researcher to get an overall view of the specific product, instead of just looking at the details of a specific length of sequence. Integration between public and private data is also provided but only in the sense that public data is integrated with private data. Thus a user with private data can makes links to and see public data, use private and public data but still prevent access to and integration with the private data.

The overall scientific goal of FunGIMS is to provide the user with a set of tools and access to a large amount of data in one convenient place. The idea is not to replace the use of each individual tool but to provide the user with results which can serve as a starting point. For some biologists this will provide enough information to allow them to continue down a specific route. Others may want to pursue a specific topic in more detail. The separate modules cater for the main types of functional genomics data used. Each module helps the user to do analysis relevant to that topic and tries to provide links to other data types in FunGIMS. Currently FunGIMS consists of Sequence, Structure, Genomic and Small molecule modules and will in the future include modules for Microarray, Genotype and Literature data. Each of these modules are specialized to deal with a different type of data. All the modules overlap with each other to some extent, but each still provides unique functions for the specific data type e.g. proteins have a sequence that is mostly dealt with in the Sequence module whereas the structural aspects are dealt with in the Structure module. Integration between the different datatypes in each module is of vital importance. A good example is that of a user interested in a specific protein and its function. The Structural module will provide access to structural data on the protein, but at the same time it will provide links to DNA sequences, genome locations, genotype data, microarray results and related literature (where available). This will allow the user to see under which conditions the protein is up or down-regulated, which SNPs have been identified in the cDNA sequence and where the DNA coding for the protein is located on the genome. FunGIMS aim to provide an environment in which a user can access different types of data that are all linked by a common element (in this case, a protein). This type of data integration is fast becoming the future of all databases and provides a far more complete overview of a specific protein.

Each of the modules was approached from the view of the researchers, what they would want to accomplish, which tools they would use and how they would use such a module. This prevented modules from being designed according to developers rather than to assist the researcher. During the design process, researchers were consulted on commonly used tools, the way in which they used the tools and ways in which they wanted data to be presented. Usability was also kept in mind to make the tools easy to use. To facilitate easy use of the system, it was decided to focus on a web-based system, rather than a standalone system. This presents the user with a familiar environment (web browser) and allows for minimal hardware and software installations. While benefiting the user, such a system will also benefit the administrators as they need to install and maintain only one server, instead of a number of computers at various workstations.

FunGIMS supplies a variety of these services but this specific study focuses on protein and protein structure-related services. The Structural module of FunGIMS aims to provide the users with three different types of tools: Explorative, Analysis and Modelling tools. Explorative tools allow the user to explore known protein structures and their features, Analysis provides a selection of general tools to allow the user to analyze a sequence or patterns found in a sequence and Modelling allows the user to build homology models and generate scripts for various molecular dynamics programs. The scripts are intended as a stepping stone to encourage user-driven investigation.

The Analysis section will provide tools such as Prosite (de Castro *et al.*, 2006) and Hidden Markov Model searches against Pfam (Finn *et al.*, 2006). Prosite is a tool used to find motifs in a sequence which may aid in identification of the protein. A motif can be defined as an element or short stretch of amino acids that is linked to a specific functional or structural protein feature such as glycosylation or protein specificity. When referring to a motif that identifies protein specificity, the amino acid sequence of the motif must be unique to that specific activity. Some motifs are very short and inaccurate. A good example of these include glycosylation sites which are often only one or two residues in length and may thus occur at random on a protein sequence. Prosite uses regular expressions to search the motifs against a sequence. A regular expression is a way to match text patterns to strings and find the matches. These text patterns may include wildcards which allows for any specific character to be found at a position as well as specific combinations of characters.

A way to improve the accuracy of motifs is to use Hidden Markov Models (HMM). The Hmmer tool used in the Analysis section is a good example of HMM use. A HMM is a probabilistic model which takes into account the residues before and after the motif as well as the order of the residues in a motif. In the calculation it may incorporate a set amount of residues before or after the current position and this is referred to as the order of the HMM. Thus a 5th order HMM would consider five residues before and after the current position during a calculation as well as the order of the residues in the pattern. This implies that the pattern and position of residues in a protein sequence can be used to identify it or to generate HMM's that can be used to search for other proteins containing the same pattern. Using HMMs a model of a protein family can be built. HMM's are discussed in detail in Bystroff and Krogh, 2008 This allows programs such as Hmmer to accurately identify the family to which a protein belongs. In the Analysis module, Hmmer is used to search a sequence against the Pfam database. Pfam is a database built up of domain HMMs of every known protein family. It uses manually curated alignments of protein families to generate HMMs of the areas that can be used to identify each family. The more members in the family, the more accurate the domain HMM in Pfam.

The Tmhmm (Sonnhammer *et al.*, 1998) and S-tmhmm (Viklund and Elofsson, 2004) tools are also incorporated into the Analysis section. Tmhmm use HMMs to classify whether a protein has membrane crossing α-helices. These are recognized using HMMs based on length and hydrophobicity. A standard transmembrane helix is usually 20 residues long as this is the minimal length needed to cross a membrane while the residues are in a helical conformation. S-tmhmm uses HMMs to identify the orientation of a transmembrane helix in the membrane. It will give each residue a probability of whether it is on the inside in the cytosol or whether it faces the outside of a membrane.

Also included in the Analysis section are tools such as PROCHECK (Laskowski *et al.*, 1993) and the WHAT IF model check (Vriend, 1990). These tools use statistical data derived from the PDB to evaluate various parameters in a protein structure or model. These include parameters such as bond lengths, bond angles, planarity of atoms and packing environments of amino acids. Each of the tools will compare the results from the submitted structure to the statistical values and then judge it as either being within or outside acceptable limits. When analyzing models this is very useful as it can identify areas which were badly modelled.

Most proteins are made up of various secondary structural elements. To identify these elements, the DSSP program (Define Secondary Structures of Proteins) measures all the angles between the atoms in a protein and classify every residue as either being in a loop, β-strand or α-helix.

The Modelling section includes tools related to homology modelling and molecular dynamics simulations. Homology modelling a method whereby a structure of an unknown protein is built based on the structure of a related or homologous protein. Protein structure is much more conserved than protein sequence and this is the basis of homology modelling. Modelling programs usually take at least two parameters, a known structure and an alignment between the sequence for which the model is to be built and the sequence of the known structure. The coordinates for every region that aligns is then copied to the new target structure. Where regions don't align or where gaps or deletions are present, the program will try to build the structure based on statistical averages in combination with forcefields. After a basic model has been built, the program needs to refine the model. There are various steps and methods but the most well know is the satisfaction of spatial restraints. This method will adjust all the interactions between atoms to satisfy known restraints such as bond lengths and bond angles. An extension to this method is the modelling of the amino acid side chains. Because the side chains can rotate and have more rotational degrees of freedom, it is a more complex task to model. One of the approaches is to use a library of observed side chain conformations and model each side chain based on those conformations. This is fairly quick but does not always take the surrounding environment into account and thus some programs optimize the side chain conformation to include environmental conditions. Loop modelling presents another challenge as they are very flexible and usually lack a template. Most programs will either use a library of observed loops to try and model a loop section or, if the loop is short enough, will try *ab initio* modelling of the loop. Because of the loop flexibility, both these approaches have their drawbacks. Loop libraries do not contain all the known conformations of a loop as *ab initio* modelling of loops is in its infancy.

As structure is so conserved, this general approach is valid for the most proteins. General homology modelling theory holds that when there is 30%-50% sequence similarity, the backbone of the protein is correct, when the similarity is between 50-%70%, the side chains are also correct, and anything above 75% similarity will result in side chain specific contacts, or sometimes atoms, to be correct. Anything below 25%-30% is considered to be in the 'twilight zone'. To build models in this range requires a lot of extra knowledge about the protein which cannot be gained from structure and alignment alone.

The field of molecular dynamics encompasses the movement of proteins as simulated by an algorithm. These simulations provide valuable information regarding the interactions between amino acids in a protein. It can also be used as a guide in designing experiments to investigate the importance of amino acids in protein movement and interactions. It must be kept in mind that molecular simulations still have some limitations and it must be used as a tool to facilitate and guide experimental work. In order to to improve the simulations, much better models of the interactions between atoms and residues needs to be built. Tools to generate molecular dynamics scripts are also included in the Modelling section. Molecular dynamics is the application of Newton's Laws of Motion to a set of atoms over time to predict how they will move. With proteins the matter becomes more complex as certain atoms are bound to one another and undergoes short and long range interactions. Various algorithms and programs have been implemented to deal with these elements. The general terms in a molecular dynamics forcefield include energetic terms for the following: bond length, bond angle, dihedral angles, long range interactions, hydrogen bond interactions and Van Der Waals interactions. Each program treats these terms differently and assigns different values to each term based on empirical or calculated data. When starting a simulation, the program will try to perform an energy minimization on the protein. This is a technique whereby the algorithm tries to obtain the minimum energy for a protein by adjusting all the physical factors such as bond length, side chain orientations and atom-atom interactions. The Modelling section provides the user with a choice of programs for dynamics as well as modelling. For dynamics only scripts are provided as running simulations are very resource intensive and are not feasible on a web server. Some simulations may run for weeks at a time and thus take up valuable resources.

All of the tools mentioned will provide the user with extra information about the protein.

These programs will produce data for the user and thus FunGIMS provides data storage and act as an interface to the data. In addition to this, it also provides group-linked user management. This feature was requested by local biologists who wanted to consolidate data storage yet retain individual and group control over the data. Such a system would allow the users to store certain subsets of data on the server and retrieve it for later analysis. It also allows a separation between private and public data, which is important as some individuals may be working on projects that are of commercial value. FunGIMS allows these users to keep their data private, yet incorporate and enrich their data with public data from various sources. FunGIMS also allows for private and public data to be kept apart in that private data cannot be accessed by users who do not have the necessary access rights.

By providing the user with exploration, analysis and modelling tools in one central location, together with allowing for storage of the results, it facilitates knowledge discovery.

1.7. Application to Foot-and-Mouth Disease Virus

Foot-and-Mouth Disease Virus (FMDV) is a highly contagious disease found in cloven hoofed animals and a range of other hosts. Infections can cause large economic losses as well as a decrease in animal productivity. Local researchers at the Agricultural Research Council (ARC) have been working on a vaccine design against FMDV but have encountered numerous problems. Most of the vaccine design work is based on the capsid proteins of the virus as these are the main proteins exposed to the humoral immune system of the animal although the cellular immune system also plays a role. Due to the different serotypes found in FMDV, it is difficult to make a general vaccine. Current vaccine efforts are serotype-specific, sometimes even subtype-specific. Sequence analysis showed that there are a few sequence differences between the capsids of the various serotypes. The capsid plays a vital role in virus stability and entry into the cell and thus any capsid sequence variation might have an effect on virus spreading. Some of the problems during vaccine design were found to be related to structural aspects of the virus capsid proteins and the researchers had no means of using experiments to identify the differences in the structure. Since the researchers had no real experience in dealing with protein structure in a three dimensional environment, they required assistance and advice to use structural bioinformatics to solve urgent problems. Analysis or simulation programs were run, based on the advice given and interpretation of the results on the basis of the protein structure were provided to them. This collaboration is of vital importance as it helps them to direct experiments and interpret the results they see in the laboratory. The results have helped them to understand how variation in the capsid protein sequences affect the structure of the capsid and its effect on virus capsid stability.

The goal of FunGIMS is to provide tools for researchers in this kind of situation, to allow them to do research in an unfamiliar field while minimizing the technical difficulties hindering them. Most of the tools in the Structural module of FunGIMS, were specifically chosen to assist the researchers in conducting the most common structural bioinformatics and analysis on the proteins of the different virus strains. The functionality of the Structural module in FunGIMS was used together with other tools to aid in the investigation of three aspects of FMDV. Each problem additionally illustrates a specific feature/s of FunGIMS and its application to a specific problem. The three problems are:

- Annotation of the FMDV proteome. The FMDV genome is small and codes for fourteen proteins on a precursor polypeptide. The motif-finding tools in FunGIMS were used to find protein motifs in the proteome and compare the distribution of these motifs on 9 serotypes.
- Variation in FMDV 3C protease and 3D RNA polymerase. These two enzymes are important in the replication of FMDV and are usually highly conserved. The homology modelling tools in FunGIMS were used to build models of various SAT serotypes. The variation found in various subtypes of each of the three SAT serotypes was then compared and mapped to the protein structure to locate variation hot spots and to identify potential surface interaction areas.
- FMDV capsid stability and variation analysis. The FMDV capsid is vital to the virus as it protects the virus from the environment and assists in cell entry. It is also the main focus of vaccine design and thus understanding the interaction and differences between the various capsid proteins is highly important. The homology modelling

and molecular dynamics tools in FunGIMS were used to build models of the capsid proteins of various SAT2 subtypes. These models were used to map variation in the capsid and, in conjunction with molecular dynamics simulations, investigate the stability of the serotype capsids at differing pH values.

The three aspects investigated help to show the variety of problems that FunGIMS can be applied to and the way in which it helps to facilitate knowledge discovery in each case. A more detailed introduction about each FMDV topic is given at the start of the relevant chapter.

Problem Statement

Foot-and-Mouth Disease Virus (FMDV) is highly contagious virus infecting cloven-hoofed animals. A few key problems were identified by local researchers, all relating to structural aspects of the virus capsid proteins but they had no structural biology experience. A system called FunGIMS was designed, which attempts to help address these problems specifically in the investigation of FMDV and also to provide other researchers with an introductory environment for structural biology investigations, leading them towards the later use of more advanced tools. FunGIMS is a Functional Genomics and Information Management System. It provides an easy to use, web-based interface to perform a variety of analysis on various different data types. This project focused on providing easy access to structural data as well as intuitive and easy-to-use interfaces to the most commonly used structural bioinformatics tools.

The complexity and setup of structural biology tools have long been a barrier for biologists who want to make use of these tools. Most structural biology tools usually run on a UNIX type operating system. The vast majority of these tools has been validated extensively in literature and by their respective authors. Each program has a different syntax and method of operating, which may be frustrating to the normal biologist. By providing access to these tools via the web and by using a simple form-type input, most of the syntax and related problems are dealt with. An ideal solution would be to provide most of the tools and data via a web interface, which is a familiar environment for most users and which will help and guide users to perform independent structural biology work. Although the system makes it easier for the biologist to use the tools, the onus is still on the user to understand the function of each tool and how to interpret the results. The responsibility of tool setup and installation will be that of an experienced person such as a system administrator thereby allowing the biologist to focus on science. The system was also designed to facilitate the addition of new tools.

The integration and ease of use of the Structural module in FunGIMS is illustrated in a series of investigations performed on FMDV. The first problem is the way in which variation differs between FMDV serotypes with regard to their full proteomes. Insights into variation can help in identifying areas prone to accumulating variation. The second problem relates the variation found in two of the most conserved proteins in FMDV, 3C protease and 3D RNA polymerase. Variation hotspots in these proteins help to identify areas where interactions with other proteins occur and can help to pinpoint areas vital to enzymatic function. The third problem involves the stability of the FMDV capsid proteins under different pH levels and the way in which variability in the VP1-3 proteins affects stability. Stability of the capsid is vital for virus distribution as well as infection.

Although the tools were used on three FMDV cases, they are generically applicable to most proteins and problems related to protein structure. An integrated system such as FunGIMS, will provide access to a variety of tools as well as allow easy application of these tools to various problems related to protein structure.

Specific Aims

The aim of this project is the development of a Structural module in the FunGIMS system and its application specific problems in FMDV. The system allows a user to perform protein structural analysis in an environment with a minimal need for local client-side computing resources. The aims of this project is balanced between providing useful interfaces and tools for the user and programming a robust, extensible environment for protein analysis which can be applied to FMDV.

In Chapter 2 the development and design methodology of FunGIMS and the Structural module will be discussed. The aim was to design a system that is easy to use, easily expandable and allows the user to store and analyze data. The problem of tool incorporation into the module will also be discussed. Tools were incorporated into the system in a modular manner.

Chapters 3-5 each deals with an investigation of a specific aspect of FMDV, illustrating the role that FunGIMS was able to play in a specific problem/area of interest identified by local researchers in the study of Foot-and-Mouth Disease Virus (FMDV).

Chapter 3 describes the use of protein sequence-based tools in the Structural module of FunGIMS to annotate and identify similar patterns and functions in the FMDV proteome. This was applied to various FMDV serotypes to characterize the different proteomes and the functional relationship between them.

Chapter 4 uses homology modelling to characterize the variation seen in the highly conserved 3C protease and 3D RNA dependant RNA polymerase proteins of FMDV. The aim is to identify hotspots in the enzymes which are more or less prone to variation and which may be linked to functional and structural differences between the South African Territories (SAT) FMDV serotypes.

Chapter 5 investigates the functional and structural effect of mutations in the capsid proteins of FMDV. Capsid proteins are used in FMDV vaccine design and thus a thorough understanding of the changes found in these proteins is necessary. The homology modelling and molecular dynamics functionality of the Structural module of FunGIMS was used to investigate the effect of the various mutations on virus capsid and pH stability.

Chapter 2

FunGIMS Design and Implementation

2.1. Overview

The FunGIMS (Functional Genomics Information Management System) is a web-based system designed to integrate most of the major data types that a researcher might encounter in a modern functional genomics experiment. These data types include sequence data, protein structure data, microarray data, small molecule data and literature data. In addition, it also provides online access to some of the more commonly used tools in each of the data type subsections. This allows the user access to data and analysis tools in one, centralized location as well as providing storage for the data generated by the analysis tools in FunGIMS.

The following sections will discuss the technologies used in FunGIMS as well as the design process and the data model used.

2.2. FunGIMS Design and Technologies

During the design phase of FunGIMS, every effort was made to find the most appropriate technologies for each section of the project. Every section involved exhaustive investigations and testing of the options currently provided by software manufacturers. Important decisions such as a specific programming language, were only made after extensive research into the support provided and the ability to allow the programmer to do a specific job.

2.2.1. Technologies

For the success of a large project such as FunGIMS, various technologies are needed to work in unison to produce the final outcome. Each of these technologies will be discussed shortly in the following few sections. For the programming languages, Java and Python were investigated extensively as well as the availability of software packages which allow for interaction with databases. Different language-dependant web frameworks were also investigated. These included JBoss, TurboGears, Java Struts and custom Python scripts on top of a CherryPy server or Apache web server. The ability of a language to interact with databases and facilitate easy data persistence led to investigations into Java Beans, Hibernate, SQLObject and SQLAlchemy. Architectures such as the Model-View-Controller and Server-Client designs were investigated to find the most suitable option for delivering data and interactivity to users. In the software world it is important to choose your technologies wisely due to the rapid rate of new developments and the decline of once-popular software. The following sections will discuss the choices made for each of the technology aspects of the project.

2.2.1.1. Python

The programming language chosen for this project was Python (http://python.org). Python has been developed by Guido van Rossum since 1991 and is a mature and stable development language. This maturity has led to it being used by the biggest search engine company at the moment, Google (http://www.google.com), on a wide range of services. The widespread use of Python and the ease with which it is learned has resulted in an extremely wide code base that caters for a vast amount of functionalities. In the last few years Python was used in developing games such as Civilization IV (Firaxis Games, http://www.2kgames.com/civ4/home.htm), high performance scientific computing packages (NumPy, http://numpy.scipy.org; SciPy, http://www.scipy.org), web development platforms (TurboGears, http://www.turbogears.org; Pylons, http://pylonshq.com), movie animations (Blender3D, http://www.blender.org) and being supported in commercial scientific packages such as Discovery Studio II (Accelrys Inc.). Python was chosen due to its stability, ease-of-use and multitude of packages.

Python is also widely used in Bioinformatics due to its ease of use. Examples over and above scripting include: PySCeS (Olivier *et al.*, 2005) that is used very successfully in modelling the kinetics and substrate flow through enzymatic pathways (Uys *et al.*, 2006), PyMol (http://pymol.sourceforge.net) that is a very successful open source python-based 3D protein structure viewer, and PyQuante (http://pyquante.sourceforge.net/) when doing quantum mechanics.

2.2.1.2. Web Development Framework

For FunGIMS it was decided to use the TurboGears web development platform. TurboGears is mature, well developed and written in Python and allows for development of projects using all the possibilities provided by the Python language. Development in TurboGears takes some time to master but should a person have previous Python programming skills, the process is far quicker. TurboGears is based on the Model-View-Controller architecture (see section 2.2.2) and uses various other packages to perform the different functions. The use of Python and the MVC architecture in TurboGears made it the perfect choice for FunGIMS, which uses the same technologies and thus allows for easy integration. Figure 2.1 shows a diagrammatic layout of the functioning of TurboGears.

2.2.1.3. Object-Relational Mapper

Often a time consuming step in programming is constructing code to represent the data queried from a database. To overcome this problem, Object-Relational Mapping (ORM) was developed. This is a method whereby a query to a relational database can be represented in an object-orientated way in the code. The programmer defines all the tables in the database using code and also defines classes for working with the tables. The ORM then uses this information to transparently connect to the database, and provide the programmer with access to the data using the predefined classes. The ORM also provides some methods, native to the database, as normal methods owned by the classes. Thus the programmer does not have to learn the syntax needed to manage the database natively, only the concepts need to be known. These methods allow the programmer to continue programming in the same style, without the need to write his own mapper be-

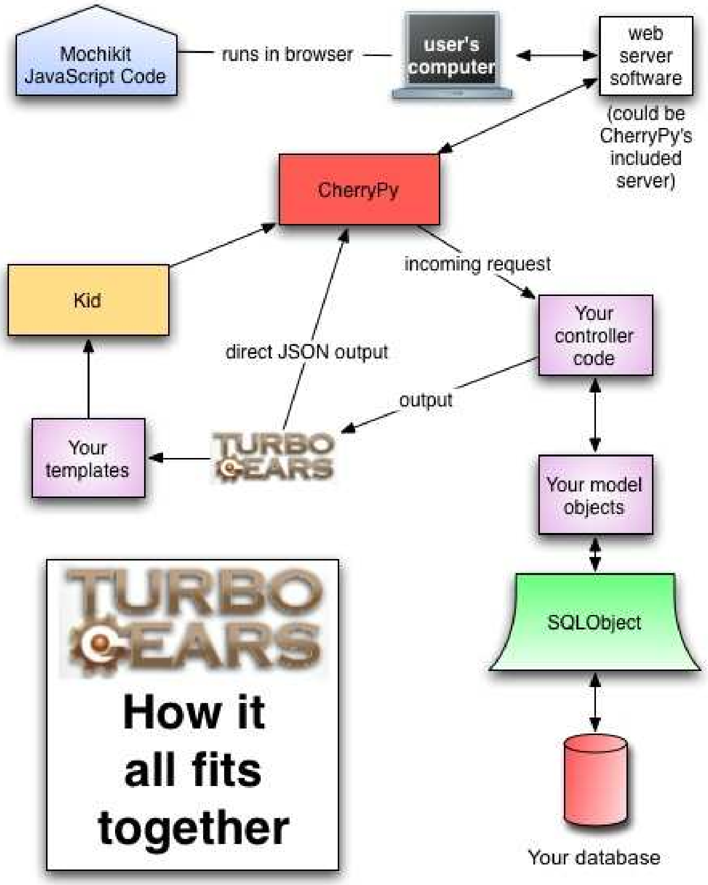


Figure 2.1: A schematic representation of how the different parts work together in TurboGears (http://docs.turbogears.org/1.0/GettingStarted/BigPicture). The user makes a request for data in the browser. This request gets directed by the controller to the model. The ORM then connects to the database, retrieves the data and returns it to the controller. The controller then provides the data to the appropriate template, which is served up as HTML code to the user's browser.

tween the database and the program. For FunGIMS it was decided to use SQLAlchemy (http://www.sqlalchemy.org). SQLAlchemy is supported in TurboGears and uses the `model.py` file to define the database, link the tables in the database to code classes and implement data class specific methods. SQLAlchemy was chosen in preference to SQLObject as it provided more advanced functions such as polymorphic joins and class creation via introspection of the database. At the time of writing, SQLAlchemy was also slated to become the default ORM for the TurboGears project. It was decided to use MySQL (http://mysql.org) as the relational database for FunGIMS. This was chosen the preferred choice rather than PostgreSQL as SQLAlchemy provided slightly better support for MySQL than for PostgreSQL when the project was started. Most of the developers also had more exposure to MySQL than PostgreSQL. MySQL provides a way to store vast amounts of data, while providing extremely fast search access to the data. All the data are stored in rows in user-defined tables, and a user can search over all fields in the tables. This provides a very powerful way of storing and querying data.

2.2.1.4. Version Control

In a project of this scope, version control is essential. Version control provides a way for the system to be backed up in increments as each part of the system changes. A developer can check out a certain part of code, work on it and then check it back into the system. The system then checks whether there was any conflict in the code, and store the changes made to the code. It also tracks the changes each developer makes as well as any changes to files. Furthermore, it prevents changes made by the different developers on the same piece of code to be checked in prior to validation thereof. An essential feature is the ability to rollback changes made to the system. It was decided to use Subversion (http://subversion.tigris.org) for this project rather than Concurrent Version System (CVS).

2.2.1.5. Templating Language

Web browsers display pages written in HyperText Markup Language (HTML). HTML uses a static code to represent items on a web page. To overcome the static element of HTML, programmers developed templating languages. These languages allow a programmer to generate static HTML content based on decisions made by the algorithm or program or even based on user input. The Kid templating system (http://www.kid.org) was used for FunGIMS. Kid is a templating system that is based on eXtensible Markup Language (XML), of which HTML is a derivative, and allows for the incorporation of Python code in the template. KID will take the XML template and the data provided by the controller, combine it and render it into HTML that is then sent to the web server. The user will then see the page as normal HTML in his browser.

2.2.2. Development and Design

The design of a large system such as FunGIMS is a complex task and requires careful development and planning to prevent a cluttered and complex code base. This is especially important when there are multiple programmers working on a project and coordination between them is vital. The first step in planning such a project is to identify the potential users and analyze their requirements. These requirements must then be implemented in a logical way to benefit the user. The programming task must also be divided amongst the programmers to speed up development.

As a first step, the use of object-orientated programming was implemented. This results in code blocks that can be reused throughout the project and facilitates faster development. A Model-View-Controller architecture was also followed (Fig. 2.2) for the software design of FunGIMS. This architecture separates a project into three different sections on the basis of the function of each section:

- Model - this contains all the code necessary for the storage of results and managing the database back end as well as handling queries to the database.
- View - this section contains all the code used in displaying results/output from the system. It contains mostly templates and usually contains very little logic code.
- Controller - this is the section in which all the functionality and the majority of the code resides. All the decision making processes in the system are stored here, and it controls input and output to the model and view. It “controls” the entire system and directs traffic and requests to the appropriate subcontrollers.

Following the MVC architecture, the project was divided into three sections namely `model.py`, `controller.py` and a folder for all the templates entitled `templates`. These are each discussed in more detail in sections 2.2.2.1, 2.2.2.2 and 2.2.2.3. In Figure 2.3 the overall design and implementation of the MVC architecture in FunGIMS is shown. This high level overview provides a clear depiction of how each part of FunGIMS fits together.

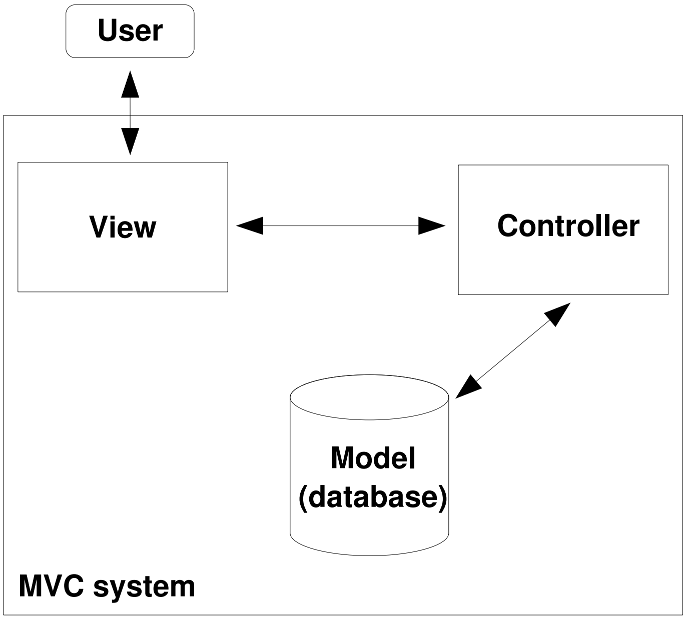


Figure 2.2: The Model-View-Controller (MVC) architecture. The Model contains the data model needed by the ORM to interact with the database. The View contain all the templates needed to display the data and the Controller controls and handles all communication between the Model and the View. The controller also calls any external programs that are needed.

During the development process, the spiral development methodology was followed. This methodology is based upon small improvements and step-wise additions of features, followed by rapid deployment and testing of the new features. This cycle is repeated as each new feature or functionality is added. The advantage of this methodology is that errors in the code and feedback from the users can be corrected and implemented quickly, which results in less effort compared to corrected errors in a project where the release and testing cycle is longer. Most of the modules were developed in conjunction with user input. Thus at each stage in the development, the user was consulted. The user was asked which functionalities he wanted, where after the programmer would implement it and the user would test it and give feedback.

During the design of FunGIMS, the usability and users of the system were always kept in mind. This forced the coding process, and the code itself, to be far more efficient and intelligent in the manner in which the different applications and functionalities were implemented. A good example of this is the System ID (sid) that is assigned to every entry of a data type. The sid should identify the specific record in such a way as to

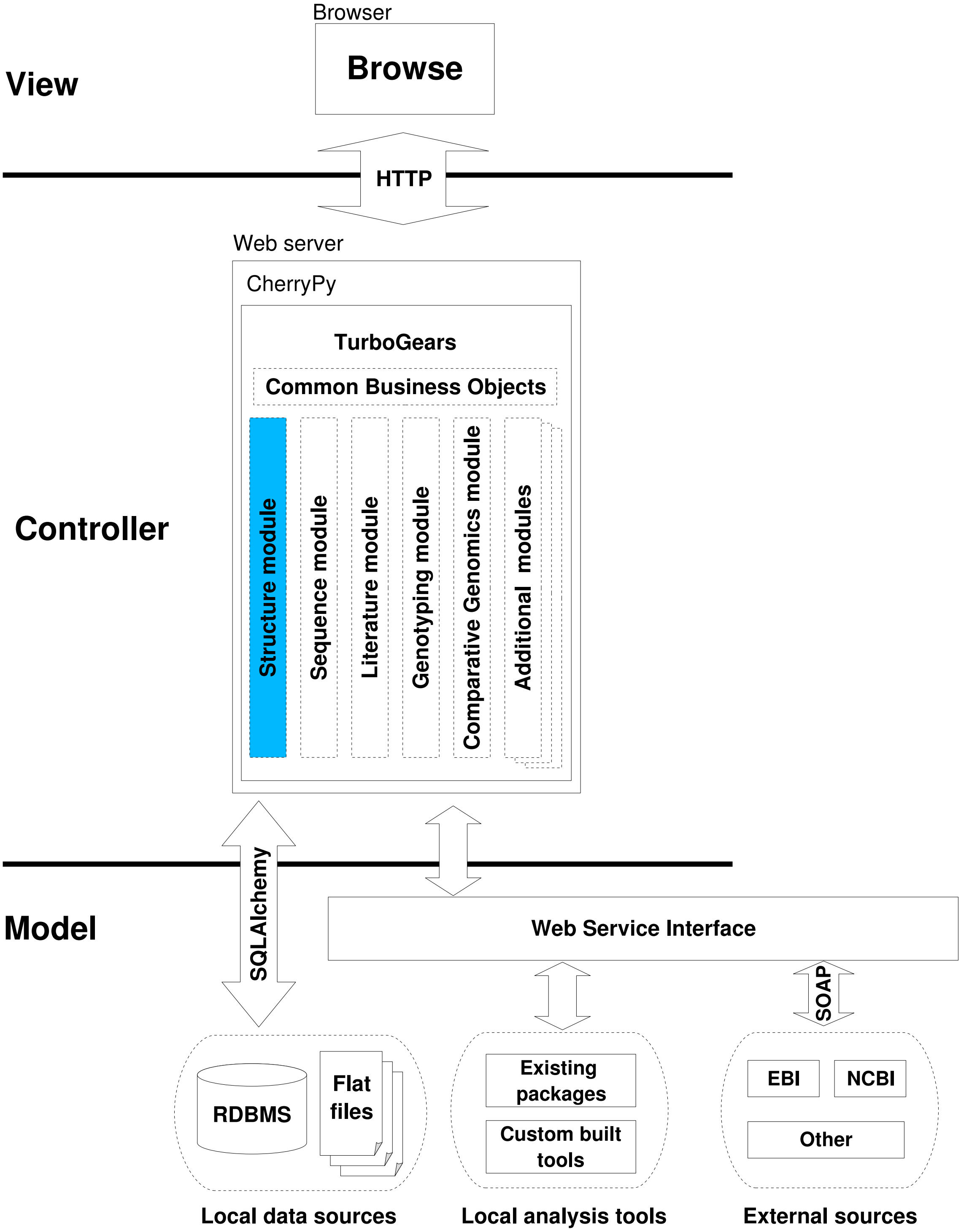


Figure 2.3: The overall design of the FunGIMS system. The design follows the Model-View-Controller architecture and uses TurboGears as the web development environment. Various other modules such SQLAlchemy provide interfaces and methods to access data and call external programs. The View provides the interface the user sees when using the system. The Controller controls and directs all requests within the system and the Model stores all the data.

facilitate easy use during coding, as well as for easy understanding thereof by the user. With FunGIMS the number of records of different data types was huge. To assist users as well as facilitate easier coding, it was decided to use a common sid format. The format, <*data type:id*>, consists of a data type identifier, followed by a :, followed by a unique number for user-generated data or the id assigned by the specific public database e.g. PDB file 1eye would have the sid: pdb:1eye. This identifies the record as a protein coordinate file and uses the more well known public database id as well. The PDB is a good example of the efficient use of a system-wide, unique id. The unique number is generated by taking the system time, in seconds since 1 January 1970, and multiplying it by a factor of ten million to get an integer number.

At the time of writing, FunGIMS catered for the following data type identifiers:

- seq - user generated/uploaded sequence
- gi - sequence from GenBank public database
- sp - sequence from SwissProt public database
- pri - user generated primer sequence
- pdb - protein structure file from the PDB
- pmid - article from the PubMed public database
- file - user uploaded generic file
- chebi - small molecule from the ChEBI database
- note - user generated note
- blast - BLAST results file
- go - Gene Ontology term
- taxon - NCBI taxon term
- trace - DNA sequence chromatogram files

These data type identifiers makes it easy for the user to see which entry they are currently working on or which entry's results they are looking at. To make the development process faster, each programmer was given responsibility for a module on FunGIMS, while core modules were developed together as they were needed.

Coding was not the main area where ease of use was of primary importance. Ease of use is the most important in the user interface. Throughout FunGIMS the interfaces were designed to be clean, intuitive and easy to use. This implies that pages do not show unnecessary information to the user. Future releases may have the option to display extra information contained in the relevant files. Each page is designed to show only the information the user needs at that moment. In the case of analysis tools, the user is asked for only the necessary information before the analysis is run.

2.2.2.1. The View

The views in FunGIMS are responsible for interacting with the user and presenting data to him. Although the views only present data, in some instances decisions on display items can only be made once the data is rendered or to alleviate more extensive coding of templates. Each view is written in the Kid templating language. Each module in FunGIMS has its own set of views and a shared subset deals with general, administrative displays such as headers, new user registration and shared items. The view files are stored in a separate directory (`templates`) and use the .kid extension. The views are compiled to Python code as needed using just-in-time (JIT) compilation.

The view also makes use of JavaScript for some visual effects and for managing the addition and deletion of notes through JSON, an AJAX library (Asynchronous JavaScript and XML) used in TurboGears to connect Python functions and JavaScript. The view also allows the inclusion of applets such as Jmol, which is used in the Structural module. These applets allow for extra functionality in the browser.

2.2.2.2. The Controller

The controller is that part of FunGIMS that regulates all the decisions regarding flow control. The controller decides what data must be retrieved, what data must be sent to the view and which commands to execute with regard to the given variables. In essence, the controller controls everything in the application. All code that make a decision resides in the controllers. In FunGIMS the responsibility of the controller has been split to facilitate collaborative coding as well as to decrease the amount of code residing in one main con-

Table 2.1: The technical specifications of FunGIMS.

Feature	
Programming Language	Python 2.4
Development Framework	TurboGears 1.0.2
Code Revision Control	Subversion 1.2.3
HTML Templating	Kid 0.9.6
Object Relational Mapping	SQLAlchemy 1.3.9
Documentation	Epydoc 3.0beta1
Back end Database	MySQL 5.0

troller. The main controller (`controller.py`) in FunGIMS decides which sub-controller (located either in the `view_controllers` or `search_controllers` folders) receives the data and which sub-controller is responsible for executing the user's commands.

In FunGIMS the following tasks are under the direct responsibility/control of the **main** controller:

- Deciding which view to present to the user
- Managing the search functionality
- Managing user access (logging in/out) and security
- Making decisions on which analysis interface to send data to
- Upload/download of files
- Generic saving of results produced by analysis methods
- Web services

The technical specifications of FunGIMS are given in Table 2.1. The choice of language (2.2.1.1), development platform (2.2.1.2) and other decisions have been discussed in the relevant sections.

2.2.2.3. The Model

The model forms the basis of all the interactions between the controller and the database in the MVC architecture. All the table definitions, table-class mappings and class-specific methods are defined in the `model.py` file. This file is used by the ORM to interact with the database and return the relevant data to the controller. The details of the data model will be discussed in section 2.4.1. There are a few main model-related methods that are used across FunGIMS. These include retrieving data for a specific entry while considering security and access restrictions on the entry, deleting privately owned data and generating new, unique identifiers for data inserted into the system.

2.3. FunGIMS Core Functionalities

FunGIMS contains a few core functionalities that are used across the board in all the different modules. These include managing users and groups, new registrations and searching of data.

2.3.1. User and Group Management

Common practice in laboratories is to divide people into work-related groups. This concept was also used in FunGIMS to manage access to data. When starting a TurboGears project, it provides you with default identity handlers. These are divided into users and groups. Each user can belong to one or multiple groups. For FunGIMS this definition was extended so that groups can also belong to other groups e.g. the different groups in an academic department. An example would be a supervisor who wants to share data with her students as well as between the students, but also wants her own private group. Under the FunGIMS identity scheme this would mean that the supervisor belongs to two groups, her own private group and the student group. This would allow the students to share data but also allow the supervisor to have private data. It is basically a concept of group of groups. Although this complicates the identity management, the advantages thereof are far more than the extra effort required to program it.

In FunGIMS each data entry belongs to either a specific user or group or, in the case of publicly available data, to the “world” group. The “world” group is accessible to everyone and all users can view and use entries belonging to this group. When data belongs to a certain group, all the users who are members of that group may access, view and use the data. This hierarchical implementation of access restrictions allows for the separation of visible data to each group. A user may also decide to browse and analyse data anonymously. This will allow him to see all public data and do analysis, but not save any results, or add notes to any entries.

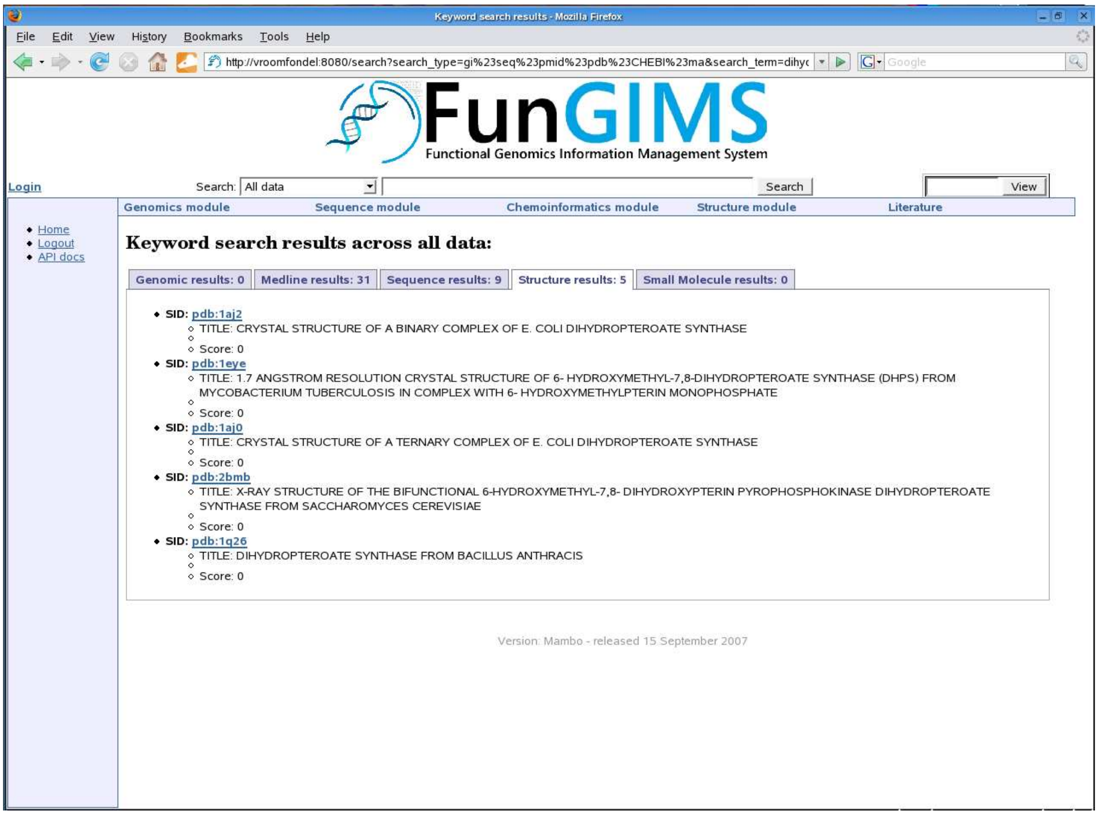
To manage users, a registration section was included. This enables the user to add new users, add users to groups and to create groups. Some restrictions are also implemented, which gives only certain users the right to add or delete users.

2.3.2. Result Management

When users generate results in FunGIMS, they are presented with the option of either storing the results in the FunGIMS database or viewing them without saving. This functionality allows users to use the FunGIMS database as a data repository. User-generated results are stored as uploaded files in the database. When the user wants to save results, they are presented with an option of selecting to which group the results will belong. The group listing includes all the groups to which the user belongs . This allows the user to share generated results with other members of the group. These results are included in any future searches that might be done against the database. If a user is browsing and analyzing data while not logged in, results cannot be saved.

2.3.3. Searching of Data and Results

FunGIMS contains a large amount of data and the best way to access a specific piece of data is to search for it. FunGIMS provides a search facility across all the data and results saved by the user. This allows the user to search for entries by means of a keyword or phrase, or simply access stored results. A user can select to search across all the data types with a keyword or a specific identifier can be entered e.g. search for “dihydropteroate synthase” or search for PDB id “1eye”. The search is implemented on two levels. The first level is a case insensitive text search across all the fields in `Identifiable` and `Description`. The results from this search are then filtered in the second level of the search, to exclude entries that the user may not see. Users can search a keyword or sid against a specific data type or across all data types. At the time of writing, FunGIMS provided searches across protein structures, sequences, literature and small molecule data sets. A keyword search across all data types will produce a page


Keyword search results - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://vroomfondel:8080/search?search_type=gi%23seq%23pmid%23pdb%23CHEBI%23ma&search_term=dihyc
Google
FunGIMS
Functional Genomics Information Management System
Login
Search: All data
Search
View
Genomics module
Sequence module
Chemoinformatics module
Structure module
Literature
Home
Logout
API docs
Genomic results: 0
Medline results: 31
Sequence results: 9
Structure results: 5
Small Molecule results: 0

Keyword search results across all data:

- **SID:** pdb:1aj2
  - TITLE: CRYSTAL STRUCTURE OF A BINARY COMPLEX OF E. COLI DIHYDROPTEROATE SYNTHASE
  -
  - Score: 0
- **SID:** pdb:1eye
  - TITLE: 1.7 ANGSTROM RESOLUTION CRYSTAL STRUCTURE OF 6- HYDROXYMETHYL-7,8-DIHYDROPTEROATE SYNTHASE (DHPS) FROM MYCOBACTERIUM TUBERCULOSIS IN COMPLEX WITH 6- HYDROXYMETHYLPTERIN MONOPHOSPHATE
  -
  - Score: 0
- **SID:** pdb:1aj0
  - TITLE: CRYSTAL STRUCTURE OF A TERNARY COMPLEX OF E. COLI DIHYDROPTEROATE SYNTHASE
  -
  - Score: 0
- **SID:** pdb:2bmb
  - TITLE: X-RAY STRUCTURE OF THE BIFUNCTIONAL 6-HYDROXYMETHYL-7,8- DIHYDROXYPTERIN PYROPHOSPHOKINASE DIHYDROPTEROATE SYNTHASE FROM SACCHAROMYCES CEREVISIAE
  -
  - Score: 0
- **SID:** pdb:1q26
  - TITLE: DIHYDROPTEROATE SYNTHASE FROM BACILLUS ANTHRACIS
  -
  - Score: 0

Version: Mambo - released 15 September 2007

Figure 2.4: The result of a search for “dihydropteroate synthase”. The results are ordered according to data type.

with results sorted according to the section they belong to e.g. sequences in the Sequence section and any structure hits in the Structure section. Should a user search for a specific identifier and it is found to be unique, the user will automatically be redirected to a view of the requested entry. Access restrictions are implemented on the searches and thus a user will not see any matches in restricted data. Figure 2.4 shows the results of a search for the keywords “dihydropteroate synthase”.

2.4. FunGIMS Data Model

2.4.1. The Data Model

FunGIMS was designed to use one database that contains all the data for each data type in separate tables. In order to incorporate the large amount of data and relationships in FunGIMS, an extensive data model had to be developed. The Functional Genomics

Experiment (FuGE) data model was used as a starting point (Jones *et al.*, 2007, Jones *et al.*, 2006) as discussed in Chapter 1. The FunGIMS data model was extended by inheriting from the `Identifiable` class in FuGE. This allowed for features in FuGE such as `Security`, `Description` and `Audit` to be accommodated in FunGIMS. `Security` implements various features related to the FuGE data model with regard to ownership of the record. `Audit` tracks changes made to a record and `Description` provides a way to add free text descriptions of the record. `Identifiable` consists of a sid, data typename, user id, group id and description id fields. These fields link an `Identifiable` entry to a user, a group, a specific description (which is linked to the `Description` class) and a specific data type. The data typename field is used when constructing the polymorphic joins for a specific module. When a new file or data entry is created in `Identifiable`, the user must also supply the fields required for `Description`. `Description` implements fields for id, description text, keywords and synonyms. When searching the database using a keyword, it is searched against `Description`.

The core data model for FunGIMS extended the FuGE data model by including additional classes to FunGIMS, all of which all inherited from `Identifiable`. These classes include `Note`, `File` and `Relationship`. `Note` is a free text field that allows a user to add free text notes to an entry. More than one `Note` may be associated with a unique `Identifiable` entry. `File` is a class that caters for any files uploaded by the user such as protein models, documents or sequences. One `File` object is linked to one `Identifiable` object. `Relationship` is a class used to link two `Identifiable` entries. This relationship is either user generated or automatically generated from the parsed data. Each specific module extends the FunGIMS data model further and by inheriting from the `Identifiable` class, allows a consistent data model to be maintained. FunGIMS currently implements the following main data type classes: `Structure`, `Sequence`, `MedlineReference` and `Compound`. The specific data model used for the Structural module will be discussed in section 2.5.2. The information in `Identifiable` was also used by SQLAlchemy to create groups of tables in the data model that contains only a certain data type using polymorphic identity joins (creating one object by joining different subclasses from the database).

The TurboGears user tracking/validation data model was used to allow the login of users and to maintain session ids during usage. TurboGears employs a set of tables for users and groups and allows users to belong to more than one group. When a user logs in, they are validated against this data model. When retrieving data belonging to a certain group, the group table is checked to assess whether a user may see the data. A unique session id is generated every time a user logs in and this allows the user to remain logged in to the system for a set amount of time (default is 20 minutes).

2.5. Structural Module

2.5.1. Overview

The Structural module caters for all protein structure data. It allows the user to investigate the protein structures, to conduct analysis on the protein sequences and structure and to generate simulation scripts for proteins. The design of the Structural module was based on the MVC design as shown and used in the rest of FunGIMS. This allows for an extensible and easily upgradable system and further allows for a maintainable code base.

The vast majority of the data in the Structural module is parsed from the MSD discussed in Chapter 1. Most protein structure data is represented in a standard column-based format known as the PDB format (http://www.pdb.org/docs.html). This text format provides structural and administrative information about the protein as well as the Cartesian coordinates of every atom in the protein. Figure 2.5 shows the column layout and an example of the latest PDB file format.

2.5.2. Data Model

The main data model used for the Structural module is based on the MSD (Boutselakis *et al.*, 2003) from the EBI at Cambridge. The MSD provides a very extensive data model to deal with protein structure data. All the data are parsed from PDB and are also linked to primary sequence providers such as GenBank.

```
 12345678901234567890123456789012345678901234567890123456789012345678901234567890
 ...
 ATOM     66 N    VAL A  14      22.866   0.219  42.591  1.00 20.77 N
 ATOM     67 CA   VAL A  14      21.639  -0.157  43.253  1.00 26.59 C
 ATOM     68 C    VAL A  14      20.898   1.039  43.832  1.00 43.97 C
 ATOM     69 O    VAL A  14      19.894   0.894  44.535  1.00 44.07 O
 ATOM     70 CB   VAL A  14      21.834  -1.310  44.228  1.00 29.30 C
 ATOM     71 CG1  VAL A  14      22.197  -2.582  43.471  1.00 28.10 C
 ATOM     72 CG2  VAL A  14      23.022  -0.961  45.095  1.00 36.14 C
 ...
```

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	“ATOM ”	Record name
7 -11	Integer	serial	Atom serial number
13-16	Atom	name	Atom name
17	Character	altLoc	Alternate location indicator
18-20	Residue name	resName	Residue name
22	Character	chainID	Chain identifier
23-26	Integer	resSeq	Residue sequence number
27	AChar	iCode	Code for insertion of residues
31-38	Real(8.3)	x	Orthogonal coordinates for X in Angstroms
39-46	Real(8.3)	y	Orthogonal coordinates for Y in Angstroms
47-54	Real(8.3)	z	Orthogonal coordinates for Z in Angstroms
55-60	Real(6.2)	occupancy	Occupancy
61-66	Real(6.2)	tempFactor	Temperature factor
77-78	LString(2)	element	Element symbol, right-justified
79-80	LString(2)	charge	Charge on the atom

Figure 2.5: Top: A protein structure file example (Valine residue 14 from 1eye.pdb). Bottom: the PDB file format specification for ATOM entries.

The MSD data model tries to provide a logical view of protein structure. It is organized into one main entity (`Structure`) that consists of 6 sub-entities (`Active Sites`, `Secondary Structure`, `External Database Links`, `Header`, `Taxonomy` and `Ligands`). Each of these sub-entities are divided into logical groups e.g. `Header` is made up of tables containing information on authors, keywords, X-ray data, etc. In this fashion each sub-entity contains different levels of information. What makes MSD unique and different from the PDB is that for every different feature in MSD, detailed data are available e.g. for every protein atom, the binding order, predicted atom valence, atom type, residue it

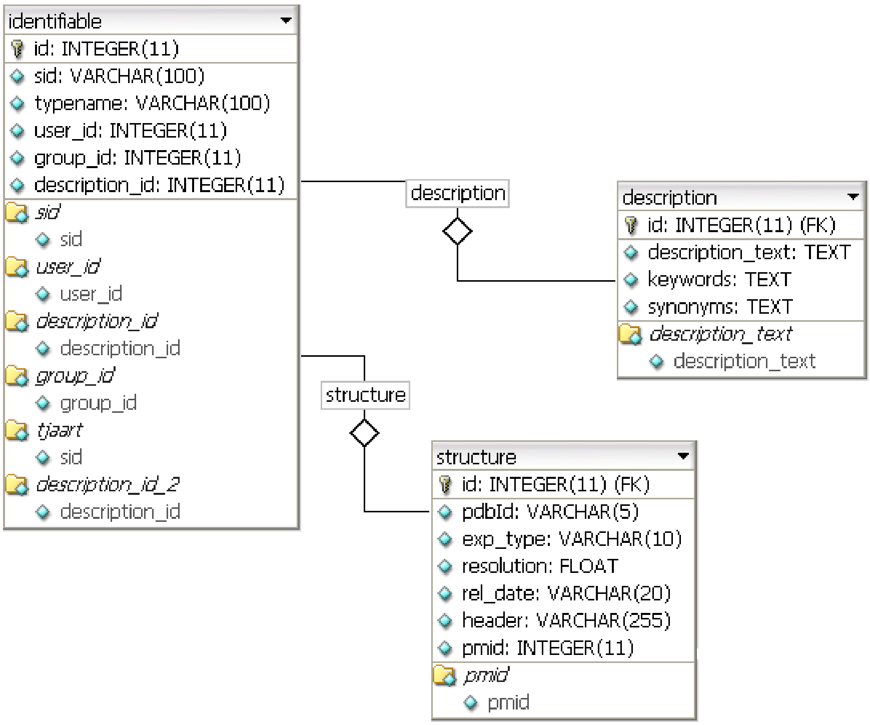


Figure 2.6: The relationship between the `Structure` object and the FuGE data model. `Identifiable` is the main data object in FuGE. `Description` provides some additional data about `Identifiable`. The `Structure` object inherits from `Identifiable` and thus also has `Description` data.

belongs to, other atoms it makes contact with, etc. This makes it one of the most complete structure databases currently available. A complete user-friendly web accessible front end to MSD has been written and is accessible at the EBI's website.

The MSD data model (figure 1.2) was extensively modified before being incorporated into FunGIMS. The Structural module data model consists of the following classes: `Residue`, `Helix`, `Sheet`, `Strand`, `Turn`, `SecondarySummary`, `Tstruc`, `Chain`, `PfamInt`, `ScopInt`, `Go`, `Ec`,`CathInt`, `SwissprotInt` and `Interpro`. All the classes inherit from `Structure` either directly or indirectly from another class. The data extracted and stored from MSD are PDB entry information (`Structure`), protein secondary structure (`SecondarySummary`) including α-helices (`Helix`), β-strands (`Strand`), β-sheets (`Sheet`) and β-turns (`Turn`), protein fold (`Tstruc`) information from CATH (`CathInt`) and SCOP (`ScopInt`), protein classification information from GO (`Go`), Interpro (`Interpro`), Pfam (`PfamInt`) and Swissprot (`SwissprotInt`) as well as EC numbers (`Ec`). Information such as the energy types of each atom and atom types were not extracted, as the Structural module only caters for a higher level of protein structure. A second set of scripts was then run on the MSD data to extract basic relationships between data types such as linking the Pubmed id with a protein entry and these were stored in the `Relationship` class. Stored relationships are between the protein, Swissprot and GO numbers as well as between the protein and Pubmed. All these generated links were also added to the FunGIMS database. Section 2.5.2.1 discusses other data sources. Most data relating to the detail such as atoms, residue planarity and energy types were omitted. This was due to the fact that the Structural module provides a basic introduction to a structure. Its main purpose is for exploratory analysis and investigation.

The FunGIMS structure data model was constructed to closely represent the actual structure levels in a protein in a top down fashion. This ensures that a protein model can be browsed by starting with the assembly, followed by the local fold, the chain specific secondary structure and finally by residue data (Figs. 2.6, 2.8 and 2.7).

2.5.2.1. Data Sources

The majority of the data in the Structure module, and also FunGIMS, are derived and parsed from public databases such as the PDB, GenBank and SwissProt. In the case of the Structure module, Python scripts were used to parse the flat file format of MSD and to add the data to the FunGIMS database.

FunGIMS also caters for user-generated data. In the Structure module specifically, user-generated data makes up a very small portion of the stored data. This is due to the fact that a model that a user generates will not be parsed and stored in the database as there is no experimental validation of the structure. All generated modelling scripts and models will be stored as files belonging to a specific user and group should the user choose to save the files.

2.5.3. Functionalities

The Structural module has various different functionalities. A user can investigate a protein structure and retrieve information about structural elements, perform motif searches and structural analysis on a protein sequence, generate homology models or generate scripts for modelling and molecular dynamics. Each of these features will be discussed separately. For the first release of the Structural module it was decided to include tools that are often used by biologists and some tools that are less used but equally valuable and that can provide new insights into their work. The design of FunGIMS and the Structural module allows for the easy addition of new tools by programmers.

The browser-based molecular viewer known as Jmol (http://jmol.sourceforge.net) is one of the features that makes the Structural module very useful. Jmol is a Java-based three dimensional molecular view that can run inside a browser as a Java applet. It uses software to render the proteins and thus does not need expensive hardware such as graphics cards. Jmol was specifically written to allow protein structure files to be displayed and manipulated inside browsers. The user can rotate the protein, zoom in, select different representations of the protein, and various other miscellaneous functions. Jmol can also be run as a standalone Java application, which allows users to download the protein files and work with them in a familiar environment.

In the Structural module, Python is used to parse the data such as residue start and end numbers in a turn or helix, and then use this data to generate buttons which controls various Jmol representations.

2.5.3.1. Structural Data Representation

The Structural module includes all structural data such as primary structure, secondary structure, tertiary structure and atomic coordinates. The first view a user would see when querying a protein is the primary sequence data. This includes the sequence of the protein, the name of the protein and other data parsed from the header such as resolution (Fig. 2.9). The primary view also shows any notes added to the specific protein as well as an atom representation (based on the coordinates in the crystallized structure) of the protein loaded into Jmol.

From the primary view the user can navigate to the secondary and tertiary structure views. The main secondary view contains a summary of all the secondary structure features found in each chain in the protein and provides links to a more detailed view of each feature. When a specific chain is selected, it takes the user to a summary of the secondary structural features for that specific chain (Fig. 2.10). This includes data on α-helices, β-strands, sheets, turns and other chain features.

A user can also see a summary of all the strands in a specific protein chain by clicking on the strand link in the secondary structure summary (Fig. 2.11). This will provide a page with a summary of the strands found in the protein chain together with their position, length and sheet id as classified in the MSD. A cartoon representation is presented in Jmol and buttons are provided to select the specific strands. These buttons are not always 100% accurate as Jmol interprets residue numbers differently than those found in the MSD due to missing residues in the protein crystal structure. This is due to the fact that sometimes part of the protein does not crystallize or only a truncated peptide was used. Thus, those residues do not get used when assigning numbers to the residues found in the crystal structure. A user can also select the sheet link and see the number of sheets in a protein structure.

A user can also access data about the α-helices in the protein chain (Fig. 2.12) from the secondary structure summary. This view gives an overview of the number of helices as well as their length, start and end residue numbers. A cartoon representation is also displayed with Jmol buttons for highlighting the helices. Information about β-strands and β-sheets can also be accessed from the secondary structure summary.

Information about all the turns in a protein can also be accessed from the secondary structure summary page. This option presents a user with a table of all the turns that occur in the protein as well as the turn type and class, start residue, end residue and a Jmol representation with Jmol buttons to select all the turns (Fig. 2.13).

In addition to the secondary structure summary, a user can also access information about the tertiary structure of the protein (Fig. 2.14). This view includes the Pfam (Finn *et al.*, 2006), CATH (Pearl *et al.*, 2005), SCOP (Conte *et al.*, 2000), GO (Ashburner *et al.*, 2000) and Interpro id's (Zdobnov and Apweiler, 2001) associated with each chain. Once again Jmol is also present but in this case the protein is shown in a ribbon representation coloured by chain.

The Structural module of FunGIMS contains tools related to secondary and tertiary structure as well as protein sequence feature prediction. Although the database (see section 2.5.2.1) provides most of the structurally derived data, a user may want to do a re-analysis of a structure or use the tools to analyze a new structure or model or protein sequence. At the time of writing, only X-ray data was supported. The structural module can be divided into roughly two parts, a structural data part and a analysis tools part.

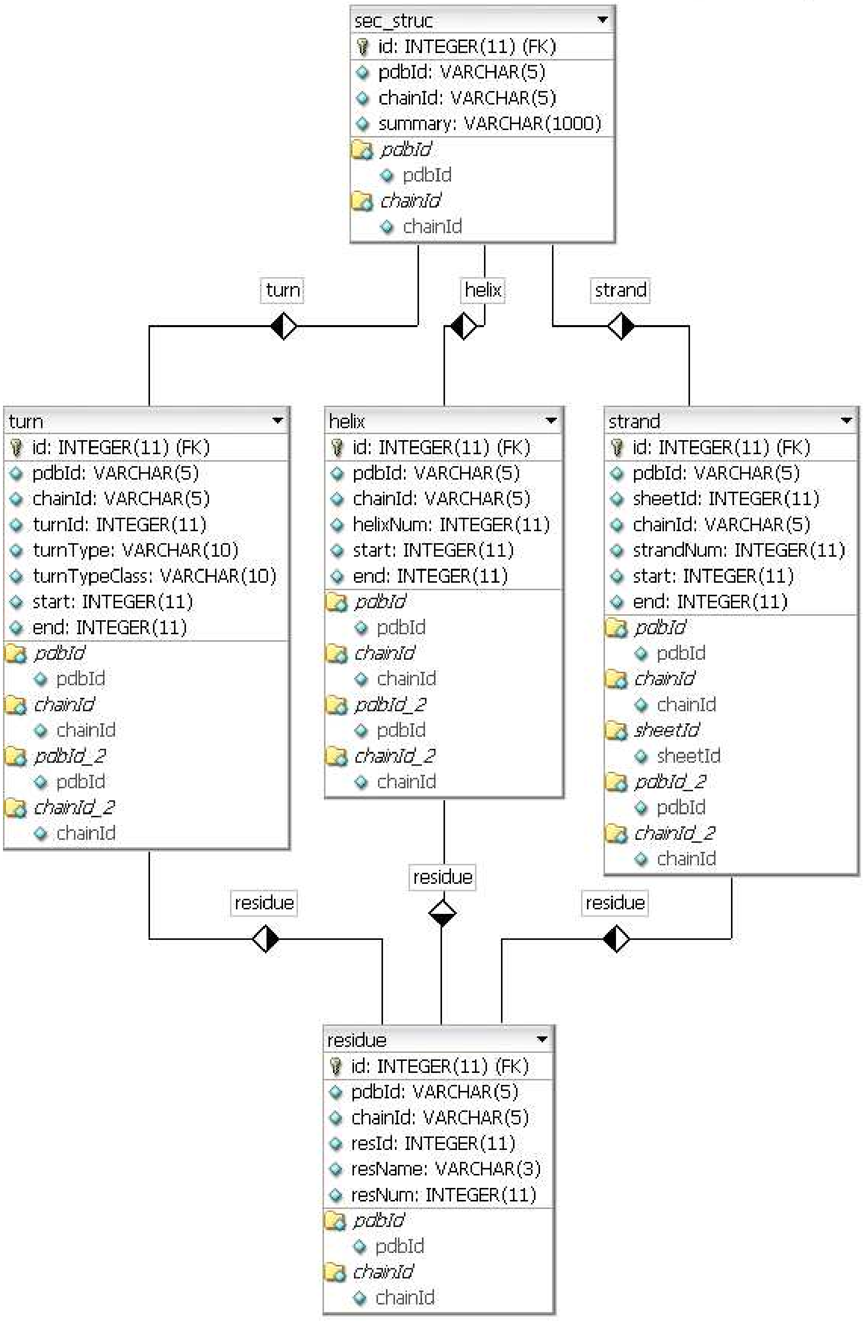


Figure 2.7: The relationship between different secondary structures in a chain and the residues in a protein. This provides the clearest example of how the data model organization follows the logical, hierarchical organization seen in a protein structure. Each secondary structure (`sec_struc`) object has several features such as a `helix` or a `strand` or a `turn`. And each of these specific secondary structural features also consists of a `residue` thus following the inherent logic in a protein structure. Due to the levels of inheritance, each `residue` object still has an `identifiable` and `description` object associated with it.

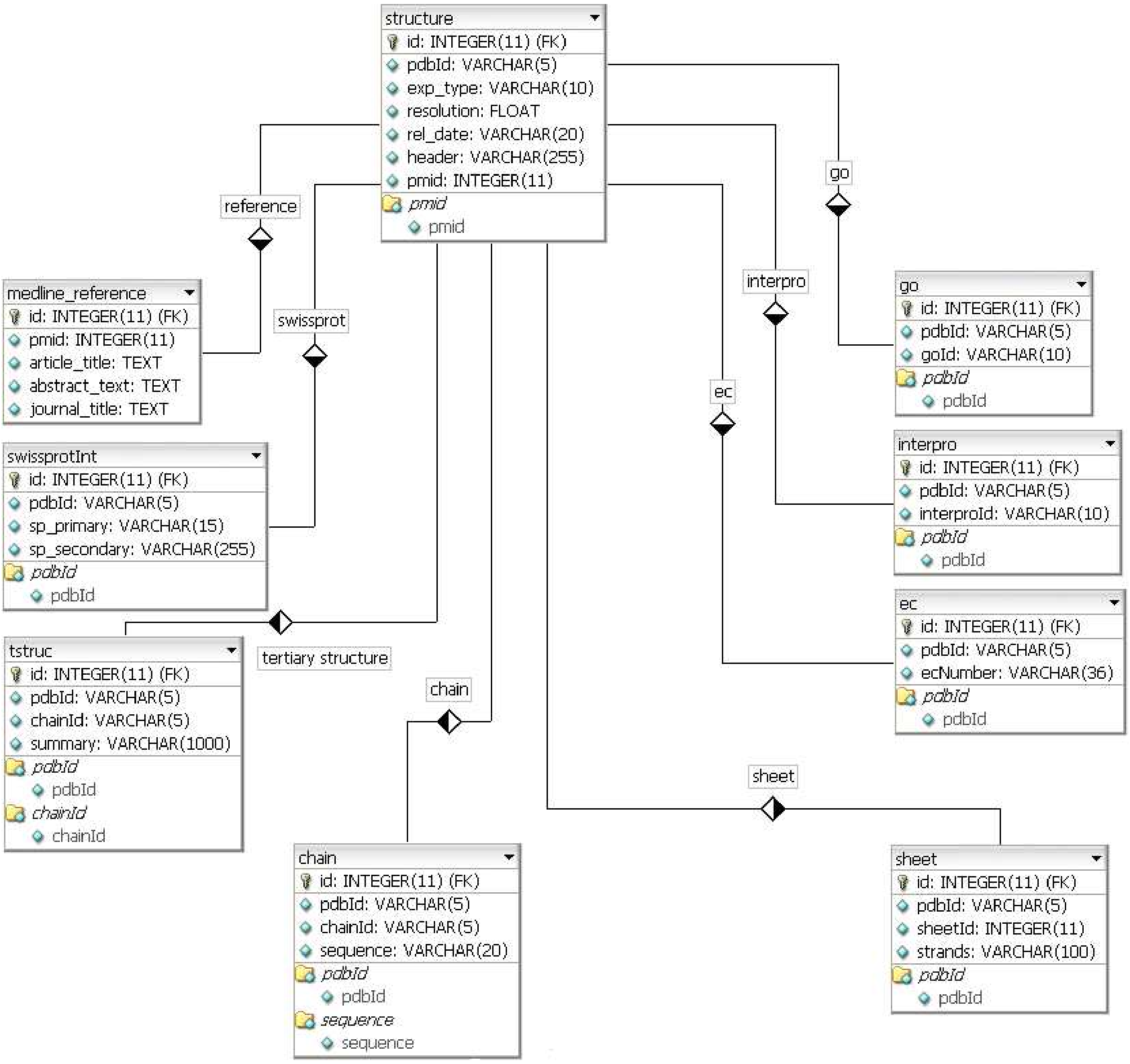
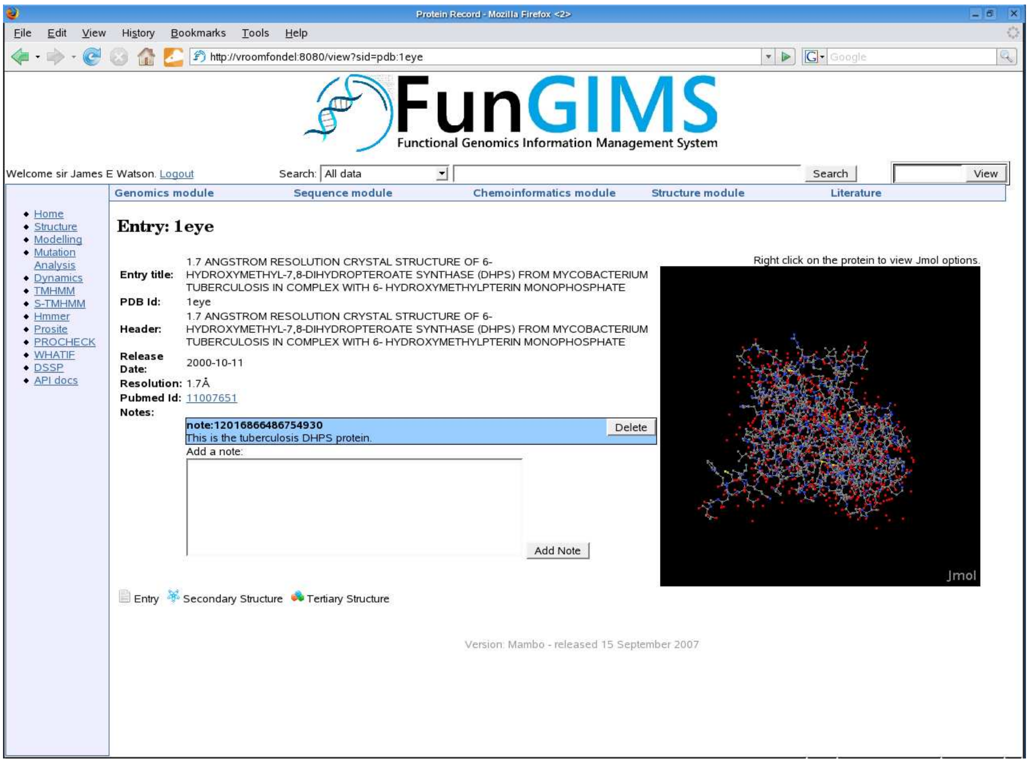


Figure 2.8: The data model for the high level `Structure` class. A `Structure` entry is linked to its reference (Pubmed) as well to high level classifiers such as `Interpro` and `GO`. The different organization levels can be seen clearly e.g. a `Structure` consists of one/many `Chain` objects and each `Structure` object also has other high level features such as a SwissProt id (`swissprotid`).

2.5.3.2. Data Analysis

The second part of the structural module is the data analysis tools (Fig. 2.15). This provides web interfaces to some commonly used tools in protein structural analysis. All these tools are external programs that are called using Python 2.4 system calls, and the


Protein Record - Mozilla Firefox <2>
File Edit View History Bookmarks Tools Help
http://vroomfondel:8080/view?sid=pdb:1eye
Google
FunGIMS
Functional Genomics Information Management System
Welcome sir James E Watson. Logout
Search: All data
Search
View
Genomics module
Sequence module
Chemoinformatics module
Structure module
Literature
Home
Structure
Modelling
Mutation Analysis
Dynamics
TMHMM
S-TMHMM
Hmmer
Prosite
PROCHECK
WHATIF
DSSP
API docs
Right click on the protein to view Jmol options.
Jmol
Delete
Add Note
Entry
Secondary Structure
Tertiary Structure
Version: Mambo - released 15 September 2007

Entry: 1eye

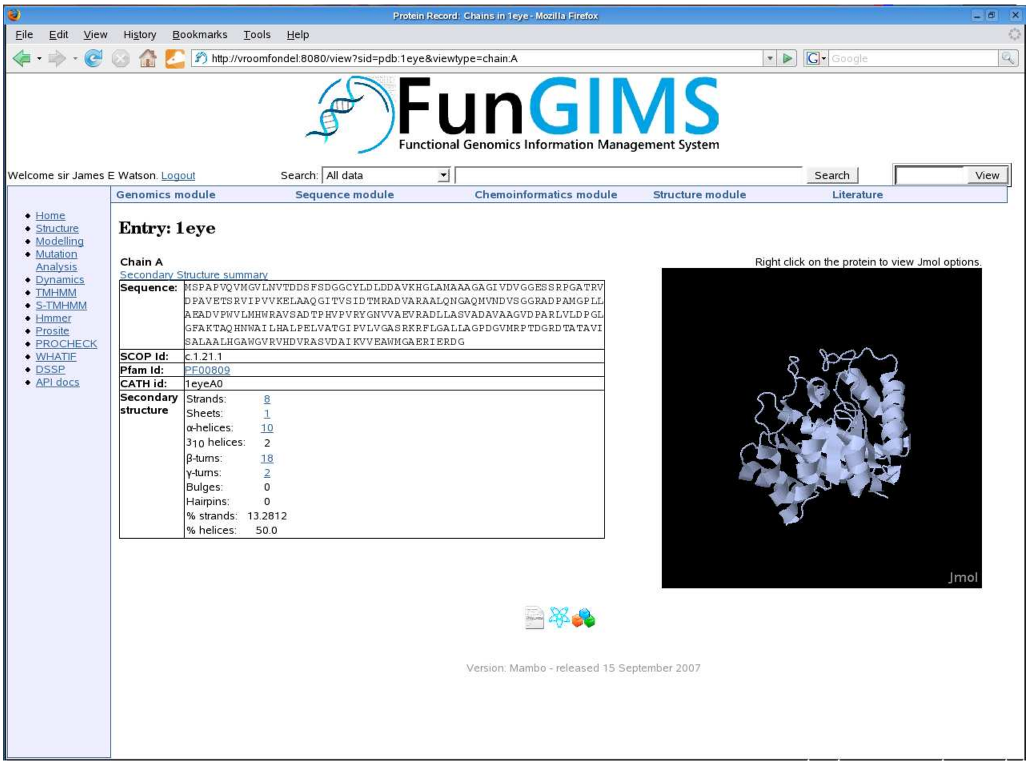
Entry title: 1.7 ANGSTROM RESOLUTION CRYSTAL STRUCTURE OF 6-HYDROXYMETHYL-7,8-DIHYDROPTEROATE SYNTHASE (DHPS) FROM MYCOBACTERIUM TUBERCULOSIS IN COMPLEX WITH 6- HYDROXYMETHYLPTERIN MONOPHOSPHATE
PDB Id: 1eye
Header: 1.7 ANGSTROM RESOLUTION CRYSTAL STRUCTURE OF 6-HYDROXYMETHYL-7,8-DIHYDROPTEROATE SYNTHASE (DHPS) FROM MYCOBACTERIUM TUBERCULOSIS IN COMPLEX WITH 6- HYDROXYMETHYLPTERIN MONOPHOSPHATE
Release Date: 2000-10-11
Resolution: 1.7Å
Pubmed Id: 11007651
Notes:
note:12016866486754930
This is the tuberculosis DHPS protein.
Add a note:

Figure 2.9: The primary view when a user views a protein. Note the general FunGIMS feature where an entry can be annotated by a note.

results are displayed to the user. Each program has a unique script located in the `utils` folder of the FunGIMS.

Users are able to analyze a protein sequence using these tools. The tools currently implemented in the Structural module are:

- Hmmer search against Pfam - Hmmer is a hidden markov model-based (HMM) search tool that tries to identify a protein sequence by matching it to a database of protein families (Finn *et al.*, 2006). Hmmer takes the sequence, an E-value cut-off and a database to search against. The output contains a list of families that matches the user submitted sequence. It also includes confidence values for every hit found to a protein family. The `hmmer.py` script in `utils` is used.
- TMHMM - TMHMM is a HMM-based tool for searching for transmembrane helices based on the amino acid sequence found in a protein sequence (Sonnhammer *et al.*, 1998). It takes a protein sequence as input and produces a graph showing which areas


Protein Record: Chains in 1eye - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://vroomfondel:8080/view?sid=pdb:1eye&viewtype=chain:A
Google
FunGIMS
Functional Genomics Information Management System
Welcome sir James E Watson. Logout
Search: All data
Search
View
Genomics module
Sequence module
Chemoinformatics module
Structure module
Literature
Home
Structure
Modelling
Mutation Analysis
Dynamics
TMHMM
S-TMHMM
Hmmer
Prosite
PROCHECK
WHATIF
DSSP
API docs
Right click on the protein to view Jmol options.
Jmol
Version: Mambo - released 15 September 2007

Entry: 1eye

Chain A

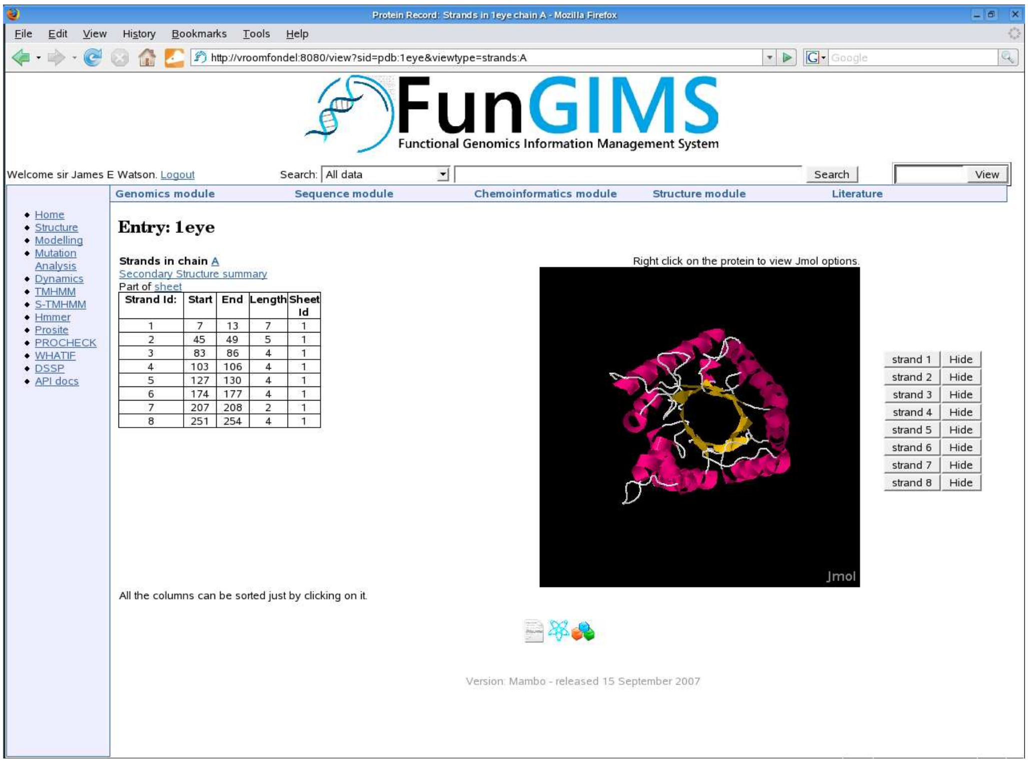
Secondary Structure summary

Sequence:	MSPAPVQVMGVLNVTDDSFSDGGCYLDLDDAVKHGLAMAAAGAGIVDVGGESSRPGATRV DPAVETSRVIPVVKELAAQGITVSIDTMRADVARAALQNGAQMVNDVSGGRADPAMGPLL AEADVPWVLMHWRAVSADTPHVPVRYGNVVAEVRADLLASVADAVAAGVDPARLVLDPGL GFAKTAQHNWAILHALPELVATGIPVLVGASRKRFLGALLAGPDGVMRPTDGRDTATAVI SALAALHGAWGVRVHDVRASVDAIKVVEAWMGAERIERDG
SCOP Id:	c.1.21.1
Pfam Id:	PF00809
CATH id:	1eyeA0
Secondary structure	Strands: 8 Sheets: 1 α-helices: 10 3_{10} helices: 2 β-turns: 18 γ-turns: 2 Bulges: 0 Hairpins: 0 % strands: 13.2812 % helices: 50.0

Figure 2.10: The chain summary view for a specific chain in a protein.

are predicted to contain transmembrane helices. The `tmhmm.py` script in `utils` is used.

- S-TMHMM - This tool tries to predict the topology (inside/outside) of any transmembrane helices found in a protein sequence (Viklund and Elofsson, 2004). It takes a protein sequence as input and produces a table showing the probability of each residue being inside or outside the membrane. The `stmhmm.py` script in `utils` is used.
- Prosite - Prosite is a database of protein motifs (de Castro *et al.*, 2006). These include short motifs such as glycosylation sites as well as longer motifs that can identify a specific protein family. To search Prosite, the ps_scan.pl script from the EBI is used. Using a protein sequence as input, it produces a list of motifs found in the protein. Flags can be set to exclude motifs with a high probability of occurrence, but this has not been implemented in the Structural module. The `prosite.py` script in `utils` is used.
- PROCHECK - This allows a user to check a protein structure file for any abnormal structural errors (Laskowski *et al.*, 1993). The checks are based on a set of normal



Entry: 1eye

Strands in chain A
Secondary Structure summary
Part of sheet

Strand Id:	Start	End	Length	Sheet Id
1	7	13	7	1
2	45	49	5	1
3	83	86	4	1
4	103	106	4	1
5	127	130	4	1
6	174	177	4	1
7	207	208	2	1
8	251	254	4	1

Right click on the protein to view Jmol options.

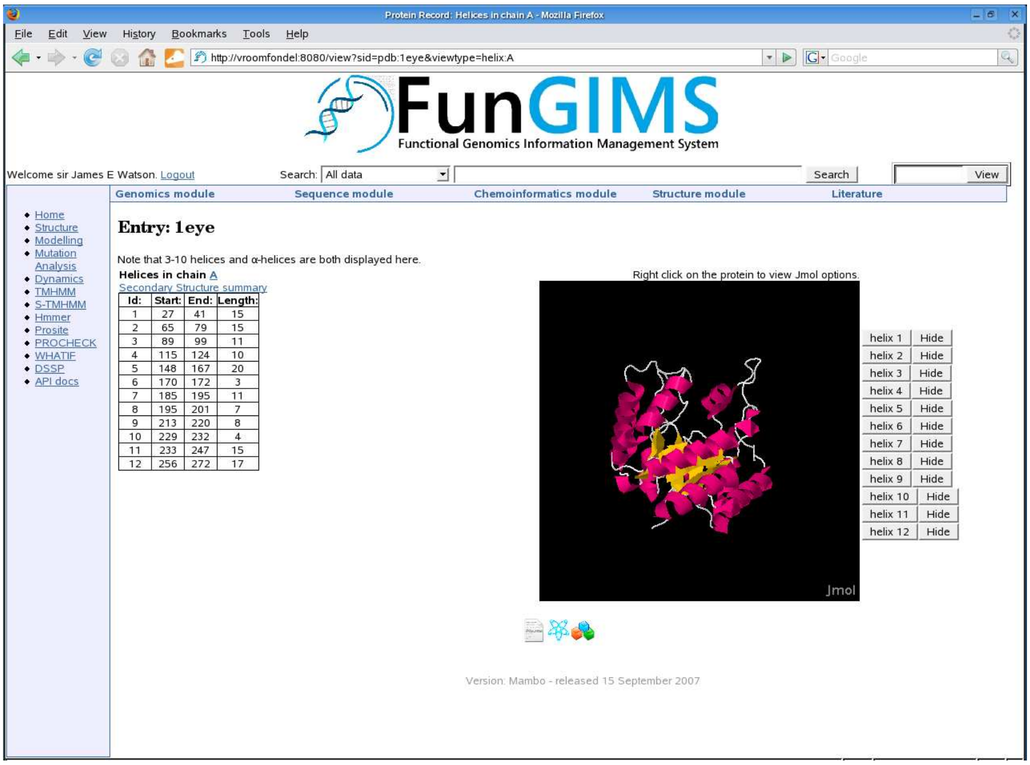
All the columns can be sorted just by clicking on it.

Version: Mambo - released 15 September 2007

Figure 2.11: The strand summary page for a protein chain.

structural parameters derived from the PDB. The input is a protein coordinate file and it produces a set of ten files that include Ramachandran plots, graphs plotting the deviation of each amino acid type from normal as well as a summary. In the Structural module the user can download each file for later use. The `procheck.py` script in `utils` is used.

- WHAT IF - WHAT IF is a comprehensive set of tools for molecular modelling and for analyzing proteins in their native environments (Vriend, 1990). The structure checking tool was implemented in the Structural module and this does a range of checks on a submitted protein file to identify possible errors and warnings. It produces a detailed report on the structure analysis that the user can download. The `whatif.py` script in `utils` is used.
- DSSP - This program calculates secondary structure based on the coordinates of the atoms in a PDB file (Kabsch and Sander, 1983). The program takes a pdb file as input and produces a report that gives the secondary structure of each amino acid. The `dssp.py` script in `utils` is used.



http://vroomfondel:8080/view?sid=pdb:1eye&viewtype=helix:A

Welcome sir James E Watson. Logout

Search: All data

Entry: 1eye

Note that 3-10 helices and α-helices are both displayed here.

Helices in chain A

Secondary Structure summary

Id:	Start:	End:	Length:
1	27	41	15
2	65	79	15
3	89	99	11
4	115	124	10
5	148	167	20
6	170	172	3
7	185	195	11
8	195	201	7
9	213	220	8
10	229	232	4
11	233	247	15
12	256	272	17

Right click on the protein to view Jmol options.

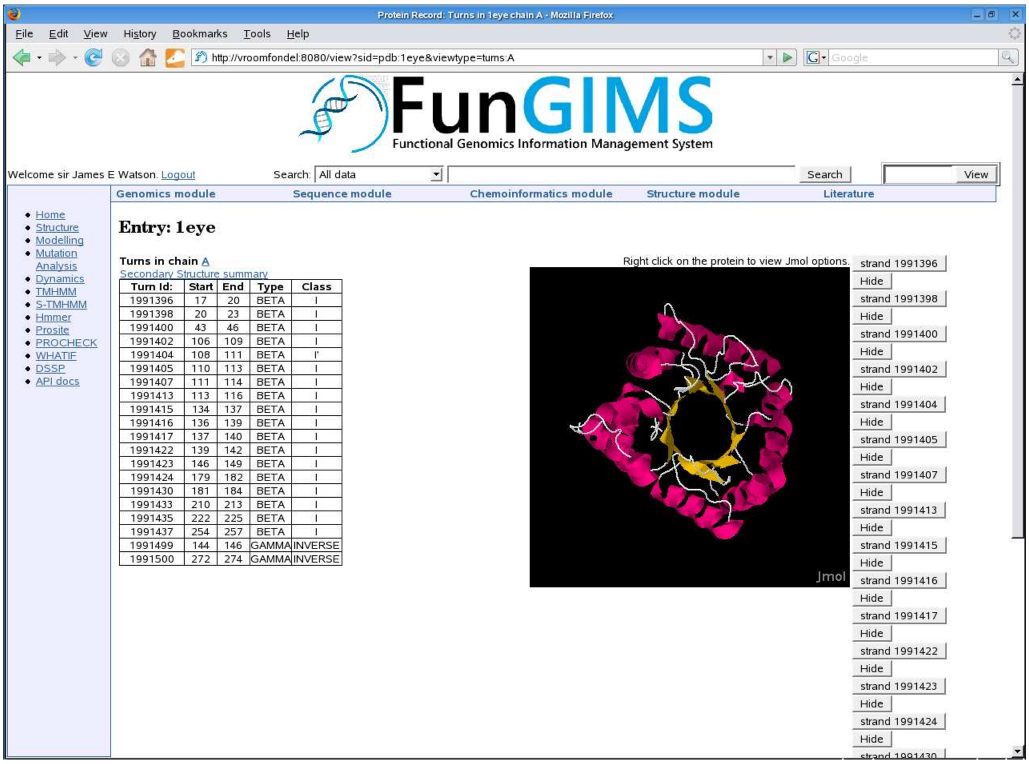
Version: Mambo - released 15 September 2007

Figure 2.12: The α-helix summary view for a protein chain.

All these tools accept either a file or a sequence from the user. The selected tool is then run via a tool-specific Python script, which thereafter uses Python system calls to run the appropriate tool on the sequence or file. The scripts for each tool are saved under the `utils` directory. All the results are saved on disk during the session. The results are also displayed to the user and the option to save the results to a certain group is available. Figures 2.16, 2.17 and 2.18 show the results from an analysis run of TMHMM, Hmmer against Pfam and a PROCHECK analysis.

2.5.3.3. Modelling and Molecular Dynamics

The third section of the Structural module has functions that allow the user to generate scripts for homology modelling and molecular dynamics (Fig. 2.15) and build models. For protein homology modelling the user has a choice between two programs, Modeller (Fiser and Sali, 2003) and WHAT IF (Vriend, 1990). The module will ask for the relevant information, pass it to the specific script located in the `utils` folder, and produce a script, using Python, which the user can download and run on his or her local machine. This



Entry: 1eye

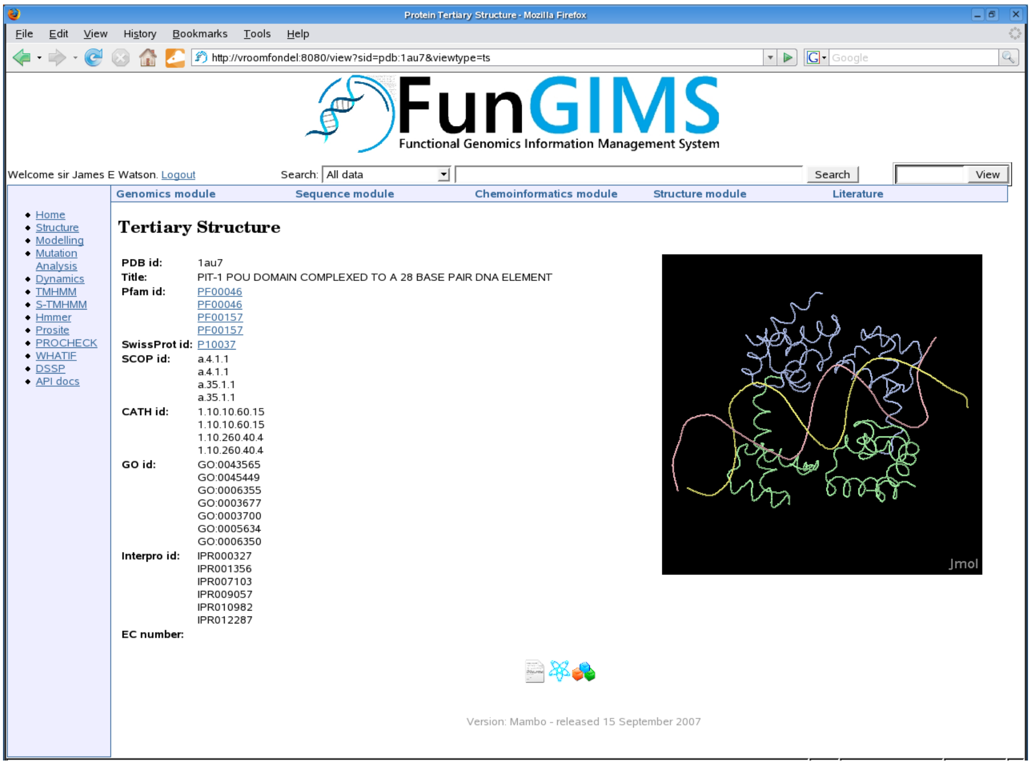
Turns in chain A
Secondary Structure summary

Turn Id:	Start	End	Type	Class
1991396	17	20	BETA	I
1991398	20	23	BETA	I
1991400	43	46	BETA	I
1991402	106	109	BETA	I
1991404	108	111	BETA	I'
1991405	110	113	BETA	I
1991407	111	114	BETA	I
1991413	113	116	BETA	I
1991415	134	137	BETA	I
1991416	136	139	BETA	I
1991417	137	140	BETA	I
1991422	139	142	BETA	I
1991423	146	149	BETA	I
1991424	179	182	BETA	I
1991430	181	184	BETA	I
1991433	210	213	BETA	I
1991435	222	225	BETA	I
1991437	254	257	BETA	I
1991499	144	146	GAMMA	INVERSE
1991500	272	274	GAMMA	INVERSE

Figure 2.13: The summary view for all the turns that occur in a protein chain.

precludes the user having to actually set up and understand the scripts and scripting language. In addition to the modelling scripts, the user may also decide to construct a model using the automatic method in the Structural module (Fig. 2.20). The user enters a template PDB id, target name, target sequence and refinement level. This will be passed to Modeller (version 9v1), which will perform an automatic alignment of the two sequences and then proceed to build a model. Currently the automated modelling process uses the first chain in a multi-chain protein as a template. When the model is ready, the user is alerted and presented with a page to download the model, modelling script and alignment file. A drawback of the automated modelling is the automated alignment performed by Modeller. When the sequences display a high identity, alignment is easy and should be accurate. However in lower identity ranges (less than 40%), automated alignment is not as accurate and it is advisable to do the alignment with manual curation of the results.

The module can also generate basic scripts, using Python, for three different molecular dynamics suites, (NAMD (Phillips *et al.*, 2005), CHARMM (Brooks *et al.*, 1983) and


Protein Tertiary Structure - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://vroomfondel:8080/view?sid=pdb:1au7&viewtype=ts
Google
FunGIMS
Functional Genomics Information Management System
Welcome sir James E Watson. Logout
Search: All data
Search
View
Genomics module
Sequence module
Chemoinformatics module
Structure module
Literature
Home
Structure
Modelling
Mutation Analysis
Dynamics
TMHMM
S-TMHMM
Hmmer
Prosite
PROCHECK
WHATIF
DSSP
API docs
Jmol
Version: Mambo - released 15 September 2007

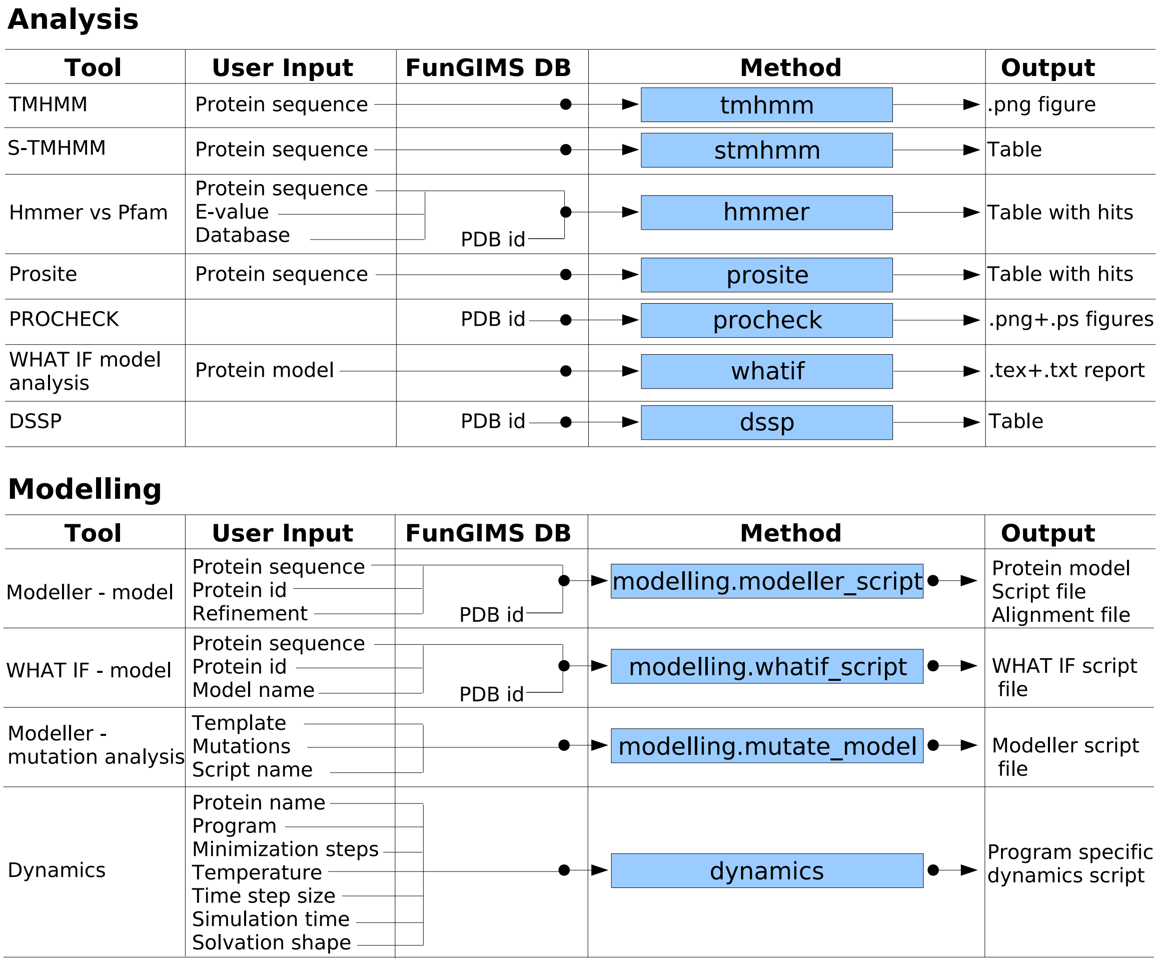
Tertiary Structure

PDB id: 1au7
Title: PIT-1 POU DOMAIN COMPLEXED TO A 28 BASE PAIR DNA ELEMENT
Pfam id: PF00046
PF00046
PF00157
PF00157
SwissProt id: P10037
SCOP id: a.4.1.1
a.4.1.1
a.35.1.1
a.35.1.1
CATH id: 1.10.10.60.15
1.10.10.60.15
1.10.260.40.4
1.10.260.40.4
GO id: GO:0043565
GO:0045449
GO:0006355
GO:0003677
GO:0003700
GO:0005634
GO:0006350
Interpro id: IPR000327
IPR001356
IPR007103
IPR009057
IPR010982
IPR012287
EC number:

Figure 2.14: The tertiary structure view of a protein. This shows information for the complete protein complex.

Yasara (http://www.yasara.com) given user input. The dynamics section only supports script generation, not running the actual simulations as this is extremely resource intensive. This allows the user to focus on the research questions without the need for technical knowledge. Figure 2.20 shows the interface for the molecular dynamics script generation section. The molecular dynamics scripts will need further editing depending on the molecule the user wants to investigate and the type of dynamics. All the modelling functionalities are located in the `utils` folder and the `modelling.py` script is used. For dynamics the `dynamics.py` script in `utils` is used. While validated homology programs are used, the quality of a model is determined by various factors such as template resolution, template-target alignment and the specific algorithm used.

The running of simulations in a UNIX environment will still require some skills and UNIX knowledge but an IT support person should be able to assist with the installation of the programs. The interpretation of the dynamics results are up to the user as automated analysis is not really a possibility yet. The intent is to provide the user with basic access



Analysis

Tool	User Input	FunGIMS DB	Method	Output
TMHMM	Protein sequence		tmhmm	.png figure
S-TMHMM	Protein sequence		stmhmm	Table
Hmmer vs Pfam	Protein sequence E-value Database	PDB id	hmmer	Table with hits
Prosite	Protein sequence		prosite	Table with hits
PROCHECK		PDB id	procheck	.png+.ps figures
WHAT IF model analysis	Protein model		whatif	.tex+.txt report
DSSP		PDB id	dssp	Table

Modelling

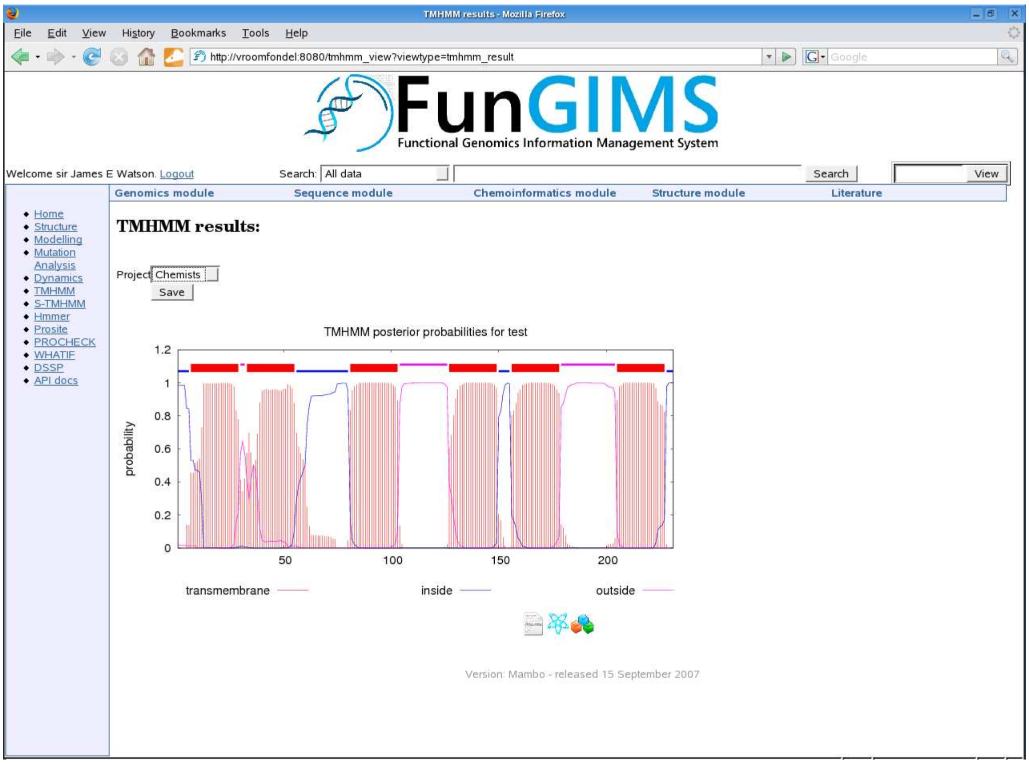
Tool	User Input	FunGIMS DB	Method	Output
Modeller - model	Protein sequence Protein id Refinement	PDB id	modelling.modeller_script	Protein model Script file Alignment file
WHAT IF - model	Protein sequence Protein id Model name	PDB id	modelling.whatif_script	WHAT IF script file
Modeller - mutation analysis	Template Mutations Script name		modelling.mutate_model	Modeller script file
Dynamics	Protein name Program Minimization steps Temperature Time step size Simulation time Solvation shape		dynamics	Program specific dynamics script

Figure 2.15: The different tools available in the Structural module. Shown are the input (user and FunGIMS supplied) required for each of the tools, the specific method called in the `utils` folder as well as the type of output the tool generates.

to molecular dynamics functionality but guidance in the interpretation of the results is currently outside the scope of the system. It is always recommended that the user consult suitable literature when engaging in any form of advanced simulations.

2.5.3.4. Help Section

FunGIMS was designed to assist biologists to conduct faster and easier analysis and exploration of data. To further this goal, a help page is provided for each function in the Structural module. This can be accessed by clicking on the link found on each page. To increase visibility it has been labeled in red. Figure 2.19 shows a typical result when a user clicked on a help link for a specific function. The help link provides a brief synopsis



TMHMM results:

Project Chemists

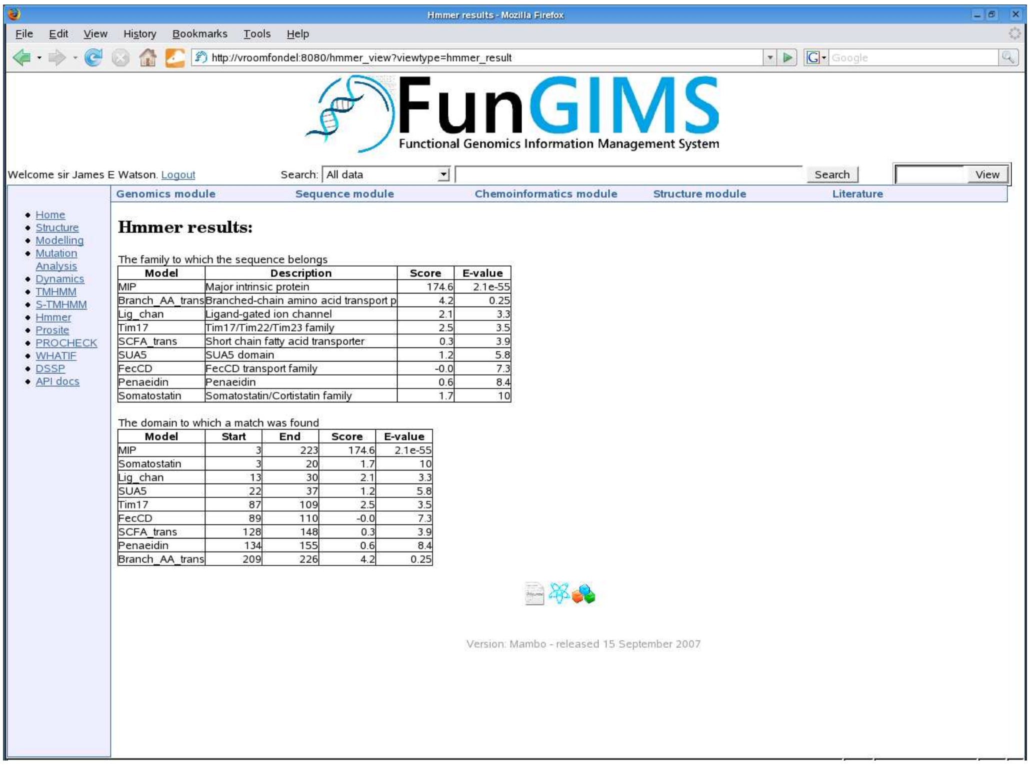
Version: Mambo - released 15 September 2007

Figure 2.16: The results from a transmembrane helix prediction on a submitted protein sequence. The drop-down menu allows the user to save the results to a specific group.

of the tool and the inputs required, as well as the output a user might expect when the tool runs successfully.

2.5.3.5. Configuration

The Structural module relies on various external programs to provide analysis methods. Installation locations and execution of these programs usually differ between machines and programs. To overcome this, a configuration file (`utils/config.py`) was created that stores all the program specific settings. This file can be edited by hand to change program properties. For each program the following properties are specified: the path to the program (executable file), a program-specific temporary directory for output, and other program specific parameters and settings. These programs are then called from inside the Structural module simply by referencing these variables. This makes system administration far easier as program settings have only to be specified and changed in one file.


Hmmer results - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://vroomfondel:8080/hmmer_view?viewtype=hmmer_result
Google
FunGIMS
Functional Genomics Information Management System
Welcome sir James E Watson. Logout
Search: All data
Search
View
Genomics module
Sequence module
Chemoinformatics module
Structure module
Literature
Home
Structure
Modelling
Mutation Analysis
Dynamics
TMHMM
S-TMHMM
Hmmer
Prosite
PROCHECK
WHATIF
DSSP
API docs
Version: Mambo - released 15 September 2007

Hmmer results:

The family to which the sequence belongs

Model	Description	Score	E-value
MIP	Major intrinsic protein	174.6	2.1e-55
Branch_AA_trans	Branched-chain amino acid transport p	4.2	0.25
Lig_chan	Ligand-gated ion channel	2.1	3.3
Tim17	Tim17/Tim22/Tim23 family	2.5	3.5
SCFA_trans	Short chain fatty acid transporter	0.3	3.9
SUA5	SUA5 domain	1.2	5.8
FecCD	FecCD transport family	-0.0	7.3
Penaeidin	Penaeidin	0.6	8.4
Somatostatin	Somatostatin/Cortistatin family	1.7	10

The domain to which a match was found

Model	Start	End	Score	E-value
MIP	3	223	174.6	2.1e-55
Somatostatin	3	20	1.7	10
Lig_chan	13	30	2.1	3.3
SUA5	22	37	1.2	5.8
Tim17	87	109	2.5	3.5
FecCD	89	110	-0.0	7.3
SCFA_trans	128	148	0.3	3.9
Penaeidin	134	155	0.6	8.4
Branch_AA_trans	209	226	4.2	0.25

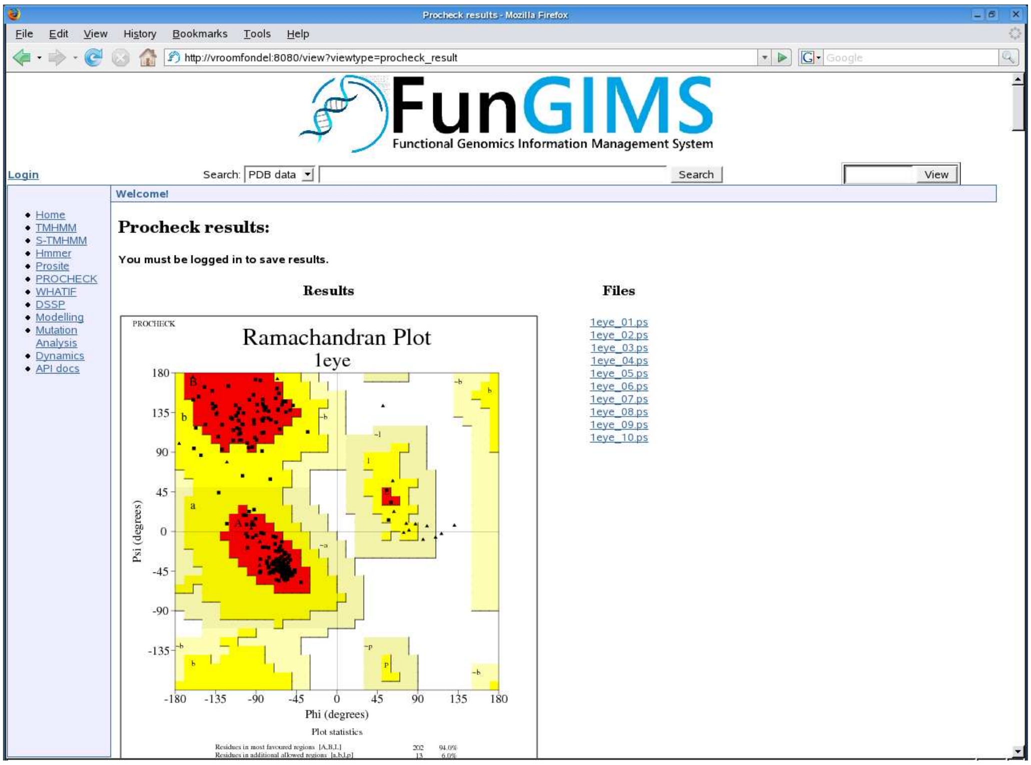
Figure 2.17: The results from a Hmmer search across Pfam using the structural module.

2.6. Future Improvements

2.6.1. FunGIMS

A system such as FunGIMS is in a constant flux of development. FunGIMS was designed to allow for the easy addition of new tools and features. There are a number of areas that can be improved upon, the database being one of them. Database table optimization would allow for queries to be dealt with faster. Distributed databases would lessen the load on the server when the database size increases significantly. In the current implementation of FunGIMS, the database size presented some challenges and smart indexing of often-queried columns in tables resulted in a decrease in query time. The database should also be expanded to include more detailed data types such as protein chip array data.

Furthermore, smart file recognition and improved file parsers would enable the user to upload a file, allow FunGIMS to parse it entirely and then insert the data into the



Procheck results:

You must be logged in to save results.

Plot statistics

Residues in most favoured regions [A,B,L] 202 94.0%
Residues in additional allowed regions [a,b,l,p] 13 6.0%

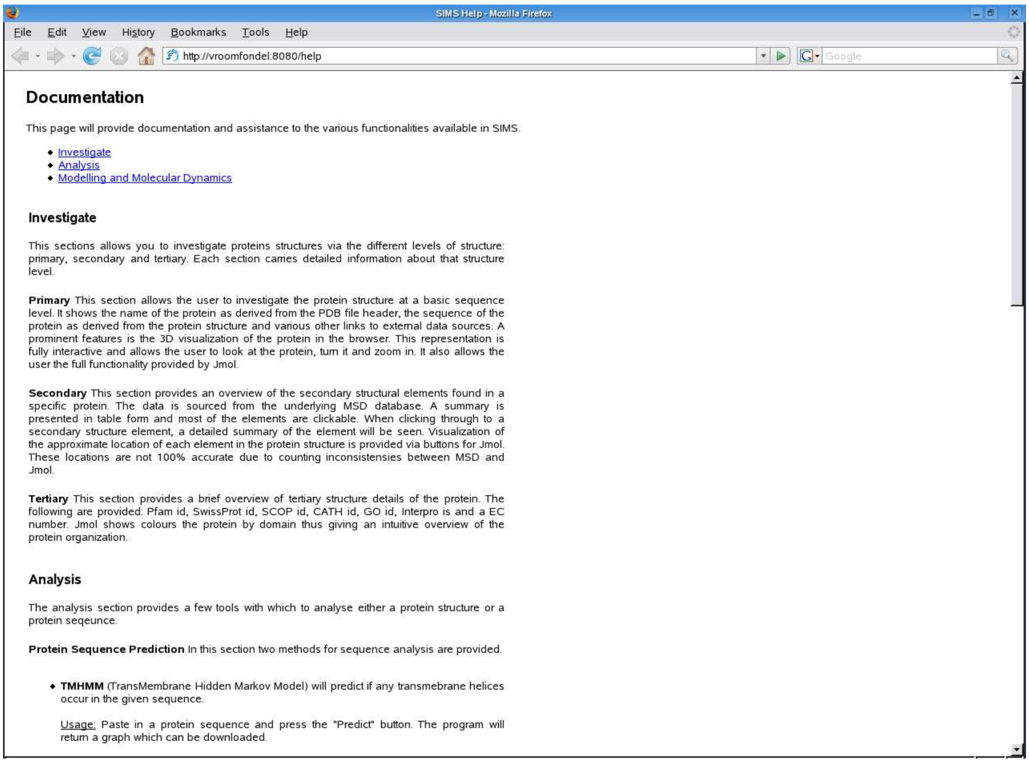
Figure 2.18: The results from a PROCHECK analysis run on PDB 1EYE.

database, not merely as a file but as a full data type. This allows queries to be more accurate as uploaded files will be parsed and stored in a data type specific manner. Automatic link generation between entries would be another major benefit to FunGIMS. Currently links between entries are generated when the database is first populated with public data and when a user links to entries with a note. Automatic link generation would navigate free text fields, notes and description text and then create the appropriate links. This automatic link generation tool should run on a daily basis so that links are always up to date.

2.6.2. Structural Module

In addition to the improvements to FunGIMS mentioned in the previous section, the Structural module also has some possible improvements.

More analysis methods can be included for different features. Tools such as consensus sec-



Documentation

This page will provide documentation and assistance to the various functionalities available in SIMS.

- Investigate
- Analysis
- Modelling and Molecular Dynamics

Investigate

This sections allows you to investigate proteins structures via the different levels of structure: primary, secondary and tertiary. Each section carries detailed information about that structure level.

Primary This section allows the user to investigate the protein structure at a basic sequence level. It shows the name of the protein as derived from the PDB file header, the sequence of the protein as derived from the protein structure and various other links to external data sources. A prominent features is the 3D visualization of the protein in the browser. This representation is fully interactive and allows the user to look at the protein, turn it and zoom in. It also allows the user the full functionality provided by Jmol.

Secondary This section provides an overview of the secondary structural elements found in a specific protein. The data is sourced from the underlying MSD database. A summary is presented in table form and most of the elements are clickable. When clicking through to a secondary structure element, a detailed summary of the element will be seen. Visualization of the approximate location of each element in the protein structure is provided via buttons for Jmol. These locations are not 100% accurate due to counting inconsistensies between MSD and Jmol.

Tertiary This section provides a brief overview of tertiary structure details of the protein. The following are provided: Pfam id, SwissProt id, SCOP id, CATH id, GO id, Interpro is and a EC number. Jmol shows colours the protein by domain thus giving an intuitive overview of the protein organization.

Analysis

The analysis section provides a few tools with which to analyse either a protein structure or a protein seqeunce.

Protein Sequence Prediction In this section two methods for sequence analysis are provided.

- **TMHMM** (TransMembrane Hidden Markov Model) will predict if any transmebrane helices occur in the given sequence.

  Usage: Paste in a protein sequence and press the "Predict" button. The program will return a graph which can be downloaded.

Figure 2.19: The help section for the Investigate section. Each function has its own help section on the Help page.

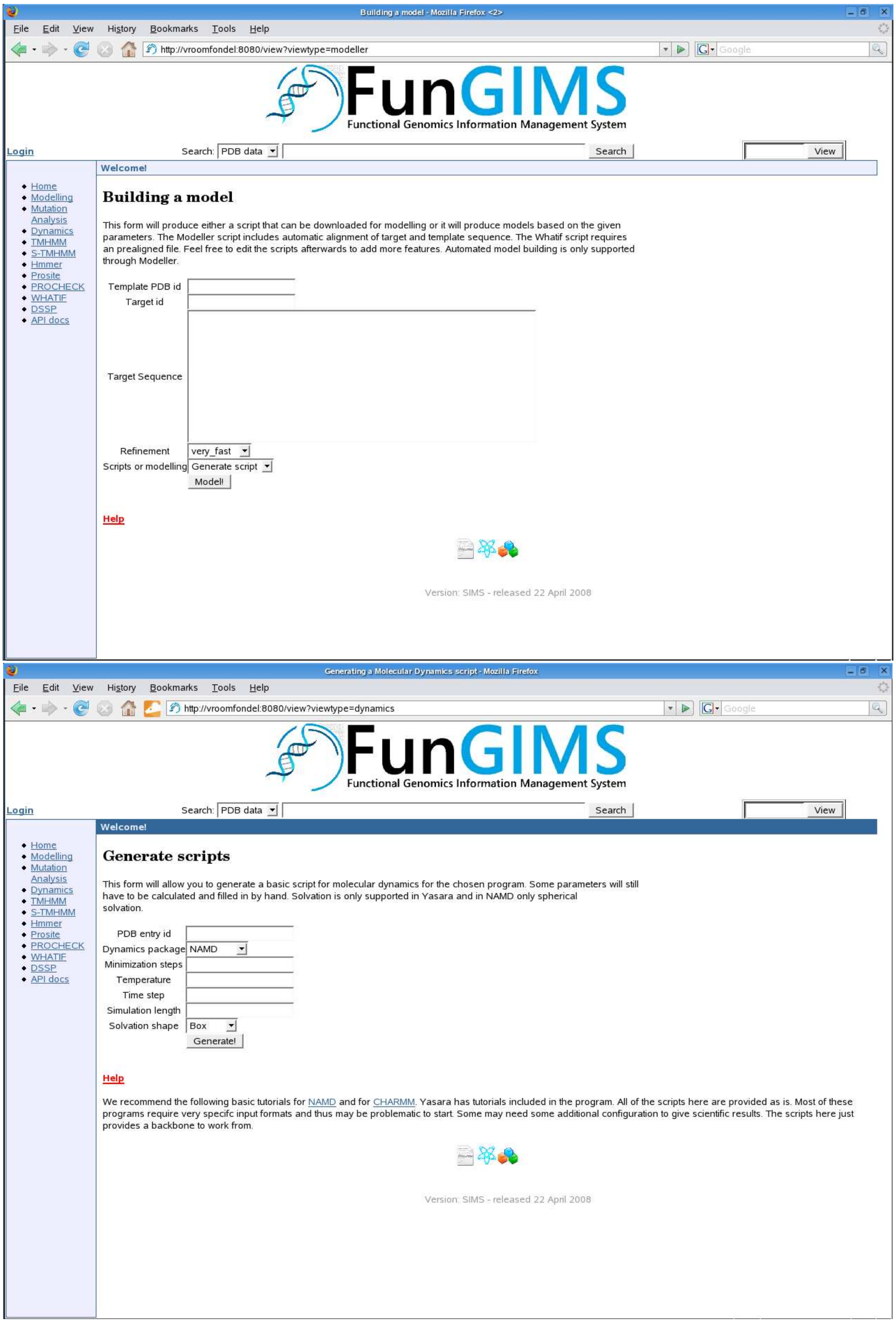
ondary structure prediction, protein export signal prediction and other protein sequence analysis tools will be a benefit to the system.

The most improvement is probably in the modelling and simulation section. The current scripts can be modified to include modelling on the selected chain of a protein, on multiple templates as well as including ligands in the modelling process. A feature could also be implemented to use alignments provided by the user. More simulation scripts with different parameters and environments could also possibly be added. A possible addition could be the implementation of a module whereby a user can start a simulation on a cluster or another computer while being able to control it from the FunGIMS system. This will allow the user to run simulations on various machines without needing the technical knowledge.

There is scope for the improvement of the user interface of the Structural module. Jmol buttons for secondary structure elements can be made more accurate. In addition a visualization library can also be included to generate scalable images of a summary of the secondary structure elements found in a protein and present them to the user in a downloadable format. A useful improvement would be scripts that facilitate a more automatic update of the database as soon as the data sources used, are updated. This would lessen the load on the site administrator and would keep the database up to date.

2.7. Conclusion

FunGIMS consists of various modules dedicated to different data types. The Structural module currently provides functions to explore structural data for a specific protein, conduct analysis on a user-submitted protein structure, including analysis such as transmembrane helix prediction, Prosite motif search and also allows the user to create homology modelling and molecular dynamics scripts. The application of the Structural module to various problems in FMDV will be discussed in the next three chapters.


Building a model - Mozilla Firefox <2>
File Edit View History Bookmarks Tools Help
http://vroomfondel:8080/view?viewtype=modeller
Google
FunGIMS
Functional Genomics Information Management System
Login
Search: PDB data
Search
View
Welcome!
Home
Modelling
Mutation Analysis
Dynamics
TMHMM
S-TMHMM
Hmmer
Prosite
PROCHECK
WHATIF
DSSP
API docs

Building a model

This form will produce either a script that can be downloaded for modelling or it will produce models based on the given parameters. The Modeller script includes automatic alignment of target and template sequence. The Whatif script requires an prealigned file. Feel free to edit the scripts afterwards to add more features. Automated model building is only supported through Modeller.

Template PDB id
Target id
Target Sequence
Refinement: very_fast
Scripts or modelling: Generate script
Model!

Help

Version: SIMS - released 22 April 2008

Generating a Molecular Dynamics script - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://vroomfondel:8080/view?viewtype=dynamics
Google
FunGIMS
Functional Genomics Information Management System
Login
Search: PDB data
Search
View
Welcome!
Home
Modelling
Mutation Analysis
Dynamics
TMHMM
S-TMHMM
Hmmer
Prosite
PROCHECK
WHATIF
DSSP
API docs

Generate scripts

This form will allow you to generate a basic script for molecular dynamics for the chosen program. Some parameters will still have to be calculated and filled in by hand. Solvation is only supported in Yasara and in NAMD only spherical solvation.

PDB entry id
Dynamics package: NAMD
Minimization steps
Temperature
Time step
Simulation length
Solvation shape: Box
Generate!

Help

We recommend the following basic tutorials for NAMD and for CHARMM. Yasara has tutorials included in the program. All of the scripts here are provided as is. Most of these programs require very specifc input formats and thus may be problematic to start. Some may need some additional configuration to give scientific results. The scripts here just provides a backbone to work from.

Version: SIMS - released 22 April 2008

Figure 2.20: Top: The automated modelling interface when building a model using Modeller. The user can decide to generate homology modelling scripts for Modeller or WHAT IF. Bottom: The molecular dynamics script-generating interface. Users can select between the different programs from the drop-down menu in the form.

Chapter 3

Reannotation of Foot-and-Mouth Disease Virus proteome

3.1. Introduction

Foot-and-Mouth Disease is a vesicular disease of cloven-hoofed animals and is caused by the Foot-and-Mouth Disease Virus (FMDV). It is a highly contagious and often fatal disease that infects economically important animals such as cattle and pigs. FMDV presents symptoms such as oral blisters and blistered hooves, which may result in lameness. In young animals infection can result in a miocarditis that can be fatal to the animal. Although most animals usually recover from FMDV infections, problems such as weight loss and swelling can continue for several months and this affects among others, milk production in cows, reduction in the availability of meat as well as affect working cattle used for ploughing in the African rural setting. FMDV is mostly transmitted via physical contact between animals kept in the same enclosure or via the clothes of the animal handlers.

FMDV occurs naturally throughout the world in wild populations but can cause economic problems when it infects domestic livestock populations (Fig. 3.1). FMDV infections can spread with great speed as seen in the outbreaks in the UK (Mason *et al.*, 2003*b*) in 2001. This outbreak resulted in an estimated loss of £4.1bn which illustrates the huge costs associated with FMD outbreaks.

FMDV is a small Aphthovirus that forms part of the Picornaviridae family (Levy *et al.*, 1994). It is non-enveloped and consists of an icosahedral capsid consisting of up to 60

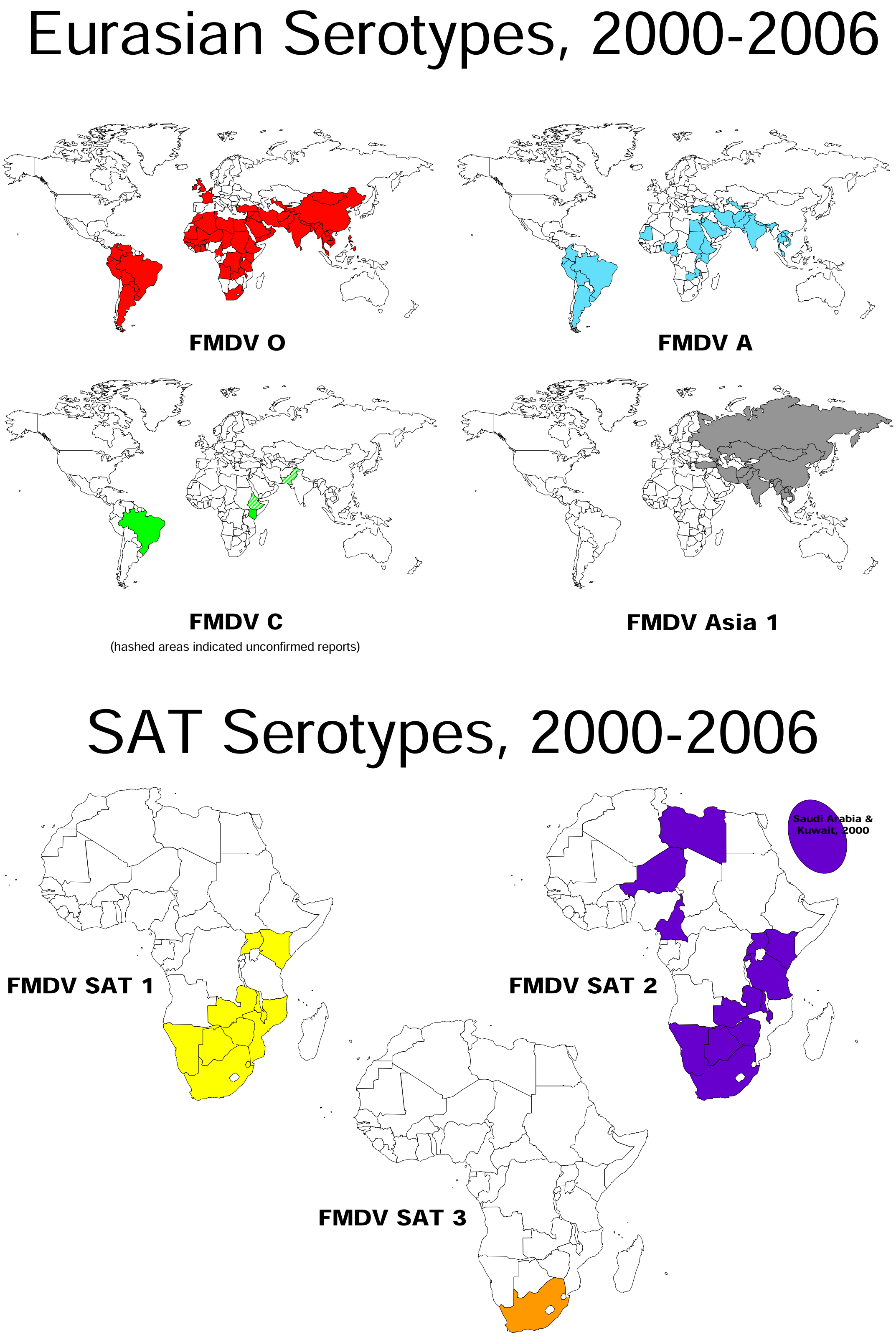

Eurasian Serotypes, 2000-2006
FMDV O
FMDV A
FMDV C
(hashed areas indicated unconfirmed reports)
FMDV Asia 1
SAT Serotypes, 2000-2006
Saudi Arabia & Kuwait, 2000
FMDV SAT 1
FMDV SAT 2
FMDV SAT 3

Figure 3.1: The distribution of FMDV outbreaks from 2000-2006 (FAO World Reference Laboratory for Foot-and-Mouth Disease, http://www.wrlfmd.org/maps/fmd_maps.htm). Top: Eurasian serotype outbreaks. Bottom: SAT serotype outbreaks.

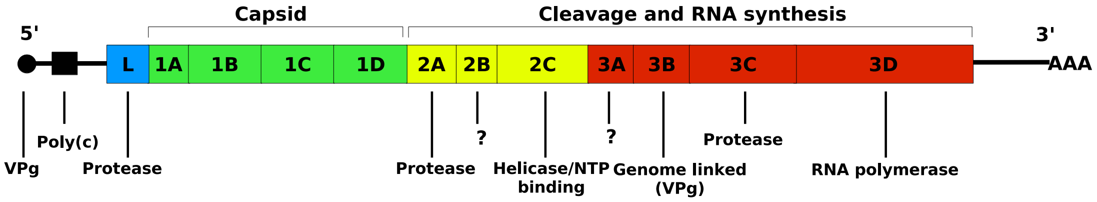


Figure 3.2: The genome organization of FMDV. It is divided into four basic sections. The 5' end is attached to the VPg protein and the 3' end is polyadenylated.

copies of four structural proteins. The structural aspects of FMDV will be discussed in more detail in chapters 4 and 5. The capsid contains a small 8.4 kb, single stranded RNA genome of positive polarity. In most cellular RNAs and some viral RNAs, a methylated G cap is usually found at the 5' terminus. In picornaviruses this is not the case and a VPg (3B) protein is bound to the 5' end (Fig. 3.2). This protein is 20-24 amino acids in length and is functionally, but not structurally, similar to several plant virus 5' terminal moieties. (Levy *et al.*, 1994). The virus also carries a polyadenylated tail at the 3' terminal. The length of this tail is encoded genetically and differs between the picornavirus members. This poly(A) tail is implicated in various roles related to genome replication.

The genome of FMDV is organized into a 5' untranslated region (5' UTR), an open reading frame (ORF) and a 3' UTR (Fig. 3.2). The ORF is divided into four basic regions: L, P1, P2, P3. The first section (L) encodes a protease that is responsible for early autocleavage of itself from the the polypeptide produced after translation. L^{pro} (Gradi *et al.*, 2003). In the L-coding region there are 2 AUG start codons. These code for proteins Lab and Lb. Both proteins appear to be present in the host but mutation studies have shown that Lb is vital to virus viability (Mason *et al.*, 2003*a*). Deletion studies have also shown that L^{pro} is needed for the virus to spread and infect its host. If L^{pro} is missing, the animal shows none of the symptoms typically associated with FMDV (Mason *et al.*, 2003*a*).

The second section produces four structural proteins (1A-D) and 2A. Post-translational cleavage by the 3C protease produces 1A-D that assembles into the icosahedral capsid. This capsid is unaffected by solvents such as ether and chloroform as there is no lipid membrane surrounding the virus (Levy *et al.*, 1994).

The third section produces three peptides after full cleavage, 2A-C. 2A seems to be an autoprotease that helps L^{pro} with early cleavage of cellular proteins and has some membrane binding ability. 2A is a short peptide consisting of only 18 residues. 2B enhances membrane permeability and blocks secretory pathways and seems to localize to sites of viral genome replication in vesicles derived from the ER (Carrillo *et al.*, 2005; Moffat *et al.*, 2005). It is also known to associate with the endoplasmic reticulum which is the site of virus genome replication. 2C appears to be associated with nucleotide binding (ATPase) and may have some helicase abilities (Mason *et al.*, 2003*a*). 2C has also been implicated in RNA synthesis initiation and localizes to virus replication vesicles. 2B and 2C are also implicated in virus-induced cytopathic effects.

The fourth section also produces 4 proteins after cleavage, namely 3A-D. The function of 3A is unknown but it seems to be involved in RNA replication (Mason *et al.*, 2003*a*) and may play a role in virus virulence (Carrillo *et al.*, 2005). Other studies have also shown that 3A directly associates with 3D and can function as a 3D co-factor (Hope *et al.*, 1997). In addition, previous studies have shown 3A to be the most invariable protein in FMDV (Carrillo *et al.*, 2005). 3A also forms a precursor with 3B i.e. 3AB, which has been implicated in RNA replication and supporting evidence comes from the fact that 3A fractionates with the ER membranes (Mason *et al.*, 2003*a*). FMDV contains 3 copies of 3B which is unique among the Picornaviridae. These 3 copies are referred to as 3B1 (23 aa), 3B2 (24 aa) and 3B3 (24 aa). The 3B becomes VPg after cleavage from a 3AB precursor. 3B appears to be associated with RNA replication, as the homologue in poliovirus helps to initiate genomic RNA synthesis (Carrillo *et al.*, 2005). Carillo and co-workers examined the variability in 3B and found that 3B1 and 3B2 are the most variable, and thus may play a role in host range and virulence. 3C is a protease of 213 amino acids, which helps to cleave the different precursor peptides from the main polypeptide produced during translation as well as cleaving host translation factors. The 3C^{pro} is responsible for ten of the thirteen cleavages of the polypeptide. Previous 3C studies have shown this protein to be conserved and thus have a limited tolerance for mutations (van Rensburg *et al.*, 2002). 3D is a virally encoded RNA dependant RNA polymerase (RdRp). It is the biggest protein encoded by the FMDV genome and is comprised of 469 amino acids. It is also one of the most highly conserved sequences in the FMDV genome (Carrillo *et al.*, 2005). 3D is responsible for the elongation of nascent RNA strands during replication. 3C and 3D will be discussed in more detail in chapters 4 and 5.

FMDV exists as various subtypes even within a serotype, a likely consequence of the high mutation rate of the virus, and although some comparisons have been done between one or two viruses, there has been no detailed proteome comparison between the different serotypes. In this section various serotype proteomes were analyzed and compared to determine if there are any major protein differences or shifts in patterns in the sequences which may help to explain the phenotypic differences seen between the serotypes. These differences include effects such as host specificity, spreading and infection speed and virulence. By identifying the differences, it should be possible to map which areas are responsible for these effects. FMDV is a devastating disease and understanding how the proteins differ from serotype to serotype will help in unraveling the important regions in each protein. In this section four methods were used to characterize each protein. A Pfam family prediction was done to identify the family. This was followed by a Prosite pattern search. The absence or presence of certain patterns can help to explain differences seen between the various serotypes. It can also help to identify structurally important areas on a protein as these areas will be conserved throughout the various serotypes. A secondary structure prediction helped to identify areas that play a vital role on the structure of the protein. It has also assisted in identifying areas where variability has a possible effect on the structure, however small that might be. A final tool that was used were hydrophobic plots. As mentioned before, various of the FMDV proteins are membrane-associated and changes in hydrophobicity of a sequence may affect the association of these proteins with the various membranes.

3.2. Methods

Dr. F. Maree (ARC) supplied 3 proteomes for annotation (SAT1/SAR/09/81, SAT1/KNP /196/91, SAT2/ZIM/07/83) and 6 more were generated from genome sequences obtained from Genbank (A24 (gi:46810792), A10 (gi:46810758), C3 (gi:46810870), O1/BFS/46 (gi:46810888), O/SAR/19/2000 (gi:30145780), SAT3/BEC/29 (gi:46810958)). Each proteome was split into its separate proteins: L, VP1, VP2, VP3, VP4, 2A, 2B, 2C, 3A, 3B1, 3B2, 3B3, 3C and 3D. All sequences are provided in the Appendix. Each protein was analyzed using the following programs: Pepwindow, garnier, Pfam and Prosite. Pepstats is part of the EMBOSS package (Rice *et al.* 2000) and calculates various protein statistics. Pepwindow is part of the EMBOSS package and was used to calculate protein hydropathy based on the Kyte-Doolittle parameters (Kyte and Doolittle, 1982). The hydrophobicity scale used is the same for every set of proteins and shows variation above and below 0, with 0 being neutral. Garnier is a secondary structure prediction tool incorporated into EMBOSS (Garnier *et al.*, 1978). Any secondary structure element longer than two residues was taken into consideration. Pfam (Finn *et al.* 2006) is a protein families database and contains Hidden Markov Models of each protein family. Hmmer (http://hmmer.janelia.org, as implemented in FunGIMS) was used to search protein sequences against the Pfam database (downloaded on 2008/05/8) with a 1e-03 cut-off value. Prosite (de Castro *et al.* 2006) is a database of patterns that identify proteins. The FunGIMS implementation of Prosite was used to scan each protein sequence.

3.3. Results and Discussion

Overall, the proteome annotation showed that the different subtypes within a serotype do not differ extensively yet local, protein specific or subtype-specific pattern changes were seen. Each set of protein sequences was submitted to the respective analysis methods. The results for each protein (L, VP1, etc.) were integrated to show any differences between the sequences (Figs. 3.3 - 3.13).

3.3.1. Pfam Results

The Pfam E-values of each protein is given in Table 3.1. The Pfam scan showed that all the proteins match the same Pfam family profile except in the case of the VP1 protein from SAT1/SAR/09/81. Upon closer inspection, it was seen that it matched the same

Pfam protein family as the other VP1 proteins but in this case it was above the cut-off of 1.0 e-03 (Table 3.1). Another interesting observation was that in VP3 (Fig. 3.6) the Pfam pattern had a far longer sequence length match in the SAT1/SAR/09/81, SAT1/KNP/196/91 and SAT3/BEC/29 subtypes. A similar situation was seen in VP1 (Fig. 3.4) where the SAT serotypes had the matching Pfam pattern split over two domains while the other serotypes had one domain match. A few proteins did not generate a match in the Pfam database. For protein 2A (Fig. 3.7) and 3B1-3 (Fig. 3.11) this is a result of their short length (about 20 amino acids in length) but for 2B (Fig. 3.8) and 3A (Fig. 3.10), each about 154 amino acids long, this is simply a matter of a lack of coverage in the Pfam database and a lack of general knowledge about the function of the protein in FMDV. The DUF1865 pattern match seen in VP4 (Fig. 3.7) is also a result of a lack of knowledge about the protein, but in this case it has already been assigned to a protein family of unknown function.

3.3.2. Prosite Results

As was to be expected, there were many Prosite hits due to certain amino acid patterns having a high probability of occurrence. Throughout most of the sequences the patterns appeared to be relatively conserved within serotypes e.g. the subtypes within SAT1 serotypes would have a certain pattern that differs slightly from the O subtypes (Figs. 3.3-3.6). It was decided not to exclude Prosite matches with a high probability of occurrence as these can provide clues to shifting patterns in the protein. There were a few interesting cases where patterns differed between proteins. The VP3 protein (Fig. 3.6) is an example of this. The VP3 protein varied from 221 to 222 amino acids in length for SAT1/3 and SAT2 isolates, respectively and with 58% overall variable aa positions. Most of the VP3 amino acid substitutions for SAT1, 2 and 3 were concentrated at four hypervariable regions, i.e. N-terminus (27-46), βB-βC loop (62-78), βE-βF loop (121-141) and βG-βH loop (165-183).

Certain matches are present in all the sequences (first two patterns) yet other patterns vary based on the genetic relatedness between the subtypes. In most of the proteins a definitive set of patterns was seen with small variations between the serotypes. An

Table 3.1: The Pfam pattern matches and E-values identified in each protein group. SAT1/KNP did not have a 3D sequence available.

		Pfam E-value								
Protein	Pfam Pattern	A24	A10	C3	O1/BFS	O/SAR	SAT1/SAR	SAT1/KNP	SAT2/ZIM	SAT3/BEC
L	Foot-and-mouth virus L-proteinase	2.2e-124	3.7e-128	8.7e-126	4.6e-130	2.2e-136	1.1e-129	1.1e-127	9.3e-128	7.7e-130
VP1	Picornavirus capsid protein	2.4e-26	4.1e-27	8.2e-25	3.7e-30	4.6e-23	Above cut-off	4.4e-05	2.9e-05	6.2e-08
VP2	Picornavirus capsid protein	4.2e-56	1.4e-56	6.2e-58	4.5e-56	1.6e-55	1.2e-42	1.3e-41	1.3e-43	8.4e-42
VP3	Picornavirus capsid protein	3.8e-41	8.9e-44	5.3e-33	3.5e-38	6.6e-38	3.3e-21	4.2e-21	1.7e-21	9.2e-25
VP4	Domain of unknown function (DUF1865)	8.8e-62	8.8e-62	3.6e-62	3.6e-62	3.6e-62	3.4e-61	3.4e-61	1.2e-60	8.5e-62
2A	None	-	-	-	-	-	-	-	-	-
2B	None	-	-	-	-	-	-	-	-	-
2C	RNA helicase	4.4e-23	4.4e-23	4.4e-23	4.4e-23	4.4e-23	7.3e-23	7.3e-23	4.4e-23	7.3e-23
3A	None	-	-	-	-	-	-	-	-	-
3B1	None	-	-	-	-	-	-	-	-	-
3B2	None	-	-	-	-	-	-	-	-	-
3B3	None	-	-	-	-	-	-	-	-	-
3C	3C cysteine protease (picornain 3C)	1.1e-80	4.8e-80	1.9e-79	2.8e-81	2.3e-79	1.5e-67	8e-69	8e-69	8.8e-68
3D	RNA dependent RNA polymerase	2.9e-162	1.4e-163	2.3e-162	1.4e-162	2.1e-161	9.2e-157	N/A	1.4e-155	2.4e-156

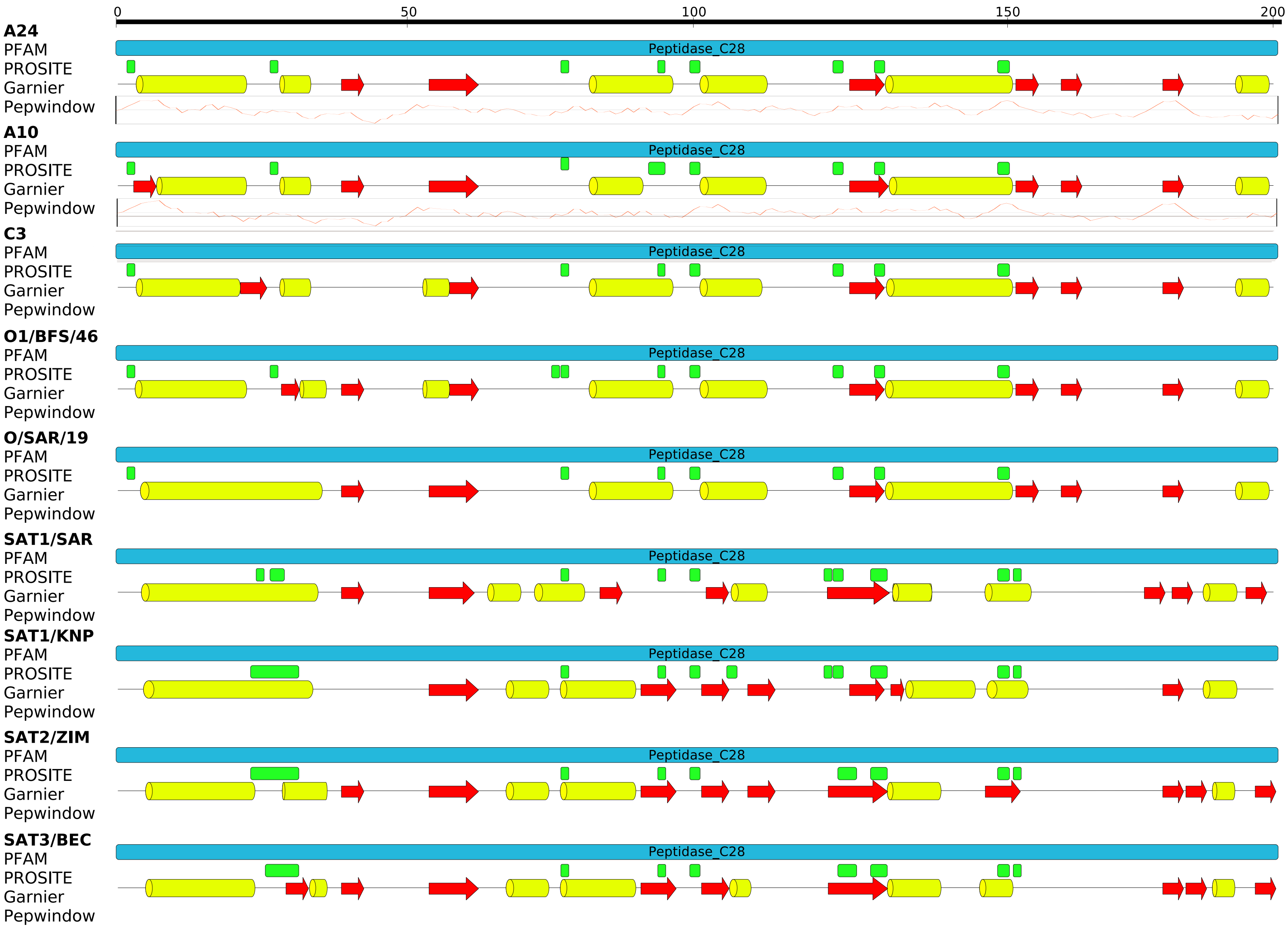


Figure 3.3: The annotation of protein L. α-helices are represented by cylinders and β-strands by red arrows.

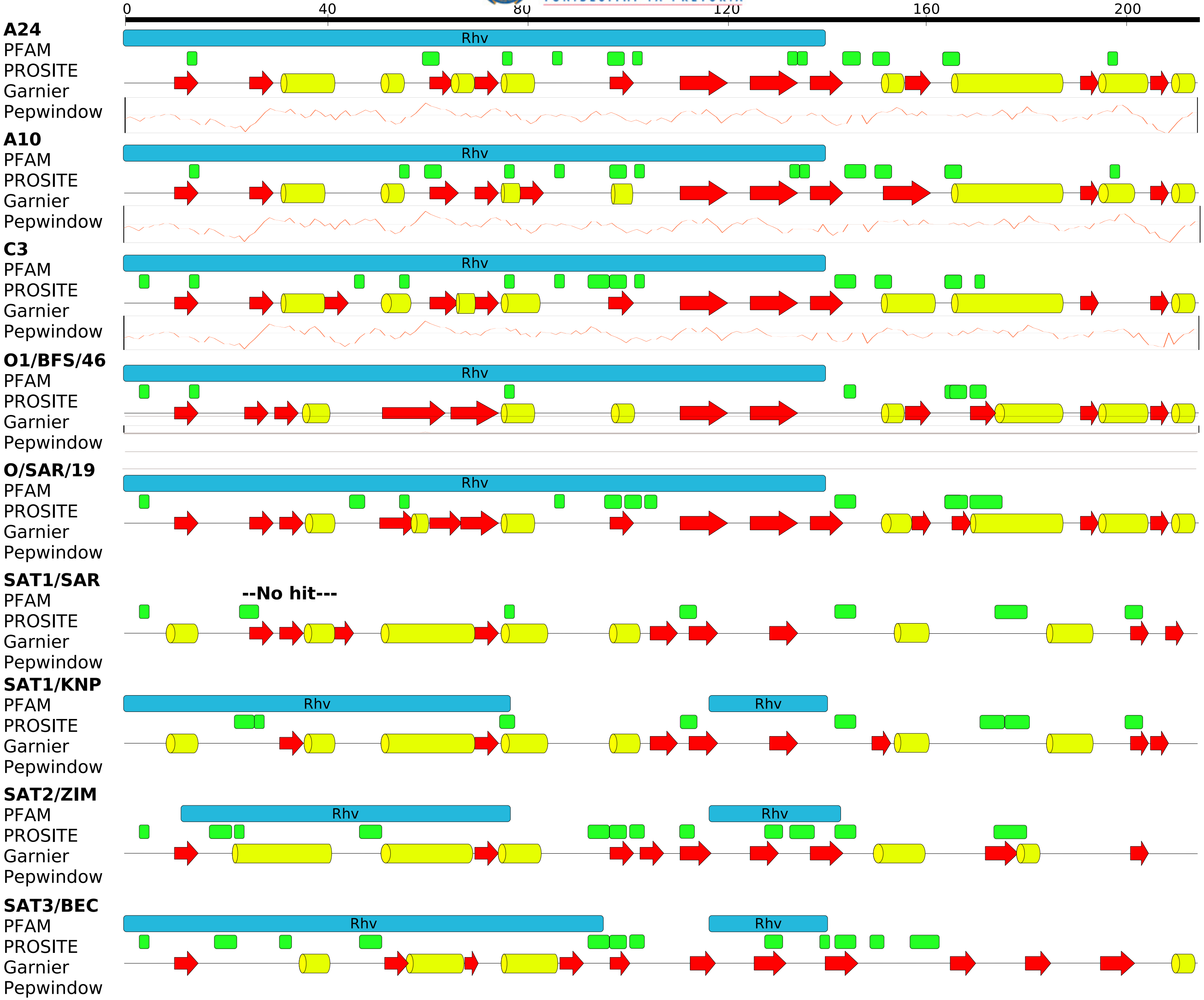


Figure 3.4: The annotation of protein VP1. α-helices are represented by cylinders and β-strands by red arrows.

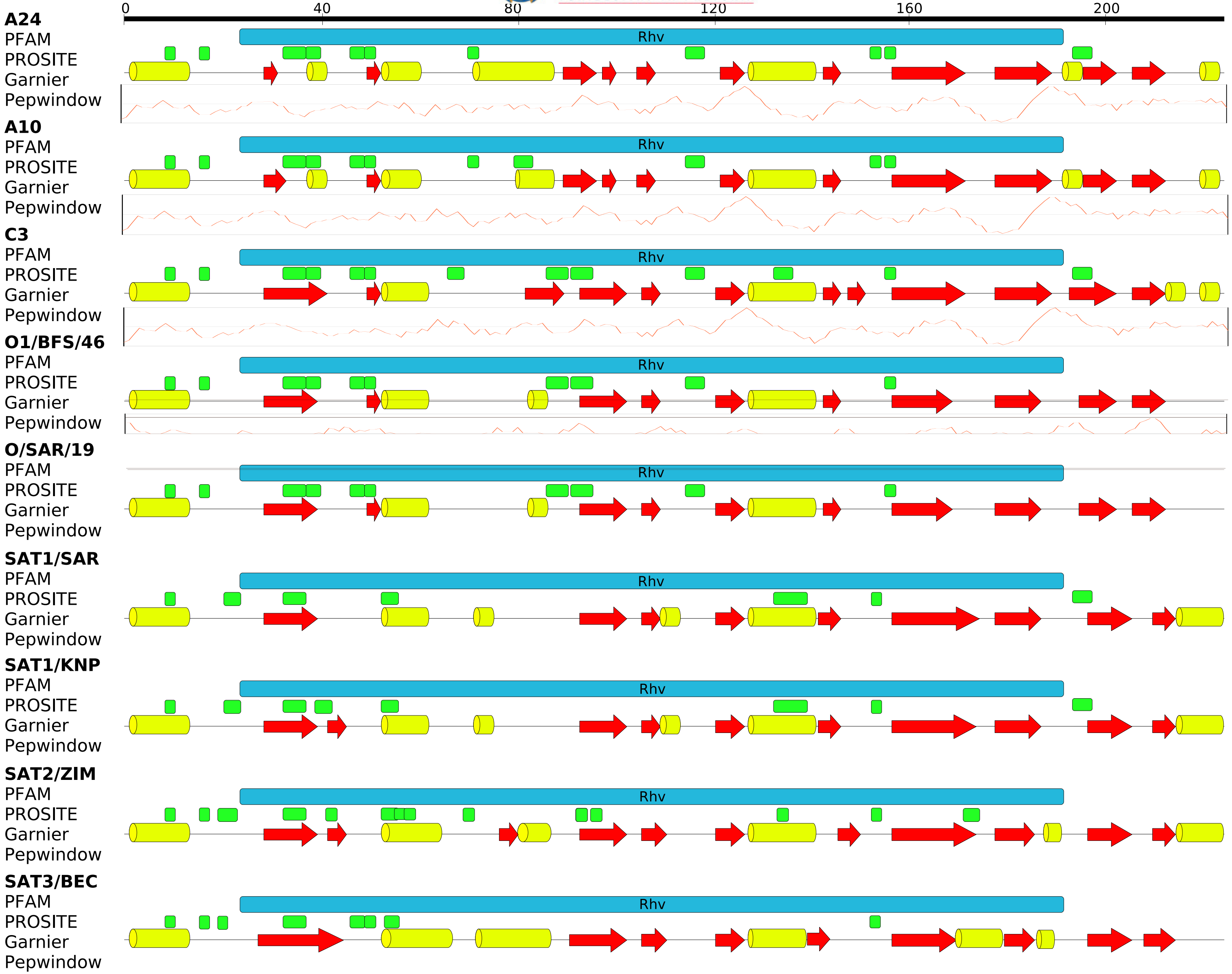


Figure 3.5: The annotation of protein VP2. α-helices are represented by cylinders and β-strands by red arrows.

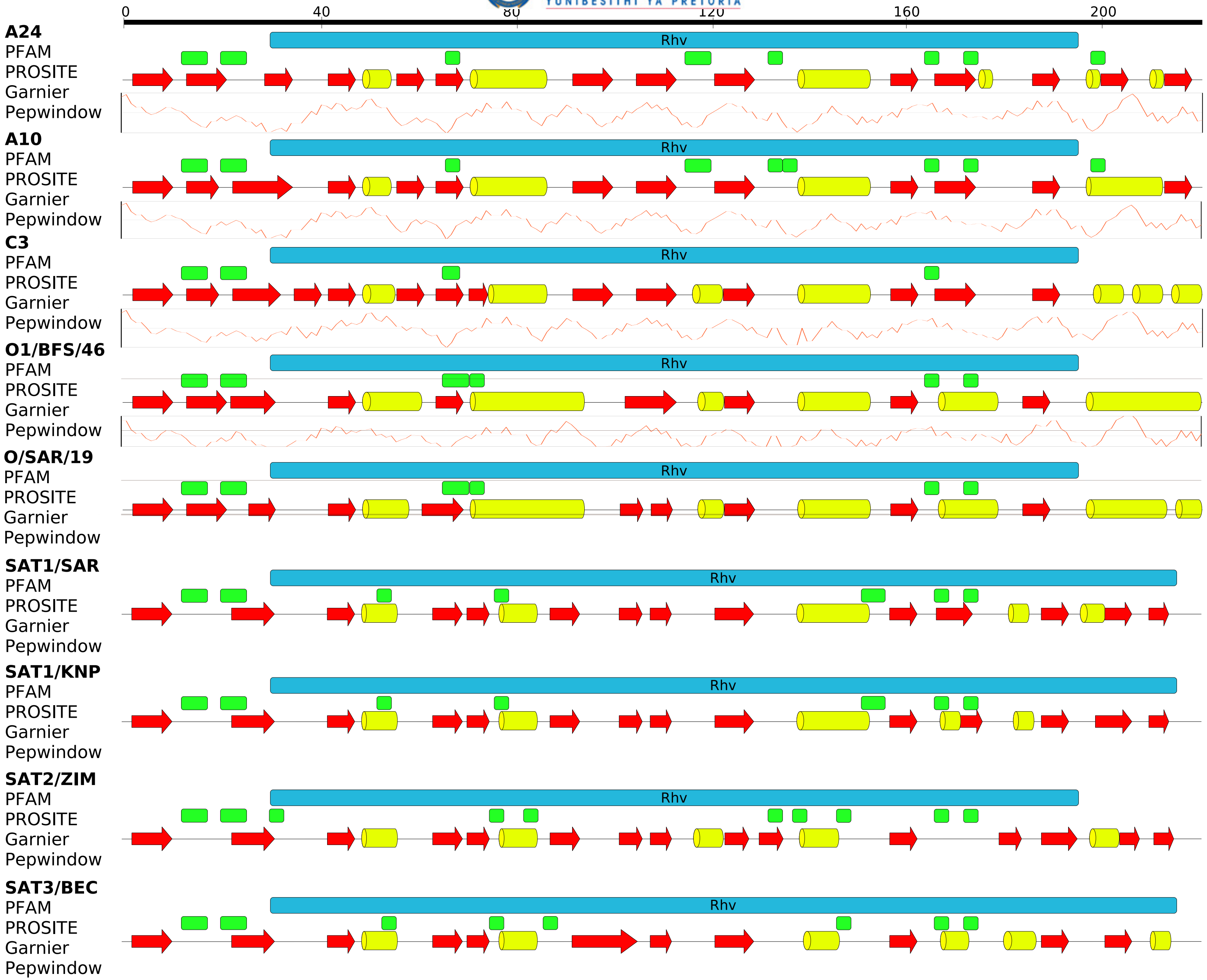


Figure 3.6: The annotation of protein VP3. α-helices are represented by cylinders and β-strands by red arrows.

example of this pattern conservation among subtypes can be seen in protein 2C (Fig. 3.9) where all the SAT serotypes share the same pattern. The SAT serotypes have an additional Prosite pattern match at the beginning and end of the sequence, which is not seen in the other serotypes analyzed. A clear pattern across all the proteins was seen for the SAT serotypes that confirms the close genetic relationship between the SAT1-3 non-structural protein coding regions. In most cases such as VP4 (Fig. 3.7) the SAT serotype displayed similar Prosite pattern hits that differ from the other serotypes. All the proteins showed a number of matches to many short patterns (3-6 residues in length) but in 2C a long pattern was found (Fig. 3.9). This pattern corresponds with the "Superfamily 3 helicase of positive ssRNA viruses domain profile". Another long pattern was found in the 3D protein (Fig. 3.13). A match to "RdRp of positive ssRNA viruses catalytic domain profile" was found, which is a RNA dependant RNA polymerase. A possible reason for these two long matches are the conserved nature of the proteins that are encoded by 2C and 3D. These proteins cannot accommodate many changes because of structural constraints and thus make it easier to construct a pattern match with a longer length.

3.3.3. Secondary Structure Results

The secondary structure prediction results showed that secondary structure is well conserved among the proteins but not as high as was expected. It was expected that the method would predict the same secondary structure for each sequence in a set, yet there were differences. This is possibly due to the method used, which is sequence-based. In most of the proteins the predicted secondary structure patterns stayed the same. In a few cases it was seen that an α-helix was split into two helices in another serotype as in the case of protein 2B (Fig. 3.8) or that an α-helix in one sequence is predicted to be a β-strand in another sequence (Fig. 3.3). Carillo and co-workers (Carrillo *et al.*, 2005) mention that a transmembrane region has been identified from position 120-140 but a transmembrane prediction using the Structural module showed no evidence of a transmembrane helix. However, hydrophobicity plots showed that the area from residue 120-140 is hydrophobic and may thus be associated with the membrane. A fact that must be kept in mind is that secondary structure prediction is a sequence-based method and thus a one residue difference, such as a proline in the middle of a α- helix, may influence the algorithm and cause it to predict two separate helices instead of a longer, bent α-helix. This is also the possible cause of secondary structure being predicted as a α-helix in one serotype but in another serotype the same region is predicted to be a β-strand as seen in a comparison of 3B2 (Fig. 3.11). Carillo and co-workers reported on variation in three hypervariable regions in 3D (aa 1-12, 64-76 and 143-153, George *et al.*, 2001, Carrillo *et al.*, 2005). These areas were found to have a low variability in the proteomes examined here. This is reflected in the secondary structure predictions that predict the same structure for these areas in all the proteomes examined (Fig. 3.13). The Prosite patterns for the last two hypervariable regions are also the same, thus indicating low variation. The amino acid and Prosite pattern variation observed for the VP3 protein was also reflected in the secondary structure prediction. Similarly, VP1, the most variable of the outer capsid proteins, showed more variation in the secondary structure prediction. The VP1 protein varied in length from 213-214 aa for SAT2, 219 aa for SAT1 and 215-217 for SAT3 with 71% overall variable amino acid positions.

It must be kept in mind that the secondary structure predictions done here was to detect patterns in the sequences and not to get residue specific accurate predictions. There is currently no tool available which does such an accurate prediction of secondary structures. Moreover the sequences used here included local strains which have not been crystallized and thus no 3D data could be used to validate predictions. Main features such as a long α-helix or a sequence of helices or sheets seem to be conserved among the sequences, but short helices and strands seem to be conserved only among closely related serotypes. The results from the Garnier predictions showed that overall secondary structure patterns can be detected by the predictions, and predictions that differ across similar sequences must be investigated with further methods (either using structures or more advanced methods such HMMSTR (Bystroff *et al.*, 2000). Crystal structure data were not used in this section as the focus was on detecting pattern similarities/differences between the various strains.

3.3.4. Pepstat Hydrophobic Plot Results

The hydrophobicity plots for each set of sequences were kept on the same scale to allow comparison between plots. Each graph shown in Figures 3.3 to 3.13 have positive values indicating hydrophobicity and negative values indicating hydrophilicity below the line. The hydrophobicity plots showed, in contrast to the secondary structure predictions, that hydrophobicity remains mostly constant even though the sequence changes. Whereas the Garnier predictions made different predictions for a section based on the residues, the hydrophobicity plot was still the same indicating that there was some measure of structural integrity being maintained in spite of sequence differences. This was especially evident with the $3C^{pro}$ (Fig. 3.12). O1/BFS/46 VP2 (Fig. 3.5) showed one of the biggest shifts in hydrophobicity around residue 180. Whereas all the other sequences have a relatively hydrophilic stretch of residues, O1/BFS/46 appears to be very neutral in that region. This area was predicted to contain a β-strand by Garnier in all the sequences and may thus indicate a buried β-strand that can afford to be less hydrophilic. An interesting feature was also seen at the beginning (around residue 20) of the SAT VP3 sequences (Fig. 3.6). All the SAT serotypes are very hydrophilic at the start of the sequence, while the other serotypes show a slight increase in hydrophobicity in the same area. The SAT serotypes showed very similar hydrophobic plots as were seen for the secondary structure predictions and the Prosite pattern matches. This provides support for a possible ancestral sequence from which the SAT serotypes emerged.

An interesting feature was seen in VP2 (Fig. 3.5). Residues 30-40 were predicted to be a β-strand in C3, O1/BFS/46, O/SAR/19/2000 and SAT1-3 but in the A serotypes it was predicted to a be short β-strand and a short α-helix. Whereas the hydrophobic plots for the rest of the proteins in VP2 are the same, this area has a different plot for each serotype. A24 and A10 start out neutral from residues 30-35 and then turn fairly hydrophilic from residues 35-40. C3's plot is relatively neutral. O1/BFS/46 and O/SAR10/2000 differ. In O1/BFS/46 residues 30-40 is hydrophilic over most of the region whereas in O/SAR/19/2000 the region is far more neutral. The two SAT1 subtypes show the same pattern but the plots for SAT2/ZIM/07/83 and SAT3/BEC/29 appear more neutral for the area. The SAT serotypes all start out with a hydrophobic area from residues 30-35 but then differ slightly from residues 35-40. Despite this difference the same Prosite pattern is conserved among all the sequences around position 35.

O1/BFS/46 shows another difference with the rest of the VP2 sequences. Overall VP2 from O1/BFS/46 is very neutral. If the hydrophobic plots are compared with the other sequences, it can be seen that O1/BFS/46 has none of the major hydrophobic plot spikes as seen at residues 120-130 and 185-195 in the other sequences. However O1/BFS/46 appears to have a unique hydrophobic area from residues 200-210 which is not seen in other subtypes.

The VP2 protein varied from 219 amino acids for SAT1 and SAT2 viruses and 218 amino acids for SAT3 viruses (52% overall variation within VP2) and the conserved N-terminal motif described by Carrillo *et al.* (2005) was supported in an alignment of SAT VP2 aa sequences, i.e. DKKTEETTLLEDRI(L/M/V)TT(S/R)H(G/N)TTT(S/T)TTQSSVG. In a structural model of the SAT type viruses this motif is located internally in the virion suggesting structural or functional constrains on this sequence and was recently mapped as a serotype-independent epitope (Filgueira *et al.*, 2000). Within the VP2 protein four hypervariable sites were identified, i.e. βA-βB loop (aa positions 31-44), βB-βC loop (aa 62-81), βC-βD loop (aa 91-101) and βE-βF loop (130-134/140 for SAT1 and 2, respectively).

3.4. Conclusion

Some authors have noted how variation in proteins such as L and 3A influence virulence and host range (Carrillo *et al.*, 2005; Mason *et al.*, 2003*a*). When looking at the annotation results, a clear picture emerges. There is variation, not only on a residue level, but also on a higher structural and potentially at a regulatory level, in almost all the proteins in the FMDV proteome. The main task now is to separate relevant and irrelevant variation. In this section global changes were looked at. Patterns such as Pfam only give a general idea of the function of the protein and thus are not as highly informative when looking at lower level differences. Lower level differences become obvious when Prosite patterns are looked at. As can be seen in the annotation results, some serotypes can be

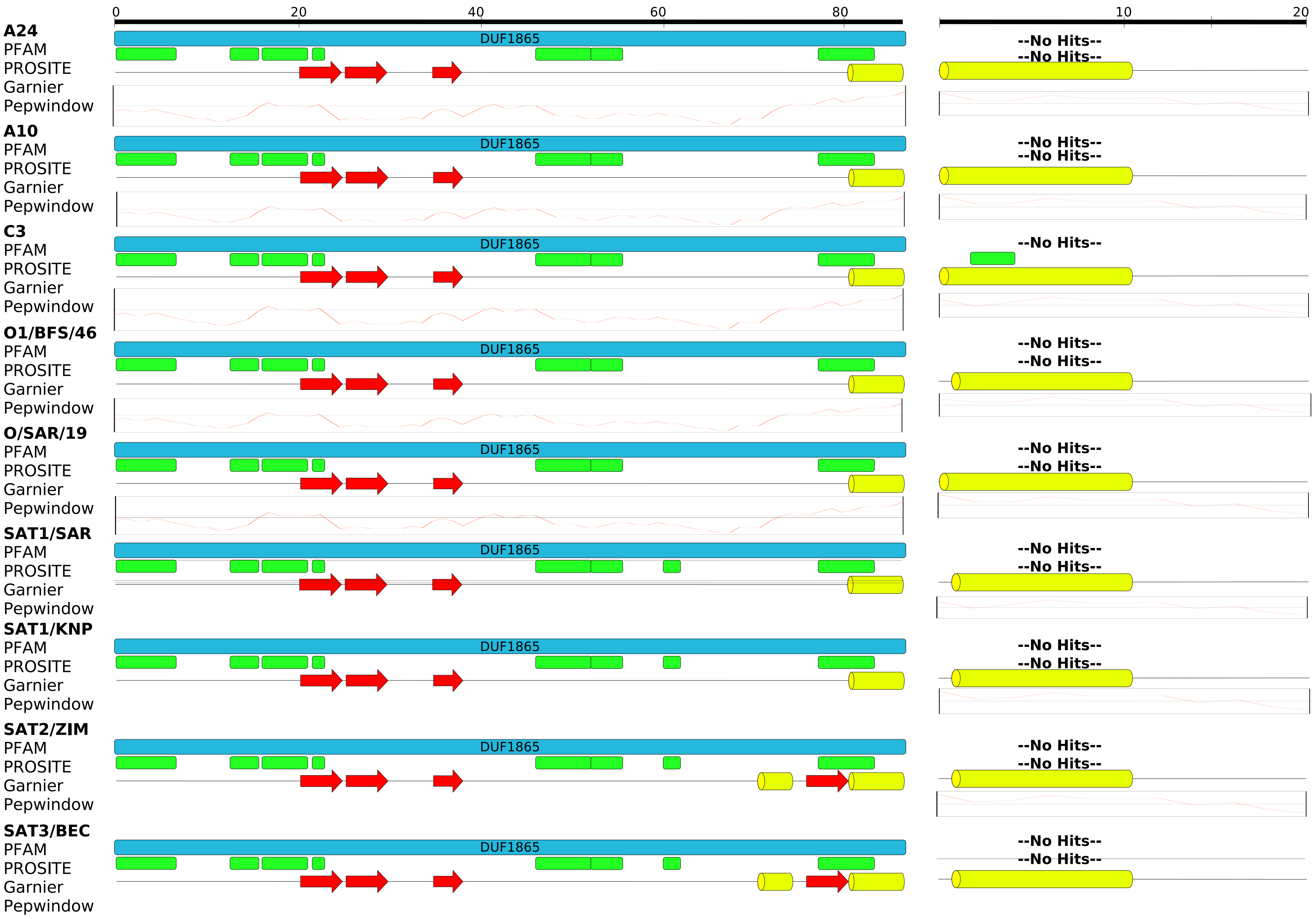


Figure 3.7: The annotation of protein VP4 (left) and 2A (right). α-helices are represented by cylinders and β-strands by red arrows.

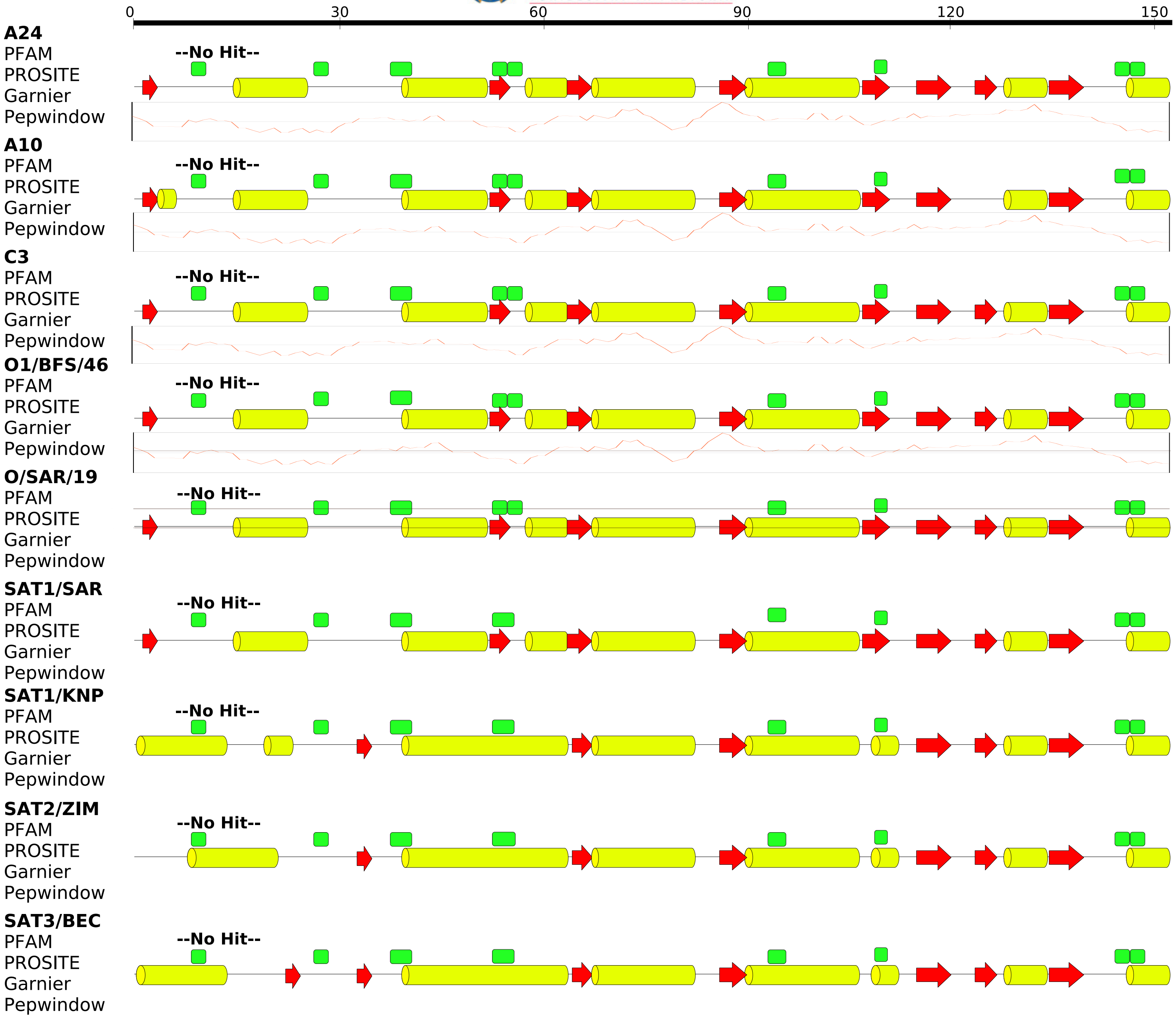


Figure 3.8: The annotation of protein 2B. α-helices are represented by cylinders and β-strands by red arrows.

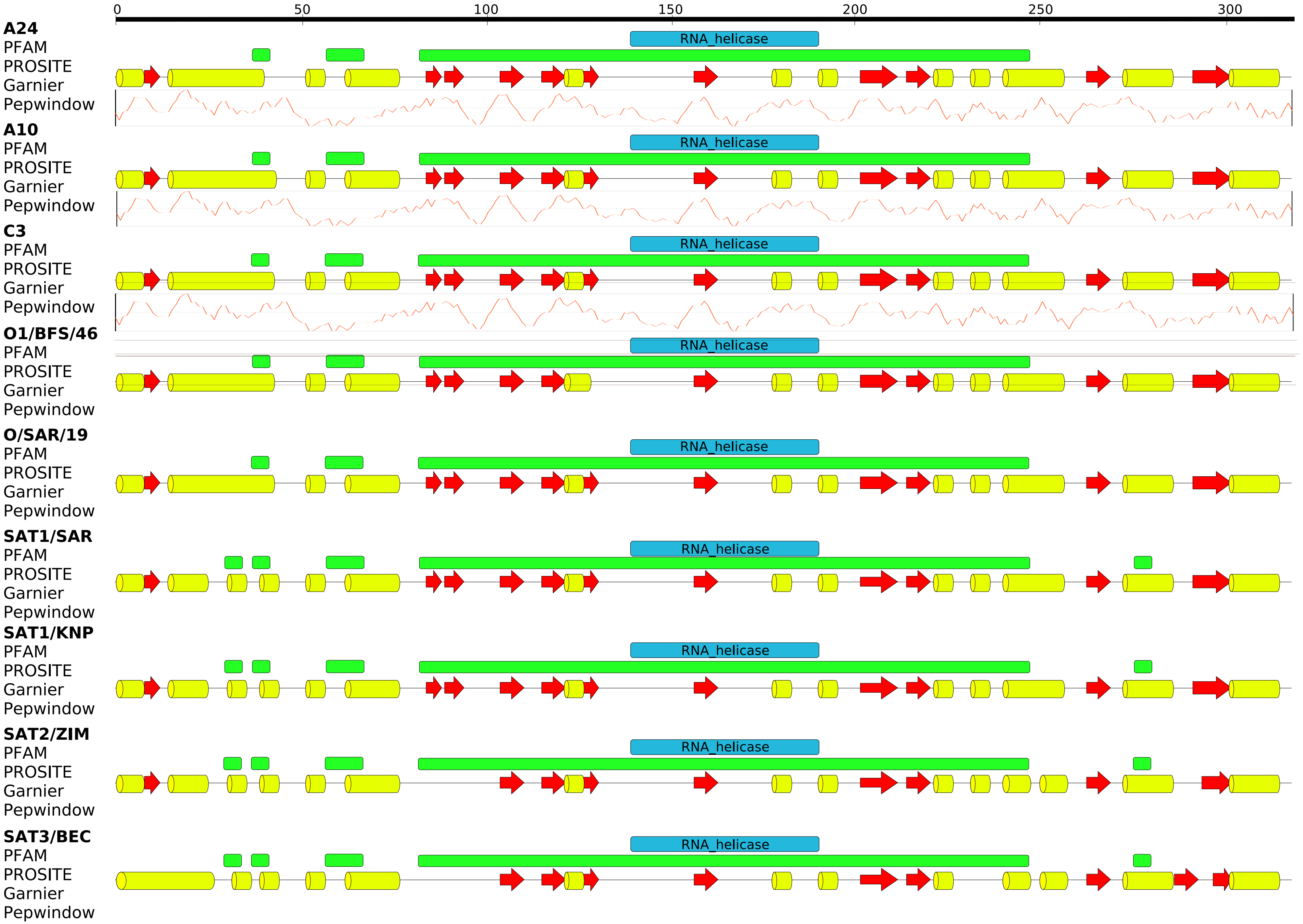


Figure 3.9: The annotation of protein 2C. α-helices are represented by cylinders and β-strands by red arrows.

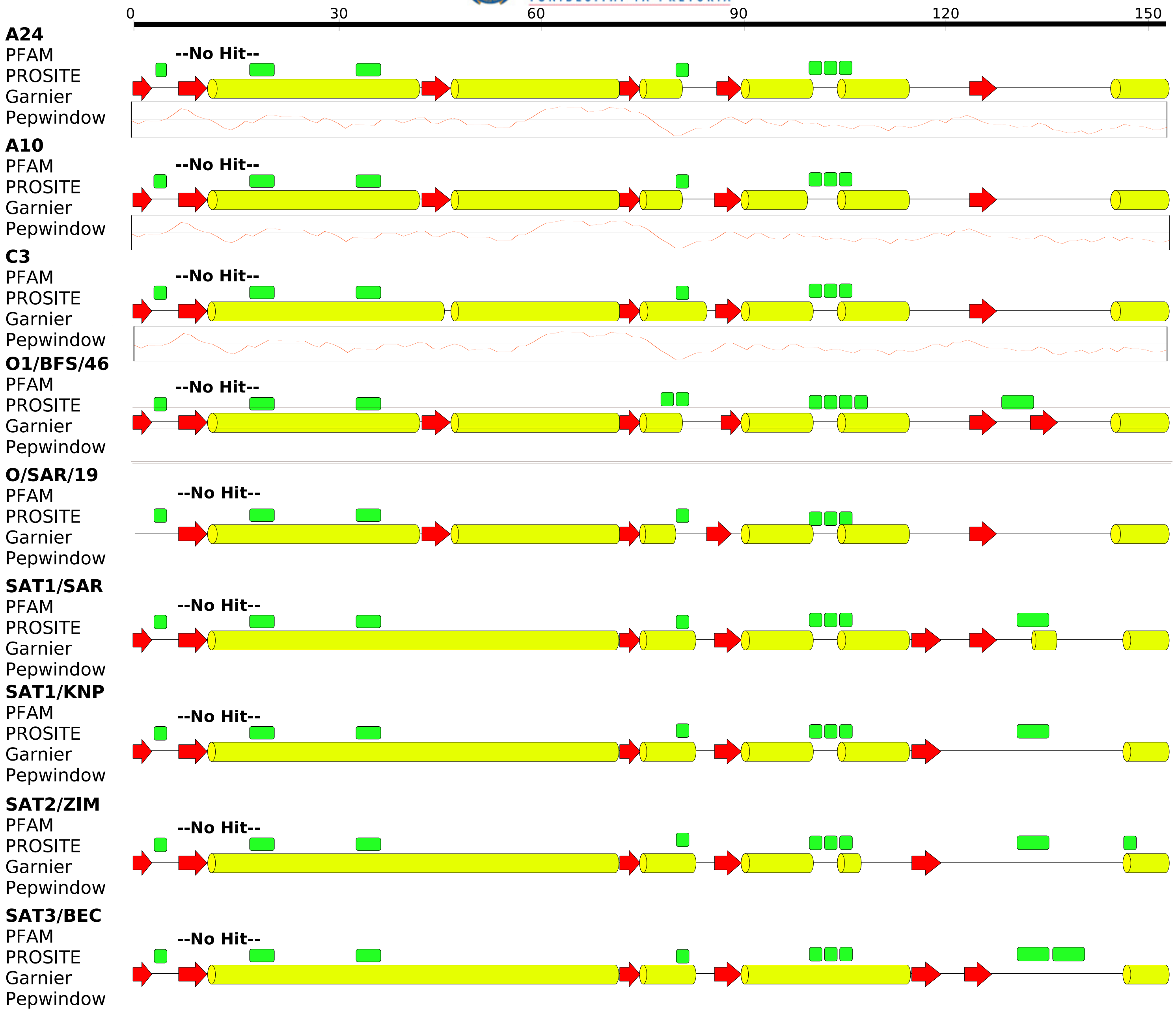


Figure 3.10: The annotation of protein 3A. α-helices are represented by cylinders and β-strands by red arrows.

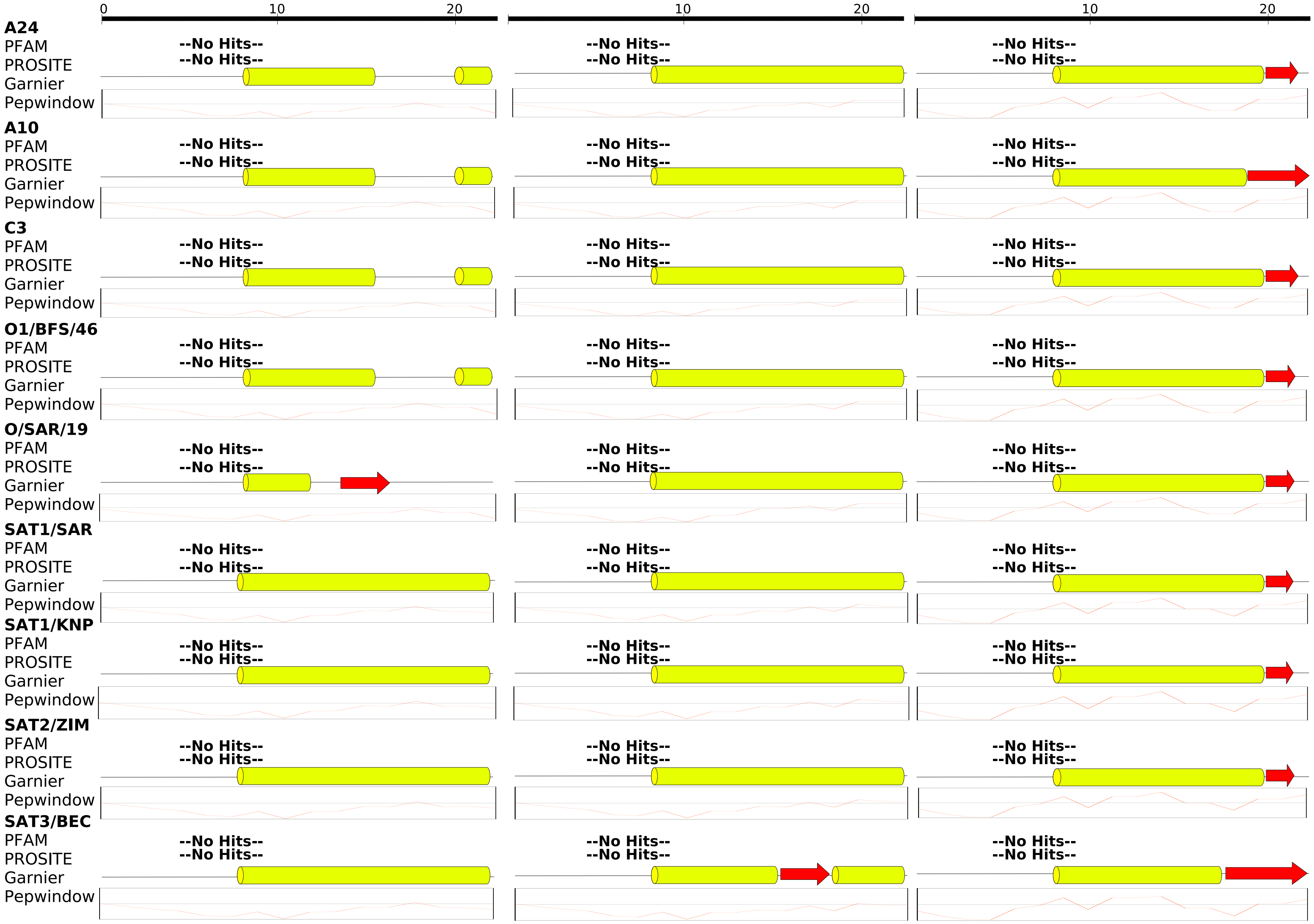


Figure 3.11: The annotation of protein 3B. Left: 3B1; middle: 3B2; right: 3B3. α-helices are represented by cylinders and β-strands by red arrows.

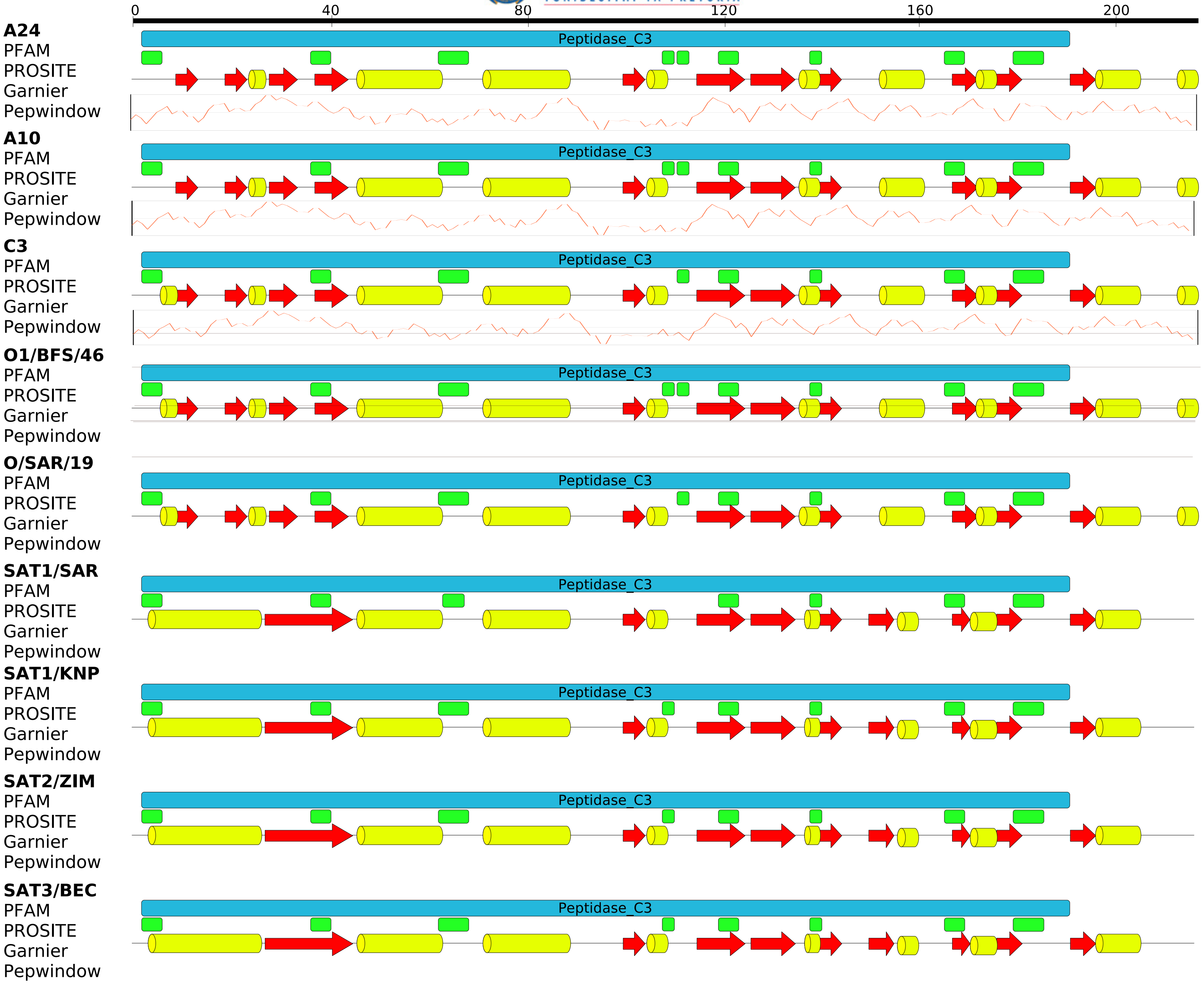


Figure 3.12: The annotation of protein 3C. α-helices are represented by cylinders and β-strands by red arrows.

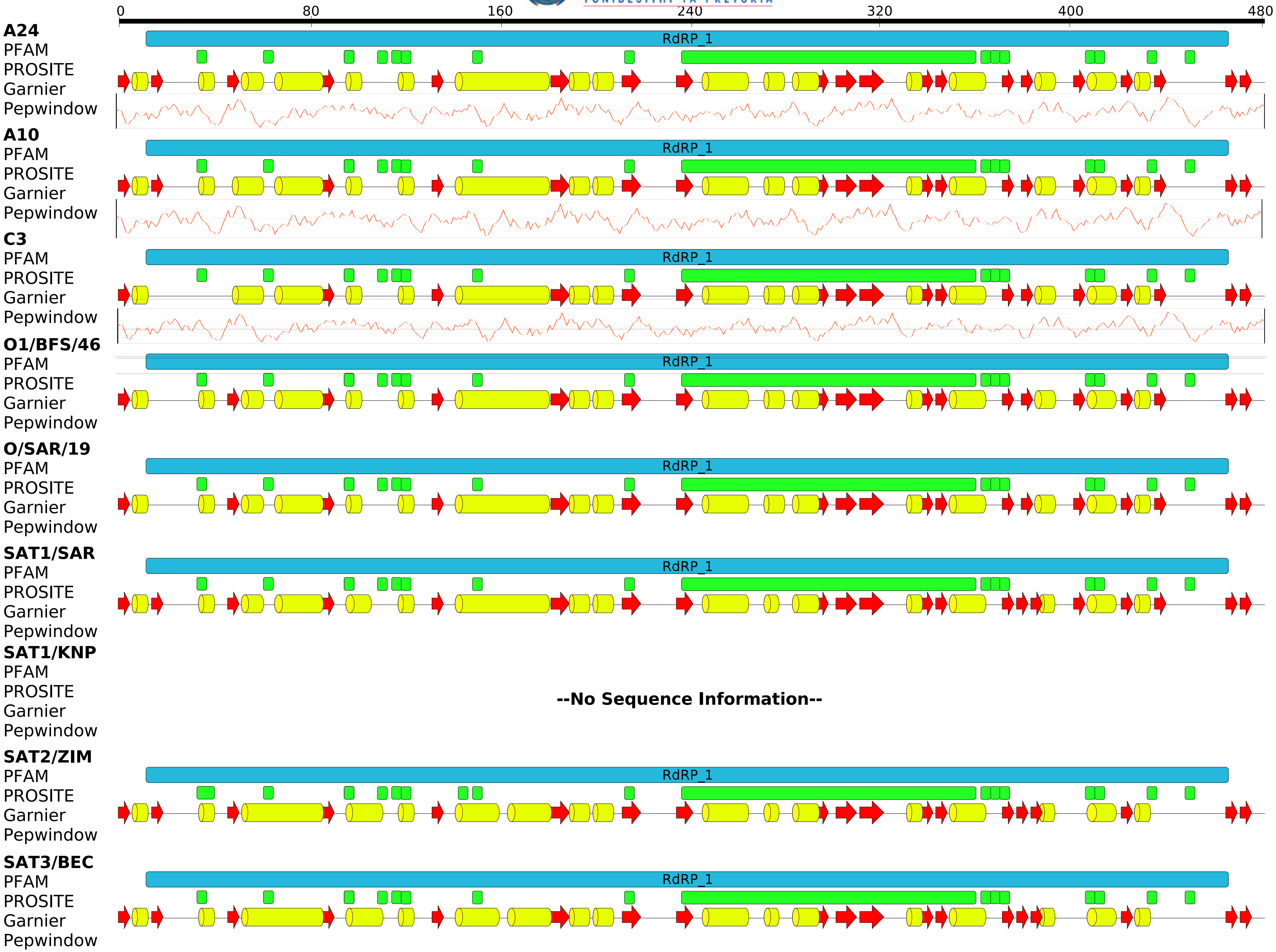


Figure 3.13: The annotation of protein 3D. α-helices are represented by cylinders and β-strands by red arrows.

grouped together on the basis of their distribution of Prosite patterns. The host uses some of these patterns for regulation and changes in the patterns may have an effect on the way the viral proteins function, on their activity or on protein-protein interactions. Changes in hydrophobicity patterns also affect the strength and affinity with which a protein, such as 2A and 3A, associates with the ER vesicle membranes and thus their duration of influence over RNA replication. A combination of all these factors may explain some of the differences seen in the host range, virulence and possibly even the spreading of the virus. The best approach to investigate these differences would be to make chimeras that contain conserved patterns found in every protein and thus determine which parts affects virus infection, translation and replication. A large-scale study involving all the sequences known for FMDV using the proteome annotation approach may yield valuable results, especially when coupled with epidemiology information such as virulence.

An important practical application of the proteome annotation is with regards to the substitution of structural proteins in the production of recombinant, chimeric viruses. The question becomes how much of the structural protein coding regions can be exchanged between serotypes in order to conserve structural constraints but be able to transfer the antigenic determinants to allow protection in the host animal. Previously it was shown that viable FMDV chimeras can be produced containing the complete or portions of the capsid coding sequence of different FMDV serotypes (Rieder *et al.*, 1994; Almeida *et al.*, 1998; van Rensburg and Mason, 2002). For example, the replacement of the pSAT2 (SAT2/ZIM/7/83) outer capsid sequences by those of A12 or SAT1/NAM/307/98 virus, rendered the resulting virus viable and stable during successive passages in BHK-21 cells (van Rensburg *et al.*, 2004; Storey *et al.*, 2007). The capsid and other sequences of the genome can be readily exchanged between serotypes and still rendered the chimeric viruses viable during successive passage *in vitro* (Almeida *et al.*, 1998; van Rensburg *et al.*, 2004; Storey *et al.*, 2007), implicating some pliability/versatility outside residues essential in the structural constraints of the virus particle. We have utilized the chimera technology in the development of recombinant FMDV vaccines specific for certain geographic locations. The virion stability, *in vitro* immunological profiles against a panel of reference sera and the receptor preferences were successfully transferred from the parental field viruses to the chimeras with the substitution of the VP1, VP2 and VP3 coding regions (Blignaut *et al.*, unpublished; Maree *et al.*, unpublished). In addition, a chimera containing the outer capsid coding region of a SAT1 virus, KNP/196/91, in the genetic background of a SAT2 virus, ZIM/7/83, protected pigs against homologous KNP/196/91 challenge (Blignaut *et al.*, unpublished).

From the proteome analysis of the capsid-coding region it became clear that the structural proteins function as a unit, a fact that is supported by numerous recombinational studies. In these studies it was found that recombination rarely occured within the structural protein coding region, that breakpoint hotspots were detected at the 1A/1B and 1D/2AB boundaries and that hot spots on either side of the structural protein coding region function as a breakpoint pair (Jackson *et al.*, 2007; Heath *et al.*, 2006; Simmonds, 2006). Both the infrequency of recombination events within the structural protein coding region and the unique secondary structure prediction and hydrophobicity profiles in this study suggest that there are severe functional constraints limiting the exchange of structural protein coding regions between divergent parental viruses. This is mostly due to interaction patterns (hydrophobic as well as electrostatic) between the different proteins in the capsid. We predict that substitution of the VP2, VP3 and VP1-2A as a complete unit may allow the best success for recovery of viable viruses in the chimera vaccine technology. The work done here a starting point for the local researchers to start comparing phenotypic traits with patterns seen on the genomes of the various local SAT strains as well as assess how these strains compare with other serotypes.

Chapter 4 will deal with a more in-depth analysis of variation in FMDV 3C and 3D and their effect on the protein structure.

Chapter 4

Modelling of Foot and Mouth Disease Virus 3C and 3D Non-structural Proteins

4.1. Introduction

One of the most important proteases in FMDV is the 3C^{pro} and its 3C^{pro}-containing precursor, 3CD. 3C^{pro} is responsible for viral polyprotein cleavage as well as some cleavage of cellular proteins such as eIF4G. The 3C^{pro} has been shown to efficiently process ten of the thirteen cleavage sites in the FMDV polyprotein (Bablanian and Grubman, 1993). 3C^{pro} is important in virus production as it cleaves the single translated polyprotein into the mature viral proteins needed for virus replication. The specificity of FMDV 3C^{pro} differs from its homologue in other picornaviruses like the Poliovirus. In polio 3C^{pro} only cleaves between Gln-Gly sites whereas in FMDV cleavage can occur between multiple dipeptides such as Gln-Gly, Glu-Gly, Gln-Leu and Glu-Ser (Palmenberg, 1990; Birtley *et al.*, 2005). Evolutionary studies have shown that the 3C^{pro} belongs to the trypsin family of Ser proteinases (Bablanian and Grubman, 1993). This is supported by the 3C^{pro} structure from FMDV, which shows a chymotrypsin-like fold (Fig. 4.1) and possesses a Cys-His-Asp catalytic triad in the active site (Birtley *et al.*, 2005). This chymotrypsin-like fold consists of two β-barrels positioned against one another with the active site between the two β-barrels. In FMDV an anti-parallel β-ribbon covers the active site. Sweeney and co-workers (Sweeney *et al.*, 2007) postulated that the β-ribbon is involved in substrate recognition. The β-ribbon is stabilized via hydrophobic contacts with the N-terminal barrel. The N-terminal barrel also contains an invariant region (residues 76-91) with

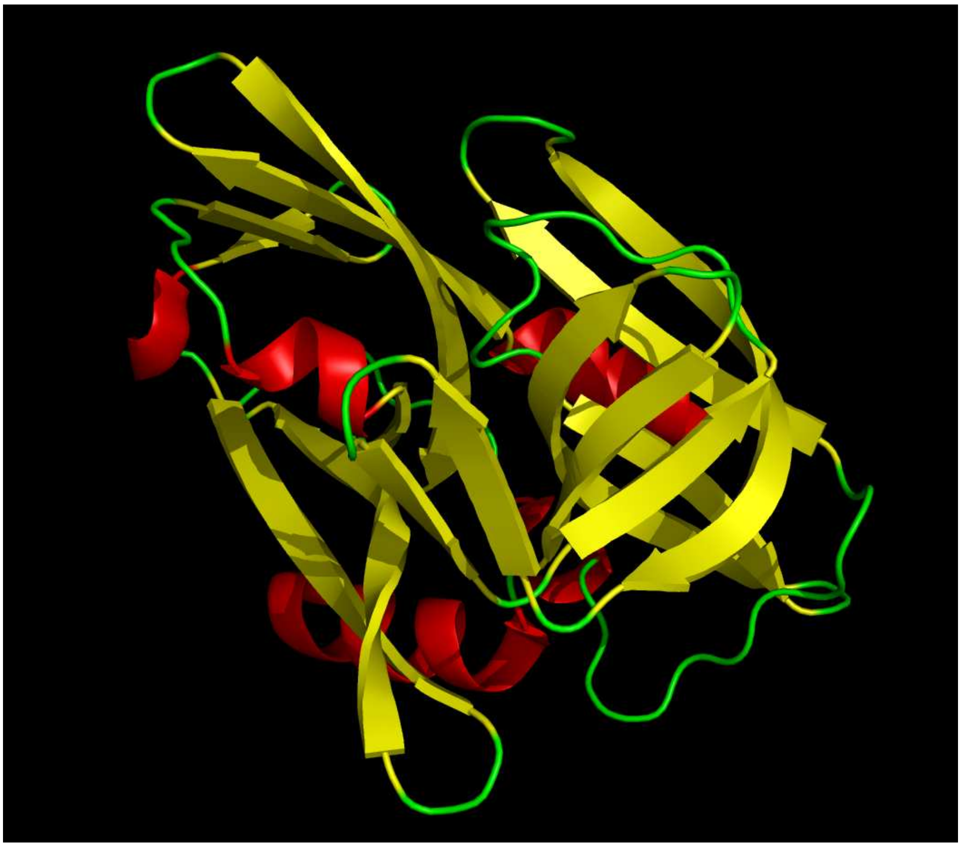


Figure 4.1: The structure of $3C^{pro}$ from FMDV serotype A (Sweeney *et al.*, 2007). Helices coloured red, strands coloured yellow. The β-ribbon can be seen in the foreground covering the active site.

the Asp at position 84 forming part of the catalytic triad (Carrillo *et al.*, 2005). The β-ribbon is quite flexible and very similar to other 14-residue β-ribbons that occur in other bacterial and viral serine proteases (Sweeney *et al.*, 2007). Most of the differences between the different β-ribbons occur neighbouring the turn in the ribbon and all the ribbons seem to be stabilized at the bottom of the ribbon via hydrophobic interactions.

The precursor, $3CD^{pro}$, has some protease activity and also participates in ribonucleoprotein complexes and influences RNA replication and translation by binding to RNA.

The $3D^{pol}$ protein that is produced from the cleavage of 3CD is a RNA dependant RNA polymerase encoded by the viral genome. The $3D^{pol}$ sequence (both RNA and protein) is conserved between the different sub- and serotypes (George *et al.*, 2001). $3D^{pol}$ is responsible for, in collaboration with host proteins, elongation of the nascent RNA chains during replication. The structure of FMDV $3D^{pol}$ is very similar to that of the poliovirus

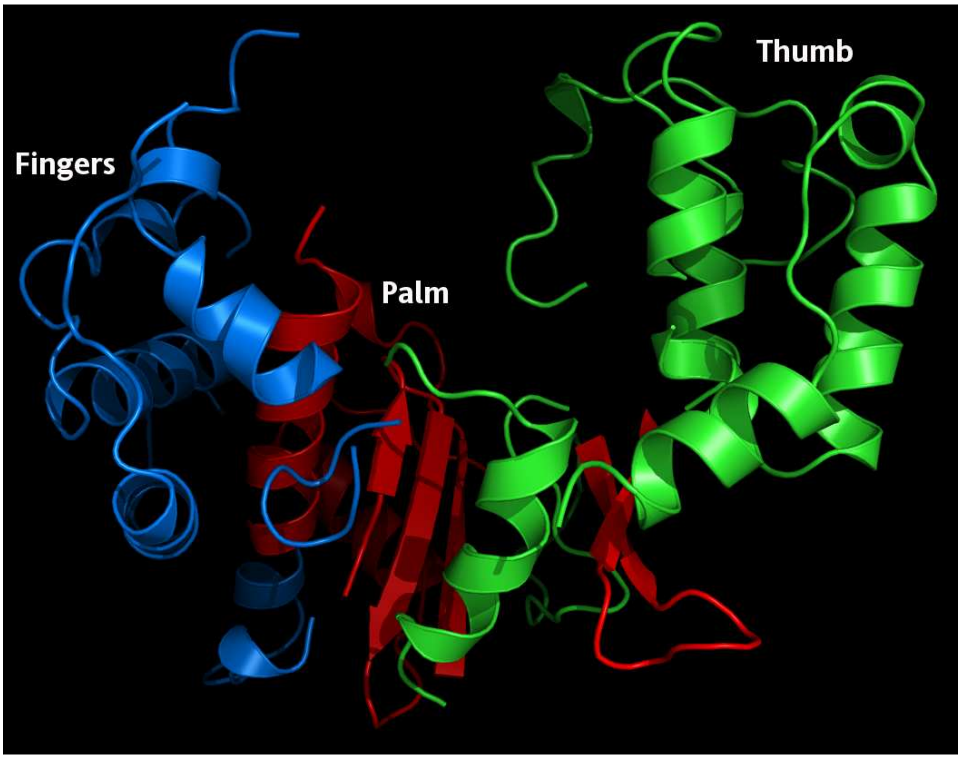


Figure 4.2: The structure of $3D^{pol}$ from the Polio virus (1RDR). Notice the 'palm' (red), 'fingers' (blue) and 'thumb' (green) subdomains (Hansen *et al.*, 1997).

$3D^{pol}$. This structure consists of a 'right-hand' polymerase consisting of 'palm', 'fingers' and 'thumb' subdomains (Fig. 4.2). It contains 17 α-helices and 16 β-strands. The palm subdomain contains some of the most highly conserved features known in all polymerases (Ferrer-Orta *et al.*, 2004). There are five conserved regions designated A-E, which are involved in phosphoryl transfer, nucleotide binding, nucleotide priming and structural integrity. A site in Motif A (Asp240 and Asp 245 in β8) helps motif C with metal ion binding as observed in the 1U09 structure. Motif B is made up of helix α11 that associates with a central β-sheet (β8, β11 and β12). Motif C, consisting of β11-turn-β12, contains the acidic sequence GDD (Gly 337-Asp338-Asp339). This acidic area is almost universally conserved and functions as a metal ion binding site during the nucleotide transfer reaction. Helix α12 forms motif D and β14 and β15 forms motif E. These motifs interact together to form the polymerase catalytic site.

Various studies have indicated the highly conserved nature of 3C and 3D (George *et al.*, 2001, Gorbalenya *et al.*, 1989, Carrillo *et al.*, 2005). In this section, the variation found in these two proteins of the South African Territories serotypes of FMDV, will be presented. The objective is to identify local variation hotspots within the two proteins. This analysis may also help to identify the 3C-3D interaction site by identifying the most conserved residues based on the structure. Highly conserved patches on the surface may indicate areas that need to be conserved for interaction between 3C and 3D.

4.2. Methods

4.2.1. 3C Protease

Dr. F. Maree (Agricultural Research Council) supplied 21 SAT1, 21 SAT2 and 9 SAT3 sequences (Table 4.1). Alignment was done with ClustalX (Thompson *et al.*, 1997) and due to the high identity the parameters were kept at the default settings. The modelling scripts were generated with the Structural module in FunGIMS and modelling done with Modeller 9v1(Fiser and Sali, 2003) including a fast model refinement step. Models of representative sequences of serotypes SAT1, SAT2 and SAT3 were built based on 2J92 (Sweeney *et al.*, 2007), which is an serotype A virus. For SAT1, KNP/196/91/1 was used with the first five and the last 6 residues removed, for SAT2, ZIM/7/83/2 was used with the first and the last 6 residues removed and for SAT3, KNP/10/90/3 was used with the first and last 6 residues removed. The start and end residues were removed due to no template match for those regions. Another possible template was found (2BHG) but it was decided to use 2J92 as an important loop was crystallized in 2J92 that is not present in the higher resolution of 2BHG (1.90 Å vs 2.20 Å).

4.2.2. 3D RNA Polymerase

Dr. F. Maree (Agricultural Research Council) supplied 9 SAT1, 4 SAT2 and 3 SAT3 sequences (Table 4.1). A FMDV 3D sequence was submitted to a Blastp search against the PDB and it identified two protein structures (1U09 and 2D7S). Both these structures are FMDV 3D structures. It was decided to use 1U09 (Ferrer-Orta *et al.*, 2004) as its resolution was 1.91Å vs 3.00Å of 2D7S. Alignment was done with ClustalX using the

Table 4.1: Top: The SAT serotypes 3C protease sequences used in the variation analysis. Bottom: The SAT serotypes used in the 3D RNA polymerase variation analysis. Provided by Dr. F. Maree of the ARC. The sequences missing a number after the '/' lack a date in the original GenBank entry.

SAT subtype 3C sequences		
SAT1	SAT2	SAT3
SAT1/UGA/3/99 (gi:62362307)	SAT2/ZIM/7/83 (gi:33332022)	SAT3/KNP/10/90 (gi:21434547)
SAT1/UGA/1/97 (gi:15419327)	SAT2/KNP/19/89 (gi:15419331)	SAT3/ZAM/4/96 (gi:62362337)
SAT1/SUD/3/76 (gi:62362303)	SAT2/SAR/16/83 (gi:62362321)	SAT3/ZIM/5/91 (gi:62362339)
SAT1/NIG/15/75 (gi:62362299)	SAT2/ANG/4/74 (gi:62362311)	SAT3/MAL/03/76 (gi:12274987)
SAT1/NIG/5/81 (gi:62362297)	SAT2/KEN/8/99 (gi:62362315)	SAT3/BEC/1/65 (gi:21328275)
SAT1/TAN/37/99 (gi:62362305)	SAT2/ZIM/14/90 (gi:62362331)	SAT3/UGA/2/97 (gi:62362335)
SAT1/TAN/1/99 (gi:15419329)	SAT2/ZIM/17/91 (gi:62362333)	SAT3/KEN/3/ (gi:46810960)
SAT1/KNP/196/91 (gi:15419321)	SAT2/2/ (gi:46810952)	SAT3/BEC/3/ (gi:46810960)
SAT1/SAR/09/81 (gi:62362301)	SAT2/SEN/7/83 (gi:62362325)	SAT3/RSA/2/ (gi:46810956)
SAT1/ZAM/2/93 (gi:62362309)	SAT2/SEN/05/75 (gi:62362323)	
SAT1/NAM/307/98 (gi:62362295)	SAT2/ANG/4/74 (gi:62362311)	
SAT1/MOZ/3/02 (gi:62362341)	SAT2/MOZ/4/83 (gi:15419321)	
SAT1/KEN/5/98 (gi:62362293)	SAT2/RHO/1/48 (gi:62362317)	
SAT1/BOT/1/68 (gi:46810946)	SAT2/KEN/3/57 (gi:6572136)	
SAT1/RSA/5/ (gi:46810940)	SAT2/RWA/2/01 (gi:62362319)	
SAT1/SWA/6/ (gi:46810942)	SAT2/SAU/6/00 (gi:21434553)	
SAT1/RHO/ (gi:46810948)	SAT2/ZAI/1/74 (gi:62362329)	
SAT1/BEC/1/ (gi:46810932)	SAT2/GHA/8/91 (gi:62362313)	
SAT1/SWA/3/ (gi:46810936)	SAT2/UGA/2/02 (gi:62362327)	
SAT1/RHO/4/ (gi:46810938)	SAT2/3KEN/21/ (gi:6810954)	
SAT1/20/ (gi:46810934)	SAT2/RHO/1/48 (gi:46810950)	

SAT subtype 3D sequences		
SAT1	SAT2	SAT3
SAR/09/81 (not yet submitted)	ZIM/7/83 (gi:33332022)	KEN/3/ (gi:46810960)
BOT/1/68 (gi:46810946)	SAT2/2/ (gi:46810952)	RSA/2/ (gi:46810956)
SWA/6/ (gi:46810942)	RHO/1//48 (gi:62362317)	
RSA/5/ (gi:46810940)	3KEN/32/ (gi:6810954)	
RHO/4/ (gi:46810938)		
SWA/3/ (gi:46810936)		
BEC/1/ (gi:46810932)		
RHO/ (gi:46810948)		
SAT1/20/ (gi:46810934)		

default parameters, modelling scripts generated with the Structural module in FunGIMS and modelling done with Modeller 9v1 including a fast model refinement step. SAR/09/81 was used as a representative sequence for SAT1, ZIM/7/83/2 was used for SAT2 and RSA/2/3 was used for SAT3. In all cases the SAT target was 6 residues shorter than the template.

4.3. Results and Discussion

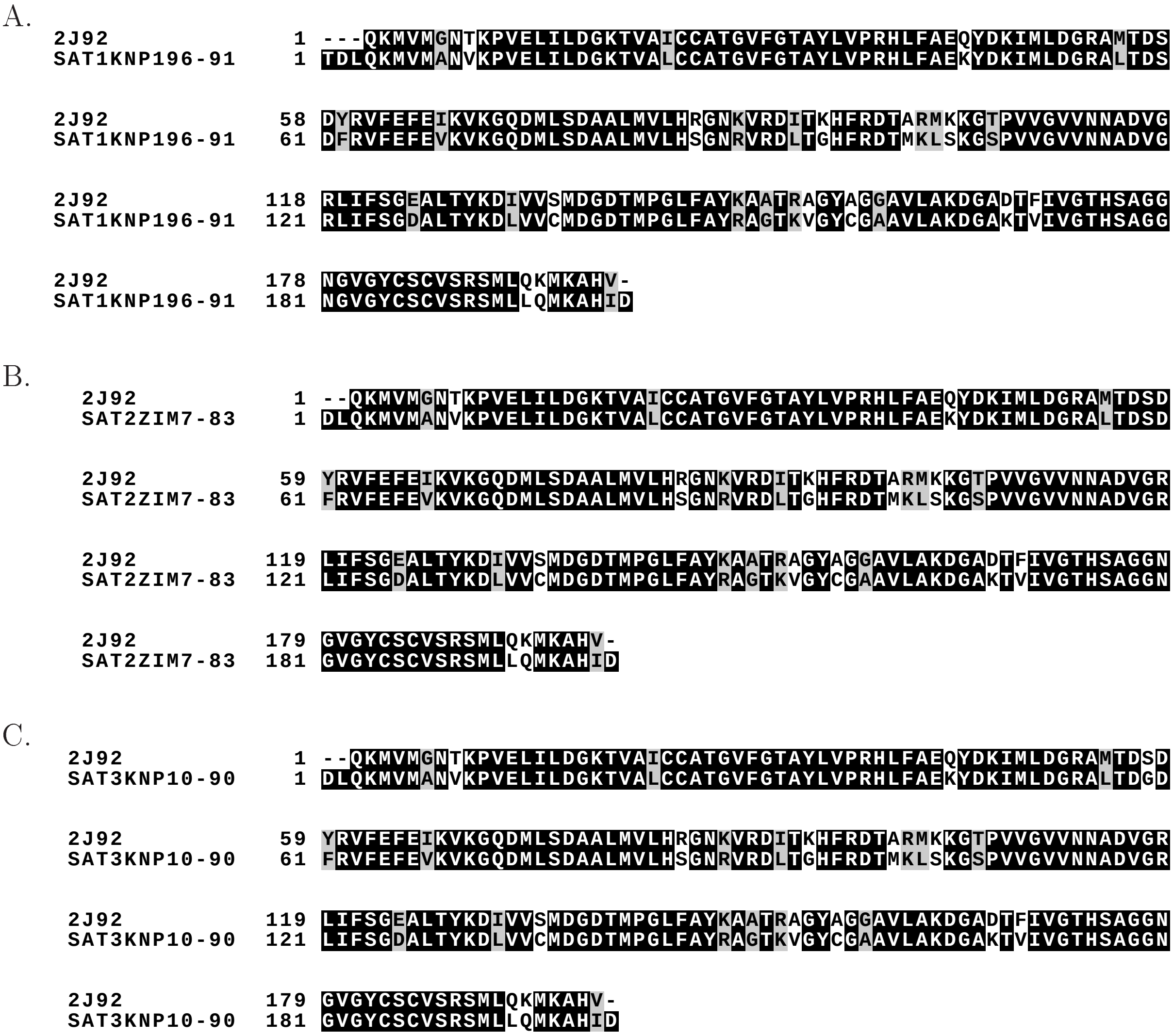
Because the various SAT serotypes are so similar, a representative model was built for each serotype (SAT1, SAT2 and SAT3). The variation for each serotype was then mapped onto the respective model.

4.3.1. 3C Protease

The SAT isolates included in this study are represented across Africa and include isolates from West, East, Central and Southern Africa. All the sequences used to build the respective models for $3C^{pro}$ showed ∼85% identity with 2J92. This was to be expected as the conservation of FMDV $3C^{pro}$ is high. The alignments that were used in modelling the $3C^{pro}$ SAT serotypes are shown in Figure 4.3 and the high identity between target and template is indicated.

After the KNP/96/91/1 SAT1 $3C^{pro}$ model was built, the variation observed in the SAT1 $3C^{pro}$ alignment was mapped onto the model (Fig. 4.5). There was variation at 45 residue positions (21%) within the 21 SAT sequences. In 76% (35) of the positions, variation was limited to 2 amino acids, 20% (9) of the positions were limited to 3 amino acids and 4% (2) limited to 4 amino acids.

ZIM/7/83/2 was used for the SAT2 model. SAT2 showed 41% more variance between the 21 SAT2 sequences compared to SAT1. Variation was observed in 63 positions (30%) and mapped to a SAT2 3C model (Fig. 4.5). In 76% (48) of the positions, variation was limited to 2 amino acids, 16% (10) of the positions was limited to 3 amino acids, 6% (4) limited to 4 amino acids and 2% (1) limited to 5 amino acids.



```
A.
2J92             1 ---QKMVMGNTKPVELILDGKTVAICCATGVFGTAYLVPRHLFAEQYDKIMLDGRAMTDS
SAT1KNP196-91    1 TDLQKMVMANVKPVELILDGKTVALCCATGVFGTAYLVPRHLFAEKYDKIMLDGRALTDS

2J92            58 DYRVFEFEIKVKGQDMLSDAALMVLHRGNKVRDITKHFRDTARMKKGTPVVGVVNNADVG
SAT1KNP196-91   61 DFRVFEFEVKVKGQDMLSDAALMVLHSGNRVRDLTGHFRDTMKLSKGSPVVGVVNNADVG

2J92           118 RLIFSGEALTYKDIVVSMDGDTMPGLFAYKAATRAGYAGGAVLAKDGADTFIVGTHSAGG
SAT1KNP196-91  121 RLIFSGDALTYKDLVVCMDGDTMPGLFAYRAGTKVGYCGAAVLAKDGAKTVIVGTHSAGG

2J92           178 NGVGYCSCVSRSMLQKMKAHV-
SAT1KNP196-91  181 NGVGYCSCVSRSMLLQMKAHID

B.
2J92             1 --QKMVMGNTKPVELILDGKTVAICCATGVFGTAYLVPRHLFAEQYDKIMLDGRAMTDSD
SAT2ZIM7-83      1 DLQKMVMANVKPVELILDGKTVALCCATGVFGTAYLVPRHLFAEKYDKIMLDGRALTDSD

2J92            59 YRVFEFEIKVKGQDMLSDAALMVLHRGNKVRDITKHFRDTARMKKGTPVVGVVNNADVGR
SAT2ZIM7-83     61 FRVFEFEVKVKGQDMLSDAALMVLHSGNRVRDLTGHFRDTMKLSKGSPVVGVVNNADVGR

2J92           119 LIFSGEALTYKDIVVSMDGDTMPGLFAYKAATRAGYAGGAVLAKDGADTFIVGTHSAGGN
SAT2ZIM7-83    121 LIFSGDALTYKDLVVCMDGDTMPGLFAYRAGTKVGYCGAAVLAKDGAKTVIVGTHSAGGN

2J92           179 GVGYCSCVSRSMLQKMKAHV-
SAT2ZIM7-83    181 GVGYCSCVSRSMLLQMKAHID

C.
2J92             1 --QKMVMGNTKPVELILDGKTVAICCATGVFGTAYLVPRHLFAEQYDKIMLDGRAMTDSD
SAT3KNP10-90     1 DLQKMVMANVKPVELILDGKTVALCCATGVFGTAYLVPRHLFAEKYDKIMLDGRALTDGD

2J92            59 YRVFEFEIKVKGQDMLSDAALMVLHRGNKVRDITKHFRDTARMKKGTPVVGVVNNADVGR
SAT3KNP10-90    61 FRVFEFEVKVKGQDMLSDAALMVLHSGNRVRDLTGHFRDTMKLSKGSPVVGVVNNADVGR

2J92           119 LIFSGEALTYKDIVVSMDGDTMPGLFAYKAATRAGYAGGAVLAKDGADTFIVGTHSAGGN
SAT3KNP10-90   121 LIFSGDALTYKDLVVCMDGDTMPGLFAYRAGTKVGYCGAAVLAKDGAKTVIVGTHSAGGN

2J92           179 GVGYCSCVSRSMLQKMKAHV-
SAT3KNP10-90   181 GVGYCSCVSRSMLLQMKAHID
```

Figure 4.3: The alignments used in the modelling of $3C^{pro}$. A: KNP/96/91/1. B: ZIM/7/82/2. C: KNP/10/90/3 with 2J92 being the template sequence (serotype A10).

KNP/10/90/3 was used as a representative for the SAT3 serotype. SAT3 showed 35% less variation than SAT1 and 54% less variation than SAT2 in the 9 sequences analyzed. There was variation in 29 positions (14%) of which 93% (27 positions) varied by 2 amino acids and 7% (2 positions) varied by 3 amino acids (Fig. 4.5). An important residue position was Asp 84 that is part of the catalytic triad. In ZIM/5/91/3 this Asp was replaced by a Tyr. This is the only occurrence in all the analyzed sequences where a mutation was present in the active site. There are 2 reasons for less variation in SAT3: SAT3 is not well represented in this study and it has a geographical distribution limited to Southern and Central Africa.

A.

```
1U09           1 -GLIVDTRDVEERVHVMRKTKLAPTVAHGVFNPEFGPAALSNKDPRLNEGVVLDEVIFSK
SAR09-81-1     1 EGLVVDTREVEERVHVMRKTKLAPTVAYGVFQPEFGPAALSNNDKRLNEGVVLDEVIFSK

1U09          60 HKGDTKMSAEDKALFRRCAADYASRLHSVLGTANAPLSIYEAIKGVDGLDAMEPDTAPGL
SAR09-81-1    61 HKGDAKMSEADKKLFRLCAADYASHLHNVLGTANSPLSVFEAIKGVDGLDAMEPDTAPGL

1U09         120 PWALQGKRRGALIDFENGTVGPEVEAALKLMEKREYKFACQTFLKDEIRPMEKVRAGKTR
SAR09-81-1   121 PWALQGKRRGALIDFENGTVGPEIEQALKLMEKKEYKFTCQTFLKDEIRPLEKVKAGKTR

1U09         180 IVDVLPVEHILYTRMMIGRFCAQMHSNNGPQIGSAVGCNPDVDWQRFGTHFAQYRNVWDV
SAR09-81-1   181 IVDVLPVEHIIYTRMMIGRFCAQMHSNNGPQIGSAVGCNPDVDWQRFGCHFAQYRNVWDI

1U09         240 DYSAFDANHCSDAMNIMFEEVFRTEFGFHPNAEWILKTLVNTEHAYENKRITVEGGMPSG
SAR09-81-1   241 DYSAFDANHCSDAMNIMFEEVFREEFGFHPNAVWILKTLINTEHAYENKRITVEGGMPSG

1U09         300 CSATSIINTILNNIYVLYALRRHYEGVELDTYTMISYGDDIVVASDYDLDFEALKPHFKS
SAR09-81-1   301 CSATSIINTILNNIYVLYALRRHYEGVELSHYTMISYGDDIVVASDYDLDFEALKPHFKS

1U09         360 LGQTITPADKSDKGFVLGHSITDVTFLKRHFHMDYGTGFYKPVMASKTLEAILSFARRGT
SAR09-81-1   361 LGQTITPADKSDKGFVLGQSITDVTFLKRHFHLDYGTGFYKPVMASKTLEAILSFARRGT

1U09         420 IQEKLISVAGLAVHSGPDEYRRLFEPFQGLFEIPSYRSLYLRWVNAVCGDAAALEHH
SAR09-81-1   421 IQEKLISVAGLAVHSGPDEYRRLFEPFQGTFEIPSYRSLYLRWVNAVCGDA------
```

B.

```
1U09           1 -GLIVDTRDVEERVHVMRKTKLAPTVAHGVFNPEFGPAALSNKDPRLNEGVVLDEVIFSK
ZIM-7-83-2     1 EGLVVDTREVEERVHVMRKTKLAPTVAHGVFQPEFGPAALSNNDKRLSEGVVLDEVIFSK

1U09          60 HKGDTKMSAEDKALFRRCAADYASRLHSVLGTANAPLSIYEAIKGVDGLDAMEPDTAPGL
ZIM-7-83-2    61 HKGDAKMSEADKRLFRLCAADYASHLHNVLGTANSPLSVFEAIKGVDGLDAMEPDTAPGL

1U09         120 PWALQGKRRGALIDFENGTVGPEVEAALKLMEKREYKFACQTFLKDEIRPMEKVRAGKTR
ZIM-7-83-2   121 PWALRGKRRGALIDFENGTVGSEIEAALKLMEKKEYKFTCQTFLKDEIRPLEKVKAGKTR

1U09         180 IVDVLPVEHILYTRMMIGRFCAQMHSNNGPQIGSAVGCNPDVDWQRFGTHFAQYRNVWDV
ZIM-7-83-2   181 IVDVLPVEHIIYTRMMIGRFCAQMHSNNGPQIGSAVGCNPDVDWQRFGTHFAQYKNVWDI

1U09         240 DYSAFDANHCSDAMNIMFEEVFRTEFGFHPNAEWILKTLVNTEHAYENKRITVEGGMPSG
ZIM-7-83-2   241 DYSAFDANHCSDAMNIMFEEVFREEFGFHPNAVWILKTLINTEHAYENKRITVEGGMPSG

1U09         300 CSATSIINTILNNIYVLYALRRHYEGVELDTYTMISYGDDIVVASDYDLDFEALKPHFKS
ZIM-7-83-2   301 CSATSIINTILNNIYVLYALRRHYEGVELSHYTMISYGDDIVVASDYDLDFEALKPHFKS

1U09         360 LGQTITPADKSDKGFVLGHSITDVTFLKRHFHMDYGTGFYKPVMASKTLEAILSFARRGT
ZIM-7-83-2   361 LGQTITPADKSDKGFVLGQSITDVTFLKRHFHLDYETGFYKPVMASKTLEAILSFARRGT

1U09         420 IQEKLISVAGLAVHSGPDEYRRLFEPFQGLFEIPSYRSLYLRWVNAVCGDAAALEHH
ZIM-7-83-2   421 IQEKLISVAGLAVHSGQDEYRRLFEPFQGTFEIPSYRSLYLRWVNAVCGDA------
```

C.

```
1U09           1 -GLIVDTRDVEERVHVMRKTKLAPTVAHGVFNPEFGPAALSNKDPRLNEGVVLDEVIFSK
RSA-2-3        1 EGLVVDTREVEERVHVMRKTKLAPTVAHGVFQPEFGPAALSNNDKRLNEGVVLDEVIFSK

1U09          60 HKGDTKMSAEDKALFRRCAADYASRLHSVLGTANAPLSIYEAIKGVDGLDAMEPDTAPGL
RSA-2-3       61 HKGDAKMSEADKKLFRLCAADYASHLHNVLGTANSPLSVFEAIKGVDGLDAMEPDTAPGL

1U09         120 PWALQGKRRGALIDFENGTVGPEVEAALKLMEKREYKFACQTFLKDEIRPMEKVRAGKTR
RSA-2-3      121 PWALQGRRRGALIDFENGTVGPEIEQALKLMEKKEYKFTCQTFLKDEIRPLEKVKAGKTR

1U09         180 IVDVLPVEHILYTRMMIGRFCAQMHSNNGPQIGSAVGCNPDVDWQRFGTHFAQYRNVWDV
RSA-2-3      181 IVDVLPVEHIIYTRMMIGRFCAQMHSNNGPQIGSAVGCNPDVDWQRFGCHFAQYKNVWDI

1U09         240 DYSAFDANHCSDAMNIMFEEVFRTEFGFHPNAEWILKTLVNTEHAYENKRITVEGGMPSG
RSA-2-3      241 DYSAFDANHCSDAMNIMFEEVFREEFGFHPNAVWVLKTLINTEHAYENKRITVEGGMPSG

1U09         300 CSATSIINTILNNIYVLYALRRHYEGVELDTYTMISYGDDIVVASDYDLDFEALKPHFKS
RSA-2-3      301 CSATSIINTILNNIYVLYALRRHYEGVELSHYTMISYGDDIVVASDYDLDFEALKPHFKS

1U09         360 LGQTITPADKSDKGFVLGHSITDVTFLKRHFHMDYGTGFYKPVMASKTLEAILSFARRGT
RSA-2-3      361 LGQTITPADKSDKGFVLGQSITDVTFLKRHFHLDYETGFYKPVMASKTLEAILSFARRGT

1U09         420 IQEKLISVAGLAVHSGPDEYRRLFEPFQGLFEIPSYRSLYLRWVNAVCGDAAALEHH
RSA-2-3      421 IQEKLISVAGLAVHSGQDEYRRLFEPFQGTFEIPSYRSLYLRWVNAVCGDA------
```

Figure 4.4: The alignments used in the modelling of 3D. A: SAR/09/81/1. B: ZIM/7/83/2. C: RSA/2/3.

Table 4.2: The changes observed in the SAT serotypes as compared to the invariant region from residue 76-91 identified by Carillo *et al.* (2005). A structural representation of the invariant region can be seen in figure 4.8.

Subtype	Variation (aa71-86)	Effect
Invariant region	VKGQDMLSDAALMVLH	-
SAT1/UGA/1/97	VKGQDMLSDAALMVL*N*	Maintains backbone H-bond and side-chain H-bond
SAT1/UGA/3/99	VKGQDMLSDAALMVL*N*	Maintains backbone H-bond and side-chain H-bond
SAT1/NIG/15/75	VKGQ*E*MLSDAALMVLH	Maintains backbone H-bond and side-chain H-bond
SAT2/ZIM/17/91	VKG*P*DMLSDAALMVLH	Maintains backbone H-bond. Might distort the loop slightly
SAT2/KNP/19/89	VKGQDMLSDAALM*G*LH	Maintains backbone H-bond
SAT2/SEN/7/83	VKGQDM*M*SDAALMVL*N*	Maintains backbone H-bond and side-chain H-bond
SAT2/SEN/05/75	VKGQDM*M*SDAALMVL*N*	Maintains backbone H-bond and side-chain H-bond
SAT2/GHA/8/91	VKGQDM*M*SDAALMVL*N*	Maintains backbone H-bond and side-chain H-bond
SAT2/UGA/2/02	VKGQDMLSDAALMVL*N*	Maintains backbone H-bond and side-chain H-bond
SAT3/ZIM/5/91	VKGQDMLS*Y*AAL*I*VLH	This includes a mutation in the active site.
SAT3/UGA/2/97	VKGQDMLSDAALMVL*N*	Maintains backbone H-bond and side-chain H-bond

Most of the variation in the SAT $3C^{pro}$ seems to occur at one end of the C-terminal β-barrel (Fig. 4.6). This region is surface-exposed and can potentially accommodate more variation without influencing the activity of the enzyme. Another interesting observation was that the inner β-sheet in the C-terminal β-barrel contained very little variation and is conserved, whereas the N-terminal β-barrel contains significantly more variation.

An invariant section (residues 76-91, VKGQDMLSDAALMVLH) in $3C^{pro}$ identified by Carillo and co-workers (Fig. 4.8), was shown to contain variation within the SAT serotypes. Table 4.2 shows the aa changes for each isolate compared to the invariant region. Eleven isolates showed variation in the invariant region. The invariant region is located on two consecutive β-strands of which the second β-sheet (residues 85-91) contains one of the catalytic triad residues (Asp). A reason for this conservation of the

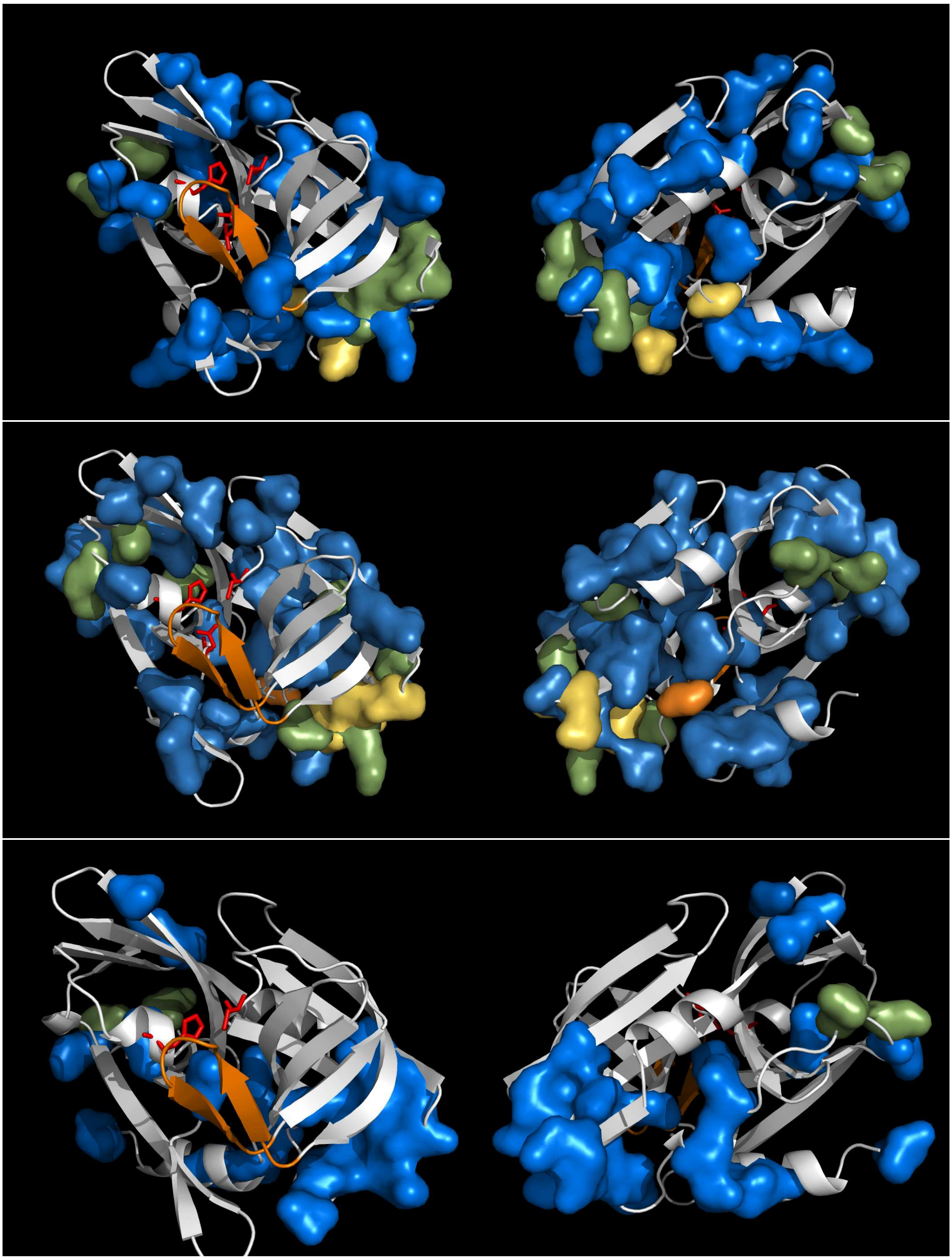


Figure 4.5: SAT 3C^{pro} variation mapped onto a SAT 3C^{pro}model. Views from both sides of the enzyme are shown. Top: SAT1, middle: SAT2, bottom: SAT3. White indicates conserved positions across all the sequences analyzed, blue indicates 2 different residues found at that position, green indicates 3 different residues found at that position and yellow indicates the presence of 4 different residues. The active site catalytic triad is coloured red and the β-ribbon is coloured orange.

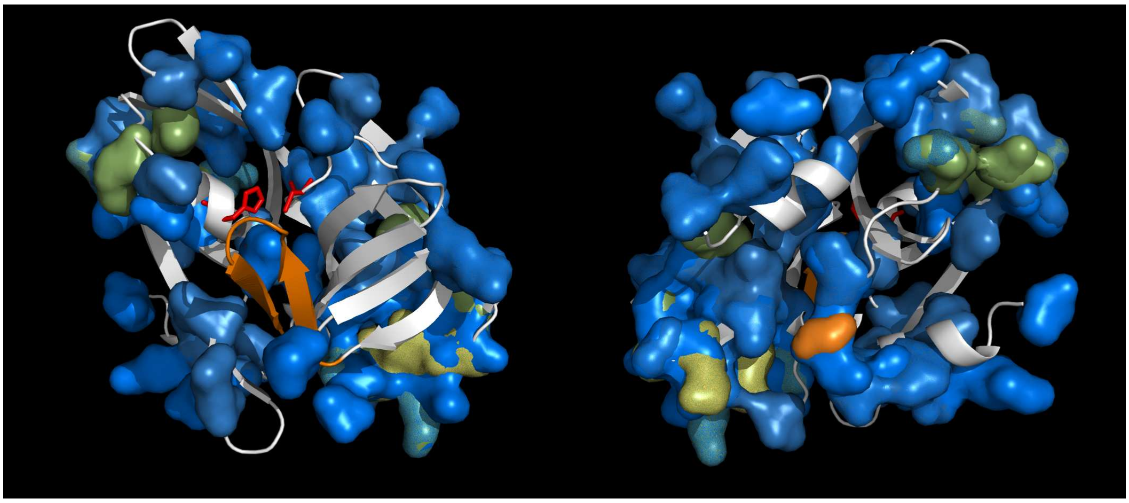


Figure 4.6: The variation seen in the 3C^{pro} protease as mapped to a cartoon representation of the enzyme. Both sides of the enzyme are shown. White indicates conserved positions across all the serotype sequences analyzed, blue indicates 2 different residues found at that position, green indicates 3 different residues found at that position and yellow indicates the presence of 4 different residues.

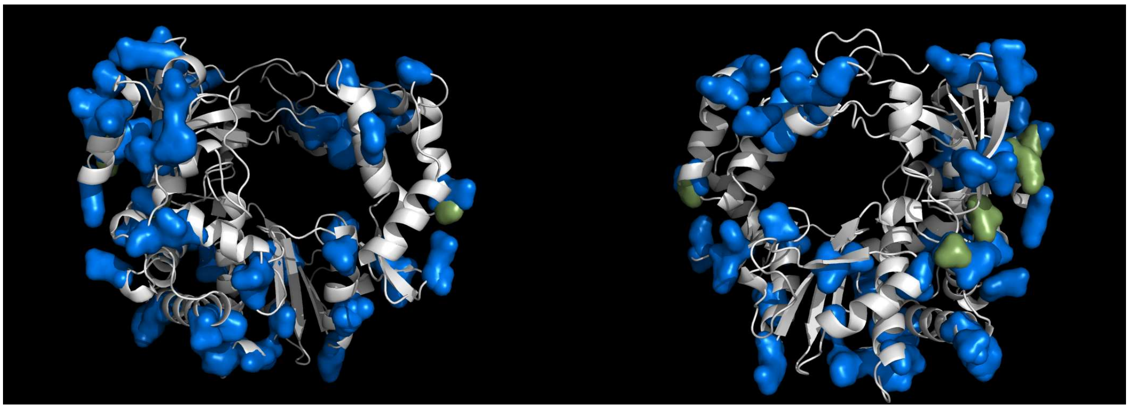


Figure 4.7: The variation seen in the 3D protease as mapped to a cartoon representation of the enzyme. Views from both sides are shown. White indicates conserved positions across all the serotype sequences analyzed, blue indicates 2 different residues found at that position and green indicates 3 different residues found at that position.

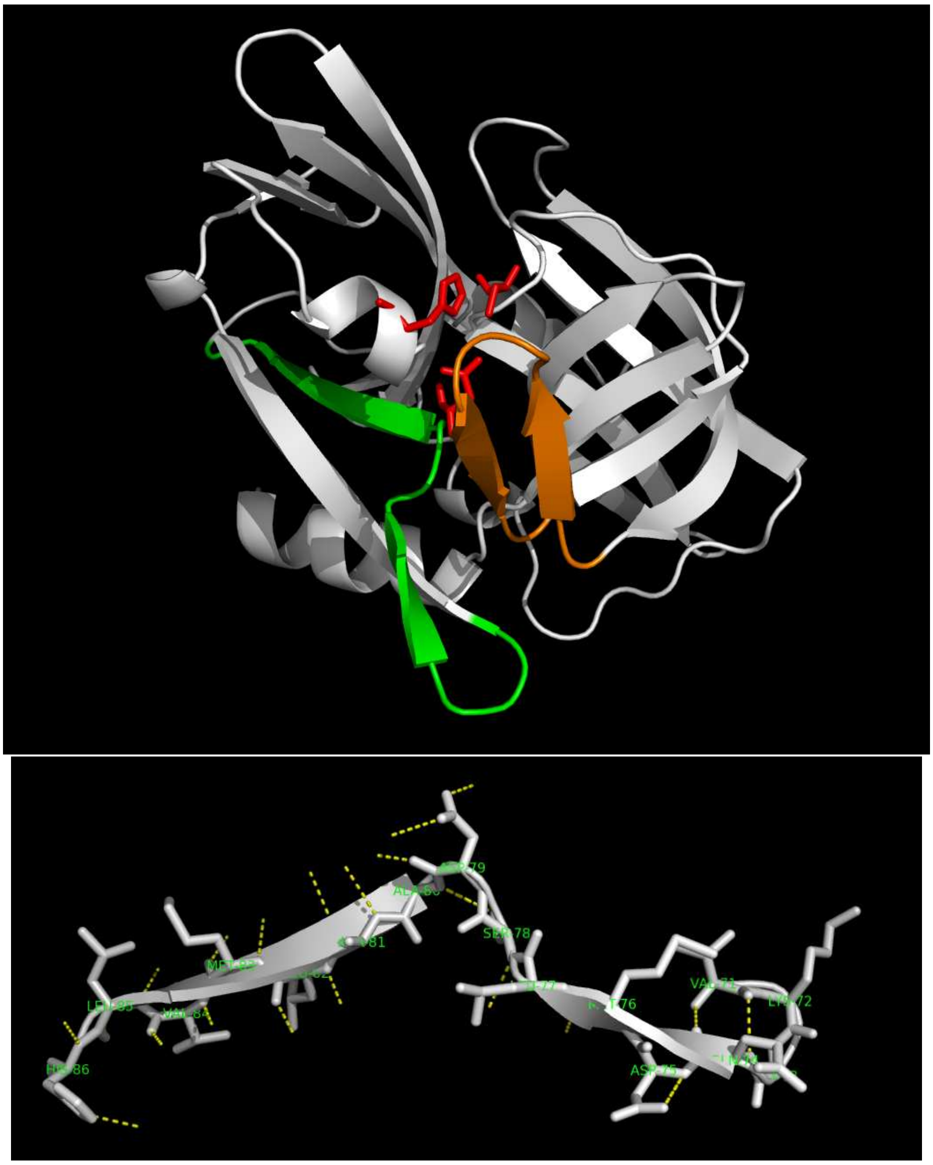


Figure 4.8: Top: The location of the invariant region identified by Carillo *et al.* in the 3C^{pro} structure. The numbers are the residue numbers used in the model and correspond to 3C^{pro} residues 76-91. Bottom: The hydrogen bond network for the invariant region. All residues are labeled according to the SAT1/KNP/96/91. Hydrogen bonds are indicated in yellow, dashed lines.

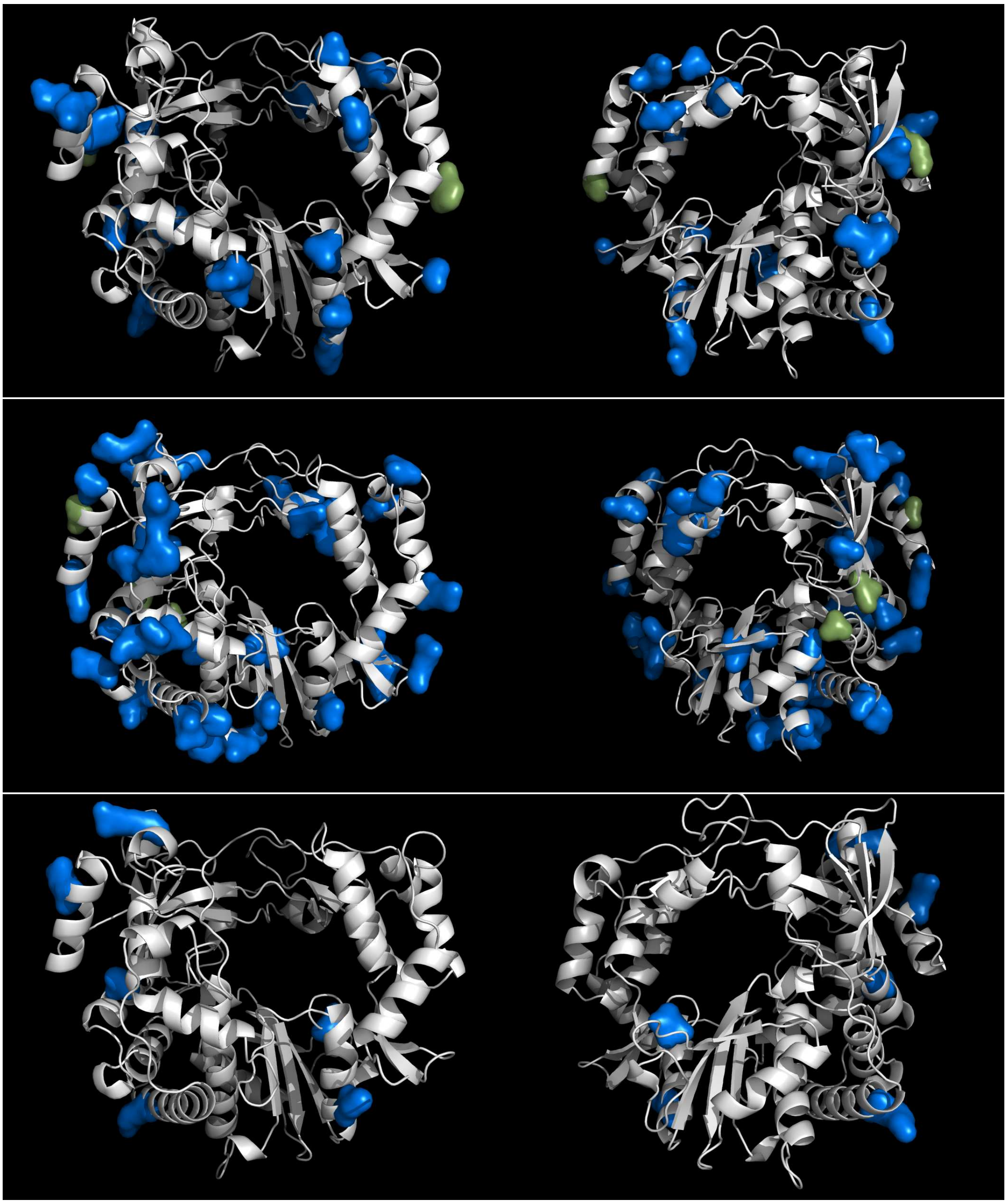


Figure 4.9: SAT 3D variation mapped onto a SAT 3D model. Views from both sides of the enzyme are shown. Top: SAT1, middle: SAT2, bottom: SAT3. White indicates conserved positions across all the sequences analyzed, blue indicates 2 different residues found at that position and green indicates 3 different residues.

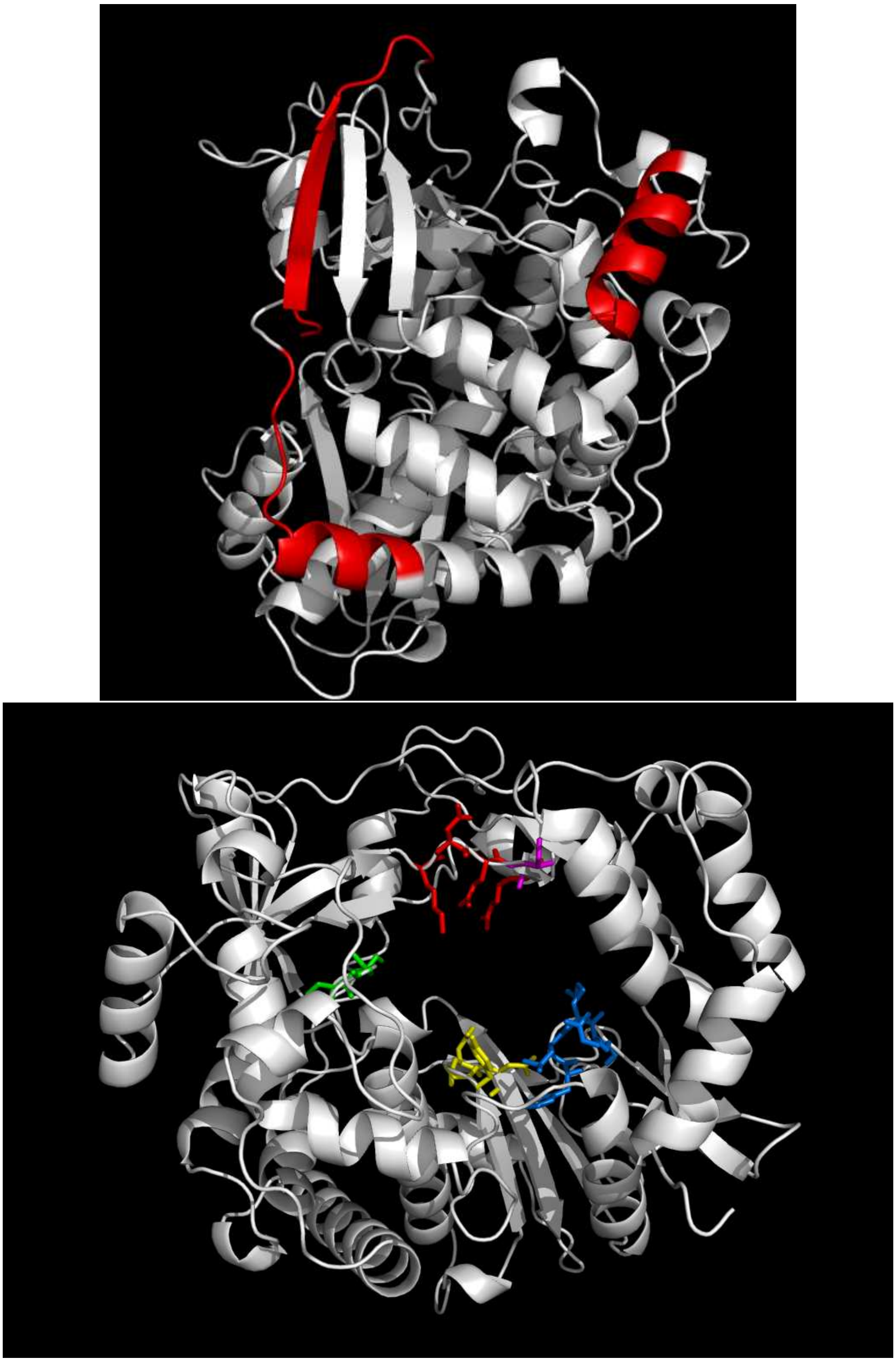


Figure 4.10: Top: The three hypervariable regions previously identified in 3D (George *et al.*, 2001). The regions coloured red and are residues 1-12 (β-strand), 64-76 (half α-helix and part of loop) and 143-153 (α-helix). Bottom: The four highly conserved motifs in 3D (Doherty *et al.*, 1999). The motifs are coloured as follows: red: KDELR; green: PSG; blue: FLKR; yellow: YGDD. The residue involved in mutation in the KDELR motif is coloured pink.

invariant region appears to be the orientation of the active site residues. The second β-strand (residues 85-91) in the invariant region associates with an adjacent β-strand (residues 40-45). This β-strand is followed by a a very short α-helix which is the location of the a second catalytic triad residue (His 46). It is involved in an extensive hydrogen bond network with two surrounding β-strands as well as with nearby residues. Figure 4.8 shows the hydrogen bond network in the region. The majority of the variable sites are involved in protein backbone hydrogen bonds. Thus, if the residue change does not involve a big physiochemical property change, it will not affect the backbone as much as the hydrogen bond network stays intact. This supports the hypothesis that the invariant region serves as an anchor region for the 3C protease. Thus, by conserving the invariant region's two β-strands, most of the active site residue orientation is also conserved.

SAT3/ZIM/5/91 showed a mutation in the active site where the Asp is converted to a Tyr. It has been previously proposed that a similar virus, Hepatitis A (HAV), may utilize a two-residue active site in 3C, which used only the Cys and His residues for catalysis (Bergmann *et al.*, 1997) but this has since been refuted (Yin *et al.*, 2005) and shown that HAV also uses a catalytic triad. This Asp-Tyr mutation has not yet been confirmed with resequencing.

In all 54 SAT 3C sequences analyzed, only one active site mutation occurred (D84Y in ZIM/5/91/3). In all the other sequences the catalytic triad and the residues surrounding them had very little, if any, variation. The analysis of the sequences showed that SAT2 3C had the most variation and that SAT3 had the least amount of variation.

4.3.2. 3D RNA Polymerase

The 3D RNA polymerase is highly conserved as mentioned before. The general sequence identity was 92% between the target and the template. This varied by no more that 1% between the three targets. The alignments used for each of the representative models are shown in Figure 4.4 and the high identity between target and template is indicated.

SAR/09/81/1 was used as the representative model for the SAT1 serotype. In the 9 SAT1 sequences provided there were 20 positions (91%) that had either one of two residues and

2 positions (9%) which had one of three residues (Fig. 4.9). The variation seemed to be limited to the outer edges of the protein.

ZIM/7/83/2 was used as the representative model for the SAT2 serotype (Fig. 4.9). SAT2 3D showed more variation compared to SAT1 and SAT3 3D. SAT2 3D had 38 positions (8%) with either one of two residues and three positions (0.8%) which had a three residue difference. This is almost double the variation seen in half the number of proteins when compared to SAT1 3D. This indicates that the 3D protein of SAT2 is more variable than that of SAT1 even though isolates from the same broad geographical region was included for both serotypes.

RSA/2/3 was used as the representative model for the SAT3 serotype (Fig. 4.9). A limited number of sequences made this serotype difficult to compare with SAT1 and SAT2. The three supplied proteins differed by two residues only in 6 positions (1.6%). The rest of the sequence was conserved.

3D variation did not seem to be limited to certain areas as seen for the 3C variation (Fig. 4.7). The results presented here suggests an average of 5% variable residues for 3D in each serotype. This is much lower than the other reported variability studies which reported variation as high as 26% variable residues (Carrillo *et al.*, 2005). This difference might be explained by the number of isolates in each serotype included in the studies as well as the geographical distribution. Intra and inter-serotype comparisons can also influence this value.

Three hypervariable regions in 3D have been identified previously (Fig. 4.10; George *et al.*, 2001). These areas did show some variability in the proteins analyzed here but it was mostly two residue differences between the proteins. The 3D hypervariable region, between residues 143-153, showed the most variability with four positions being variable. This area corresponds to a surface exposed α-helix. As can be expected, the variability are located on the exposed side of the α-helix. An α-helix important in inter-protein dimer interaction was identified from residue 68-89 (Ferrer-Orta *et al.*, 2004). The alignment of SAT 3D sequences revealed four residue positions that contained either one of two residues. The changes were located in two variable hot spots occurring at the ends of the α-helix (two mutations per site), which still conserves the important central region involved in 3D dimer interaction.

Previously four conserved motifs were described in 3D polymerases of FMDV (Doherty *et al.*, 1999; Carrillo *et al.*, 2005). These four motifs are: KDELR (residues 159-163), PSG (residues 289-291), YGDD (residues 324-327) and FLKR (residues 371-374). The location of the conserved motifs can be seen in figure 4.10. Three of the motifs were also conserved in the SAT 3D sequences used here. However, the first motif, KDELR was present in the SAT sequences as either KDEIR or KDEVR. KDEIR was found to be conserved in all the SAT 3D sequences used except for SAT2/3KEN/21 that used the KDEVR motif. When looking at the orientation and location of the KDELR/KDEIR motif on the structure (Fig. 4.10) it is evident that the variable residue (L) is pointing away from the active site. The two mutations seen here (Leu->Ile, Leu->Val) are both similar in size and hydrophobicity, which maintain the physiochemical properties probably required for a residue in this location.

In comparison, the sequences used here showed that 3D also has less variation than 3C^{pro}. The SAT 3D variation followed the trend seen in SAT 3C^{pro} where SAT2 had the most variation. This is explained by the fact that SAT2 is more prevalent in wildlife in Africa and has caused the most outbreaks. This results in an increased chance for variation accumulation in the genome, which can possibly be an indication of the age of the SAT2 serotype. If SAT2 was the ancestral SAT serotype, it would have acquired more variation over time. But without a detailed phylogenetic study of the relationship between the SAT types, this is pure speculation.

4.4. Conclusion

The replication of FMDV is dependent on several factors, including cell entry via receptors, replication of the RNA genome, translation, the correct polyprotein processing by viral encoded proteases, and packaging of the RNA into virions. A recent study investigated possible factors involved in the replication of SAT isolates which presented with diverse growth kinetics. The implication of this is in the implementation of engineered virus to be used as custom-made vaccine specific for a geographic region. In principle infectious cDNA technology can be used to produce foot-and-mouth disease viruses with improved biological properties if the antigenic determinants of the outer capsid of a good vaccine strain with the desirable biological properties in a production plant are substituted by that of an outbreak isolate (Zibert *et al.*, 1990; Rieder *et al.*, 1993; Almeida *et al.*, 1998; Beard and Mason, 2000; van Rensburg *et al.*, 2004; Storey *et al.*, 2007). In practice we have found that the resulting chimera virus mostly took on the growth performance of the parental field isolate, although some improvement was observed by the presence of the better genetic background of the vaccine strain. Even with improvement of the cell entry pathway by introduction of alternative receptor entry mechanisms the growth performance was not significantly enhanced (Blignaut et al., unpublished; Maree, personal communication). To investigate whether these amino acid differences impact on the ability of the 3C$^{\text{pro}}$ to recognise different cleavage sites within the P1 polyprotein, several chimeric viruses were engineered and the analysis of these are underway. In this study we investigated the amount of variation within the 3C$^{\text{pro}}$ responsible for ten of the twelve proteolytic processing events of the FMDV polyprotein to support a present study on the amount of variation within the 3C cleavage sites and the activity of the enzyme within the cleavage site variation.

A study of the heterogeneity of the FMDV 3C$^{\text{pro}}$ revealed 32% variant amino acid positions, whilst 57%, 65% and 75% variant amino acids were observed for the external capsid proteins (1B to 1D) (van Rensburg *et al.*, 2004). Similar to other picornaviral 3C$^{\text{pro}}$, FMDV 3C$^{\text{pro}}$ belongs to an unusual family of chymotrypsin-like cysteine proteases, containing a serine protease fold, as confirmed by the recently solved FMDV 3C$^{\text{pro}}$ crystal structure (Birtley *et al.*, 2005). The catalytic mechanism of 3C$^{\text{pro}}$ involves a Cys-His-Asp triad which has a very similar conformation to the Ser-His-Asp triad found in serine proteases. It is important to note that the third member of the triad is also an Asp residue in HAV, but a Glu in HRV (Curry *et al.*, 2007). The FMDV 3C $^{\text{pro}}$ cleavage specificity exhibits great heterogeneity, but similar to other picornaviral 3C$^{\text{pro}}$, the enzyme requires a hydrophobic residue at P4 (Curry *et al.*, 2007). Whereas other picornavirus 3C proteases accept only Gln at the P1 position, the FMDV 3C$^{\text{pro}}$ differs in that it is able to accept both Gln and Glu in this position. It has been suggested that correlations between the different sub-sites in the substrate binding pocket of 3C^{pro} exist. By analysing FMDV sequences (Carrillo *et al.*, 2005), Curry and co-workers (2007) suggested correlations between P1, P2 and P1'. For instance, if P1 is a Gln, P2 would usually be a Lys and P1' a hydrophobic residue. Small amino acids (Gly or Ser) are however present in the P1' position for all the viruses analysed when P1 is Glu. Important roles for P2 and P4' have also been implicated (Birtley *et al.*, 2005).

In addition to processing of the viral polyprotein, 3C^{pro} has been shown to cleave host cell proteins in cell culture. Cleavage of histone H3, resulting in a down-regulation of transcription, has been demonstrated (Falk *et al.*, 1990; Tesar and Marquardt, 1990), although an unusual cleavage site was suggested. The enzyme has also been reported to cleave host cell translation initiation proteins, eIF4G and eIF4A (Belsham and Sonenberg, 2000; Li *et al.*, 2001; Strong and Belsham, 2004). These cleavage events occur rather late in the infection cycle and their role in viral replication is unclear. A recent report indicated that PTB, eIF3a,b and PABP RNA-binding proteins are cleaved during FMDV infection in cell culture, although no evidence for 3C^{pro} involvement was established (Pulido *et al.*, 2007).

Mapping the variation found within 53 SAT viruses representative across Africa onto the 3C^{pro} structure reveals that these are almost entirely peripheral to the substrate-binding site, supportive to previous finding by Birtley *et al.* (2005). There was some variation close-by the active site in the invariant region but all the variation still preserved the backbone hydrogen bond structure needed to keep the catalytic triad in the correct conformation for catalysis. This emphasizes the highly conserved nature of 3C^{pro} and the likeliness that chimeric viruses containing the outer capsid region of a disparate virus within the genetic background of an existing SAT2 genome-length clone (van Rensburg *et al.*, 2004) will be processed by the SAT2 3C^{pro}. The rate of processing might however be influenced by the sequence variation within the 3C cleavage sites in the P1 polyprotein.

The 3D RdRp is extremely conserved and is needed for virus replication. All of the variation were seen to occur outside of the binding cavity (Fig. 4.9) in the central part of the enzyme. Some of the variation may influence the activity of 3D but this study found that the majority of the differences are natural variation. The few differences in the invariant regions (KDEI/V/LR) were found not to significantly influence the overall activity as they have similar physiochemical properties. Another factor was that the side chains of the different residues in the invariant regions pointed away from the active site. All the variation seen in the different serotypes may have a small effect on the activity of the enzymes or on interaction cellular proteins, and this in turn could affect the replication speed of the virus. The variation may simply be a result of natural variation in SAT serotype enzymes. After analysis of the models and variation, there does not appear to be a reasonable site where 3C-3D interaction occurs. Although 3C presents an area on the C-terminal β-barrel where there is almost no variation, it does not necessarily imply an interaction site. 3D has a flattish area on the protein which, although it is sometimes used in protein-protein interaction, is not conclusive proof of an interaction site. The crystal structure of polio 3CD has been published (Marcotte *et al.*, 2007) but upon analysis it was found that the crystal structure provides no evidence for the interaction between 3C and 3D as they are separated by a 7-residue linker region. Further studies into co-variation was not done as it falls outside the scope of this specific study. The variation seen in 3C confirms the conserved nature of 3C yet it highlights that the variation that does occur, are limited to certain areas. Chapter 5 investigates the effect of variation on the capsid protein stability and its structure.

Chapter 5

FMDV Capsid Stability and Variation Analysis

5.1. Introduction

The capsid of the FMD virus consists of 60 copies of a four chain protomer derived from polypeptide P1 (Fig. 5.1) and is ca. 300Å in diameter. The P1 polypeptide is cleaved into three parts by the 3C^{pro} or 3CD^{pro} protease complex which results in the VP3, VP1, and VP0 peptides. Autocatalytic cleavage of VP0 into VP4 and VP2 is the last step in capsid assembly. One of the 60 protomers consists of chains VP1-4 encoded by the 1A-D coding regions on the FMDV genome. VP1-3 each consists of a β-barrel (8-stranded) in a jelly-roll topology (Fig. 5.1; Acharya *et al.*, 1989). The pentamers formed by five protomers, associate through an α-helix situated in the VP3 protein (Ellard *et al.*, 1999). This helix associates with its reciprocal helix as well as with His 142 in the opposite pentamer. Curry and co-workers have proposed that His 142 is vital in keeping the protomers together (Curry *et al.*, 1995). His 142 in VP3 of one pentamer associates with the positive dipole formed at the one end of the α-helix in VP2 of the opposite pentamer. It was speculated that the protonation of His 142 may prevent capsid assembly. Other histidine residues (His 145 on VP3 and His 21 on VP2) in close proximity were thought to also play a role in capsid assembly and uncoating. Mutation studies showed that if His 142 is replaced with an arginine, there is almost no capsid formation (Ellard *et al.*, 1999).

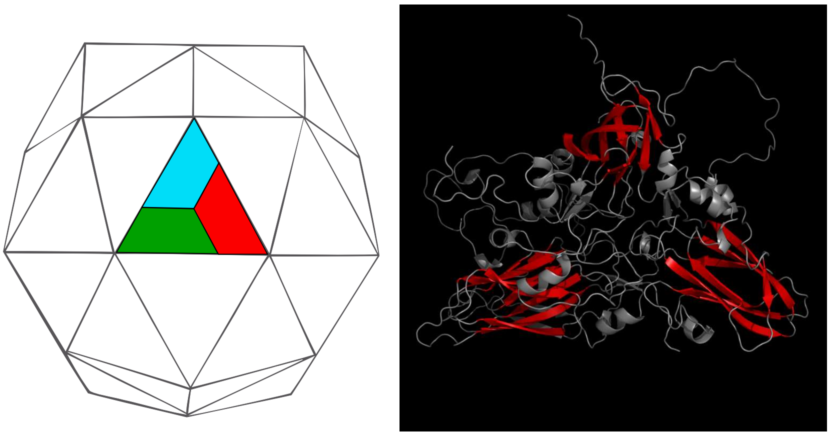


Figure 5.1: Left: A schematic representation of the FMDV icosahedral capsid. The capsid consists of 60 copies of each of the protomers. VP1: blue, VP2: green, VP3: red. Right: The 8 stranded β-barrel in a jelly-roll topology in VP1-3. β-barrels are coloured red.

The arrangement of the structural proteins in the capsid provides the antigenic sites important for eliciting neutralizing antibodies following infection or vaccination. VP1-3 forms the outside of the capsid while VP4 is completely buried inside the capsid. The FMDV capsid, unlike other Picornaviridae, also functions as a general scaffold to keep the RNA protected from the *in vivo* environment and mediates the binding to cellular receptors during cell entry. Interaction with cellular receptors is via the flexible βG-βH loop of VP1. This exposed loop contains an RGD (Arg-Gly-Asp) motif (Logan *et al.*, 1993) involved in binding integrin-receptors of which the αvβ1, αvβ3, αvβ6 and αvβ8 are known to be utilized by FMDV. In the well-studied O serotype, this GH-loop contains a cysteine residue at its base that allows the formation of a disulphide bond with a cysteine in VP2. This adds some stability to the loop and may aid in the receptor preference of the virus. Although field viruses use the integrin-receptors for infection, cell culture-adapted viruses obtain the ability to utilize an alternative receptor *i.e.*. heparan sulfate proteoglycans (HSPG) to enter cells (Fry *et al.*, 1999).

Previous structural work on FMDV includes crystallization of serotype O, A and C capsid at various resolutions (Acharya *et al.*, 1989;Curry *et al.*, 1992; Fry *et al.*, 1993; Lea *et al.*, 1994). These structures were used to identify the important areas such as the RGD-containing GH-loop in VP1. Later studies (Fry *et al.*, 1999; Fry *et al.*, 2005) used crystal structures to identify binding sites for HSPG on the capsid. The HSPG binding site was identified as a shallow depression at the junction of VP1, VP2 and VP3. Residue 56 of VP3 was identified as being important in the interaction with HSPG (Jackson *et al.*, 1996). In the wild type, residue 56 is a histidine but upon cell culture adaptation, this changes to an arginine which associates with high affinity to HSPG. Curry and co-workers (1996) postulated that the GH-loop flexibility affects the movement and interaction of VP3 and in turn mutations in VP2 affect the GH-loop on VP1.

Dr. F. Maree and co-workers have shown *in vitro* with FMDV that two of the SAT2 serotype capsids (ZIM/5/83/2, ZIM/7/83/2) differ by only six residues (located on the surface), yet ZIM/5/83/2 has more infectious particles and is more stable following treatment at pH 6.0 than ZIM/7/83/2 (unpublished work). At pH 6.0 infectious particles of ZIM/5/83/2 could still be detected while ZIM/7/83/2 lost infectivity. ZIM/7/83/2 also adapted to using HSPG to infect cells and kills cells with a high efficiency. In contrast, ZIM/5/83/2 does not use HSPG to infect cells and has a low cell-killing efficiency. The HSPG adaptation is a known result from viral passage through cultured cells (Sa-Carvalho *et al.*, 1997; Fry *et al.*, 1999). This is important in their vaccine research work and implies that small mutations on the capsid can play a vital role in capsid stability and infectivity. The work presented here will try to characterize the variation and link it to the structure of the capsid proteins in an attempt to explain the results seen *in vitro*.

5.2. Methods

The complete modelling of a virus capsid is very time consuming and resource intensive. An alternative approach is to use a protomer (in this case the assembly formed by VP1-4) and then, using symmetry operations, generate the complete capsid assembly. 1ZBE (Fry *et al.*, 2005) from the PDB was used to generate the complete capsid. This resulted in a complete virus capsid which showed the interactions between the different chains. It also showed the pore structures that are involved in ion movement into and out of the virus.

5.2.1. Capsid Protomer

Protomer models of six SAT2 strains were constructed (ZAM/7/96/2, ZIM/14/90/2, ZIM/17/91/2, ZIM/5/83/2, ZIM/7/83/2, SAU/6/00/2) based on the crystallographic coordinates of O1BFS (1FOD) (Logan *et al.*, 1993). With the exception of SAU/6/00/2, the remaining strains have been found to be prevalent serotypes of the SAT2 family in the western and northern geographical regions of Southern Africa. The sequence data for the strains were provided by Dr. Francois Maree from the TADP, Agricultural Research Council, South Africa. Alignments for all the models were done with ClustalX using the default parameters, the modelling scripts were generated using the Structural module in FunGIMS and models were built using Modeller 9v1 (Fiser and Sali, 2003).

A PROPKA (Li *et al.*, 2005) analysis of each protomer (ZIM/5/83/2 and ZIM/7/83/2) was also done to assist in identifying major protonation states affected by a pH of 6.0. Yasara was used to analyze any hydrogen bond networks present.

5.2.2. Capsid Pentamer

A model of a pore (capsomer, 5-fold symmetry) consisting of five protomers was selected from the generated capsid model by deleting all unnecessary chains. This was used as a basis to investigate the effect of the different mutations found in the various strains and the way in which they influence chain-chain interactions as well as protomer-protomer interactions. Pentamer models of ZIM/5/83/2 and ZIM/7/83/2 were also built using the 1ZBE-generated capsid (strain A1061) as template. Alignments were done with ClustalX using the default parameters, modelling scripts were generated with the Structural module of FunGIMS and models with Modeller 9v1. The template lacked certain residues and for the modelling process these residues had to be removed from the targets due to no template matching (residues 140-158 from VP1, the first 7 residues of VP2, the first 14 residues from VP4 and residues 40-59).

To investigate pH-dependant differences between ZIM/5/83/2 and ZIM/7/83/2 pentamers, a molecular dynamics simulation was done for $\sim$2.5ns. Yasara was used to do the dynamics. The simulation was run at a pH of 6.0, water density of 0.997 g/ml, a NaCl concentration of 0.9%, using the Amber99 forcefield with periodic boundary conditions at a temperature of 298K. These simulation conditions were applied to both the respective protomers as well as the pentamers. A molecule consisting of two protomers (henceforth called the dimer) was also generated to analyze the interface between two pentamers.

5.3. Results and Discussion

5.3.1. Capsid Modelling

The complete capsid was generated with symmetry operations and used as a template for the investigation of the various proteins involved in capsid assembly (Fig. 5.2). The pore is located at the 5-fold axis (Fig. 5.3) and is comprised of five protomers. One VP1 chain from each protomer forms the pore. This was used as a basis for investigating the interactions between the five protomers and the chains in the protomers. Figure 5.3 shows the interaction between the VP2 and VP3 chain in the five protomers.

After analysis and structural mapping of the variation it is clear that the core of the capsids is quite conserved. The observed variation is probably the result of the quasi-species nature of the FMDV genome and positive selection pressure exerted on phenotype level. Variation seemed to occur mostly on surface areas and areas close to protein-protein interfaces. Although most of the variation is neutral, some of the variable residues result in the addition or loss of interactions. These specific differences may change the capsid assembly and disassembly dynamics slightly but none of the conserved amino acids identified as playing a role in capsid stability were affected. A far more detailed study of variation and structure would be required to identify individual interactions deemed to be important in capsid structure.

5.3.2. Protomer Modelling and Variation Mapping

Recently there has been considerable interest in the structural basis of the effect of pH on FMDV (Curry *et al.*, 1995). Furthermore, Doel and Baccarini (1981) reported a direct correlation between thermal stability of 146S particles and the protective ability of a

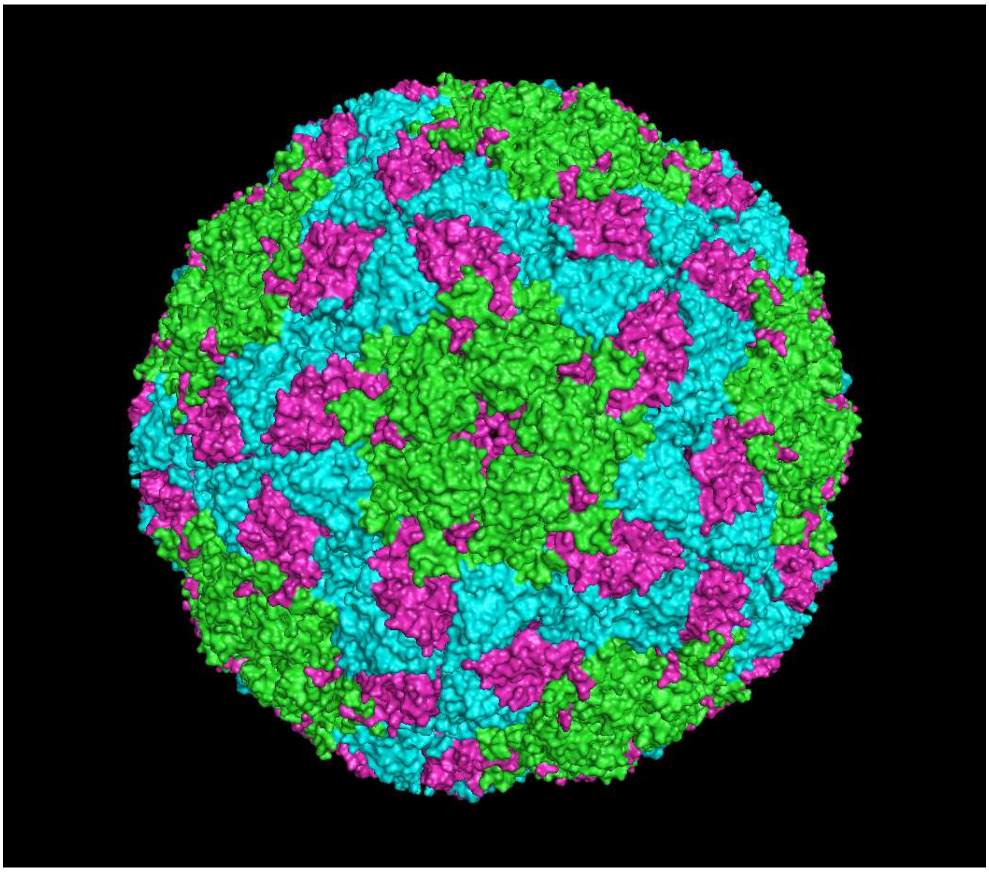


Figure 5.2: The capsid as generated from 1ZBE using symmetry operations. Green: VP1, Cyan: VP2, Magenta: VP3, VP4 - hidden on the inside of the capsid. The pore at the 5-fold axis is the in the centre of the image surrounded by 5 VP1 chains (green).

vaccine. Dr. F.F. Maree and colleagues examined the stability of SAT2 viruses from different topotypes in southern Africa (Haydon *et al.*, 2001; Bastos *et al.*, 2003) as well as a SAT3 virus to different pH environments to compare the phenotypic variance within these serotypes. The southern Africa SAT2 and SAT3 isolates can be divided into three lineages based on 1D phylogenetics, supporting a southern, western and northern clusters (Haydon *et al.*, 2001; Bastos *et al.*, 2003). Two of the viruses, i.e. SAT2/ZIM14/90 and SAT2/ZIM/17/91, belong to the western lineage of SAT2 viruses. A third SAT2 virus, belonging to the northern lineage of southern Africa SAT2 isolates, i.e. SAT2/ZAM/7/93 was included in this study. Also available was a SAT3 virus from the same geographical region, designated as SAT3/ZAM/4/96. Treatment of the SAT2 and SAT3 viruses with a buffer of pH 6.0 revealed differences toward there stability in mild acidic environment even within a serotype. Both the SAT2 and SAT3 Zambian isolates lost their infectivity

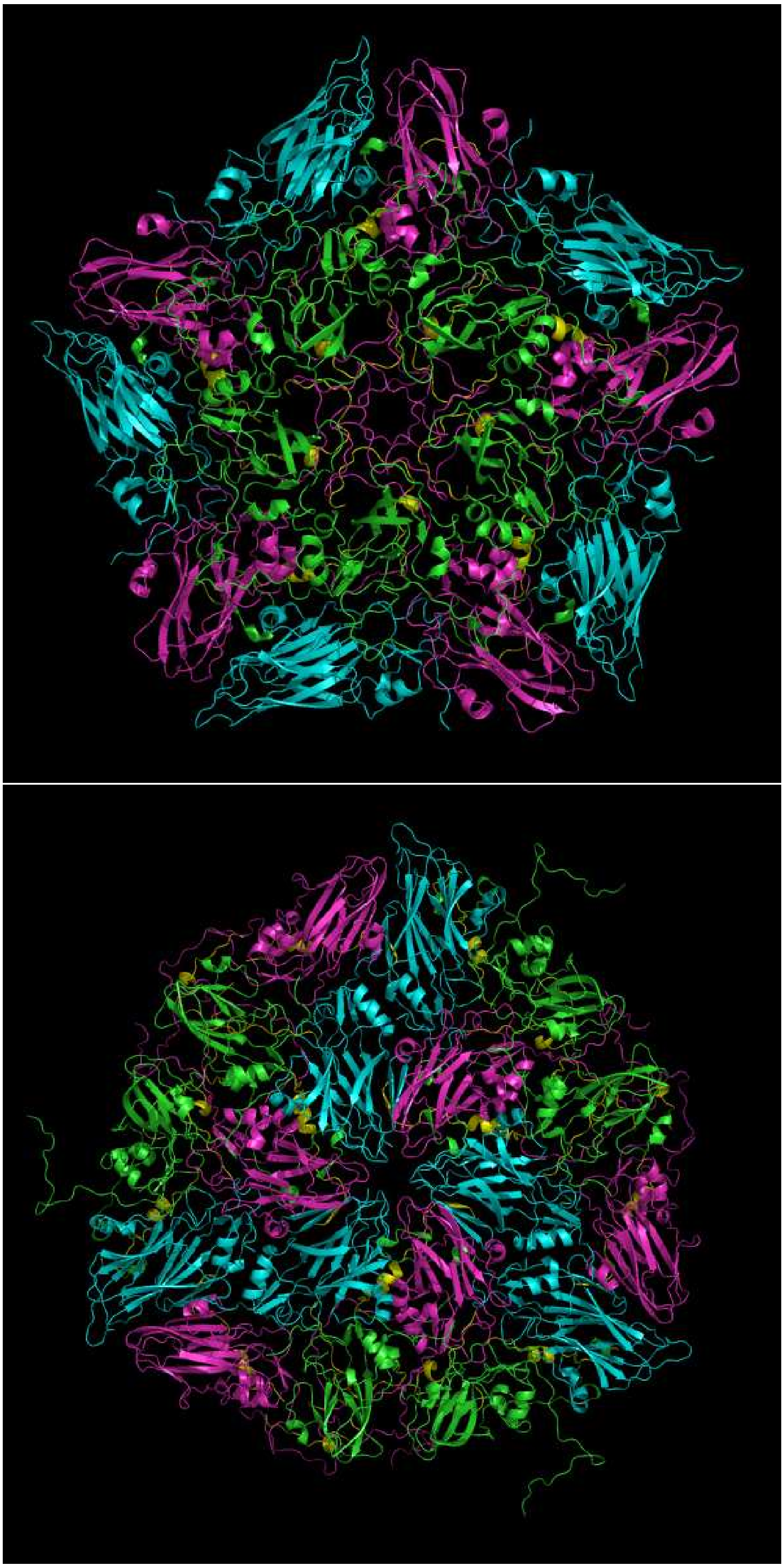


Figure 5.3: Top: A complete 5-fold pore assembly comprised of 5 protomers. Bottom: A 6 protomer complex surrounding the 3-fold pore showing the association between VP2 and VP3. Green: VP1, Cyan: VP2, Magenta: VP3, Yellow: VP4.

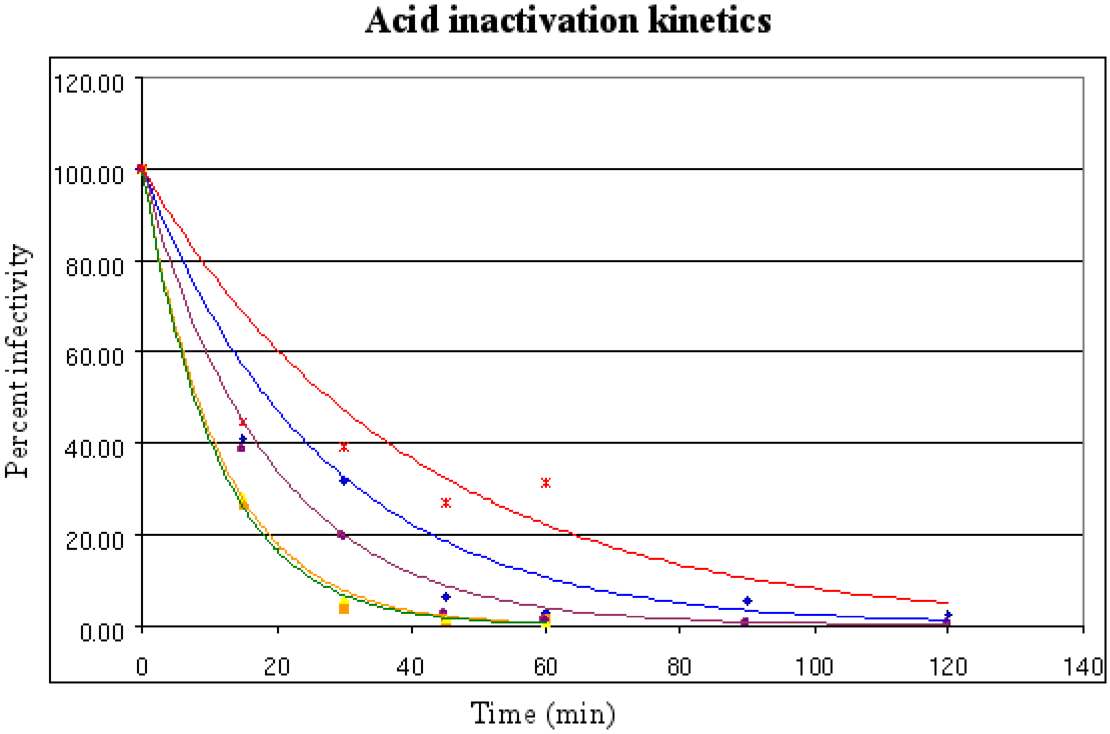


Figure 5.4: The sucrose density gradient purified viruses at an approximate titre of 4-9^{106} were treated at pH 6.0 for different lengths of time following a 1:50 dilution in the appropriate NET buffer (150mM NaCl, 10mM EDTA and 100mM Tris). The percentage of infectious particles remaining after treatment was determined by plaque titrations on BHK-21 cells and plotted against time. The exponential declines were used to calculate the inactivation rate constants (described by Mateo *et al.*, 2003).The acid inactivation kinetics of the viruses were reflected by the inactivation rate constants at pH 6.0, which were 0.025, 0.044, 0.065, 0.090 and 0.085 for ZIM/7/83, ZIM/14/90, ZIM/17/91, ZAM/7/96 and SAT3/ZAM/4/96, respectively. Red: ZIM/7/83. SAT2: blue - ZIM/17/91, purple - ZIM/14/90, green -ZAM/7/96. SAT3: Yellow - ZAM/4/96. Data courtesy of Dr F.F. Maree.

completely at a pH of 6.0 with 30 minutes treatment (Fig. 5.4). In contrast all three the SAT2 isolates from the western lineage of southern Africa isolates revealed significant drop in titres following 30 minutes at pH6.0 but in one instance, i.e. ZIM/7/83, at least 40% infectivity was still present after 1h incubation. Since FMDV relies on acid induced disassembly of the capsid proteins for infection and release of RNA this variation in the acid stability of SAT2 virions was further investigated by mapping the amino acid variation on the modelled 3D structure of a SAT2 virion.

Protomer models for six SAT2 strains, summarised in Table 5.1 - 5.3, were built using the alignments in Figure 5.5. The resulting models were compared to the pore model as well as to one another. All differences were classed into three categories: no effect (normal variation or surface exposed without any change in local structure or interactions), effect on intra-protomer association and effect on inter-protomer association. The results are summarized in Tables 5.1, 5.2 and 5.3. As can be expected, most of the variation was found in the VP1 chain, the most variable of the capsid proteins. VP4 did not show any significant differences. Most of the differences seen could have a possible effect on protomer-protomer interaction although in isolation, single differences might have a very small effect. Overall it seems that most of the differences in the chain could have have a small effect on the inter-protomer interaction and to a far lesser degree, intra-protomer interaction.

The variation in the capsid was also mapped to a model of the pentamer (Fig. 5.6). The variation only included mutations that would change the type or amount of interaction. This showed that such variation mostly occurs on interfaces and, significantly, around the pore and pore wall at the 5-fold axis. Most of the mutations do not appear to influence the structure, but some of the variation around the pore wall could have effects with regard to other virus functions such as adhesion and ion movement.

5.3.3. Pentamer Molecular Dynamics

SAT2/ZIM/7/83 was considered an efficient vaccine strain for many years in the southern Africa region in view of the fact that it produced high yields of 146S antigen, was considered to be a stable virus in the production process and elicited a strong immune response (Esterhuysen *et al.*, 1988). The recent inability to produce sufficient yields of 146S particles in cell culture monolayers lead us to investigate genetic changes that may affect the stability of the virus. ZIM/5/83/2 and ZIM/7/83/2 showed a difference of 6 residues (Table 5.4) and this resulted in a differing stability at pH 6. The pH_{50} can be described as the half-way point in the transition of 146S infectious particles into 12S pentamers and was described by Curry *et al.*, 1995 as a measure of pH sensitivity. The pH_{50} for both the SAT2 viruses (Fig. 5.7) were similar at pH 6.6 and comparable to serotype A viruses (Curry *et al.*, 1995). Nevertheless, between pH 5.8 and 6.3 the infectious particles deteriorate rapidly, probably as a result of break down into 12S

Table 5.1: The results from a comparison of the VP1 chain of the 6 SAT2 strains used in this study. Differences that do not have an influence on interaction were ignored (e.g. Ile -> Val). Strains: 1: ZAM/7/96, 2: ZIM/14/90, 3: ZIM/17/91, 4: ZIM/5/83, 5: ZIM/7/83, 6: SAU/6/00. The ZIM/7/83 proteome sequence was used as a reference sequence.

VP1	**Strains**						
#	**1**	**2**	**3**	**4**	**5**	**6**	**Effect**
6	E	E	G	E	E	E	Possible ionic interaction with VP2 (intra-protomer), Gly would disrupt this interaction.
21	R	S	S	A	A	N	Interaction with VP2 (inter-protomer), Arg, Asn might show stronger interaction with VP2.
23	V	A	M	T	T	Q	Interaction with VP2 (inter-protomer), different side chains might have different interaction strengths, Gln introduces a charge.
28	M	M	M	M	V	K	Interaction with VP3 (inter-, intra-protomer), Lys introduces a $\delta+$ charge.
39	F	F	F	F	F	S	Ser completely lacks the hydrophobic interaction present in other strains.
43	H	H	H	L	L	H	Exposed to surface. His may gain ionic interaction with VP1 (inter-protomer), Leu may disrupt interaction with VP1 (inter-protomer).
57	K	N	N	N	N	K	Ionic residue can interact with VP3 (inter-protomer), Lys lacks a $\delta-$ charge.
83	E	D	T	E	E	D	Situated on the exposed outer edge of the pore. Interacts with receptors in conjunction with residue 85. Thr lacks the $\delta-$ charge.
85	A	K	K	E	E	T	Situated in the wall of the pore, exposed to surface and might interact with receptors. Ala lacks any charge, Lys only presents $\delta+$ charge while Glu presents $\delta-$ charge.
101	G	R	R	R	R	G	Pore wall, ionic interaction with VP1, VP3 (inter-protomer), Gly lacks any side chain charge.
111	K	S	S	N	N	G	On the outer edge of the pore, longer side chains such as Lys may gain interaction with VP1 (interprotomer). Gly loses all possible charged interactions.
129	R	R	R	R	R	V	Val loses the charged interaction with between VP1 and VP2.
147	R	R	W	R	R	R	On surface, interactions with VP2, VP3 (intra-protomer).
200	S	G	G	A	A	T	Interaction with VP3 (inter-protomer), Ser might have one more hydrogen bond.

```
1FOD          1 TTSAGESADPVTTTVENYGGETQIQRRQHTDVSFIMDRFVKVTPQNQINILDLMQVPSHTLVGGLLRASTYYFSDLEIAVKHEGDLTWVPNGAPEKALDNTTNPTAYHKAPLTRLALPYT
SAU-6-00-2    1 TTSAGESADVVTTDPSTHGGNVQEGRRKHTEVAFLLDRSTHVHTNKTSFVVDLMDTKEKALVGAILLRASTYYFCDLEIACVGDHTRAFWQPNGAPRTTQLGDNPMVFAKGGVTRFAIPFT
ZAM-07-96-2   1 TTSAGEGADVVTTDPSTHGGRVVEKRRMHTDVAFVLDRFTHVHTNKTTFNVDLMDTKEKTLVGALLRASTYYFCDLEIACVGEHARVYWQPNGAPRTTQLGDNPMVFSHNKVTRFAIPYT
ZIM-14-90-2   1 TTSSGEGADVVTTDPSTHGGSVAEKRRMHTDVAFVMDRFTHVHTNKTAFAVDLMDTNEKTLVGALLRASTYYFCDLEIACIGDHKRVWWQPNGAPRTTQLRDNPMVFSHNSVTRFALPYT
ZIM-17-91-2   1 TTSSGGGADVVTTDPSTHGGSVMEKRRMHTDVAFVMDRFTHVHTNKTSFVIDLMDTNEKTLVGALLRASTYYFCDLEVACIGTHKRVWWQPNGAPRTTQLRDNPMVFSHNSVTRFALPYT
ZIM-05-83-2   1 TTSSGEGADVVTTDPSTHGGAVTEKKRMHTDVAFVMDRFTHVLTNRTAFAVDLMDTNEKTLVGALLRAATYYFCDLEIACLGEHERVWWQPNGAPRTTTLRDNPMVFSHNNVTRFAVPYT
ZIM-07-83-2   1 TTSSGEGADVVTTDPSTHGGAVTEKKRVHTDVAFVMDRFTHVLTNRTAFAVDLMDTNEKTLVGGLLRAATYYFCDLEIACLGEHERVWWQPNGAPRTTTLRDNPMVFSHNNVTRFAVPYT

1FOD        121 APHRVLATVYNGECRYSR--NAVPNLRGDLQVLAQKVARTLPTSFNYGAIKATRVTELLYRMKRAETYCPRPLLAIHPTE--ARHKQKIVAPVK/EETTLLEDRILTTRNGHTTSTTQSS
SAU-6-00-2  121 APHRLLSTVYNGECVYKKTPTAIRGDRAALAVKYADSTHTLPSTFNFGFVTVDKPVDVYYRMKRAELYCPRPLLPAYEHTGGDRFDAPIGVERQ/EETTLLEDRILTTRHGTTTSTTQSS
ZAM-07-96-2 121 APHRLLATRYNGECKYTQEARAIRGDRAVLAAKYAGAKHSLPSTFNFGHVTADAAVDVYYRMKRAELYCPRPLLPAYEHSDRDRFDAPIGVEKQ/EETTLLEDRIVTTRHGTTTSTTQSS
ZIM-14-90-2 121 APHRLLSTRYNGECNYTQRSPAIRGDRAVLAAKYANVKHELPSTFNFGFVTADKPVDVYFRMKRTELYCPRPLLPAYDHGDRDRFDAPIGVEKQ/EETTLLEDRIVTTRHGTTTSTTQSS
ZIM-17-91-2 121 APHRLLSTRYNGECKYTERATAIRGDWAVLAAKYANTKHELPSTFNFGFVTADEPVDVYYRMERAELYCPRPLLPVYDHGNRDRFDAPIGVEKQ/EETTLLEDRIVTTRHGTTTSTTQSS
ZIM-05-83-2 121 APHRLLSTRYNGECKYTQQSTAIRGDRAVLAAKYANTKHKLPSTFNFGHVTADKPVDVYYRMKRAELYCPRPLLPGYDHADRDRFDSPIGVEKQ/EETTLLEDRIVTTRHGTTTSTTQSS
ZIM-07-83-2 121 APHRLLSTRYNGECKYTQQSTAIRGDRAVLAAKYANTKHKLPSTFNFGHVTADKPVDVYYRMKRAAVYCPRPLLPGYDHADRDRFDSPIGVEKQ/EETTLLEDRIVTTRHGTTTSTTQSS

1FOD        236 VGVTYGYATAEDFVSGPNTSGLETRVVQAERFFKTHLFDWVTSDSFGRCHLLELPTDHKGVYGSLTDSYAYMRNGWDVEVTAVGNQFNGGCLLVAMVPELCSIQKRELYQLTLFPHQFIN
SAU-6-00-2  240 VGVTLGYADADSFRPGPNTSGLETRVQQAERFFKEKLFDWTSDKPFGTLYVLELPKDHKGIYGKLTDSYTYMRNGWDVQVSATSTQFNGGSLLVAMVPELSSLKSREEFQLTLYPHQFIN
ZAM-07-96-2 240 VGITYGYADADSFRPGPNTSGLETRVEQAERFFKEKLFDWTSDKPFGTLYVLELPKDHKGIYGSLTDAYAYMRNGWDVQVTATSTQFNGGSLLVALVPELCSLREREEFQLTLYPHQFIN
ZIM-14-90-2 240 VGITYGYADSDSFRSGPNTSGLETRVEQAERFFKEKLFDWTSDKPFGTLYILELPKDHKGIYGSLTESYAYMRNGWDVQVSATSTQFNGGSLLVAMVPELCSLRAREEFQLSLYPHQFIN
ZIM-17-91-2 240 VGITYGYADSDSFRPGPNTSGLETRVEQAERFFKEKLFDWTSDKPFGALYVLELPKDHKGIYGSLTESYAYMRNGWDVQVSATSTQFNGGSLLVAMVPELCSLRDREEFQLSLYPHQFIN
ZIM-05-83-2 240 VGITYGYADADSFRPGPNTSGLETRVEQAERFFKEKLFDWTSDKPFGMLYVLELPKDHKGIYGSLTDAYTYMRNGWDVQVSATSTQFNGGSLLVAMVPELCSLKDREEFQLSLYPHQFIN
ZIM-07-83-2 240 VGITYGYADADSFRPGPNTSGLETRVEQAERFFKEKLFDWTSDKPFGTLYVLELPKDHKGIYGSLTDAYTYMRNGWDVQVSATSTQFNGGSLLVAMVPELCSLKDREEFQLSLYPHQFIN

1FOD        356 PRTNMTAHITVPFVGVNRYDQYKVHKPWTLVVMVVAPLTVNT-EGAPQIKVYANIAPTNVHVAGEFPSKE/GIFPVACSDGYGGLVTTDPKTADPVYGKVFNPPRNQLPGRFTNLLDVAE
SAU-6-00-2  360 PRTNTTAHIQVPYLGVNRHDQGKRHHAWSLVVMVLTPLTTEAQMNSGTVEVYANIAPTNVVVAGELPGKQ/GIVPVAAADGYGGFQNTDPKTADPIYGYVYNPSRNDCHGRFSNLLDVAE
ZAM-07-96-2 360 PRTNTTAHIQVPYLGVNRHDQGKRHQAWSLVVMVLTPLTTETQMTSGTVEVYANIAPTNVFVAGEMPAKQ/GIVPVACADGYGGFQNTDPKTADPIYGYVYNPSRNDCHGRYSNLLDVAE
ZIM-14-90-2 360 PRTNTTAHIQVPYLGVNRHDQGKRHQAWSLVVMVLTPLTTEAQMNSGTVEVYANIAPTNVFVAGEMPAKQ/GIIPVACSDGYGGFQNTDPKTADPIYGYVYNPSRNDCHGRYSNLLDVAE
ZIM-17-91-2 360 PRTNTTAHIQVPYLCVNRHDQGKRHQTWSLVVMVLTPLTTEAQMNSGTVEVYANIAPTNVFVAGEKPAKQ/GIVPVACSDGYGGFQNTDPKTADPIYGYVYNPSRNDCHGRYSNLLDVAE
ZIM-05-83-2 360 PRTNTTAHIQVPYLGVNRHDQGKRHQAWSLVVMVLTPLTTEAQMQSGTVEVYANIAPTNVFVAGEKPAKQ/GIIPVACFDGYGGFQNTDPKTADPIYGYVYNPSRNDCHGRYSNLLDVAE
ZIM-07-83-2 360 PRTNTTAHIQVPYLGVNRHDQGKRHQAWSLVVMVLTPLTTEAQMQSGTVEVYANIAPTNVFVAGEKPAKQ/GIIPVACFDGYGGFQNTDPKTADPIYGYVYNPSRNDCHGRYSNLLDVAE

1FOD        474 ACPTFLRFEGGVPYVTTKTDSDRVLAQFDMSLAAKHMSNTFLAGLAQYYTQYSGTINLHFMFTGPTDAKARYMVAYAPPGME---PPKTPEAAAHCIHAEWDTGLNSKFTFSIPYLSAAD
SAU-6-00-2  479 ACPTLLDFD-GKPYIVTKNNGDKVMTSFDVAFTHKVHRNTFLAGLADYYTQYSGSLNYHFMYTGPTHHKAKFMVAYVPPGVETAQLPTTPEDAAHCYHAEWDTGLNSSFSFAVPYISAAD
ZAM-07-96-2 479 ACPTLLNFD-GKPYVVTKNNGDKVMTCFDVAFTHKVHKNTFLAGLADYYTQYQGSLNYHFMYTGPTHHKAKFMVAYIPPGVETDKLPKTPEDAAHCYHSEWDTGLNSQFTFAVPYVSASD
ZIM-14-90-2 479 ACPTFLDFD-GKPYVVTKNNGDKVMTCFDVAFTHKVHKSTFLAGLADYYTQYQGSLNYHFMYTGPTHHKAKFMVAYIPPGTATDKLPKTPEDAAHCYHSEWDTGLNSQFTFAVPYVSASD
ZIM-17-91-2 479 ACPTFLNFD-GKPYVVTKNNGDKVMTCFDVAFTHKVHKNTFLAGLADYYTQYQGSLNYHFMYTGPTHHKAKFMVAYIPPGVETDKLPKTPEDAAHCYHSEWDTGLNSQFTFAVPYVSASD
ZIM-05-83-2 479 ACPTFLNFD-GKPYVFTKNNGDKVMTCFDVAFTHKVHKNTFLAGLADYYAQYQGSLNYHFMYTGPTHHKAKFMVAYIPPGIETDRLPKTPEDAAHCYHSEWDTGLNSQFTFAVPYVSASD
ZIM-07-83-2 479 ACPTFLNFD-GKPYVVTKNNGDKVMTCFDVAFTHKVHKNTFLAGLADYYAQYQGSLNYHFMYTGPTHHKAKFMVAYIPPGIETDRLPKTPEDAAHCYHSEWDTGLNSQFTFAVPYVSASD

1FOD        591 YTYTASDVAETTNVQGWVCLFQITHGKADGDALVVLASAGKDFELRLPVDARA---------------E/SGNTGSIINNYYMQQYQNSMDTQLGDN-----------------------
SAU-6-00-2  598 FSYTHTDTPAMATTNGWVIVLQVTDTHSAEAAVVVSVSAGPDLEFRFPIDPVRQ/GAGQSSPATGSQDQ-SGNTGSIINNYYMQQYQNSMDTQLGDNAISGGSNEGSTDTTSTHTNNTQN
ZAM-07-96-2 598 FSYTHTDTPAMATTNGWVAVYQVTDTHSAEAAVVVSVSAGPDLEFRFPIDPVRQ/GAGQSSPATGSQNQ-SGNTGSIINNYYMQQYQNSMDTQLGDNAISGGSNEGSTDTTSTHTNNTQN
ZIM-14-90-2 598 FSYTHTDTPAMATTNGWVAVYQVTDTHSAEAAVVVSVSAGPDLEFRFPIDPIRQ/GAGQSSPATGSQNQ-SGNTGSIINNYYMQQYQNSMDTQLGDNAISGGSNEGSTDTTSTHTNNTQN
ZIM-17-91-2 598 FSYTHTDTPAMATTNGWVAVYQVTDTHSAEAAVVVSVSAGPDLEFRFPIDPVRQ/GAGQSSPATGSQNQ-SGNTGSIINNYYMQQYQNSMDTQLGDNAISGGSNEGSTDTTSTHTNNTQN
ZIM-05-83-2 598 FSYTHTDTPAMATTNGWVAVFQVTDTHSAEAAVVVSVSAGPDLEFRFPVDPVRQ/GAGHSSPATGSQNQ-SGNTGSIINNYYMQQYQNSMDTQLGDNAISGGSNEGSTDTTSTHTNNTQN
ZIM-07-83-2 598 FSYTHTDTPAMATTNGWVAVFQVTDTHSAEAAVVVSVSAGPDLEFRFPVDPVRQ/GAGHSSPVTGSQNQ-SGNTGSIINNYYMQQYQNSMDTQLGDNAISGGSNEGSTDTTSTHTNNTQN

1FOD        672 -DWFSKLASSAFSGLFGALLA----
SAU-6-00-2  716 NDWFSKLAQSAISGLFGALLADKKT
ZAM-07-96-2 716 NDWFSKLAQSAISGLFGALLADKKT
ZIM-14-90-2 716 NDWFSKLAQSAISGLFGALLADKKT
ZIM-17-91-2 716 NDWFSKLAQSAISGLFGALLADKKT
ZIM-05-83-2 716 NDWFSKLAQSAISGLFGALLADKKT
ZIM-07-83-2 716 NDWFSKLAQSAISGLFGALLADKKT
```

Figure 5.5: The alignments used to model the six SAT2 capsid protomers. The identity between the targets and template are all around 60% with a variation of 1%.

Table 5.2: The results of a comparison of the VP2 chain of the 6 strains used in this study. Differences that do not have an influence on interaction were ignored (e.g. Ile -> Val). Strains: 1: ZAM/7/96, 2: ZIM/14/90, 3: ZIM/17/91, 4: ZIM/5/83, 5: ZIM/7/83, 6:SAU/6/00. The ZIM/7/83 proteome sequence was used as a reference sequence.

VP2	**Strains**						
#	**1**	**2**	**3**	**4**	**5**	**6**	**Effect**
51	E	E	E	E	E	Q	Charged Gln can affect protomer interaction.
88	S	S	S	S	S	K	Charged Lys can interact more strongly with other protomers.
91	A	S	S	A	A	S	Interaction with VP2 (inter-protomer), Ser might induce an extra hydrogen bond.
93	A	A	A	T	T	T	Interaction with VP3 (intra-protomer), Thr might induce an extra hydrogen bond.
188	T	N	N	Q	Q	N	Interaction with VP3 (inter-protomer), Thr might disrupt the ionic interactions seen in Gln and Glu.
209	M	M	K	K	K	L	Interaction with VP2 (inter-protomer), Met, Leu might disrupt the ionic interactions seen in Lys.

pentameric units (Curry *et al.*, 1995; Knipe *et al.*, 1997; Mateo *et al.*, 2003). The rate of loss of infectious particles was not equal for ZIM/7/83 and ZIM/5/83 at the low pH range (Fig. 5.7), with the infectivity of ZIM/7/83 deteriorating more rapidly than ZIM/5/83 below pH 6.2. Although the starting titer of the two viruses was normalized, the ZIM/5/83 repeatedly end up with approximately 10-80 infectious particles at pH 5.8 and 5.6 respectively, while no ZIM/7/83 infectious particles were present below pH 6.0. However, no infectious particles was repeatedly observed for ZIM/7/83 at these pH conditions. The biological significance of these difference were investigated using models of the 12S pentamers.

Molecular dynamics simulations of the ZIM/5/83/2 and ZIM/7/83/2 pentamers were run for ~2.5ns. Figure 5.8 shows the RMSD variation over time for each of the two different pentamers over the simulation time at pH 6.0. The protomer simulations of ZIM/5/83/2 and ZIM/7/83/2 were run for ~2.2ns. The RMSD variation over time are shown in Figure 5.8. There was no significant difference in the RMSD of either the pentamers or the protomers of ZIM/5/83/2 and ZIM/7/83/2. Any significant difference such as a pentamer dissociation would have shown highly divergent RMSD values.

The PROPKA results showed that there were four interesting His residues to investigate.

Table 5.3: The results from a comparison of the VP3 chain of the 6 strains used in this study. Differences that do not have an influence on interaction were ignored (e.g. Ile -> Val). Strains: 1: ZAM/7/96, 2: ZIM/14/90, 3: ZIM/17/91, 4: ZIM/5/83, 5: ZIM/7/83, 6:SAU/6/00. The ZIM/7/83 proteome sequence was used as a reference sequence.

VP3	**Strains**						
#	**1**	**2**	**3**	**4**	**5**	**6**	**Effect**
3	V	I	V	I	I	V	Forms part of the central pore, has an effect on the size of the pore. Other serotypes have a Phe in this position.
8	A	S	S	F	F	A	Situated in the pore opening. The Phe will close up the pore and might be a compensatory mutation for position 3. Other serotypes have an Ala in this position. The Ala and Ser is smaller in size and thus allows for a slightly bigger pore.
54	L	F	F	F	F	L	Hydrophobic interactions with VP2 (intra-protomer), Leu lacks ring which reduces hydrophobicity.
64	V	V	V	F	V	V	Surface exposed but possible interaction with VP2 (inter-protomer). Phe might disrupt sheet formation slightly.
87	N	S	N	N	N	N	Surface exposed, interaction with VP1 (inter-protomer). The Ser interaction might be slightly less due to the OH group.
98	T	T	T	A	A	T	Ionic interaction with VP1 (intra-protomer). The Ala lacks an OH group to form hydrogen bonds.
129	V	T	V	I	I	V	In combination with site 130, this forms the binding area for Heparan sulfate, interaction with VP2 (inter-protomer). The Thr OH group might form extra interactions thus compensation for the Ala in position 130.
130	E	A	E	E	E	E	In combination with site 129, this forms the binding area for Heparan sulfate, interaction with VP2 (inter-protomer). The Ala will result in a loss of hydrogen bonds when compared to Glu.
137	K	K	K	K	K	T	Thr disrupts charged inter-protomer interaction.

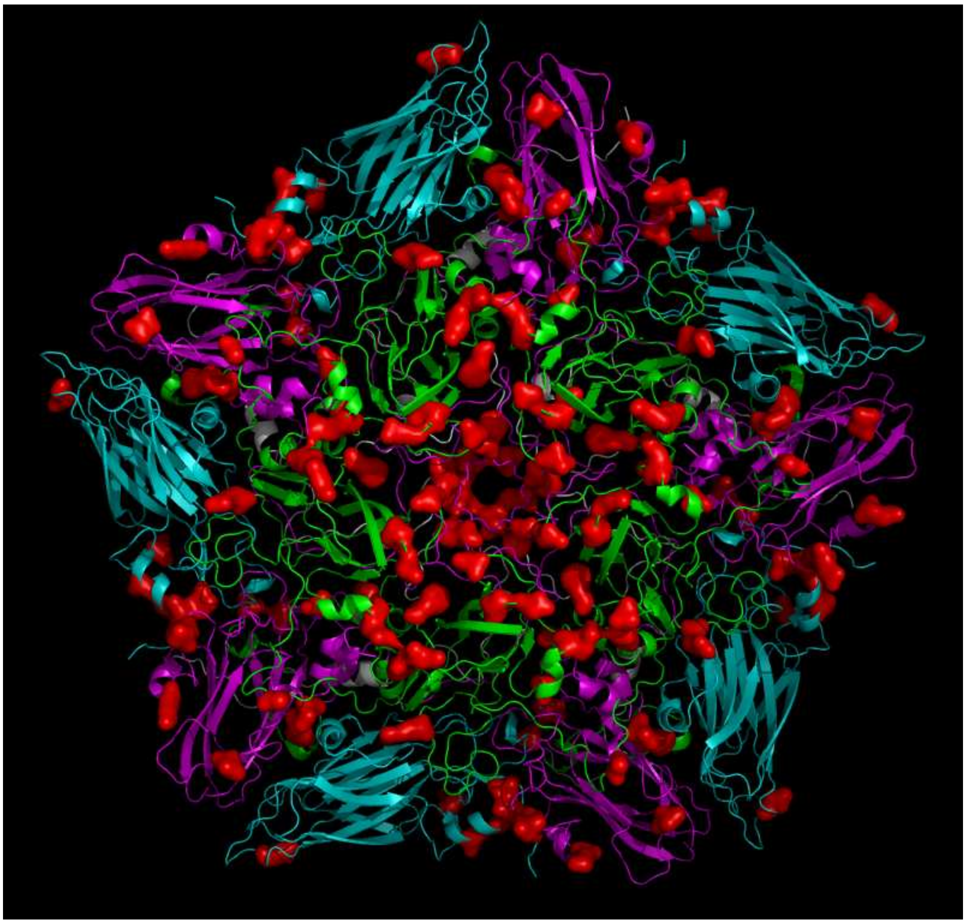


Figure 5.6: The variation seen in VP1-3 mapped to a 5-fold axis model of the protomers. Variable positions are coloured red. Green: VP1, Cyan: VP2, Magenta: VP3.

These had a shift from below a pKa of 6.0 to above a pKa of 6.0 between ZIM/5/83/2 and ZIM/7/83/2. The four His residues were: His 511 (81), His 545 (115), His 575 (145) and His 602 (172). The numbers in brackets are the residue numbers as referred to in Ellard *et al.*, 1999 (Table 5.5). The PROPKA results for the generated dimer showed different results as most of the His residues identified in the protomers were buried in the dimer interface. This difference in result was due to the fact that when the pentamers associate to form the dimer, the His residues identified are buried in the interface and thus excluded from any water contact. This changed the solvent environment around the His residues.

Molecular dynamics simulation was done for each for each of the pentamers of ZIM/5/83/2 and ZIM/7/83/2. The pentamer models were both built on the same template and thus

Table 5.4: The differences between the P1 peptide of ZIM/5/83/2 and ZIM/7/83/2.

Res #	**ZIM/5/83/2**	**ZIM/7/83/2**
28	Met	Val
64	Ala	Gly
186	Glu	Ala
187	Leu	Val
287	Thr	Met
493	Phe	Val

Table 5.5: The pKa values for the four His residues identified by PROPKA as undergoing protonation changes at pH 6.0. The protomer of both ZIM strains were used for the respective pKa predictions and a diner generated from the ZIM/5/82/2 protomer.

	pKa		
His #	**ZIM/5/83/2**	**ZIM/7/83/2**	**Dimer**
511	3.21	7.07	3.21
545	6.15	5.94	6.15
575	5.92	7.62	-1.12
602	6.51	5.27	6.51

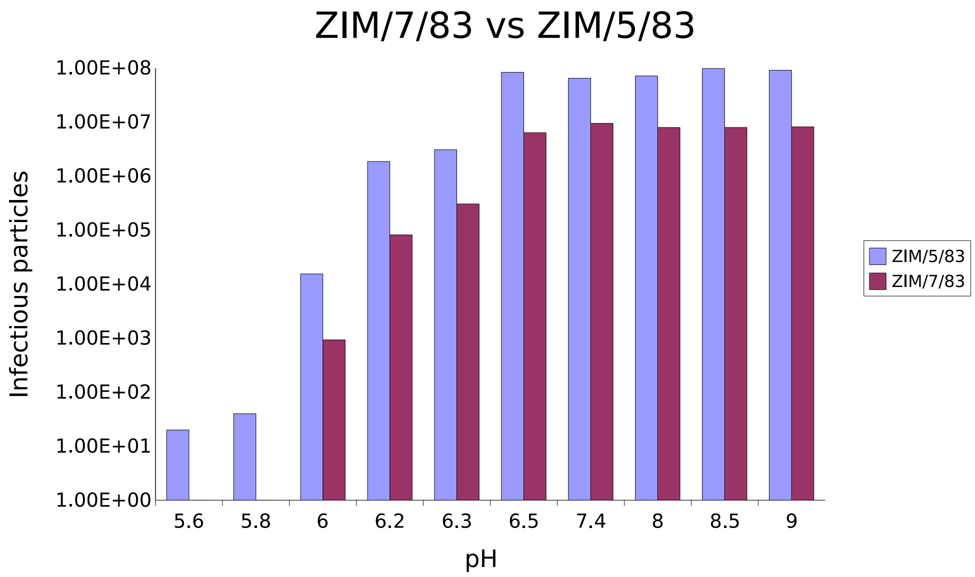
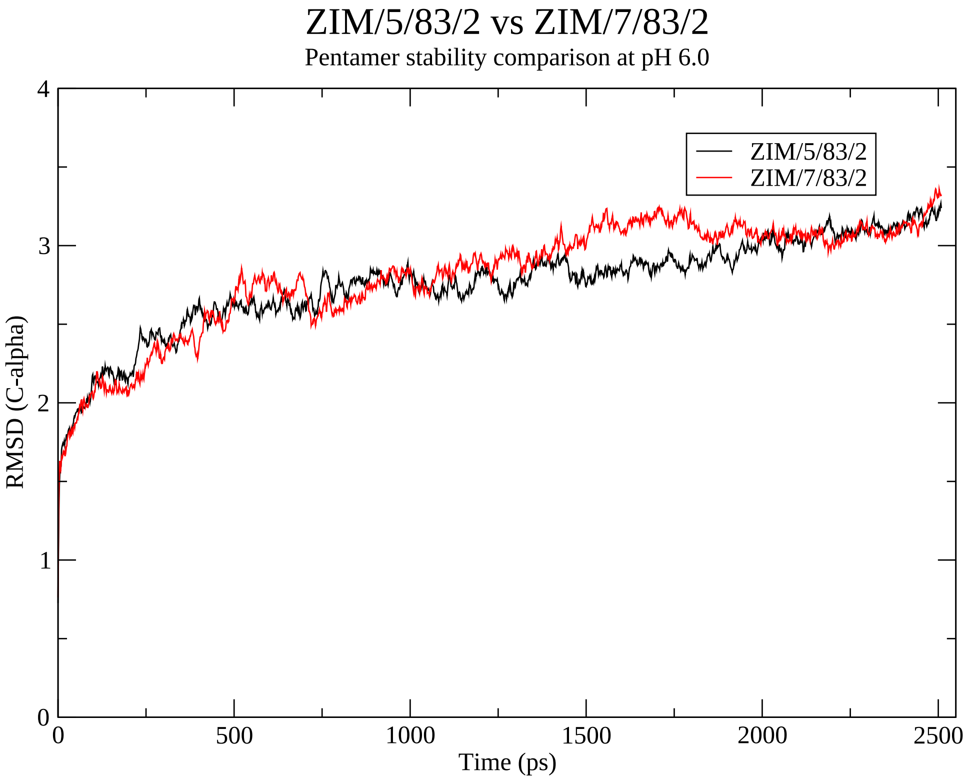


Figure 5.7: The sucrose density gradient purified ZIM/7/83 and ZIM/5/83 infectious 146S particles were incubated in buffered solutions spanning a pH range of 5.6 to 9.0. Following 30 min incubation the pH of the solution was restored and the amount of infectious particles remaining determined by titration on BHK-21 cells. Both SAT2 infectious particles were stable at a wide range of pH conditions from 6.5 to 9.0 for a period of 30 min and up to 2 hours (data not shown). At pH 6.5 at least 35% and 38% of infectious particles for ZIM/7/83 and ZIM/5/83 respectively, were still present after 30 minutes. Data courtesy of Dr F.F. Maree.



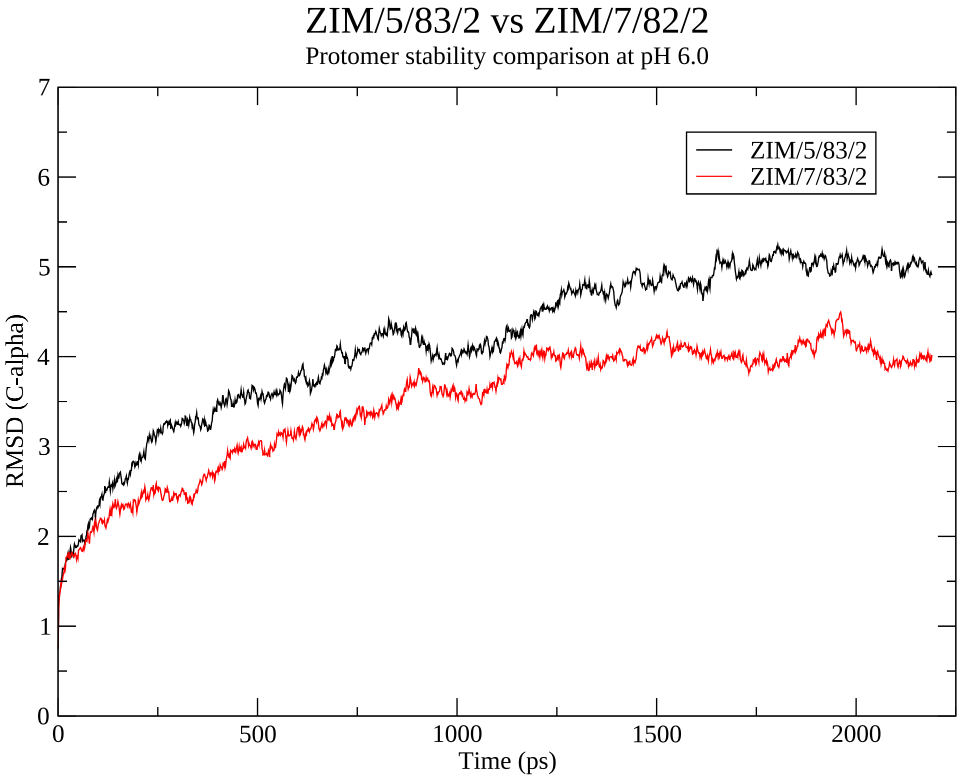


Figure 5.8: The Cα RMSD variation of ZIM/5/83/2(black) and ZIM/7/83/2 (red) over the ∼2.5ns simulation time at pH 6.0. Top: Pentamer stability. Bottom: Protomer stability.

after a dynamics simulation, the results in terms of RMSD deviation from the model could be compared. After the 2.5ns simulation run, it was seen that there was no significant difference in RMSD between the pentamers. The graphs showed that the RMSD deviation started to flatten out and thus is was concluded that the pentamers were stable. *In vitro* evidence showed at pH 6.0 ZIM/7/83/2 was less stable than ZIM/5/83/2. There was only a 6 residue difference between the two pentamers and none of these residues were predicted to show any change in protonation state from pH 7.0 to pH 6.0. Thus it was speculated that the lower pH may disrupt general association between the protomers in the pentamers. The simulation showed that no major changes occurred as can be seen in Figure 5.8. A major change, such as the pentamer dissociating, would have showed prominently on a graph plotting RMSD. In order to investigate whether the disruption occurs at protomer level, molecular dynamics simulations were run on the respective protomers as well. The results showed that the protomers were stable and thus the residues had no effect on protomer stability (Fig. 5.8). The difference between the RMSD levels of the two plots are as a result of the presence of loops in the protomer. The movement of these loops will affect the RMSD calculations but not to such an extent as to mask an unstable protomer. A factor to consider is that some residues, mainly in VP4, could not be resolved from the electron density maps during structure determination and could thus not be modelled. This simulation showed that when considering RMSD, the protomers stayed relatively stable with no major increase in RMSD as would have been expected for a protomer dissociating. This implies that the pH disruption occurs at another level. This dissociation may be investigated in the future by using binding interaction studies on the individual components using equipment such as a biosensor.

It was decided to perform a pKa prediction on both protomers as well as the dimer to see whether there is any change in pKa. The PROPKA pKa prediction results for the protomers indicated four interesting His residues which change protonation states around pH 6.0. These residues were inspected manually. When considering the pattern of binding by these His residues, it would appear that ZIM/5/83/2 and ZIM/7/83/2 do not gain or lose a nett amount of bonds (Table 5.6). However when the residues are mapped to the

Table 5.6: The changes in pKa for the four His residues in the protomer identified by PROPKA as undergoing protonation changes at pH 6.0. All residue numbers refer to the residues in the full model.

His #	ZIM/5/83/2	ZIM/7/83/2	
511	3.21	7.07	This His is exposed to the surface and thus the solvent environment would affect this pKa massively. No conclusions can be drawn about this residue.
545	6.15	5.94	In ZIM/5/83/2 the His is pointing towards solvent, whereas in ZIM/7/83/2 it is pointing inwards towards the protein. It also interacts with the adjacent pentamer.
575	5.92	7.62	This His is pointing towards the interface with another protomer. It is implicated in interprotomer association.
602	6.51	5.27	Exposed to the surface.

structure a different picture emerges. From the structure it can be seen that all these changes occur on the VP3 chain (Fig. 5.9).

His 575 (145 in chain C) in VP3 is associated in interprotomer interaction (Curry *et al.*, 1995; Ellard *et al.*, 1999). Hydrogen bond analysis of the dimer molecule indicated that His 575 (145) interacts with Ala 571 (141 on chain C of protomer 1) and with Lys 273 (63 on chain B of protomer 2). The PROPKA results for the dimer molecule show the pKa for His 575 to be -1.12. It must be kept in mind that this is a statistical calculation (in a non-water environment) and implies that the residue is deprotonated most of the time. The fact that it is not exposed to solvent influences the pKa predictions as well. Thus, from these results it appears that a pH below 6.0 would prevent the formation of the capsid, as pentamers cannot assemble. It appears that a significant proportion of His 575 (145) needs to be neutral for pentamers to assemble into a capsid. This confirms the work done by van Vlijmen and co-workers (1998) in which they calculated that His 575 (145) may play a role in capsid disassembly (van Vlijmen *et al.*, 1998) and the work of Twomey and co-workers (Twomey *et al.*, 1995) on FMDV vaccine stability. Various authors (Curry *et al.*, 1995; Ellard *et al.*, 1999) also showed that His 572 (142) was important in association between the pentamers. The hydrogen bond analysis showed that His 575 (145) made a hydrogen bond with the backbone of Ala 571 (141), which is

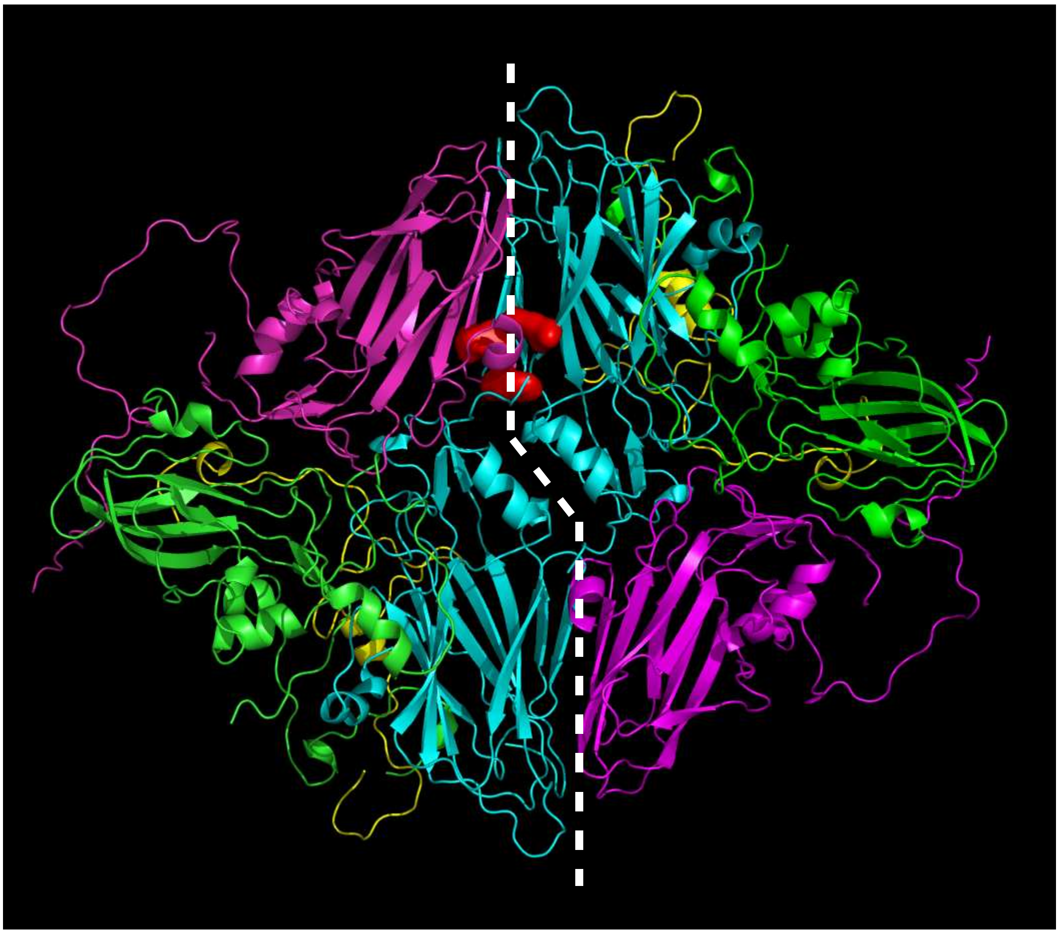


Figure 5.9: The interaction interface between two pentamer sections. One protomer of each pentamer is shown. The dashed line indicates the interaction surface. His 575 (145), His 572 (142) and Lys 273 (63) are coloured red using Van der Waals surfaces.

located right next to His 572 (142) (Fig. 5.10). This backbone hydrogen bond seems to be important is helping to orientate the His 572 (142) containing loop correctly to form the association with the charged dipole of the α-helix. His 575 (145) also makes a hydrogen bond with Lys 273 (63) in the adjacent pentamer, thus providing extra interaction and stabilization between the pentamers. The loss of the hydrogen bonds with either Lys 273 (63) or Ala 571 (141) would have a significant effect on the interaction interface.

The PROPKA pKa analysis predicts that the V493F mutation affects the pKa values of the ZIM/5/83/2 and His 575 and thus makes it neutral above pH 5.92, while ZIM/7/83/2 is neutralized at a higher pH. Although the distance between Phe 493 (63 on chain C) and His 575 (145) is ~17.5 Å, long distance effects transmitted through the β-sheet cannot

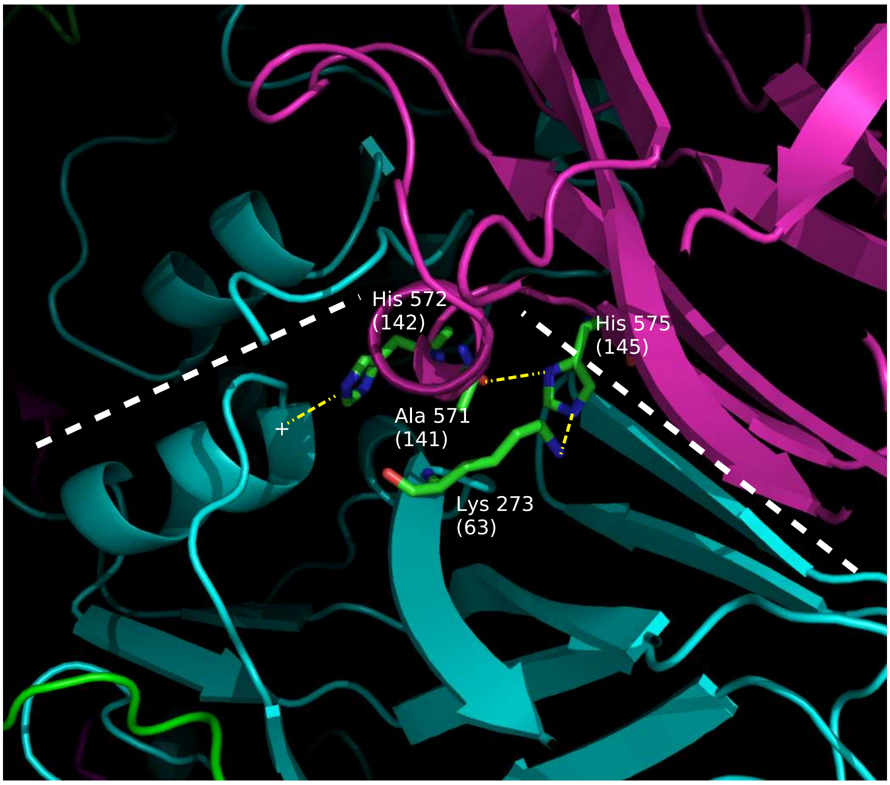


Figure 5.10: The hydrogen bond network found in the pentamer interface. When His 575 (145) is neutral, it makes a hydrogen bond with Lys 273 (63) and Ala 571 (141). The neutral state seem to prevent pentamer association through His 572 (142) and His 575 (145). Yellow dashed lines indicate hydrogen bonds and white dashed line indicates pentamer interface. The "+" indicates the charged dipole of the α-helix.

be ruled out (Fig. 5.11). Thus the neutral His 575 (145) seems to be vital for pentamer assembly. This result is similar to the one noted by Curry and co-workers (Curry *et al.*, 1995) in which it was found that subtype A10 was more stable than A22 by 0.5 pH units and shows that there is variation . It must be kept in mind that these results are based on mostly statistical predictions and that experimental work is required to confirm the results.

5.4. Conclusion

Protein-protein recognition mediates many fundamental biological processes. A detailed knowledge of these processes requires the determination of the structural, energetic, and functional roles of individual amino acid residues and interactions in protein-protein in-

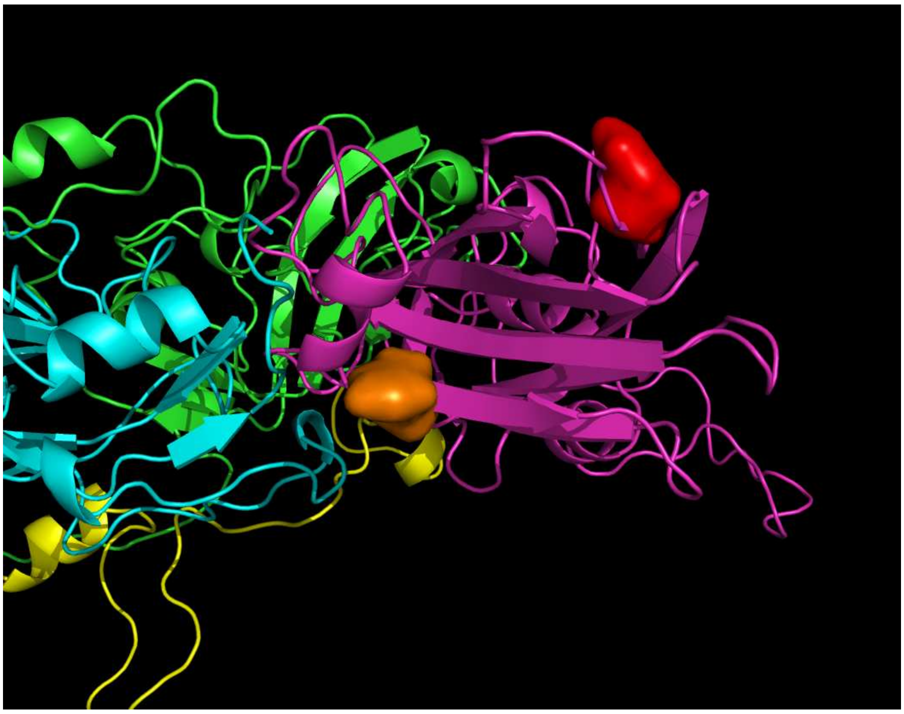


Figure 5.11: A side-on view of VP3 with the location of Phe 493 (63) in relation to His 575 (145). The distance between the residues are ∼17.5 Å. VP1: green, VP2: cyan, VP3: magenta, Phe 493: red and His 575: orange.

terfaces. These studies have been generally undertaken by using small protein-ligand complexes or oligomeric proteins of moderate size (Reguera *et al.*, 2004). In contrast, for multimeric protein complexes, such as viral capsids (Liljas, 1986; Hadfield *et al.*, 1997) or large cellular assemblies, little is known about the specific molecular determinants of protein association and stability. Mutational studies of virus capsids, generally focused on a few specific amino acid residues, have provided important insights (Ellard *et al.*, 1999; Mateo *et al.*, 2003). However, exhaustive experimental studies on the relative importance of residues and molecular interactions in viral capsid assembly, disassembly, and or stability are still limited. These studies contribute also to the understanding of protein structure-function relationships and they could be exploited possibly in the design of thermostable vaccines and antiviral agents promoting capsid disassembly or interfering with assembly (Wien *et al.*, 1996; Hadfield *et al.*, 1997; Diana *et al.*, 1997; Belnap *et al.*, 2000).

Many viruses, including viruses of medical or veterinary significance, have capsids of icosahedral symmetry (Reguera *et al.*, 2004). FMDV is a small non-enveloped virus with a pseudo T=3 icosahedral capsid formed by 60 copies each of four nonidentical polypeptide chains, i.e. VP1, VP2, VP3 and VP4. There has been considerable interest in the structural basis of the effect of pH on FMDV (Curry *et al.*, 1995). Multiple evidence on the structural data of FMDV had been gathered by high resolution X-ray crystallography in recent years that allow the identification of residues involved in stabilising the virion structure. Assembly of the picornaviral capsid proceeds in several steps (Rueckert, 1996). The capsid proteins VP0 (1AB), VP3 (1C), and VP1 (1D) are translated as a polyprotein precursor (P1), may fold co-translationally (Rossmann and Johnson, 1989), and are proteolytically processed by 3C^{pro} (Birtley *et al.*, 2005) to yield the mature protomer. Five protomers are assembled to form a pentameric intermediate, and finally, 12 pentamers are assembled to form the icosahedral capsid (Fig. 5.1). After encapsidation of the RNA genome most VP0 molecules are processed to give VP4 (1A, the N terminus of VP0) and VP2 (1B). Disassembly of the FMDV virion *in vivo* begins with its dissociation into pentamers (Vasquez *et al.*, 1979) by acidification in the endosomes (Carrillo *et al.*, 1984). Furthermore, Doel and Baccarini (1981) reported on a direct correlation between thermal stability of 146S particles and the protective ability of an antigen/vaccine. It was found that mild heating of FMDV virions leads to irreversible dissociation into stable pentamers (Rueckert, 1996), an event that appears as the main cause for the need of a cold chain to preserve FMD vaccines. Analysis of the crystal structure of the FMDV capsid (Acharya *et al.*, 1989; Lea *et al.*, 1994; Lea *et al.*, 1995; Curry *et al.*, 1996; Fry *et al.*, 1999) indicates that the pentameric intermediate subunits interact mainly through a relatively limited number of electrostatic interactions; a role of His-142 of VP3 in the acid-induced disassembly of FMDV has already been demonstrated (Ellard *et al.*, 1999).

A variety of approaches have been used to study the effects of acid. X-ray crystallographic techniques have been used to determine acid-induced structural changes in mengo virus (Kim *et al.*, 1990) and HRV (Giranda *et al.*, 1992). Amino acid changes which affect acid lability, have been identified by the generation and sequencing of acid stable mutants of HRV (Giranda *et al.*, 1992; Skern *et al.*, 1991). Another approach involved computer modelling of the effects of pH on electrostatic interactions within poliovirus and HRV

(Warwicker, 1992). In the present study, we did a side-by-side comparison of the pH stability of SAT2 and SAT3 viruses. The results revealed that SAT2 infectious particles showed similar or even more stability in mild acidic conditions than was previously described for viruses belonging to the A, O and C serotypes, stable in solutions with high ionic strength, but was sensitive to heat (Maree *et al.*, unpublished). Even though the SAT2 viruses used in this study differed by less than 11% in there amino acid sequence of the capsid proteins, the SAT2 virions had a diverse range of sensitivities toward mild acidic conditions. A SAT3 isolate from the same geographic distribution were much more sensitive to acidic environment (Maree *et al.*, unpublished).

Using the tools provided by the Structural module, it was possible to construct models as well as run molecular dynamics simulations. The variation mapping showed that most of the variation in the protomers occurs in areas on the surface as well as close to interface areas. Despite the differences, each individual difference plays only a small part in the overall interaction. The molecular dynamics results showed no real difference in the stability of the pentamers or the protomers at pH 6.0. However a pKa analysis showed that the difference in pH stability of ZIM/5/83/2 and ZIM/7/83/2 was due to the change in pKa of His 575 (145) in VP3. These results indicated that although pentamer association is mediated by many different interactions, there are usually one or two very important residues in the interaction interface. In the case of FMDV the role of His 142 has been proven (van Vlijmen *et al.*, 1998; Ellard *et al.*, 1999). This work predicts that His 145 is important in the initial association of the pentamers and also shows the effect of pH on pentamer association. The simulations conducted also support the current theories that the protonation states of His 142 and His 145 are the determining factors in pentamer assembly.

The work done here provides the local vaccine researchers with data about the predicted behavior of the capsid proteins under certain pH conditions. It provided possible explanations for their results as well as opened up new avenues of research into designing stable vaccines by exploiting the knowledge gained in analysing capsid interactions.

Chapter 6

Concluding Discussion

Structural biology forms the basis of our understanding of the relationship between the structure and function of a protein. The one cannot be studied without the other. Traditional structural biology involved time-consuming experiments to characterize a protein and its structure. In the modern age of structural biology this task has been made easier by the presence of databases that contain a vast amount of data related to the structure and function of a protein. These databases are usually specialized for a specific function such as the PDB, which accepts only three dimensional coordinates of protein structures. Although most of these databases are available on the Internet, they are underutilized by biologists. The opposite is also true in that computational biologists does not always utilize all the data and expertise of experimental biologists.

Structural biology consists of two parts: the experimental part in which structures are determined and proteins are characterized and the computational part in which computers are used to analyze and interpret structures. Experimental biologists tend to shy away from the computational side, citing reasons such as the complexity of the programs and the vast amount of data that is available. In an effort to alleviate these problems, a web-based system known as FunGIMS was designed.

FunGIMS is a Functional Genomics Information Management System that consists of various modules, each specialized for a specific type of data, yet integrating the different data types in a transparent manner. FunGIMS currently consists of modules for Structure, Sequence, Genomics and Small molecules. This study focused on the Structural module, its design and the way in which it can help experimental biologists enrich and guide their experiments to achieve more successful results. In the future the system may include more aspect to educate the users about the limitations inherent in the specific tools that they use.

FunGIMS was designed for ease of use by both programmers and biologists. During the design phase, it was decided to use the MVC architecture for FunGIMS, which allows for easy expansion of the program as well as addition of new programs and analysis methods. This type of architecture separates the display of data, control functions and data management into three separate sections, allowing for easy maintenance or upgrading of a section of FunGIMS. The design architecture was applied not only to the overall FunGIMS section but also to the more specific Structural module.

The main focus during the design of FunGIMS was not the programmers but the end users of the program. To alleviate the problems encountered by biologists when using structural biology programs, the interfaces were designed to be intuitive and easy to use. All the syntax and specific subtleties of running a program have been hidden from the user and only the basic information is required. A user can access this information by simply uploading files or using data from the databases already present in FunGIMS. The program is then run and the results presented to the user in a clean interface with the option to download or save the results. Security was also of concern as some users preferred to keep data private or share it with only a certain subset of users. To overcome this issue a system was created whereby users belong either to a single or multiple groups and every data entry belongs to a certain group. Public data are visible to all users and belong to a World group. Whenever a user saves data, he/she can decide to which group the data belongs and thus share it with the members of that group while preventing any other user from accessing it.

Easy access to data is important and this was well catered for in FunGIMS. It provides a search function that allows the user to search across all data or a selected subset with either keywords or a specific entry identifier. When searching, access rights to data entries are taken into consideration and a user will only be able to view results which he has access to.

The data are stored in a relational database that allows for the creation of complex queries to return specific results. The database is populated by parsing public data from the PDB, MSD and GenBank as well as storing user-generated data. Links between the data are also generated to allow for better integration between the data types.

The Structural module caters exclusively for structural and protein data as well as the analysis of proteins. It provides access to all the known protein structure files in the PDB as well as the enhanced data from the MSD. This allows a user to explore the protein structure in detail while also presenting an interactive display of the protein in the browser. Jmol is used in this regard and allows the user to interact with the protein in a three dimensional environment inside his web-browser. Data such as the secondary structure composition, SCOP, Pfam and other relevant information are presented to the user in a clear and consistent format.

The Structural module also provides protein structure and sequence analysis tools. There are tools for predicting transmembrane helices (TMHMM), for predicting protein families on the basis of sequence (Hmmer search against Pfam) as well as searching for conserved motifs in a sequence (Prosite). In addition to the analysis tools there are also tools that allow the user to build homology models and generate scripts for molecular dynamics simulations. In the homology modelling section a user simply enters the basic required information and thereafter generate scripts for homology modelling using Modeller or WHAT IF. Homology models can also be built online using Modeller, with the user providing a protein sequence and a template PDB structure id as well as refinement levels. The Structural module then proceeds to do an automated alignment and model building using Modeller. The resulting model, alignment file and script file used are then supplied to the user to download or save in the system.

Due to the computationally intensive nature of molecular dynamics simulations, the Structural module only provides a script generation capability. Scripts can then be run on a local machine. The user simply enters the required information and can then select to generate a script for either CHARMM, NAMD or Yasara. Thereafter the script is prepared and the user can download it or save it in the system.

The functionality of the Structural module was used in three investigations on FMDV. The first objective was related to proteome differences between different serotypes of

FMDV. Using the tools in the Structural module, each proteome was analyzed for various features such as secondary structure, conserved motifs and hydrophobicity. The results were then compared on an individual protein level between the different serotypes. Various differences were found such as changes in the hydrophobicity patterns on proteins 2A and 3A. These changes may affect the way in which certain proteins associate with each other as well as with membranes such as the ER and hence may have an influence on replication rates.

The second hurdle encountered by the local researchers was related to differences in the replication rate and plaque morphology between the different serotypes. Experimental evidence pointed to variation in the 3C protease and 3D RNA polymerase proteins of FMDV. Using the Structural module, models of the 3C and 3D proteins were built and differences between various SAT serotypes were mapped to the structure. For 3C 51 SAT serotype sequences were used and for 3D 16 SAT serotype sequences were used. After the differences were mapped to the protein models, it was found that a region in 3C, previously believed to be invariant, contained 9 differences. When locating the differences on the protein model, it was found that although these differences did occur, the hydrogen bond network in the local area was preserved. This preservation allows 3C to accept these differences without a major change in the activity of the protein.

Previous studies showed that 3D contained four invariant regions. After mapping the differences to the structure it was found that three of the four invariant regions were also conserved in the SAT serotypes. However, one region showed some variation. When mapping these differences to the protein model, it was found that these differences did not affect the structure since the different amino acids involved all have the same physiochemical characteristics and size. These changes will also not have a major effect on the activity of the protein, but subtle differences may explain the differences seen in replication rate and plaque morphology.

A third problem faced by the researchers during FMDV vaccine design, was the stability of two FMDV SAT2 subtype capsids. There were five differences between the proteins making up the capsid, but during experiments it was seen that one capsid was consistently more stable at pH 6.0 than the other. To investigate this observation, the Structural module was used to search for relevant structure and to construct homology models of the capsid protomers. Molecular dynamics simulation scripts for the Yasara program were also generated to investigate at which level of capsid assembly the difference had an effect. After building the models and running simulations of capsid protomers as well as capsid pentamer assemblies, it was found that there were no differences between the stability of the protomer and that of the pentamer. This prompted other avenues of investigation that resulted in performing pKa predictions of the residues predicted to be involved in the pentamer association interface. The pKa predictions showed that the pKa value of His145 on chain 1C, involved in interpentamer interactions (Ellard *et al.*, 1999), changed when a Val493Phe mutation occurred on chain 1C, structurally close approximation to His145. This resulted in a pKa shift of 0.5 units and thus made the ZIM/5/83/2 slightly more stable at pH 6.0 than ZIM/7/83/2.

The results obtained for FMDV allow researchers to understand the results reflected in their experimental work with regard to slight differences in FMDV replication rates. It also allows for a new understanding of the interaction between the different protein chains in the capsid as well as understanding the effect of seemingly innocuous differences in the amino acids sequence. In conclusion, these small differences in the capsid protein sequence affect pentamer-pentamer association and not the assembly of protomers or pentamers.

Introducing the local researchers to these tools, allowed them to become more comfortable with using structural biology tools and lead to the use of more advanced programs. Throughout the various chapters in this study, it was seen that structural biology plays a vital role in understanding the biological world. By providing easy access to structural data and analysis tools, biologists can now explore a new world that was previously considered to be a complex environment and so improve and guide future experimental work. This work expanded the knowledge of local researchers by providing new information about conserved patterns and features in local SAT strains, variation levels and effects in SAT 3C and 3D enzymes as well as providing new avenues for improving vaccine design based on viral capsid interaction analysis.

Bibliography

Acharya, R., Fry, E., Stuart, D., Fox, G., Rowlands, D. and Brown, F. (1989) The three-dimensional structure of foot-and-mouth disease virus at 2.9 A resolution. *Nature* **337**, 6209, 709–716.

Almeida, M. R., Rieder, E., Chinsangaram, J., Ward, G., Beard, C., Grubman, M. J. and Mason, P. W. (1998) Construction and evaluation of an attenuated vaccine for foot-and-mouth disease: difficulty adapting the leader proteinase-deleted strategy to the serotype O1 virus. *Virus Res* **55**, 1, 49–60.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M. and Sherlock, G. (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet* **25**, 1, 25–29.

Bablanian, G. M. and Grubman, M. J. (1993) Characterization of the foot-and-mouth disease virus 3C protease expressed in *Escherichia coli*. *Virology* **197**, 1, 320–327.

Bastos, A. D. S., Anderson, E. C., Bengis, R. G., Keet, D. F., Winterbach, H. K. and Thomson, G. R. (2003) Molecular epidemiology of SAT3-type foot-and-mouth disease. *Virus Genes* **27**, 3, 283–290.

Beard, C. W. and Mason, P. W. (2000) Genetic determinants of altered virulence of Taiwanese foot-and-mouth disease virus. *J Virol* **74**, 2, 987–991.

Belnap, D. M., Filman, D. J., Trus, B. L., Cheng, N., Booy, F. P., Conway, J. F., Curry, S., Hiremath, C. N., Tsang, S. K., Steven, A. C. and Hogle, J. M. (2000) Molecular tectonic model of virus structural transitions: the putative cell entry states of poliovirus. *J Virol* **74**, 3, 1342–1354.

Belsham, G. J. and Sonenberg, N. (2000) Picornavirus RNA translation: roles for cellular proteins. *Trends Microbiol* **8**, 7, 330–335.

Benson, D., Karsch-Mizrachi, I., Lipman, D., Ostell, J. and Wheeler, D. L. (2006) GenBank. *Nucleic Acids Res* **34**, Database issue, D16–D20.

Bergmann, E. M., Mosimann, S. C., Chernaia, M. M., Malcolm, B. A. and James, M. N. (1997) The refined crystal structure of the 3C gene product from hepatitis A virus: specific proteinase activity and RNA recognition. *J Virol* **71**, 3, 2436–2448.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N. and Bourne, P. E. (2000) The Protein Data Bank. *Nucleic Acids Res* **28**, 1, 235–42.

Birtley, J. R., Knox, S. R., Jaulent, A. M., Brick, P., Leatherbarrow, R. J. and Curry, S. (2005) Crystal structure of foot-and-mouth disease virus 3C protease. New insights into catalytic mechanism and cleavage specificity. *J Biol Chem* **280**, 12, 11520–11527.

Boutselakis, H., Dimitropoulos, D., Fillon, J., Golovin, A., Henrick, K., Hussain, A., Ionides, J., John, M., Keller, P. A., Krissinel, E., McNeil, P., Naim, A., Newman, R., Oldfield, T., Pineda, J., Rachedi, A., Copeland, J., Sitnov, A., Sobhany, S., Suarez-Uruena, A., Swaminathan, J., Tagari, M., Tate, J., Tromm, S., Velankar, S. and Vranken, W. (2003) E-MSD: the European Bioinformatics Institute Macromolecular Structure Database. *Nucleic Acids Res* **31**, 1, 458–462.

Brooks, B. R., Bruccoleri, R. E., Olafson, B. D., States, D. J., Swaminathan, S. and Karplus, M. J. (1983) CHARMM: a Program for Macromolecular Energy, Minimization and Dynamics Calculations *J Comput Chem* **4**, 187–217.

Bystroff, C. and Krogh, A. (2008) Hidden Markov Models for prediction of protein features. *Methods Mol Biol* **413**, 173–198.

Bystroff, C., Thorsson, V. and Baker, D. (2000) HMMSTR: a hidden Markov model for local sequence-structure correlations in proteins. *J Mol Biol* **301**, 1, 173–190.

Carrillo, C., Tulman, E. R., Delhon, G., Lu, Z., Carreno, A., Vagnozzi, A., Kutish, G. F. and Rock, D. L. (2005) Comparative genomics of foot-and-mouth disease virus. *J Virol* **79**, 10, 6487–6504.

Carrillo, E. C., Giachetti, C. and Campos, R. H. (1984) Effect of lysosomotropic agents on the foot-and-mouth disease virus replication. *Virology* **135**, 542–545.

Conte, L. L., Ailey, B., Hubbard, T. J., Brenner, S. E., Murzin, A. G. and Chothia, C. (2000) SCOP: a structural classification of proteins database. *Nucleic Acids Res* **28**, 1, 257–259.

Curry, S., Abrams, C. C., Fry, E., Crowther, J. C., Belsham, G. J., Stuart, D. I. and King, A. M. (1995) Viral RNA modulates the acid sensitivity of foot-and-mouth disease virus capsids. *J Virol* **69**, 1, 430–438.

Curry, S., Abu-Ghazaleh, R., Blakemore, W., Fry, E., Jackson, T., King, A., Lea, S., Logan, D., Newman, J. and Stuart, D. (1992) Crystallization and preliminary X-ray analysis of three serotypes of foot-and-mouth disease virus. *J Mol Biol* **228**, 4, 1263–1268.

Curry, S., Fry, E., Blakemore, W., Abu-Ghazaleh, R., Jackson, T., King, A., Lea, S., Newman, J., Rowlands, D. and Stuart, D. (1996) Perturbations in the surface structure of A22 Iraq foot-and-mouth disease virus accompanying coupled changes in host cell specificity and antigenicity. *Structure* **4**, 2, 135–145.

Curry, S., RoquÃ©-Rosell, N., Sweeney, T. R., Zunszain, P. A. and Leatherbarrow, R. J. (2007) Structural analysis of foot-and-mouth disease virus 3C protease: a viable target for antiviral drugs? *Biochem Soc Trans* **35**, Pt 3, 594–598.

de Castro, E., Sigrist, C. J. A., Gattiker, A., Bulliard, V., Langendijk-Genevaux, P. S., Gasteiger, E., Bairoch, A. and Hulo, N. (2006) ScanProsite: detection of PROSITE signature matches and ProRule-associated functional and structural residues in proteins. *Nucleic Acids Res* **34**, Web Server issue, W362–W365.

Diana, P., Barraja, P., Almerico, A. M., Dattolo, G., Mingoia, F., Loi, A. G., Congeddu, E., Musiu, C., Putzolu, M. and Colla, P. L. (1997) Acyclic glycosidopyrroles analogues of ganciclovir: synthesis and biological activity. *Farmaco* **52**, 5, 281–282.

Doel, T. R. and Baccarini, P. J. (1981) Thermal stability of foot-and-mouth disease virus. *Arch Virol* **70**, 1, 21–32.

Doherty, M., Todd, D., McFerran, N. and Hoey, E. M. (1999) Sequence analysis of a porcine enterovirus serotype 1 isolate: relationships with other picornaviruses. *J Gen Virol* **80 (Pt 8)**, 1929–1941.

Donofrio, N., Rajagopalon, R., Brown, D., Diener, S., Windham, D., Nolin, S., Floyd, A., Mitchell, T., Galadima, N., Tucker, S., Orbach, M. J., Patel, G., Farman, M., Pampanwar, V., Soderlund, C., Lee, Y.-H. and Dean, R. A. (2005) 'PACLIMS': a component LIM system for high-throughput functional genomic analysis. *BMC Bioinformatics* **6**, 94.

Doyle, S. (2001) *Understanding Information & Communication Technology for AS Level.* Nelson Thornes Publishers.

Droit, A., Hunter, J., Rouleau, M., Ethier, C., Picard-Cloutier, A., Bourgais, D. and Poirier, G. (2007) PARPs Database: A LIMS systems for protein-protein interaction data mining or Laboratory Information management system. *BMC Bioinformatics* **8**, 1, 483.

Ellard, F. M., Drew, J., Blakemore, W. E., Stuart, D. I. and King, A. M. (1999) Evidence for the role of His-142 of protein 1C in the acid-induced disassembly of foot-and-mouth disease virus capsids. *J Gen Virol* **80 (Pt 8)**, 1911–1918.

Esterhuysen, J. J., Thomson, G. R., Ashford, W. A., Lentz, D. W., Gainaru, M. D., Sayer, A. J., Meredith, C. D., van Rensburg, D. J. and Pini, A. (1988) The suitability of a rolled BHK21 monolayer system for the production of vaccines against the SAT types of foot-and-mouth disease virus. I. Adaptation of virus isolates to the system, immunogen yields achieved and assessment of subtype cross reactivity. *Onderstepoort J Vet Res* **55**, 2, 77–84.

Falk, M. M., Grigera, P. R., Bergmann, I. E., Zibert, A., Multhaup, G. and Beck, E. (1990) Foot-and-mouth disease virus protease 3C induces specific proteolytic cleavage of host cell histone H3. *J Virol* **64**, 2, 748–756.

Ferrer-Orta, C., Arias, A., Perez-Luque, R., Escarmis, C., Domingo, E. and Verdaguer, N. (2004) Structure of foot-and-mouth disease virus RNA-dependent RNA polymerase and its complex with a template-primer RNA. *J Biol Chem* **279**, 45, 47212–47221.

Filgueira, M. P., Wigdorovitz, A., Romera, A., Zamorano, P., Borca, M. V. and Sadir, A. M. (2000) Detection and characterization of functional T-cell epitopes on the structural proteins VP2, VP3, and VP4 of foot and mouth disease virus O1 campos. *Virology* **271**, 2, 234–239.

Finn, R. D., Mistry, J., Schuster-Bockler, B., Griffiths-Jones, S., Hollich, V., Lassmann, T., Moxon, S., Marshall, M., Khanna, A., Durbin, R., Eddy, S. R., Sonnhammer, E.

L. L. and Bateman, A. (2006) Pfam: clans, web tools and services. *Nucleic Acids Res* **34**, Database issue, D247–D251.

Fiser, A. and Sali, A. (2003) Modeller: generation and refinement of homology-based protein structure models. *Methods Enzymol* **374**, 461–491.

Fry, E., Acharya, R. and Stuart, D. (1993) Methods used in the structure determination of foot-and-mouth disease virus. *Acta Crystallogr A* **49 (Pt 1)**, 45–55.

Fry, E. E., Lea, S. M., Jackson, T., Newman, J. W., Ellard, F. M., Blakemore, W. E., Abu-Ghazaleh, R., Samuel, A., King, A. M. and Stuart, D. I. (1999) The structure and function of a foot-and-mouth disease virus-oligosaccharide receptor complex. *EMBO J* **18**, 3, 543–554.

Fry, E. E., Newman, J. W. I., Curry, S., Najjam, S., Jackson, T., Blakemore, W., Lea, S. M., Miller, L., Burman, A., King, A. M. Q. and Stuart, D. I. (2005) Structure of Foot-and-mouth disease virus serotype A10 alone and complexed with oligosaccharide receptor: receptor conservation in the face of antigenic variation. *J Gen Virol* **86**, Pt 7, 1909–1920.

Fulton, K. F., Ervine, S., Faux, N., Forster, R., Jodun, R. A., Ly, W., Robilliard, L., Sonsini, J., Whelan, D., Whisstock, J. C. and Buckle, A. M. (2004) CLIMS: crystallography laboratory information management system. *Acta Crystallogr D Biol Crystallogr* **60**, Pt 9, 1691–1693.

Garnier, J., Osguthorpe, D. and Robson, B. (1978) Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *J Mol Biol* **120**, 97–120.

Gasteiger, E., Gattiker, A., Hoogland, C., Ivanyi, I., Appel, R. D. and Bairoch, A. (2003) ExPASy: The proteomics server for in-depth protein knowledge and analysis. *Nucleic Acids Res* **31**, 13, 3784–3788.

George, M., Venkataramanan, R., Pattnaik, B., Sanyal, A., Gurumurthy, C. B., Hemadri, D. and Tosh, C. (2001) Sequence analysis of the RNA polymerase gene of foot-and-mouth disease virus serotype Asia1. *Virus Genes* **22**, 1, 21–26.

Gille, C. and Frömmel, C. (2001) STRAP: editor for STRuctural Alignments of Proteins. *Bioinformatics* **17**, 4, 377–378.

Giranda, V. L., Heinz, B. A., Oliveira, M. A., Minor, I., Kim, K. H., Kolatkar, P. R., Rossmann, M. G. and Rueckert, R. R. (1992) Acid-induced structural changes in human rhinovirus 14: possible role in uncoating. *Proc Natl Acad Sci U S A* **89**, 21, 10213–10217.

Gorbalenya, A. E., Donchenko, A. P., Blinov, V. M. and Koonin, E. V. (1989) Cysteine proteases of positive strand RNA viruses and chymotrypsin-like serine proteases. A distinct protein superfamily with a common structural fold. *FEBS Lett* **243**, 2, 103–114.

Gradi, A., Svitkin, Y. V., Sommergruber, W., Imataka, H., Morino, S., Skern, T. and Sonenberg, N. (2003) Human rhinovirus 2A proteinase cleavage sites in eukaryotic initiation factors (eIF) 4GI and eIF4GII are different. *J Virol* **77**, 8, 5026–5029.

Hadfield, A. T., Lee, W., Zhao, R., Oliveira, M. A., Minor, I., Rueckert, R. R. and Rossmann, M. G. (1997) The refined structure of human rhinovirus 16 at 2.15 A resolution: implications for the viral life cycle. *Structure* **5**, 3, 427–441.

Hansen, J. L., Long, A. M. and Schultz, S. C. (1997) Structure of the RNA-dependent RNA polymerase of poliovirus. *Structure* **5**, 8, 1109–1122.

Haydon, D. T., Bastos, A. D., Knowles, N. J. and Samuel, A. R. (2001) Evidence for positive selection in foot-and-mouth disease virus capsid genes from field isolates. *Genetics* **157**, 1, 7–15.

Heath, L., van der Walt, E., Varsani, A. and Martin, D. P. (2006) Recombination patterns in aphthoviruses mirror those found in other picornaviruses. *J Virol* **80**, 23, 11827–11832.

Hogue, C. W. (1997) Cn3D: a new generation of three-dimensional molecular structure viewer. *Trends Biochem Sci* **22**, 8, 314–6.

Hope, D. A., Diamond, S. E. and Kirkegaard, K. (1997) Genetic dissection of interaction between poliovirus 3D polymerase and viral protein 3AB. *J Virol* **71**, 12, 9490–9498.

Jackson, A. L., O'Neill, H., Maree, F., Blignaut, B., Carrillo, C., Rodriguez, L. and Haydon, D. T. (2007) Mosaic structure of foot-and-mouth disease virus genomes. *J Gen Virol* **88**, Pt 2, 487–492.

Jackson, T., Ellard, F. M., Ghazaleh, R. A., Brookes, S. M., Blakemore, W. E., Corteyn, A. H., Stuart, D. I., Newman, J. W. and King, A. M. (1996) Efficient infection of cells in culture by type O foot-and-mouth disease virus requires binding to cell surface heparan sulfate. *J Virol* **70**, 8, 5282–5287.

Jones, A. R., Miller, M., Aebersold, R., Apweiler, R., Ball, C. A., Brazma, A., Degreef, J., Hardy, N., Hermjakob, H., Hubbard, S. J., Hussey, P., Igra, M., Jenkins, H., Julian, R. K., Laursen, K., Oliver, S. G., Paton, N. W., Sansone, S.-A., Sarkans, U., Stoeckert, C. J., Taylor, C. F., Whetzel, P. L., White, J. A., Spellman, P. and Pizarro, A. (2007) The Functional Genomics Experiment model (FuGE): an extensible framework for standards in functional genomics. *Nat Biotechnol* **25**, 10, 1127–1133.

Jones, A. R., Pizarro, A., Spellman, P., Miller, M. and Group, F. E. W. (2006) FuGE: Functional Genomics Experiment Object Model. *OMICS* **10**, 2, 179–184.

Kabsch, W. and Sander, C. (1983) Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* **22**, 2577 – 2637.

Kim, S., Boege, U., Krishnaswamy, S., Minor, I., Smith, T. J., Luo, M., Scraba, D. G. and Rossmann, M. G. (1990) Conformational variability of a picornavirus capsid: pH-dependent structural changes of Mengo virus related to its host receptor attachment site and disassembly. *Virology* **175**, 1, 176–190.

Knipe, T., Rieder, E., Baxt, B., Ward, G. and Mason, P. W. (1997) Characterization of synthetic foot-and-mouth disease virus provirions separates acid-mediated disassembly from infectivity. *J Virol* **71**, 4, 2851–2856.

Kyte, J. and Doolittle, R. F. (1982) A simple method for displaying the hydropathic character of a protein. *J Mol Biol* **157**, 1, 105–132.

Laskowski, R., MacArthur, M., Moss, D. and Thornton, J. (1993) PROCHECK: a program to check the stereochemical quality of protein structures. *J Appl Cryst* **26**, 283–291.

Lea, S., Abu-Ghazaleh, R., Blakemore, W., Curry, S., Fry, E., Jackson, T., King, A., Logan, D., Newman, J. and Stuart, D. (1995) Structural comparison of two strains of foot-and-mouth disease virus subtype O1 and a laboratory antigenic variant, G67. *Structure* **3**, 6, 571–580.

Lea, S., Hernandez, J., Blakemore, W., Brocchi, E., Curry, S., Domingo, E., Fry, E., Abu-Ghazaleh, R., King, A. and Newman, J. (1994) The structure and antigenicity of a type C foot-and-mouth disease virus. *Structure* **2**, 2, 123–139.

Levy, J. A., Fraenkel-Conrat, H. and Owens, R. A. (1994) *Virology* Prentice Hall.

Li, H., Robertson, A. D. and Jensen, J. H. (2005) Very fast empirical prediction and rationalization of protein pKa values. *Proteins* **61**, 4, 704–721.

Li, W., Ross-Smith, N., Proud, C. G. and Belsham, G. J. (2001) Cleavage of translation initiation factor 4AI (eIF4AI) but not eIF4AII by foot-and-mouth disease virus 3C protease: identification of the eIF4AI cleavage site. *FEBS Lett* **507**, 1, 1–5.

Liljas, L. (1986) The structure of spherical viruses. *Prog Biophys Mol Biol* **48**, 1, 1–36.

Logan, D., Abu-Ghazaleh, R., Blakemore, W., Curry, S., Jackson, T., King, A., Lea, S., Lewis, R., Newman, J. and Parry, N. (1993) Structure of a major immunogenic site on foot-and-mouth disease virus. *Nature* **362**, 6420, 566–568.

Marcotte, L. L., Wass, A. B., Gohara, D. W., Pathak, H. B., Arnold, J. J., Filman, D. J., Cameron, C. E. and Hogle, J. M. (2007) Crystal structure of poliovirus 3CD protein: virally encoded protease and precursor to the RNA-dependent RNA polymerase. *J Virol* **81**, 7, 3583–3596.

Mason, P. W., Grubman, M. J. and Baxt, B. (2003*a*) Molecular basis of pathogenesis of FMDV. *Virus Res* **91**, 1, 9–32.

Mason, P. W., Pacheco, J. M., Zhao, Q.-Z. and Knowles, N. J. (2003*b*) Comparisons of the complete genomes of Asian, African and European isolates of a recent foot-and-mouth disease virus type O pandemic strain (PanAsia). *J Gen Virol* **84**, Pt 6, 1583–1593.

Mateo, R., DÃaz, A., Baranowski, E. and Mateu, M. G. (2003) Complete alanine scanning of intersubunit interfaces in a foot-and-mouth disease virus capsid reveals critical contributions of many side chains to particle stability and viral function. *J Biol Chem* **278**, 42, 41019–41027.

Moffat, K., Howell, G., Knox, C., Belsham, G. J., Monaghan, P., Ryan, M. D. and Wileman, T. (2005) Effects of foot-and-mouth disease virus nonstructural proteins on the structure and function of the early secretory pathway: 2BC but not 3A blocks endoplasmic reticulum-to-Golgi transport. *J Virol* **79**, 7, 4382–4395.

Monnier, S., Cox, D. G., Albion, T. and Canzian, F. (2005) T.I.M.S: TaqMan Information Management System, tools to organize data flow in a genotyping laboratory. *BMC*

Bioinformatics **6**, 246.

Morisawa, H., Hirota, M. and Toda, T. (2006) Development of an open source laboratory information management system for 2-D gel electrophoresis-based proteomics workflow. *BMC Bioinformatics* **7**, 430.

Olivier, B. G., Rohwer, J. M. and Hofmeyr, J.-H. S. (2005) Modelling cellular systems with PySCeS. *Bioinformatics* **21**, 4, 560–561.

Palmenberg, A. C. (1990) Proteolytic processing of picornaviral polyprotein. *Annu Rev Microbiol* **44**, 603–623.

Pearl, F., Todd, A., Sillitoe, I., Dibley, M., Redfern, O., Lewis, T., Bennett, C., Marsden, R., Grant, A., Lee, D., Akpor, A., Maibaum, M., Harrison, A., Dallman, T., Reeves, G., Diboun, I., Addou, S., Lise, S., Johnston, C., Sillero, A., Thornton, J. and Orengo, C. (2005) The CATH Domain Structure Database and related resources Gene3D and DHS provide comprehensive domain family information for genome analysis. *Nucleic Acids Res* **33**, Database issue, D247–D251.

Phillips, J. C., Braun, R., Wang, W., Gumbart, J., Tajkhorshid, E., Villa, E., Chipota, C., Skeel, R. D., Kale, L. and Schulten, K. (2005) Scalable molecular dynamics with NAMD. *J Comput Chem* **26**, 1781–1802.

Prlic, A., Down, T. A. and Hubbard, T. J. P. (2005) Adding some SPICE to DAS. *Bioinformatics* **21 Suppl 2**, ii40–ii41.

Pulido, M. R., Serrano, P., Saiz, M. and Martinez-Salas, E. (2007) Foot-and-mouth disease virus infection induces proteolytic cleavage of PTB, eIF3a,b and PABP RNA-binding proteins. *Virology* **364**, 466–474.

Reguera, J., Carreira, A., Riolobos, L., Almendral, J. M. and Mateu, M. G. (2004) Role of interfacial amino acid residues in assembly, stability, and conformation of a spherical virus capsid. *Proc Natl Acad Sci U S A* **101**, 9, 2724–2729.

Rice, P., Longden, I. and Bleasby, A. (2000) EMBOSS: the European Molecular Biology Open Software Suite. *Trends Genet* **16**, 6, 276–277.

Rieder, E., Baxt, B., Lubroth, J. and Mason, P. W. (1994) Vaccines prepared from chimeras of foot-and-mouth disease virus (FMDV) induce neutralizing antibodies and protective immunity to multiple serotypes of FMDV. *J Virol* **68**, 11, 7092–7098.

Rieder, E., Bunch, T., Brown, F. and Mason, P. W. (1993) Genetically engineered foot-and-mouth disease viruses with poly(C) tracts of two nucleotides are virulent in mice. *J Virol* **67**, 9, 5139–5145.

Rossmann, M. G. and Johnson, J. E. (1989) Icosahedral RNA virus structure. *Annu Rev Biochem* **58**, 533–573.

Rueckert, R. R. (1996) *Virology* Lippincott-Raven Publishers, Philadelphia.

Sa-Carvalho, D., Rieder, E., Baxt, B., Rodarte, R., Tanuri, A. and Mason, P. W. (1997) Tissue culture adaptation of foot-and-mouth disease virus selects viruses that bind to heparin and are attenuated in cattle. *J Virol* **71**, 7, 5115–5123.

Simmonds, P. (2006) Recombination and selection in the evolution of picornaviruses and other Mammalian positive-stranded RNA viruses. *J Virol* **80**, 22, 11124–11140.

Skern, T., Torgersen, H., Auer, H., Kuechler, E. and Blaas, D. (1991) Human rhinovirus mutants resistant to low pH. *Virology* **183**, 2, 757–763.

Sonnhammer, E. L., von Heijne, G. and Krogh, A. (1998) A hidden Markov model for predicting transmembrane helices in protein sequences. *Proc Int Conf Intell Syst Mol Biol* **6**, 175–182.

Storey, P., Theron, J., Maree, F. F. and O'Neill, H. G. (2007) A second RGD motif in the 1D capsid protein of a SAT1 type foot-and-mouth disease virus field isolate is not essential for attachment to target cells. *Virus Res* **124**, 1-2, 184–192.

Strong, R. and Belsham, G. J. (2004) Sequential modification of translation initiation factor eIF4GI by two different foot-and-mouth disease virus proteases within infected baby hamster kidney cells: identification of the 3Cpro cleavage site. *J Gen Virol* **85**, 2953–2962.

Sweeney, T. R., Roque-Rosell, N., Birtley, J. R., Leatherbarrow, R. J. and Curry, S. (2007) Structural and mutagenic analysis of foot-and-mouth disease virus 3C protease reveals the role of the beta-ribbon in proteolysis. *J Virol* **81**, 1, 115–124.

Tesar, M. and Marquardt, O. (1990) Foot-and-mouth disease virus protease 3C inhibits cellular transcription and mediates cleavage of histone H3. *Virology* **174**, 2, 364–374.

Thompson, J., Gibson, T., Plewniak, F., Jeanmougin, F. and Higgins, D. (1997) The ClustalX windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools. *Nucleic Acids Research* **24**, 4876–4882.

Thompson, J. D., Muller, A., Waterhouse, A., Procter, J., Barton, G. J., Plewniak, F. and Poch, O. (2006) MACSIMS: multiple alignment of complete sequences information management system. *BMC Bioinformatics* **7**, 318.

Twomey, T., Newman, J., Burrage, T., Piatti, P., Lubroth, J. and Brown, F. (1995) Structure and immunogenicity of experimental foot-and-mouth disease and poliomyelitis vaccines. *Vaccine* **13**, 16, 1603–1610.

Uys, L., Hofmeyr, J. H. S., Snoep, J. L. and Rohwer, J. M. (2006) Software tools that facilitate kinetic modelling with large data sets: an example using growth modelling in sugarcane. *Syst Biol (Stevenage)* **153**, 5, 385–389.

van Rensburg, H., Haydon, D., Joubert, F., Bastos, A., Heath, L. and Nel, L. (2002) Genetic heterogeneity in the foot-and-mouth disease virus Leader and 3C proteinases. *Gene* **289**, 19–29.

van Rensburg, H. G., Henry, T. M. and Mason, P. W. (2004) Studies of genetically defined chimeras of a European type A virus and a South African Territories type 2 virus reveal growth determinants for foot-and-mouth disease virus. *J Gen Virol* **85**, Pt 1, 61–68.

van Rensburg, H. G. and Mason, P. W. (2002) Construction and evaluation of a recombinant foot-and-mouth disease virus: implications for inactivated vaccine production. *Ann N Y Acad Sci* **969**, 83–87.

van Vlijmen, H. W., Curry, S., Schaefer, M. and Karplus, M. (1998) Titration calculations of foot-and-mouth disease virus capsids and their stabilities as a function of pH. *J Mol Biol* **275**, 2, 295–308.

Vasquez, C., Denoya, C. D., Torre, J. L. L. and Palma, E. L. (1979) Structure of foot-and-mouth disease virus capsid. *Virology* **97**, 1, 195–200.

Viklund, H. and Elofsson, A. (2004) Best alpha-helical transmembrane protein topology predictions are achieved using hidden Markov models and evolutionary information. *Protein Sci* **13**, 7, 1908–1917.

Voegele, C., Tavtigian, S. V., de Silva, D., Cuber, S., Thomas, A. and Calvez-Kelm, F. L. (2007) A Laboratory Information Management System (LIMS) for a high throughput genetic platform aimed at candidate gene mutation screening. *Bioinformatics* **23**, 18, 2504–2506.

Vriend, G. (1990) WHAT IF: A molecular modeling and drug design program. *Journal of Molecular Graphics* **8**, 52–56.

Warwicker, J. (1992) Model for the differential stabilities of rhinovirus and poliovirus to mild acidic pH, based on electrostatics calculations. *J Mol Biol* **223**, 1, 247–257.

Wien, M. W., Chow, M. and Hogle, J. M. (1996) Poliovirus: new insights from an old paradigm. *Structure* **4**, 7, 763–767.

Yin, J., Bergmann, E. M., Cherney, M. M., Lall, M. S., Jain, R. P., Vederas, J. C. and James, M. N. G. (2005) Dual modes of modification of hepatitis A virus 3C protease by a serine-derived beta-lactone: selective crystallization and formation of a functional catalytic triad in the active site. *J Mol Biol* **354**, 4, 854–871.

Zdobnov, E. M. and Apweiler, R. (2001) InterProScan–an integration platform for the signature-recognition methods in InterPro. *Bioinformatics* **17**, 9, 847–848.

Zibert, A., Maass, G., Strebel, K., Falk, M. M. and Beck, E. (1990) Infectious foot-and-mouth disease virus derived from a cloned full-length cDNA. *J Virol* **64**, 6, 2467–2473.

Appendix

L

>A24

MNTTDCFIALVHAIREIRAFFLPRATGRMEFTLHNGERKVFYSRPNNHDNCWLNTILQLFRYVGEPFFDWVYDSPENLTLEAIEQLEELTGLELHEGGPPALV

IWNIKHLLHTGIGTASRPSEVCMVDGTNMCLADFHAGIFLKGQEHAVFACVTSNGWYAIDDEDFYPWTPDPSDVLVFVPYDQEPLNGEWKTKVQQKLK

>A10

MNTTNCFIALVYLIREIKTLFRSRTTGKMEFTLHNGEKKTFYSRPNNHDNCWLNTILQLFRYVDEPFFDWVYNSPENLTLDAIKQLENFTGLELHEGGPPALV

IWNIKHLLQTGIGTASRPSEVCMVDGTDMCLADFHAGIFMKGQEHAVFACVTSDGWYAIDDEDFYPWTPDPSDVLVFVPYDQEPLNGDWKTLVQRKLK

>C3

MNTTDCFIALVHAIREIIAIFFPRTAGKMEFTLYTGEKKTFYSRPNNHDNCWLNAILQLFRYVDEPFFDWVYNSPENLTLEAIKQLEELTGLELHEGGPPALV

IWNIKHLLNTGIGTASRPSEVCMVDGTDMCLADFHAGIFLKGQEHAVFACVTSNGWYAIDDEDFYPWTPDPSDVLVFVPYDQEPLNGEWKTKVQQKLK

>O1

MNTTDCFIALVQAIREIKALFLPRTTGKMELTLYNGEKKTFYSRPNNHDNCWLNAILQLFRYVEEPFFDWVYSSPENLTLEAIKQLEDLTGLELHEGGPPALV

IWNIKHLLHTGIGTASRPSEVCMVDGTDMCLADFHAGIFLKGQEHAVFACVTSNGWYAIDDEDFYPWTPDPSDVLVFVPYDQEPLNGEWKAKVQRKLK

>O/SAR

MSTTDCFIALLYAFREIKTLFLSRAQGKMEFTLHNGEKKTFYSRPNNHDNCWLNTILQLFRYVDEPFFDWVYYSPENLTLDAIKQLEEITGLELHEGGPPALV

IWNIKHLLNTGIGTASRPNEVCMVDGTDMCLADFHAGIFLKGQEHAVFACVTSNGWYAIDDEDFYPWTPDPSDVLVFVPYDQEPLNGEWKAKVQKRLR

>SAT1

MKTTDCFNVLFEIFHRLRHTFKAERKMEFTLYNGEKKTFYSRPNEHGNCWLNSLLQLFRYVDEPLFESEYLSPENKTLDMIRQLSDYTKLDLSDGGPPALVLW

LIKDCLQTGVGTSTRPSEICVINGVVMTLADFHAGIFIKGTEHAVFALNTSEGWYAIDDEVFYPWTPDPENVLAYVPYDQEPLDVDWQDRAGLFLR

>KNP1

MKTTDCFSVLFEIFHRLRHTLKTERKMEFTLYNGERKTFYSRPNKHGNCWLNSLLQLFRYVDEPLFESEYLSPENKTLDMIKQLSDYTKLDLSDGGPPALVLW

LIKGCLQTGVGTSTRPSEICVINGVTMTLADFHAGIFIKGTEHAVFALNTSEGWYAIDDEVFYPWTPDPENVLAYVPYDQEPLDVDWQERAGLFLR

>SAT2

MKTTDCFNVLLEIIYRFRHTFKTDRKMEFTLYNGEKKTFYSRPNKHGNCWLNSLLQLFRYVDEPLFESEYLSPENKTLDMIKQLSDYTKLDLSDGGPPALVLR

LIKDCLQTGVGTSTRPSEICVINGVVMTLADFHAGIFIKGTGHAVFALNTSEGWYAIDDEVFYPWTPDPENVLAYVPYDQEPLDVDWQDRAGLFLR

>SAT3

MKTTDCFNALLEIFHRFRQTLNTNRKMEFTLYNGEKKTFYSRPNTHGNCWLNSLLQLFRYVDEPLFESEYLSPENKTLDMIKQLSDYTKLDLTDGGPPALVLW

LIKDCLQTGVGTSTRPSEICVINGVVMTLADFHAGIFIKGTEHAVFALNTSEGWYAIDDEVFYPWTPDPENVLAYVPYDQEPLDVDWQDRAGLFLR

VP1

>A24

TTATGESADPVTTTVENYGGETQIQRRHHTDIGFIMDRFVKIQSLSPTHVIDLMQTHQHGLVGALLRAATYYFSDLEIVVRHEGNLTWVPNGAPESALLNTSN

PTAYNKAPFTRLALPYTAPHRVLATVYNGTSKYAVGGSGRRGDMGSLAARVVKQLPASFNYGAIKADAIHELLVRMKRAELYCPRPLLAIEVSSQDRHKQKII

APAKQ

>A10

TTATGESADPVTTTVENYGGETQVQRRHHTDVGFIMDRFVKINSLSPTHVIDLMHTHKHGIVGALLRAATYYFSDLEIVVRHDGNLTWVPNGAPEAALSNTSN

PTAYNKAPFTRLALPYTAPHRVLATVYNGTSKYSASGSRRGDLGSLATRVATQLPASFNYGAIKAQAIHELLVRMKRAELYCPRPLLAIEVSSQDRYKQKIIA

PAKQ

>C3

TTTTGESADPVTTTVENYGGETQVQRRHHTDVAFVLDRFVKVPVSDRQQHTLDVMQVHKDSIVGALLRAATYYFSDLEIAVTHTGKLTWVPNGAPVSALDNTT

NPTAYHKGPLTRLALPYTAPHRVLATTYTGTTTYTTSARRGDSAHLAAAHARHLPTSFNFGAVKAETVTELLVRMKRAELYCPRPILPIQPTGDRHKQPLIAP

AKQ

>O1

TTSAGESADPVTTTVENYGGETQIQRRQHTDVSFIMDRFVKVTPQNQINILDLMQVPSHTLVGALLRASTYYFSDLEIAVKHEGDLTWVPNGAPEKALDNTTN

PTAYHKAPLTRLALPYTAPHRVLATVYNGECRYSRNAVPNLRGDLQVLAQKVARTLPTSFNYGAIKATRVTELLYRMKRAETYCPRPLLAIHPTEARHKQKIV

APVKQ

>O/SAR

TTSTGESADPVTATVENYGGETQVQRRQHTDVSFILDRFVKVTPKDQINVLDLMQTPAHTLVGALLRTATYYFADLEVAVKHEGNLTWVPNGAPETALDNTTN

PTAYHKAPLTRLALPYTAPHRVLATVYNGNCKYGESPVTNVRGDLQVLAQKAARTLPTSFNYGAIKATRVTELLYRMKRAETYCPRPLLAIHPSEARHKQKIV

APVKQ

>SAT1

TTSAGEGAEPVTVDASQHGGNSRGVHRQHTDVSFLLDRFTLVGKTQNNKMTLDLLQTKEKALVGAILRAATYYFSDLEVACLGENKWVGWTPNGAPELEEVGD

NPVVFSNRGATRFALPFTAPHRCLATTYNGDCKYKPAGTAPRDNIRGDLAVLAQRIAGETHIPTTFNYGRIYTEAEVDVYVRMKRAELYCPRPLLTHYDHNGK

DRYKTAITKPAKQ

>KNP1

TTSAGEGAEPVTTDASQHGGDRRTTRRHHTDVSFLLDRFTLVGKTQDNKLTLDLLQTKEKALVGAILRAATYYFSDLEVACVGDNKWVGWTPNGAPELAEVGD

NPVVFSKGRTTRFALPYTAPHRCLATAYNGDCKYKPTGTAPRENIRGDLATLAARIASETHIPTTFNYGRIYTDTEVDVYVRMKRAELYCPRPVLTHYDHGGR

DRYRTAITKPVKQ

>SAT2

TTSSGEGADVVTTDPSTHGGAVTEKKRVHTDVAFVMDRFTHVLTNRTAFAVDLMDTNEKTLVGALLRAATYYFCDLEIACLGEHERVWWQPNGAPRTTTLRDN

PMVFSHNNVTRFAVPYTAPHRLLSTRYNGECKYTQQSTAIRGDRAVLAAKYANTKHKLPSTFNFGYVTADKPVDVYYRMKRAELYCPRPLLPGYDHADRDRFD

SPIGVKKQ

>SAT3

TTSAGEGADVVTTDVTTHGGEVSVPRRQHTNVEFLLDRFTHIGTINGHRTICLLDTKEHTLVGAILRSATYYFCDLEVAVLGNAKYAAWVPNGCPHTDRVEDN

PVVHSKGSVVRFALPYTAPHGVLATVYNGNCKYSTTQRVAPRRGDLGALSRRVENETTRCIPTTFNFGRLLCESGDVYYRMKRTELYCPRPL RVRYTHTADR

YKTPLVKPEKQ

VP2

>A24

DKKTEETTLLEDRILTTRNGHTTSTTQSSVGVTHGYSTEEDHVAGPNTSGLETRVVQAERFYKKYLFDWTTDKAFGHLEKLELPSDHHGVFGHLVDSYAYMRN

GWDVEVSAVGNQFNGGCLLVAMVPEWKEFDTREKYQLTLFPHQFISPRTNMTAHITVPYLGVNRYDQYKKHKPWTLVVMVVSPLTVNNTSAAQIKVYANIAPT

YVHVAGELPSKE

>A10

DKKTEETTLLEDRILTTRNGHTTSTTQSSVGVTYGYSTEEDHVAGPNTSGLETRVVQAERFFKKFLFDWTTDKPFGHLTKLELPTDHHGVFGHLVDSYAYMRN

GWDVEVSAVGNQFNGGCLLVAMVPEWKEFDTREKYQLTLFPHQFISPRTNMTAHITVPYLGVNRYDQYKKHKPWTLVVMVLSPLTVSNTAATQIKVYANIAPT

YVHVAGELPSKE

>C3

DKKTEETTLLEDRILTTRNGHTTSTTQSSVGVTYGYATAEDSSSGPNTSGLETRVHQAERFFKMTLFDWVPSQNFGHMHKVVLPTDPKGVYGGLVKSYAYMRN

GWDVEVTAVGNQFNGGCLLVALVPEMGDISDREKYQLTLYPHQFINPRTNMTAHITVPYVGVNRYDQYKQHKPWTLVVMVVAPLTVNTSGAQQIKVYANIAPT

NVHVAGELPSKE

>O1

DKKTEETTLLEDRILTTRNGHTTSTTQSSVGVTYGYATAEDFVSGPNTSGLETRVVQAERFFKTHLFDWVTSDSFGRYHLLELPTDHKGVYGSLTDSYAYMRN

GWDVEVTAVGNQFNGGCLLVAMVPELCSIQKRELYQLTLFPHQFINPRTNMTAHITVPFVGVNRYDQYKVHKPWTLVVMVVAPLTVNTEGAPQIKVYANIAPT

NVHVAGEFPSKE

>O/SAR

DKKTEETTLLEDRILTTRNGHTTSTTQSSVGVTYGYATAEDFVSGPNTSGLETRVVQAERFFKTHLFDWVTSDPFGRLLELPTDHKGVYGSLTDSYAYMRNGW

DVEVTAVGNQFNGGCLLVAMVPELCSIDKRELYQLTLFPHQFINPRTNMTAHITVPFVGVNRYDQYKVHKPWTLVVMVVAPLTVNTEGAPQIKVYANIAPTNV

HVAGEFPSKE

>SAT1

DKKTEETTLLEDRILTTSHGTTTSTTQSSVGVTYGYAESDHFLPGPNTNGLETRVEQAERFFKHKLFDWTLEQQFGTTHILELPTDHKGIYGQLVDSHSYIRN

GWDVEVSATATQFNGGCLLVAMVPELCKLADREKYQLTLFPHQFLNPRTNTTAHIQVPYLGVDRHDQGTRHKAWTLVVMVVAPYTNDQTIGSTKAEVYVNIAP

TNVYVAGEKPAKQ

>KNP1

DKKTEETTLLEDRILTTSHGTTTSTTQSSVGITYGYADSDRFLPGPNTNGLETRVEQAERFFKHKLFDWTLEQRFGTTHVLELPTDHKGIYGQLVDSHSYIRN

GWDVEVSATATQFNGGCLLVAMVPELCKLSEREKYQLTLFPHQFLNPRTNTTAHIQVPYLGVDRHDQGTRHKAWTLVVMVVAPYTNDQTIGSNKAEVYVNIAP

TNVYVAGEKPAKQ

>SAT2

DKKTEETTLLEDRILTTRHGTTTSTTQSSVGITYGYADADSFRPGPNTSGLETRVEQAERFFKEKLFDWTSDKPFGTLYVLELPKDHKGIYGSLTDAYTYMRN

GWDVQVSATSTQFNGGSLLVAMVPELCSLKDREEFQLSLYPHQFINPRTNTTAHIQVPYLGVNRHDQGKRHQAWSLVVMVLTPLTTEAQMQSGTVEVYANIAP

TNVFVAGEKPAKQ

>SAT3

DKKTEETTHLEDRILTTRHNTTTSTTQSSVGVTYGYVSADRFLPGPNTSGLESRVEQAERFFKERLFTWTASQEYAHVHLLELPTDHKGIYGVMVDSHAYVRN

GWDVQVTATSTQFNGGTLLVAMVPELHSMDTRDVSQLTLFPHQFINPRTNTTAHIVVPYVGVNRHDQVQMHKAWTLVVAVMAPLTTASMGQDNVEVYANIAPT

NVYVAGERPSKQ

VP3

>A24

GIFPVACADGYGGLVTTDPKTADPAYGKVYNPPRTNYPGRFTNLLDVAEACPTFLCFDDGKPYVTTRTDDTRLLAKFDLSLAAKHMSNTYLSGIAQYYTQYS

GTINLHFMFTGSTDSKARYMVAYIPPGVETPPDTPERAAHCIHAEWDTGLNSKFTFSIPYVSAADYAYTASDTAETINVQGWVCIYQITHGKAENDTLVVSV

SAGKDFELRLPIDPRQQ

>A10

GIFPVACADGYGGLVTTDPKTADPVYGKVYNPPRTNYPGRFTNLLDVAEACPTFLCFDDGKPYVVTRTDDTRLLAKFDVSLAAKHMSNTYLSGIAQYYTQYS

GTINLHFMFTGSTDSKARYMVAYIPPGVETPPDTPEEAAHCIHAEWDTGLNSKFTFSIPYVSAADYAYTASDTAETTNVQGWVCVYQITHGKAENDTLVVSA

SAGKDFELRLPIDPRPQ

>C3

GIFPVACADGYGNMVTTDPKTADPAYGKVYNPPRTALPGRFTNYLDVAEACPTFLVFENVPYVSTRTDGQRLLAKFDVSLAARHMSNTYLAGLAQYYTQYAG

TINLHFMFTGPTDAKARYMVAYVPPGMEAPENPEEAAHCIHAEWDTGLNSKFTFSIPYISAADYAYTASNEAETTCVQGWVCVYQITHGKADADALVISASA
GKDFELRLPVDARQQ

>O1

GIFPVACSDGYGGLVTTDPKTADPVYGKVFNPPRNQLPGRFTNLLDVAEACPTFLHFEGDVPYVTTKTDSDRVLAQFDMSLAAKHMSNTFLAGLAQYYTQYS
GTINLHFMFTGPTDAKARYMIAYAPPGMEPPKTPEAAAHCIHAEWDTGLNSKFTFSIPYLSAADYAYTASDVAETTNVQGWVCLFQITHGKADGDALVVLAS
AGKDFELRLPVDARAE

>O/SAR

GIFPVACSDGYGGLVTTDPKTADPAYGKVFNPPRNMLPGRFTNFLDVAEACPTFLHFEGGVPYVTTKTDSDRVLAQFDLSLAAKHMSNTFLAGLAQYYTQYS
GTINLHFMFTGPTDAKARYMIAYAPPGMEPPKTPEAAAHCIHAEWDTGLNSKFTFSIPYLSAADYAYTASDAAETTNVQGWVCLFQITHGKADGDALVVLAS
AGKDFELRLPVDARTQ

>SAT1

GILPVAVSDGYGGFQNTDPKTSDPVYGHVYNPARTGLPGRFTNLLDVAEACPTFLDFNGVPYVTTQSNSGSKVLTRFDLAFGHKNLKNTFMSGLAQYYAQYS
GTLNLHFMYTGPTNNKAKYMVAYIPPGTHPLPETPEMASHCYHAEWDTGLNSTFTFTVPYVSAADYAYTYSDEPEQASVQGWVGVYQVTDTHEKDGAVVVSI
SAGPDFEFRMPISPSRQ

>KNP1

GILPVAVSVGYGGFQNTDPKTSDPVYGHVYNPARTGLPGRFTNLLDVAEACPTLLDFNGVPYVTTQANSGSKVLTCFDLAFGHKNLKNTFMSGLAQYYTQYS
GTLNLHFMYTGPTNNKAKYMVAYIPPGTHPLPETPEMASHCYHAEWDTGLNSTFTFTVPYVSAADFAYTYSDEPEQASVQGWVGVYQVTDTHEKDGAVVVSV
SAGPDFEFRMPISPSRQ

>SAT2

GIIPVACFDGYGGFQNTDPKTADPIYGYVYNPSRNDCHGRYSNLLDVAEACPTFLNFDGKPYVVTKNNGDKVMTCFDVAFTHKVHKNTFLAGLADYYAQYQG
SLNYHFMYTGPTHHKAKFMVAYIPPGIETDRLPKTPEDAAHCYHSEWDTGLNSQFTFAVPYVSASDFSYTHTDTPAMATTNGWVAVFQVTDTHSAEAAVVVS
VSAGPDLEFRFPVDPVRQ

>SAT3

GIIPVACNDGYGGFQNTDPKTADPIYGLVSNPPRTAFPGRFTNLLDVAEACPTFLDFDGVPYVKTTHNSGSKILTHIDLAFGHKSFKNTYLAGLAQYYAQYS
GSINLHFMYTGPTQSKARFMVAYIPPGTTPVPNTPEQAAHCYHSEWDTGLNSKFTFTVPYMSAADFAYTYCDEPEQASAQGWVTLYQITDTHDPNSAVLVSV
SAGADFELRLPINPTAQ

VP4

>A24

GAGQSSPATGSQNQSGNTGSIINNYYMQQYQNSMDTQLGDNAISGGSNEGSTDTTSTHTTNTQNNDWFSKLASSAFTGLFGALLA

>A10

GAGQSSPATGSQNQSGNTGSIINNYYMQQYQNSMDTQLGDNAISGGSNEGSTDTTSTHTTNTQNNDWFSKLASSAFTGLFGALLA

>C3

GAGQSSPATGSQNQSGNTGSIINNYYMQQYQNSMDTQLGDNAISGGSNEGSTDTTSTHTTNTQNNDWFSKLASSAFSGLFGALLA

>O1

GAGQSSPATGSQNQSGNTGSIINNYYMQQYQNSMDTQLGDNAISGGSNEGSTDTTSTHTTNTQNNDWFSKLASSAFSGLFGALLA

>O/SAR

GAGQSSPATGSQNQSGNTGSIINNYYMQQYQNSMDTQLGDNAISGGSNEGSTDTTSTHTTNTQNNDWFSKLASSAFSGLFGALLA

>SAT1

GAGQSSPATGSQNQSGNTGSIINNYYMQQYQNSMDTQLGDNAISGGSNEGSTDTTSTHTNNTQNNDWFSKLAQSAFSGLVGALLA

>KNP1

GAGQSSPATGSQNQSGNTGSIINNYYMQQYQNSMDTQLGDNAISGGSNEGSTDTTSTHTNNTQNNDWFSKLAQSAFSGLVGALLA

>SAT2

GAGHSSPVTGSQNQSGNTGSIINNYYMQQYQNSMDTQLGDNAISGGSNEGSTDTTSTHTNNTQNNDWFSKLAQSAISGLFGALLA

>SAT3

GAGQSSPATGSQNQSGNTGSIINNYYMQQYQNSMDTQLGDNAISGGSNEGSTDTTSTHTNNTQNNDWFSKLAQSAISGLFGALLA

2A

>A24

LLNFDLLKLAGDVESNPG

>A10

LLNFDLLKLAGDVESNPG

>C3

LSNFDLLKLAGDVESNPG

>O1

TLNFDLLKLAGDVESNPG

>O/SAR

LLNFDLLKLAGDVESNPG

>SAT1

LGNFELLKLAGDVESNPG

>KNP1

LCNFDLLKLAGDVESNPG

>SAT2

LCNFDLLKLAGDVESNPG

>SAT3

LCNFDLLKLAGDVESNPG

2B

>A24

PFFFSDVRSNFSKLVDTINQMQEDMSTKHGPDFNRLVSAFEELATGVKAIRTGLDEAKPWYKLIKLLSRLSCMAAVAARSKDPVLVAIMLADTGLEILDSTF

VVKKISDSLSSLFHVPAPVFSFGAPILLAGLVKVASSFFRSTPEDLERAEKQ

>A10

PFFFADVRSNFSKLVDTINQMQEDMSTKHGPDFNRLVSAFEELATGVKAIRTGLDEAKPWYKLIKLLSRLSCMAAVAARSKDPVLVAIMLADTGLEILDSTF

VVKKISDSLSSLFHVPAPAFSFGAPILLAGLVKVASSFFRSTPEDLERAEKQ

>C3

PFFFSDVRSNFSKLVETINQMQEDMSTKHGPDFNRLVSAFEELATGVKAIRTGLDEAKPWYKLIKLLSRLSCMAAVAARSKDPVLVAIMLADTGLEILDSTF

VVKKISDSLSSLFHVPAPVFSFGAPILLAGLVKVASSFFRSTPEELERAEKQ

>O1

PFFFSDVRSNFSKLVETINQMQEDMSTKHGPDFNRLVSAFEELAIGVKAIRTGLDEAKPWYKLIKLLSRLSCMAAVAARSKDPVLVAIMLADTGLEILDSTF

VVKKISDSLSSLFHVPAPVFSFGAPVLLAGLVKVASSFFRSTPEDLERAEKQ

>O/SAR

PFFFSDVRSNFSKLVETINQMQEDMSTKHGPDFNRLVSAFEELATGVKAIRTGLDEAKPWYKLIKLLSRLSCMAAVAARSKDPVLVAIMLADTGLEILDSTF

VVKKISDSLSSLFHVPAPVFSFGAPILLAGLVKVASSFFRSTPEDLERAEKQ

>SAT1

PFFFSDVRENFTKLVDSINSMQQDMSTKHGPDFNRLVSAFEELTQGVKAIKEGLDEAKPWYKVIKLLSRLSCMAAVAARSKDPVLVAIMLADTGLEILDSTF

VVKKISDALSSVFHVPAPVFSFGAPILLAGLVKVASTFFRSTPEDLERAEKQ

>KNP1

PFFFADVRENFTKLVDSINNMQHDMSTKHGPDFNRLVSAFEELTKGVKAIKDGLDEAKPWYKVIKLLSRLSCMAAVAARSKDPVLVAIMLADTGLEILDSTF

VVKKISDALSSVFHVPAPVFSFGAPILLAGLVKVASTFFRSTPEDLERAEKQ

>SAT2

PFFFSDVRENFTKLVESINNMQQDMSTKHGPDFNRLVSAFEELTKGVKAIKDGLDEAKPWYKVIKLLSRLSCMAAVAARSKDPVLVAIMLADTGLEILDSTF

VVKKISDALSSVFHVPAPVFSFGAPILLAGLVKVASTFFRSTPEDLERAEKQ

>SAT3

PFFFADVRENFTKLVDSINSMQQDISTKHGPDFNRLVSAFEELTKGVKAIKDGLDEAKPWYKIIKLLSRLSCMAAVAARSKDPVLVAIMLADTGLEILDSTF

VVKKISDALSSVFHVPAPVFSFGAPVLLAGLVKVASTFFRSTPEDLERAEKQ

2C

>A24

LKARDINDIFAILKNGEWLVKLILAIRDWIKAWIASEEKFVTTTDLVPGILEKQRDLNDPSKYKEAKEWLDNARQACLKSGNVHIANLCKVVAPAPSRSRPE

PVVVCLRGKSGQGKSFLANVLAQAISTHFTGRTDSVWYCPPDPDHFDGYNQQTVVVMDDLGQNPDGKDFKYFAQMVSTTGFIPPMASLEDKGKPFNSKVIIA

TTNLYSGFTPRTMVCPDALNRRFHFDIDVSAKDGYKINNKLDIIKALEDTHTNPVAMFQYDCALLNGMAVEMKRMQQDMFKPQPPLQNVYQLVQEVIERVEL

HEKVSSHPIFKQ

>A10

LKARDINDIFAILKNGEWLVKLILAIRDWIKAWIASEEKFVTMTDLVPGILEKQRDLNDPGKYKEAKEWLDNARQACLKSGNVHIANLCKVVAPAPSKSRPE

PVVVCLRGKSGQGKSFLANVLAQAISTHFTGRTDSVWYCPPDPDHFDGYNQQTVVVMDDLGQNPDGKDFKYFAQMVSTTGFIPPMASLEDKGKPFNSKVIIA

TTNLYSGFTPRTMVCPDALNRRFHFDIDVSAKDGYKINNKLDIIKALEDTHTNPVAMFQYDCALLNGMAVEMKRLQQDMFKPQPPLQNVYQLVQEVIERVEL

HEKVSSHPIFKQ

>C3

LKARDINDIFAILKNGEWLVKLILAIRDWIKAWIASEEKFVTMTDLVPGILEKQRDLNDPSKYKEAKEWLDNARQACLKSGNVHIANLCKVVAPAPSKSRPE

PVVVCLRGKSGQGKSFLANVLAQAISTHFTGRTDSVWYCPPDPDHFDGYNQQTVVVMDDLGQNPDGKDFKYFAQMVSTTGFIPPMASLEDKGKPFNSKVIIA

TTNLYSGFTPRTMVCPDALNRRFHFDIDVSAKDGYKINNKLDIIKALEDTHTNPVAMFQYDCALLNGMAVEMKRMQQDVFKPQPPLQNVYQLVQEVIERVEL

HEKVSSHPIFKQ

>O1

LKARDINDIFAILKNGEWLVKLILAIRDWIKAWIASEEKFVTMTDLVPGILEKQRDLNDPSKYKEAKEWLDNARQACLKSGNVHIANLCKVVAPAPSKSRPE

PVVVCLRGKSGQGKSFLANVLAQAISAHFTGRTDSVWYCPPDPDHFDGYNQQTVVVMDDLGQNPDGKDFKYFAQMVSTTGFIPPMASLEDKGKPFNSKVIIA

TTNLYSGFTPRTMVCPDALNRRFHFDIDVSAKDGYKINNKLDIIKALEDTHTNPVAMFQYDCALLNGMAVEMKRMQQDMFKPQPPLQNVYQLVQEVIDRVEL

HEKVSSHPIFKQ

>O/SAR

LKARDINDIFAILKNGEWLVKLILAIRDWIKAWIASEEKFVTMTDLVPGILEKQRDLNDPSKYKEAKEWLDNARQACLKSGNIHIANLCKVVAPAPSRSRPE

PVVVCLRGKSGQGKSFLANVLAQAISTHFTGRTDSVWYCPPDPDHFDGYNQQTVVVMDDLGQNPDGKDFKYFAQMVSTTGFIPPMASLEDKGKPFNSKVIIA

TTNLYSGFTPRTMVCPDALNRRFHFDIDVSAKDGYKINNKLDIIKALEDTHTNPVAMFQYDCALLNGMAVEMKRMQQDMFKPQPPLQNVYQLVQEVIDRVEL

HEKVSSHPIFKQ

```
>SAT1
LKARDINDIFAILKNGEWLVKLILAIRDWIKAWISSEEKYISMTDLVPRILECQRNLNDPSKYQESKEWLENAREACLKNGNVHIANLCKVNAPAPSKSRPE
PVVVCLRGKSGQGKSFLANVLAQAISTHFTGRVDSVWYCPPDPDHFDGYNQQAVVVMDDLGQNPDGKDFKYFAQMVSTTGFIPPMASLEDKGKPFNSKVIVA
TSNLYSGFTPRTMVCPDALNRRFHFDIDVSAKDGYKVNNRLDIIKALEDTHTNAPAMFNYDCALLNGSAVEMKRLQQDVFKPLPPLNSLYQLVDEVIERVKL
HEKVSSHPIFKQ

>KNP1
LKARDINDIFAILKNGEWLVKLILAIRDWIKAWISSEEKYISMTDLVPRILECQRNLNDPSKYQESKEWLENAREACLKNGNVHIANLCKVNAPAPSKSRPE
PVVVCLRGKSGQGKSFLANVLAQAISTHFTGRVDSVWYCPPDPDHFDGYNQQAVVVMDDLGQNPDGKDFKYFAQMVSTTGFIPPMASLEDKGKPFNSKVIIA
TSNLYSGFTPRTMVCPDALNRRFHFDIDVSAKDGYKVNNRLDIIKALEDTHTNAPAMFNYDCALLNGSAVEMKRLQQDVFKPLPPLNSLYQLVDEVIERVKL
HEKVSSHPIFKQ

>SAT2
LKARDINDIFAILKNGEWLVKLILAIRDWIKAWISSEEKYISMTDLVPRILECQHNLNDPSKYQESKEWLENAREACLKNGNHHIANLCKVNAPAPSRSRPE
PVVVCLRGKSGQGKSFLANVLAQAISTHFTGRTDSVWYCPPDPDHFDGYNQQTVVVMDDLGQNPDGKDFKYFAQMVSTTGFIPPMASLEDKGKPFNSKVIIA
TSNLYSGFTPRTMVCPDALNRRFHFDIDVSAKDGYKVNNRLDIIKALEDTHTNAPAMFNYDCALLNGSAVEMKRLQQDVFKPLPPLNSLYQLVDEVIERVKL
HEKVSSHPIFKQ

>SAT3
LKARDINDVFAILKNGEWLVKLILAIRDWIKAWISSEEKYISMTDLVPRILECQHNLNDPSKYQESKEWLENAREACLKNGNHHIANLCKVNAPAPSKSRPE
PVVVCLRGKSGQGKSFLANVLAQAISTHFTGRTDSVWYCPPDPDHFDGYNQQAVVVMDDLGQNPDGKDFKYFAQMVSTTGFIPPMASLEDKGKPFNSKVIIA
TSNLYSGFTPRTMVCPDALNRRFHFDIDVSARDGYKVNNRLDIIKALEDTHTNAPAMFNYDCALLNGSAVEMKRLQQDVFKPLPPLNSLYQLVDEVIERVKL
HEKVSSHPIFKQ
```

3A

```
>A24
ISIPSQKSVLYFLIEKGQHEAAIEFFEGMVHDSIKEELRPLIQQTSFVKRAFKRLKENFEIVALCLTLLANIVIMIRETRKRQKMVDDAVSEYIERANITTD
DKTLDEAEKNPLETSGASTVGFRERPLPGQKARNDENSEPAQPAEEQPQAE

>A10
ISIPSQKSVLYFLIEKGQHEAAIEFFEGMVHDSVKEELRPLIQQTSFVKRAFKRLKENFEIVALCLTLLANIVIMIRETRKRQKMVDDAVNDYIERANITTD
DKTLDEAEKNPLETSGASTVGFRERSLTGQKARDDVNSEPAQPAEDQPQAE

>C3
ISIPSQKSVLYFLIEKGQHEAAIEFFEGMVHDSIKEELRPLIQHTSFAKRAFKRLKENFEIVALCLTLLANIVIMVRETRKRQKMVDDAVNEYIEKANITTD
DKTLDEAEKNPLETSGASTVGFRERTLPGQKARDDVNSEPAQPVEEQPQAE

>O1
ISIPSQKSVLYFLIEKGQHEAAIEFFEGMVHDSIKEELQPLIQQTSFVKRAFKRLKENFEIVALCLTLLANIVITVRETRKRQKMVDDAVNEYIEKANITTD
DKTLDEAEKSPLETSGASTVGFRERTLPGQKACDDVNSEPAQPVEEQPQAE

>O/SAR
ISIPSQKAVLYFLIEKGQHEAAIEFFEGMVHDSIKEELRPLIQQTSFVKRAFKRLKENFEIVALCLTLLANIVIMIRETRKRQQMVDDAVNEYIEKANITTD
DKTLDEAEKNPLETSGATTVGFREKTLPGHKAGDDVNSEPTKPVEEQPQAE

>SAT1
ISIPSQKSVLYFLIEKGQHEAAIEFYEGMVHDSIKEELKPLLEQTSFAKRAFKRLKENFEIVALVVVLLANIVIMIRETRKRQKMVDDALDEYIEKANITTD
DKTLDEAERNPQEVVDKPTVGFRERRLPGHKTDDEVNTEPVKPAERPQAE

>KNP1
ISIPSQKSVLYFLIEKGQHEAAIEFYEGMVHDSIKEELKPLLEQTSFAKRAFKRLKENFEIVALVVVLLANIIIMIRETRKRQKMVDDALDEYIEKANITTD
DKTLEEAEKNPREVVDKPTVGFRERKLPGHKTDDEVNSEPVKPVDKPQAE
```

>SAT2

ISIPSQKSVLYFLIEKGQHEAAIEFYEGMVHDSIKEELKPLLEQTSFAKRAFKRLKENFEIVALVVVLLANIIIMIRETRKRQKMVDDALDEYIEKANITTD
DKTLEEAGRNPQEVVDKPTVGFRERKLPGHKTDDEVNSEPAKPTEKPQAE

>SAT3

ISIPSQKSVLYFLIEKGQHEAAIEFYEGMVHDSIKEELKPLLEQTSFAKRAFKRLKENFEIVALVVVLLANIVIMIRETRKRQKMVDDALDEYIEKANITTD
DKTLDEAEKNPQEVVDKPTVGFRKRELPGQKTGNEVNSEPTKPVEKPQAE

3B1

>A24

GPYAGPLERQKPLKVRAKLPQQE

>A10

GPYAGPLERQKPLKVRAKLPQQE

>C3

GPYAGPLERQKPLKVRAKLPQQE

>O1

GPYAGPLERQKPLKVRAKLPQQE

>O/SAR

GPYTGPLERQKPLKVRTKLPQQE

>SAT1

GPYAGPLERQQPLKLKAKLPRAE

>KNP1

GPYAGPLERQQPLKLKAKLPKAE

>SAT2

GPYAGPLERQQPLKLKAKLPQAE

>SAT3

GPYAGPLERQQPLKLKAKLPRAE

3B2

>A24

GPYAGPMERQKPLKVKAKAPVVKE

>A10

GPYAGPMERQKPLRVKAKAPVVKE

>C3

GPYAGPMERQKPLKVKAKAPVVKE

>O1

GPYAGPMERQKPLKVKAKAPVVKE

>O/SAR

GPYAGPMERQKPLKVKVKAPVVKE

>SAT1

GPYAGPLEKQQPLKLKARLPVAKE

>KNP1

GPYAGPLEKQQPLKLKAKLPVAKE

>SAT2

GPYAGPLEKQQPLKLKARLPVAKE

>SAT3

GPYAGPLEKQQPLKLKTRLPVAKE

3B3

>A24

GPYEGPVKKPVALKVKAKNLIVTE

>A10

GPYEGPVKKPVALKVKARNLIVTE

>C3

GPYEGPVKKPVALKVKAKNLIVTE

>O1

GPYEGPVKKPVALKVKAKNLIVTE

>O/SAR

GPYEGPVKKPVALKVKAKNLIVTE

>SAT1

GPYEGPVKKPVALKVKAKAPIVTE

>KNP1

GPYEGPVKKPVALKVKAKAPIVTE

>SAT2

GPYEGPVKKPVALKVKAKAPIVTE

>SAT3

GPYEGPVKKPVALKVKTKAPIVTE

3C

>A24

SGAPPTDLQKLVMGNTKPVELILDGKTVAICCATGVFGTAYLVPRHLFAEKYDKIMLDGRAMTDSDYRVFEFEIKVKGQDMLSDAALMVLHRGNRVRDITKH
FRDTARMKKGTPVVGVINNADVGRLIFSGEALTYKDIVVCMDGDTMPGLFAYKAATKAGYCGGAVLAKDGADTFIVGTHSAGGNGVGYCSCVSRSMLLKMKA
HVDPEPHHE

>A10

SGAPPTDLQKLVMGNTKPVELILDGKTVAICCATGVFGTAYLVPRHLFAEKYDKIMLEGRAMTDSDYRVFEFEIKVKGQDMLSDAALMVLHRGNRVRDITKH
FRDTARMKKGTPVVGVVNNADVGRLIFSGEALTYKDIVVCMDGDTMPGLFAYKAATKAGYCGGAVLAKDGADTFIVGTHSAGGNGVGYCSCVSRSMLQKMKA
HVDPEPHHE

>C3

SGAPPTDLQKMVMGNTKPVELILDGKTVAICCATGVFGTAYLVPRHLFAEKYDKIMLDGRAMTDSDYRVFEFEIKVKGQDMLSDAALMVLHRGNRVRDITKH
FRDVARMKKGTPVVGVINNADVGRLIFSGEALTYKDIVVCMDGDTMPGLFAYKAATKAGYCGGAVLAKDGAETFIVGTHSAGGNGVGYCSCVSRSMLLKMKA
HIDPEPHHE

>O1

SGAPPTDLQKMVMGNTKPVELILDGKTVAICCATGVFGTAYLVPRHLFAEKYDKIMLDGRAMTDSDYRVFEFEIKVKGQDMLSDAALMVLHRGNRVRDITKH

FRDTARMKKGTPVVGVINNADVGRLIFSGEALTYKDIVVCMDGDTMPGLFAYRAATKAGYCGGAVLAKDGADTFIVGTHSAGGNGVGYCSCVSRSMLLKMKA

HIDPEPHHE

>O/SAR

SGAPPTDLQKMVMGNTKPVELILDGKTVAICCATGVFGTAYLVPRHLFAEKYDKIMLDGRAMTDSDYRVFEFETKVKGQDMLSDAALMVLHRGNRVRDITKH

FRDVARMKKGTPVVGVINNADVGRLIFSGEALTYKDIVVCMDGDTMPGLFAYKAATKAGYCGGAVLAKDGAETFIVGTHSAGGNGVGYCSCVSRSMLLKMKA

HIDPEPHHE

>SAT1

SGCPPTDLQKMVMANVKPVELILDGKTVALCCATGVFGTAYLVPRHLFAEKYDKIMLDGRALTDSDFRVFEFEVKVKGQDMLSDAALMVLHSGNRVRDLTGH

FRDIMKLSKGSPVVGVVNNADVGRLIFSGDALTYKDLVVCMDGDTMPGLFAYRAGTKVGYCGAAVLAKDGAKTVIVGTHSAGGNGVGYCSCVSRSMLLQMKA

HIDPPPHTE

>KNP

SGCPPTDLQKMVMANVKPVELILDGKTVALCCATGVFGTAYLVPRHLFAEKYDKIMLDGRALTDSDFRVFEFEVKVKGQDMLSDAALMVLHSGNRVRDLTGH

FRDTMKLSKGSPVVGVVNNADVGRLIFSGDALTYKDLVVCMDGDTMPGLFAYRAGTKVGYCGAAVLAKDGAKTVIVGTHSAGGNGVGYCSCVSRSMLLQMKA

HIDPPPHTE

>SAT2

SGCPPTDLQKMVMANVKPVELILDGKTVALCCATGVFGTAYLVPRHLFAEKYDKIMLDGRALTDSDFRVFEFEVKVKGQDMLSDAALMVLHSGNRVRDLTGH

FRDTMKLSKGSPVVGVVNNADVGRLIFSGDALTYKDLVVCMDGDTMPGLFAYRAGTKVGYCGAAVLAKDGAKTVIVGTHSAGGNGVGYCSCVSRSMLLQMKA

HIDPPPHTE

>SAT3

SGCPPTDLQKMVMANVKPVELILDGKTVALCCATGVFGTAYLVPRHLFAEKYDKIMLDGRALTDSDFRVFEFEVKVKGQDMLSDAALMVLHSGNRVRDLTGH

FRDTMKLSKGSPIVGVVNNADVGRLIFSGDALTYKDLVVCMDGDTMPGLFAYRAGTKVGYCGAAVLAKDGAKTVIVGTHSAGGNGVGYCSCVSRSMLLQMKA

HIDPPPHTE

3D

>A24

GLIVDTRDVEERVHVMRKTKLAPTVAHGVFNPEFGPAALSNKDPRLNDGVVLDEVIFSKHKGDTKMSEEDKALFRRCAADYASRLHSVLGTANAPLSIYEAI

KGVDGLDAMEPDTAPGLPWALQGKRRGALIDFENGTVGPEVEAALKLMEKREYKFACQTFLKDEIRPMEKVRAGKTRIVDVLPVEHILYTRMMIGRFCAQMH

SNNGPQIGSAVGCNPDVDWQRFGTHFAQYRNVWDVDYSAFDANHCSDAMNIMFEEVFRTEFGFHPNAEWILKTLVNTEHAYENKRITVEGGMPSGCSATSII

NTILNNIYVLYALRRHYEGVELDTYTMISYGDDIVVASDYDLDFEALKPHFKSLGQTITPADKSDKGFVLGHSITDVTFLKRHFHMDYGTGFYKPVMASKTL

EAILSFARRGTIQEKLISVAGLAVHSGPDEYRRLFEPFQGLFEIPSYRSLYLRWVNAVCGDA

>A10

GLIVDTRDVEERVHVMRKTKLAPTVAHGVFNPEFGPAALSNKDPRLNEGVVLDEVIFSKHKGDVKMTEEDKALFRRCAADYASRLHSVLGTANAPLSIYEAI

KGVDGLDAMEPDTAPGLPWALQGKRRGALIDFENGTVGPEVEAALKLMEKREYKFACQTFLKDEIRPMEKVRAGKTRIVDVLPVEHILYTRMMIGRFCAQMH

SNNGPQIGSAVGCNPDVDWQRFGTHFAQYRNVWDVDYSAFDANHCSDAMNIMFEEVFRTDFGFHPNAEWILKTLVNTEHAYENKRITVEGGMPSGCSATSII

NTILNNIYVLYALRRHYEGVELDTYTMISYGDDIVVASDYDLDFEALKPHFKSLGQTITPADKSDKGFVLGHSITDVTFLKRHFHMDYGTGFYKPVMASKTL

EAILSFARRGTIQEKLISVAGLAVHSGPDEYRRLFEPFQGLFEIPSYRSLYLRWVNAVCGDA

>C3

GLIVDTRDVEERVHVMRKTKLAPTVAHGVFNPEFGPAALSNRDPRLNEGVVLDEVIFSKHKGDTKMSEEDKALFRRCAADYASRLHSVLGTANAP

LSIYEAIKGVDGLDAMEPDTAPGLPWALQGKRRGALIDFENGTVGPEVEAALKLMEKREYKFACQTFLKDEIRPMEKVRAGKTRIVDVLPVEHILYTRMMIGR

FCAQMHSNNGPQIGSAVGCNPDVDWQRFGTHFAQYRNVWDVDYSAFDANHCSDAMNIMFEEVFRTEFGFHPNAEWILKTLVNTEHAYENKRITVEGGMPSGCS

```
ATSIINTILNNIYVLYALRRHYEGVELDTYTMISYGDDIVVASDYDLDFEALKPHFKSLGQTITPADKSDKGFVLGHSITDVTFLKRHFHMDYGTGFYKPVMA
SKTLEAILSFARRGTIQEKLISVAGLAVHSGPDEYRRLFEPFQGLFEIPSYRSLYLRWVNAVCGDA

>O1
GLIVDTRDVEERVHVMRKTKLAPTVAHGVFNPEFGPAALSNKDPRLNEGVVLDEVIFSKHKGDTKMSEEDKALFRRCAADYASRLHSVLGTANAPLSIYEAIK
GVDGLDAMEPDTAPGLPWALQGKRRGALIDFENGTVGPEVEAALKLMEKREYKFACQTFLKDEIRPMEKVRAGKTRIVDVLPVEHILYTRMMIGRFCAQMHSN
NGPQIGSAVGCNPDVDWQRFGTHFAQYRNVWDVDYSAFDANHCSDAMNIMFEEVFRTEFGFHPNAEWILKTLVNTEHAYENKRITVEGGMPSGCSATSIINTI
LNNIYVLYALRRHYEGVELDTYTMISYGDDIVVASDYDLDFEALKPHFKSLGQTITPADKSDKGFVLGHSITDVTFLKRHFHMDYGTGFYKPVMASKTLEAIL
SFARRGTIQEKLISVAGLAVHSGPDEYRRLFEPFQGLFEIPSYRSLYLRWVNAVCGDA

>O/SAR
GLIVDTRDVEERVHVMRKTKLAPTVAHGVFNPEFGPAALSNKDPRLNEGVVLDEVIFSKHKGNTKMSEEDKALFRRCAADYASRLHSVLGTANAPLSTYEAIK
GVDGLDAMEPDTAPGLPWALQGKRRGALIDFENGTVGPEVEAALKLMEKREYKFTCQTFLKDEIRPMEKVRAGKTRIVDVLPVEHILYTRMMIGRFCAQMHSN
NGPQIGSAVGCNPDVDWQRFGTHFAQYRNVWDVDYSAFDANHCSDAMNIMFEEVFNTDFGFHPNAEWILKTLVNTEHAYENKRITVEGGMPSGCSATSIINTI
LNNIYVLYALRRHYEGVELDSYTMISYGDDIVVASDYDLDFEALKPHFKSLGQTITPADKSDKGFVLGHSITDVTFLKRHFHMDYGTGFYKPVMASKTLEAIL
SFARRGTIQEKLTSVAGLAVHSGPDEYRRLFEPFQGLFEIPSYRSLYLRWVNAVCGDA

>SAT1
GLVVDTREVEERVHVMRKTKLAPTVAYGVFQPEFGPAALSNNDKRLNEGVVLDEVIFSKHKGDAKMSEADKKLFRLCAADYASHLHNVLGTANSPLSVFEAIK
GVDGLDAMEPDTAPGLPWALQGKRRGALIDFENGTVGPEIEQALKLMEKKEYKFTCQTFLKDEIRPLEKVKAGKTRIVDVLPVEHIIYTRMMIGRFCAQMHSN
NGPQIGSAVGCNPDVDWQRFGCHFAQYRNVWDIDYSAFDANHCSDAMNIMFEEVFREEFGFHPNAVWILKTLINTEHAYENKRITVEGGMPSGCSATSIINTI
LNNIYVLYALRRHYEGVELSHYTMISYGDDIVVASDYDLDFEALKPHFKSLGQTITPADKSDKGFVLGQSITDVTFLKRHFHLDYGTGFYKPVMASKTLEAIL
SFARRGTIQEKLISVAGLAVHSGPDEYRRLFEPFQGTFEIPSYRSLYLRWVNAVCGDA

>SAT2
GLVVDTREVEERVHVMRKTKLAPTVAHGVFQPEFGPAALSNNDKRLSEGVVLDEVIFSKHKGDAKMSEADKRLFRLCAADYASHLHNVLGTANSPLSVFEAIK
GVDGLDAMEPDTAPGLPWALRGKRRGALIDFENGTVGSEIEAALKLMEKKEYKFTCQTFLKDEIRPLEKVKAGKTRIVDVLPVEHIIYTRMMIGRFCAQMHSN
NGPQIGSAVGCNPDVDWQRFGTHFAQYKNVWDIDYSAFDANHCSDAMNIMFEEVFREEFGFHPNAVWILKTLINTEHAYENKRITVEGGMPSGCSATSIINTI
LNNIYVLYALRRHYEGVELSHYTMISYGDDIVVASDYDLDFEALKPHFKSLGQTITPADKSDKGFVLGQSITDVTFLKRHFHLDYETGFYKPVMASKTLEAIL
SFARRGTIQEKLISVAGLAVHSGQDEYRRLFEPFQGTFEIPSYRSLYLRWVNAVCGDA

>SAT3

GLVVDTREVEERVHVMRKTKLAPTVAHGVFQPEFGPAALSNNDKRLNEGVVLDEVIFSKHKGDAKMSEADKRLFRLCAADYASHLHNVLGTANSPLSVFEAIK

GVDGLDAMEPDTAPGLPWALQGKRRGALIDFENGTVGPEIEAALKLMEKKEYKFTCQTFLKDEIRPLEKVKAGKTRIVDVLPVEHIIYTRMMIGRFCAQMHSN

NGPQIGSAVGCNPDVDWQRFGTHFAQYKNVWDIDYSAFDANHCSDAMNIMFEEVFREEFGFHPNAVWILKTLINTEHAYENKRITVEGGMPSGCSATSIINTI

LNNIYVLYALRRHYEGVELSHYTMISYGDDIVVASDYDLDFEALKPHFKSLGQTITPADKSDKGFVLGQSITDVSFLKRHFHLDYETGFYKPVMASKTLEAIL

SFARRGTIQEKLISVAGLAVHSGQDEYRRLFEPFQGTFEIPSYRSLYLRWVNAVCGDA
```