A web-based protein structure prediction viewer for the *Plasmodium falciparum* exportome.

Tom Newport

October 5, 2014

Abstract

The intracellular parasite *Plasmodium falciparum* significantly modifies human red blood cells to create an environment suitable for parasite proliferation. Both the modification of red blood cells and proliferation of the parasite are responsible for the most severe symptoms of malaria in humans. In order to effect this modification, *P. falciparum* exports a set of proteins termed the exportome, of which approximately 360 unique members have now been identified. Understanding the interaction network for host and parasite proteins within the red blood cell represents both an interesting scientific challenge and an important step in understanding severe malaria at a molecular scale, but experimental efforts are hampered by a lack of structural information, and computational efforts are hampered by a lack of homologues for *P. falciparum* genes in model organisms.

Here I present work towards a software tool which will automatically apply a range of pre-existing structure prediction tools to proteins in the *P. falciparum* exportome and display the results in a visually consistent user friendly web based interface. Once complete, this tool will aid in the design of protein constructs used to experimentally determine the interaction network for parasite and host proteins.

Contents

1 Introduction **2**
1.1 Protein Structure Prediction 4

2 Software implementation **7**
2.1 Aims 7
2.2 Overview 9
2.2.1 Python server 9
2.2.2 Public facing server 11
2.2.3 Remote JavaScript client 12
2.2.4 The `data_manager` class 13
2.2.5 The `ui_bindings` class 15

2.3 Implementation 17
2.3.1 Milestones 18

3 Discussion **19**

Acknowledgments **23**

1 Introduction

The apicomplexan parasite *Plasmodium falciparum* is the most virulent causative agent of malaria, responsible for over 600,000 deaths annually [1]. Along with other members of the *Plasmodium* family, *P. falciparum* has a complex lifecycle, moving between several different tissues in both mammalian and arthropod hosts. Symptomatic disease in humans occurs when *P. falciparum* undergoes rounds of asexual reproduction inside human red blood cells [2].

In some respects, the intracellular environment of a red blood cell is an ideal location for parasite proliferation. The cells' lack of an MHC (Major Histocompatibility Complex) system, which would otherwise be used to identify intracellular pathogens to the host immune system, renders parasites immunologically invisible [3], whilst the vascular system allows the parasite to travel throughout the body. The highly specialised nature of red blood cells, however, means that the intracellular environment also presents significant challenges to parasite survival.

Mature red blood cells lack protein production and export machinery, and are a nutritionally poor environment, with a proteome dominated by haemoglobin, which typically accounts for around 98% of the protein content of the cell [4]. *P. falciparum* is able to digest red blood cell proteins, however haemoglobin lacks several amino acids required for protein production. Red blood cells are

also subject to regular 'quality control' in the spleen, where damaged or infected cells are killed and recycled [5].

In order to survive and proliferate inside red blood cells, *P. falciparum* exports a range of proteins which radically transform the red blood cell, collectively termed the **exportome**. Many of these proteins are involved in setting up a parasite-derived protein export system, capable of directing exported *P. falciparum* proteins to sites both inside and outside the red blood cell. *P. falciparum* resides inside a parasitophorous vacuole, and exports proteins via structures termed Maurer's Clefts [6], which bear some similarity to golgi apparatus [7].

Many exported proteins are associated with the formation of knobs, specialised structures which form at the red blood cell membrane and promote cytoadherence to epithelial cells, platelets and other red blood cells [8] to avoid detection and destruction in the spleen. Severe forms of malaria, including cerebral malaria, are believed to be caused by sequestration of infected red blood cells in deep tissues, as well as overinduction of inflammatory cytokines caused by cell adhesion [2]. Other exported components have been associated with increasing red blood cell membrane permeability to facilitate nutrient and waste exchange and strengthening the red blood cell cytoskeleton (Reviewed in [5]).

To date, at least 10% of the protein products of the *P. falciparum* genome have been shown to be exported to the host cell [9]. Within this exportome, there exist 360 distinct proteins once close duplicates are excluded.

Whilst the *P. falciparum* genome has been available since 2002 [10], comparatively few genes have been studied in depth, and many remain of unknown function. The discovery of a motif termed the PEXEL (Plasmodium Export Element) shared between many exportome components has made it possible to

reliably predict some components of the *P. falciparum* exportome in the absence of other information [11].

The PEXEL motif is pentameric, located near the N-terminal of the protein, and can be generalised as the amino acid sequence `RxLxE/Q/D` [12], where `x` is any non-charged amino acid [13] although the non-canonical PEXEL motif `KxLxE/Q/D` and relaxed PEXEL motif `RxxLxE/Q/D` are also seen occasionally [5]. It is known that the amino acid sequence is cleaved after the leucine residue in the parasite ER [12] although how the PEXEL motif targets the protein for export remains unclear. A subset of PEXEL-carrying proteins have been shown to be cleaved by the *P. falciparum* protein Plasmepsin V [14].

In addition to exported PEXEL proteins, several PEXEL Negative exported proteins have been identified using transcription profiling [15]. Based on experimental evidence, a cryptic signal is though to exist near the N-terminal of both mature (cleaved) PEXEL proteins and PNEPs [16], although the nature of this sequence remains unclear.

Understanding the protein-protein interactions responsible for the transformation of the infected red blood cell is both an interesting scientific challenge and an important step towards a better understanding of malaria in humans. Whilst an interactome for *P. falciparum* proteins has been produced using a yeast-two-hybrid [17] it is of low quality and contains many false positives.

1.1 Protein Structure Prediction

The design of constructs for experimental use is hampered by a lack of structural information about *P. falciparum* exportome components. The considerable evolutionary distance between *P. falciparum* and eukaryotic model organ-

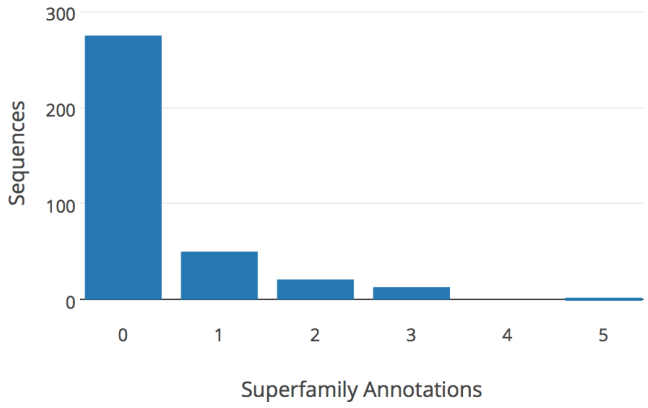

Figure 1: **Superfamily annotation frequency for *P. falciparum* exportome components** — Histogram showing frequency of superfamily annotations for all currently known components of the *P. falciparum* exportome. Over 75% of sequences did not match any domains.

isms means that around 60% of genes in the *P. falciparum* genome have no clear homologues[18, 19]. Using the SUPERFAMILY database to search for previously identified structural domains matches fewer than 25% of the components of the *P. falciparum* exportome (own research, summarised in Figure 1). To date, only 4 components of the *P. falciparum* exportome have solved structures in the PDB (own research, `https://gist.github.com/tomnewport/a04868602d3482d33921`). Knowledge based modelling using a tool such as MODELLER [20] is therefore not applicable. There exist several tools, however, which are able to predict features and domains of proteins using a variety of approaches which are less reliant on the existence of homologous genes in model organisms.

Secondary Structure and Disorder Prediction

Parts of a protein such as helices and strands will have a fixed 3D structure and pattern of amino acids and are said to be ordered. Other parts of the protein, especially loops, may not possess such an ordered or fixed structure and so are said to be disordered. Disordered parts of a protein often connect domains or features of the protein secondary structure and so disorder prediction can be used to find domain boundaries. Several algorithms exist to predict both secondary structure and disorder from amino acid sequence, and the tool metaPrDOS [21] can be used to obtain a consensus from a selection of disorder prediction algorithms.

Coiled Coil Prediction

Coiled coils are a common structural motif whereby two or more alpha helices are wound together, often important in the formation of obligomers and complexes. The COILS software tool uses a database of known coiled coil motifs to predict the likelihood of a coiled coil at particular sites on in an amino acid sequence [22].

Transmembrane Prediction

Proteins may include several domains which cross or interact with the hydrophobic environment of the lipid membrane of a cell or compartments within the cell. The software tool TMHMM uses a hidden Markov model based approach to predict these domains based on the protein's amino acid sequence alone [23].

Combined Approaches

Some software tools combine several different approaches to structure prediction. InterPro [24] performs coiled coil prediction and transmembrane prediction, and searches for known domains with similar sequences. Results are presented in a single graphic showing predicted domains along the amino acid sequence.

Phyre2 [25] performs disorder prediction, secondary structure prediction and transmembrane prediction before comparing against a fold database and attempting to build a model of the tertiary structure based on the amino acid sequence. Results are presented in a series of figures as well as 3D models based on different templates in a PDB format.

2 Software implementation

2.1 Aims

This project aims to build software tools to automate bioinformatic analysis of *P. falciparum* exported proteins and red blood cell proteins and present the results in a web-based interface. This will be used to aid in the design of protein constructs used to produce a high quality interactome for exported *P. falciparum* proteins.

The *P. falciparum* genome and other associated data are already available from several online databases, including the dedicated PlasmoDB [26] which provides functional genomic data for genes found in several *Plasmodium* species and includes annotations such as gene location, polymorphisms and expression data, as well as some limited structural annotations (e.g. Figure 2). No database presently provides a broad range of structural annotations required for construct

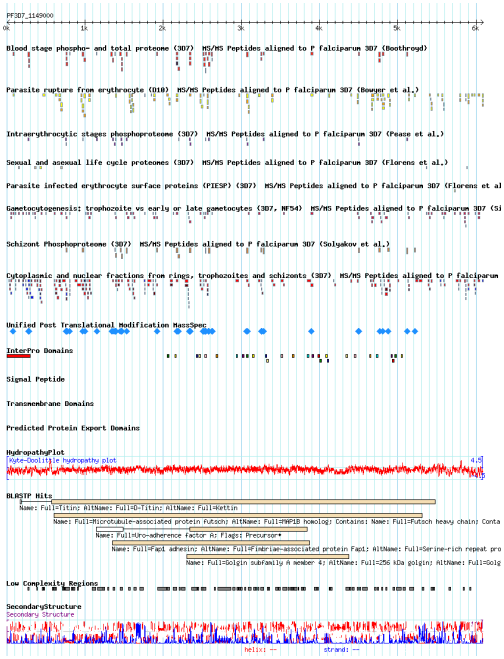

Figure 2: **PlasmoDB Structural information for PF3D7_1149000** — This includes, amongst others, InterPro domains, secondary structure prediction and export domains. Note that the protein product of PF3D7_1149000 has been shown to be exported.

design, or allows the user to look only at proteins known to be exported to the red blood cell.

Protein structure prediction is a fast evolving field. As existing tools are updated and new tools are published, both the inputs and outputs of such a tool may change. It is therefore vital to consider modularity and flexibility at an early stage to ensure future development is not compromised by design decisions made earlier in development.

2.2 Overview

The finished software will consist of three distinct parts. A local client implemented in Python will request analyses of newly added sequences from remote servers and then parse and store the response, whilst a remote client implemented in JavaScript will load and display protein files to the end user. A simple HTTP server will serve and cache JavaScript files necessary for the remote client and data files generated by the Python client.

2.2.1 Python server

The local python client, which is responsible for requesting, retrieving and parsing structure predictions, may be provided with a list of gene names which correspond to genes available through PlasmoDB (`http://plasmodb.org/`) [26]. These will then be retrieved in FASTA format from PlasmoDB and submitted several servers: TMHMM for transmembrane prediction; Phyre2 for fold recognition and secondary structure predication; metaPrDOS for disorder prediction; COILS for coiled coil prediction and InterPro for feature annotation. Submission to web-based servers will be performed using HTTP requests to CGI (Common

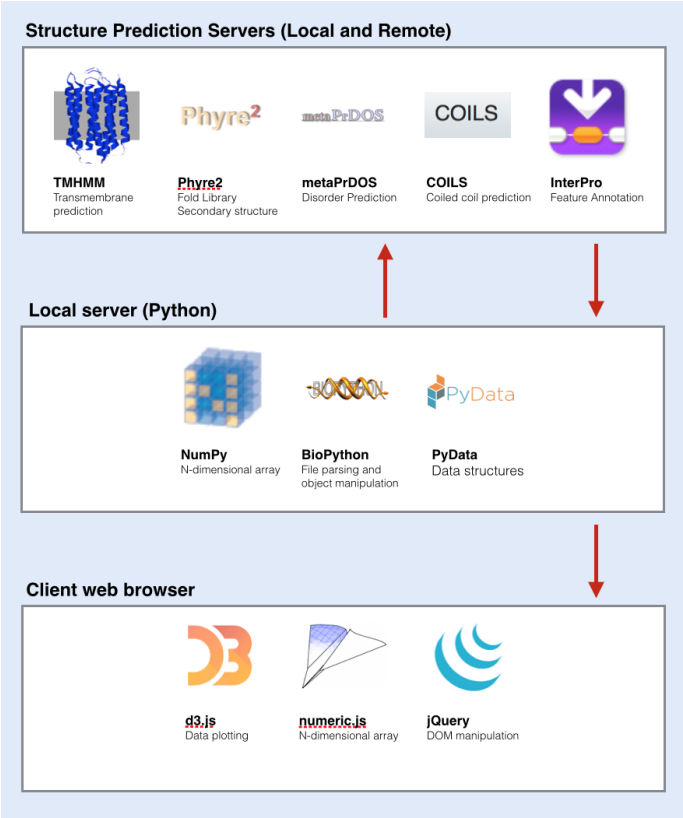

Figure 3: **High-level overview of planned software** — Sequence data is submitted to the servers TMHMM, Phyre2, metaPrDOS, COILS and InterPro using a python script running on a local server. The data is then retrieved and processed. Once processed and stored, data may be requested by the client and displayed using client-side scripts for visualisation.

Gateway Interface) scripts on respective remote servers using the Python Requests library (`http://python-requests.org`). Where possible, the client will be notified by email when results are available, however if this feature is not offered by a particular server the client will make periodic HTTP requests to determine if the job has been completed. Once the job is completed, it may be retrieved using an additional HTTP request.

Parsing of results will be performed primarily using the Pandas dataframe provided by PyData (`http://pandas.pydata.org`), assisted by BioPython [27], used to open files in biological formats such as PDB and FASTA, and Numpy [28], which implements a computationally efficient n-dimensional array for the Python language. Results from many servers will be saved to two files in comma-separated values (CSV) format, one (**series**) storing series with a defined value for each amino acid (a prediction of the likelihood of each amino acid forming part of a coiled coil, for example), and one (**annotations**) storing annotations identifying features stretching across several amino acids (domains identified based on similarity to common motifs in a database, for example).

Position	**Amino Acid**	**Series 1**	**Series 2**	**...**	**Series N**
1	S	0.2	20	...	0.9
2	N	0.3	14	...	0.1
3	I	0.2	12	...	0.3

Table 1: **Series Table Example** — Each series has a defined value for each amino acid. Each column represents a single series, and the table possesses an unlimited number of columns. Each row represents a single amino acid.

2.2.2 Public facing server

Since the public-facing HTTP server only retrieves files (and is not able to trigger any additional server-side processing), any off-the-shelf HTTP server

Start Position	End Position	Source	Description
10	143	InterPro	Transmembrane Region
53	197	Phyre2	Duffy receptor, alpha form

Table 2: **Annotation Table Example** — Each annotation consists of a start position and an end position, a source (the server which identified it) and a description (the feature which was identified by the server). Each row is an annotation, and the table may contain an unlimited number of rows.

may be used without modification, greatly simplifying the problems of installing, maintaining and securing a publicly accessible server on the web.

Unlike most existing web-based tools for viewing genomic data where visualisations are rendered on the server and transmitted as images, visualisations will be rendered client-side based on raw data. Users will therefore require a recently updated web browser in order to view visualisations of data using the remote client. The client-side approach removes the need for the server to render images, reducing load on the server, and also removes the need to transmit images, reducing load on the network. Additionally, advanced users are able to tweak visualisations as required simply by changing rules in the CSS (Cascading StyleSheet) files supplied with the remote client.

2.2.3 Remote JavaScript client

An HTML (HyperText Markup Language) document provided by the server will download the remote client and all dependencies (including default style sheets) to the user's web browser. The remote client will consist of two main classes, a `data_manager` class which will load data from the server, perform some client side processing and then provide an object which can be used to retrieve data for plotting, and a `ui_bindings` class, which will provide an interface which may be used to create, update and link different UI elements.

2.2.4 The `data_manager` class

The `data_manager` class will first download a schema from the server in JSON format which will describe where data required for visualisations may be found. This will provide a machine readable description of the contents of the data files and the relationships between them. All proteins will likely share a single schema file, however updates to the schema file can be used to change the way the remote client interprets data files. A visualisation of an example schema file is shown in Figure 4. Several objects appear once per schema, including an array of required csv files, a link to protein metadata (properties of the protein such as its length and gene name), an identifier for the position series which contains the position of each amino acid in the series table, and an identifier for the sequence series, which defines the residue at each position.

An unlimited number of three different data types: **series**; **series groups** and **annotation lists** may then be defined.

The `Series` class The series class is named and will contain an identifier locating that series within the required csv files. It will also contain minimum and maximum y-value thresholds to annotate features, and the standard deviation of a gaussian kernel convolution kernel which may be used to smooth the series before annotating features, and a description for features annotated in this manner. In the example shown in Figure 4, the named series "disorder" may be convolved using a gaussian kernel of standard deviation 1, and values higher than 0.5 but lower than 2 (unrealistically high to cause upper bound to be ignored) may be annotated as "Disordered Region".

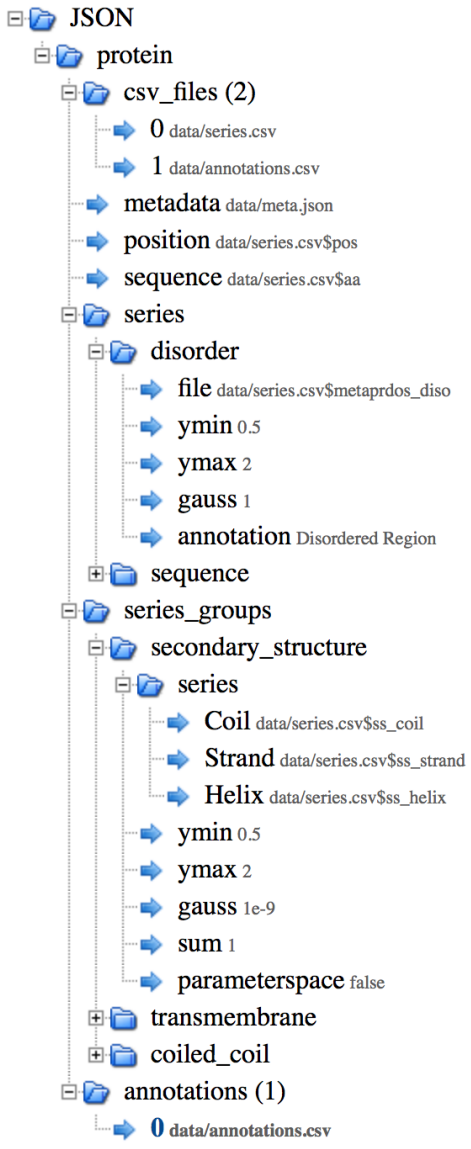

Figure 4: **Tree visualisation of JSON Schema** — Each protein object contains a list of required CSV files and a link to a protein metadata file. Position identifies a spreadsheet column which contains the amino acid position, and sequence identifies a spreadsheet column which contains a single-letter amino acid code for each position. Series are then defined, followed by series groups and finally annotations. Only one object of each type is shown expanded. Visualised using `http://www.jsontree.com`.

The `Series_Group` class The series group class is named and will contain several named series objects. The properties of these objects for the purpose of automatic annotation will be shared, and additional properties may be used to indicate that the series taken together should sum to a particular value for all amino acids, or that the series each represent a different point in parameter space for some algorithm.

The `Annotation` object The annotation object will simply contain a list of annotations, with a start, end, source and description. Each protein possesses only one annotation list, which can be populated from multiple files.

2.2.5 The `ui_bindings` class

The UI Bindings class will consist of a UI Manager, which coordinates initialising and updating UI elements in response to user actions, a viewscope class which defines the part of the sequence being viewed by the user, and a plot class, which keeps an individual plot up to date. Plotting is primarily performed in a vector format using Data Driven Documents (d3.js) [29].

The UI manager and viewscope The UI manager will provide a simple way to populate the user interface with interactive, data-driven elements. Creating a new plot will involve binding a new UI element to the UI manager using the DOM position (in CSS selector format), the data to be plotted, the class used to draw the plot and any additional options. An example of this is shown in Figure 5.

Plots may subscribe to changes in a viewscope object through the UI Manager. For example, a UI element may show disorder in a region of the gene

```
ui.bind(
        "#fig_overview_disorder",
        protein_data.series.disorder,
        D3SingleSeriesOverview,
        {round:true, cscheme:"darkred"}
);
```

Figure 5: **Example usage of `bind` method of UI Manager class** — Arguments are 1) DOM Position as CSS selector 2) Data to plot 3) Class used to draw plot 4) Options passed to plot.

specified by a viewscope object. If the update (redraw) method of the UI element is subscribed to the viewscope object, it will be called each time the user changes the view, redrawing the plot.

Plotting In the sense used here, a plot may be any method which uses a piece of data to alter the user interface. This may be as simple as printing a piece of text or toggling the visibility of some UI element, or as complex as an interactive visualisation of large amounts of data. The only required feature of a plot is that it should exist at a specific fixed point or set of points in the DOM (Document Object Model) specified using a CSS selector, and be able to initialise and update (draw and redraw) itself.

Several different subclasses of plot will be implemented. Those prefixed with "D3" primarily use the Data Driven Documents JavaScript library to simplify the generation of SVG (Scalable Vector Graphics) plots.

D3 Annotation Plot Annotations arranged along an x-axis representing residue number.

D3 Single Series Overview A single series plotted along an x-axis representing residue number.

D3 Series Group Overview A group of series plotted along an x-axis representing residue number.

D3 Secondary Structure A plot showing the amino acid code at each residue number as well as the secondary structure feature (coil, helix or strand) predicted at that residue.

Annotation List A simple list of annotations, not plotted on an axis.

Write Meta Inserts a particular piece of protein metadata at a point (or points) in the DOM.

Match Visibility Shows or hides a DOM element based on the truth value of a piece of data.

View Slider An interactive element which may be used to both change and display region of the protein currently being viewed by the user.

2.3 Implementation

Whilst the present implementation is incomplete, significant progress has been made and several parts are now fully functional. The remote JavaScript client is close to completion (see Figure 6), and can provide an interactive user interface which can be used to view protein data generated by the local Python client. The local Python client is incomplete, and whilst methods have been written to carry out individual steps, the client is not yet capable of automatically carrying out protein structure prediction in a high throughput manner. Software engineering

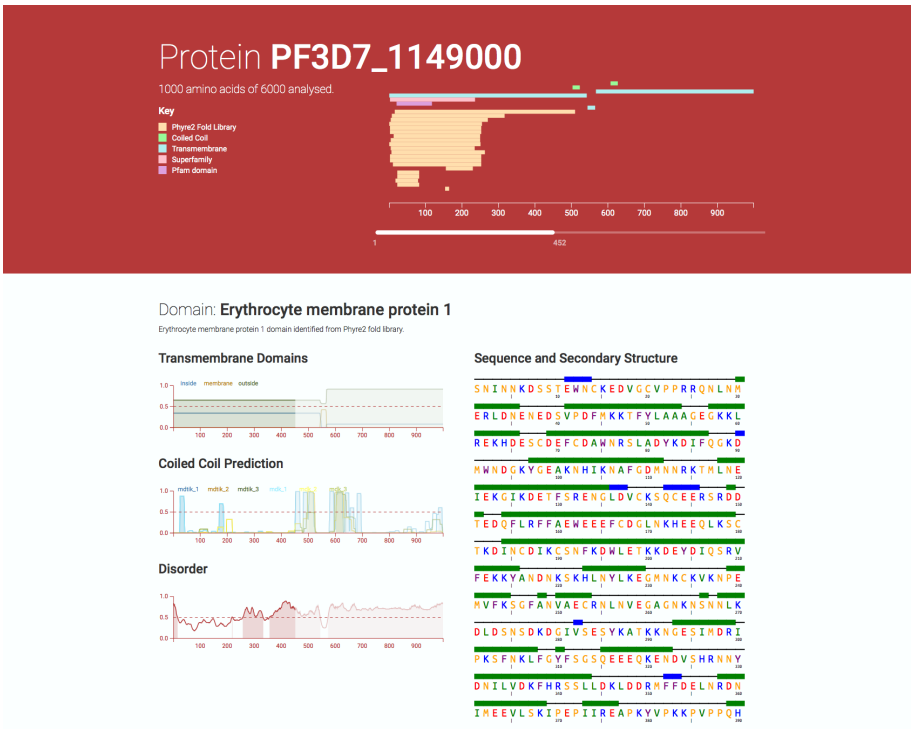

Figure 6: **Remote client** — An example page generated by the remote client, showing examples of the D3 Annotation plot, Write Meta plot, View Slider plot, D3 Single Series plot, D3 Series Group plot and D3 Secondary Structure plot.

challenges have been almost entirely solved in the general case, and it is expected to be trivial to implement remaining specific cases based on these solutions.

2.3.1 Milestones

Local python client The local python client is able to submit sequences to TMHMM, Phyre2, metaPrDOS and COILS but is not able to submit sequences to InterPro.

The local python client is able to check the status of a job submitted to Phyre and metaPrDOS using an IMAP email server, but is not able to check the status of jobs submitted to InterPro, TMHMM or COILS.

The local python client is able to track the status of jobs submitted to Phyre, metaPrDOS and TMHMM and retrieve the results, but is not able to retrieve jobs submitted to COILS or InterPro.

The local python client is able to parse output from Phyre, metaPrDOS, COILS, TMHMM and InterPro and save the combined output.

Public-facing HTTP Server The public facing HTTP Server works when implemented using the built in Python SimpleHTTPServer class. Implementation using any other off-the-shelf server package will be trivial.

Remote Client With the exception of a few refinements, the remote JavaScript client (as shown in Figure 6) is complete. Protein data can be loaded and interpreted using the schema JSON file and used to generate several interactive plots. New plots may be added to the user interface with a single line of JavaScript, the modularity of the code new types of plots to be added to the UI manager easily.

Some examples of plots are shown in Figure 7 (D3 Single Series), Figure 8 (D3 Series Group), Figure 9 (D3 Secondary Structure) and Figure 10 (D3 Annotation).

3 Discussion

Once complete, it is hoped that the system presented here will effectively present data which will aid in the design of protein constructs which can be used to experimentally test interactions between both protein components of the human red blood cell proteome and exported *P. falciparum* proteome (exportome).

Previously, attempts to map interactions between *P. falciparum* genes (e.g.

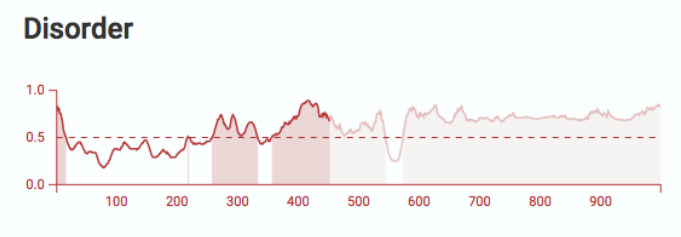

Figure 7: **D3 Series Plot** — Disorder is plotted for the first 1000 residues of the protein. In order to identify disordered regions (shaded in red) the series is first convolved using a gaussian kernel of standard deviation equal to 1 (as per the JSON schema file). The region between residue 1 and 452 is selected and emphasised in the plot.

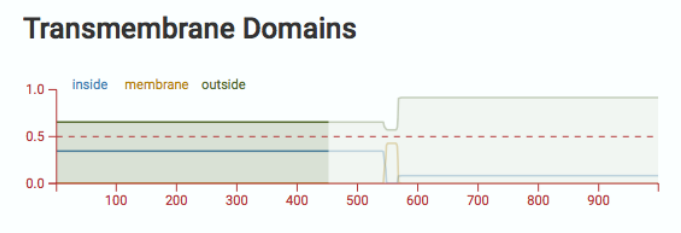

Figure 8: **D3 Series Group Plot** — This plot shows TMHMM predictions for each residue. The region between 1 and 452 is selected and emphasised.

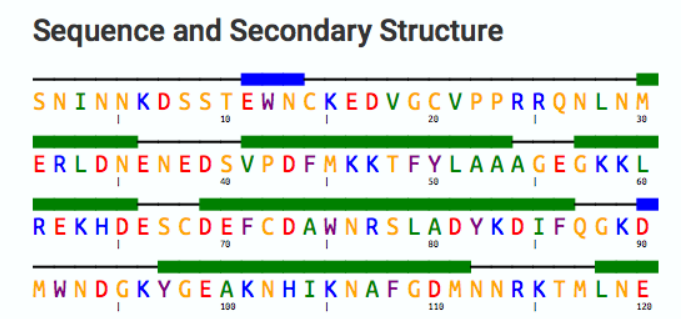

Figure 9: **D3 Secondary Structure Plot** — Sequence and secondary structure are plotted. Helices are shown in green and strands are shown in blue.

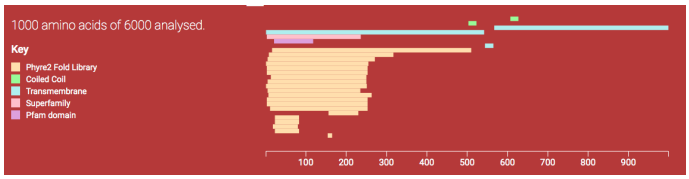

Figure 10: **D3 Annotation Plot** — Annotations are shown coloured by source.

[17]) have produced poor quality data, whilst computational approaches (e.g. [18, 19]) have been hampered by a lack of homologous genes in model organisms. By focusing on the exportome, a small subset of the *P. falciparum* genome which accounts for much of the virulence of malaria in humans, it will be possible to produce higher quality data from an experimental system.

The tool presented here will combine the results of several different protein structure prediction tools in one visually consistent interface, which will allow an experienced structural biologist to gain an understanding of the basic architecture of the protein (such as domain boundaries, transmembrane regions and common structural motifs) sufficient to optimise constructs for experimental use. The flexibility inherent in the design of the software will allow new tools to be added and existing tools to be rearranged and reconfigured to meet new requirements and integrate new or improved approaches in the same visually consistent manner.

Whilst detailed structural information is absent from many genetic databases, this tool will provide a database of *P. falciparum* exportome proteins with structural predictions drawn from a range of sources. Information provided will be similar to the output provided by tools such as InterPro and Phyre2, although optimised both for a specific set of proteins; the *P. falciparum* exportome, and a specific application; design of experimental protein constructs.

The system as proposed has several limitations which will provide a focus for future development. For example, it is not yet possible for users to add personal annotations for later reference. The ability to select an area of a gene, add notes and download the amino acid sequence would greatly improve the experience of a user designing protein constructs using this system. This will,

however, require considerable increases in the complexity of the public-facing HTTP server.

Use of client-side rendering for graphics means that displaying more than 6 plots becomes taxing for a modern computer and so it is necessary to reduce the graphical intensity of plots (by removing animated behaviours, for example). The system does not yet have any means of handling data which does not conform to the two implemented data models; series (with a single defined value at each residue) or annotation (a span with a defined start and end). Notably, there is presently no support for predictions in the form of 3D models such as those generated by Phyre2.

Several protein structure prediction tools specify an upper limit on sequence length, and this limitation (often around 1000 residues) will initially be carried through to the tool proposed here. Splitting a large protein into several overlapping segments could overcome this limitation, however in several cases it is unclear how (or indeed whether) the results of these multiple analyses could be reliably reassembled to produce a single valid output for the protein.

Whilst an increasing number of increasingly powerful protein structure prediction tools are available to researchers, each tool typically uses a distinct visual language, complicating direct comparison and cross referencing using the results of several tools. Additionally, many tools are built to handle a single protein at a time, complicating attempts to batch-process several proteins. The local python client described here will allow a set of methods to be applied to a set of proteins automatically, whilst the remote JavaScript client is able to present the output of these methods in a visually consistent and appealing manner.

Acknowledgments

I am grateful to John Vakonakis for his insight, enthusiasm and encouragement whilst supervising this project.

References

[1] World Health Organisation. *World Malaria Report 2013*. World Health Organization, 2013.

[2] Qijun Chen, Martha Schlichtherle, and Mats Wahlgren. Molecular Aspects of Severe Malaria. *Clinical Microbiology Reviews*, 13(3):439–450, July 2000.

[3] Karin Kirchgatter and Hernando a Del Portillo. Clinical and molecular aspects of severe malaria. *Anais da Academia Brasileira de Ciências*, 77(3):455–75, September 2005.

[4] Angelo D'Alessandro, Pier Giorgio Righetti, and Lello Zolla. The red blood cell proteome and interactome: an update. *Journal of proteome research*, 9(1):144–63, January 2010.

[5] Brendan Elsworth, Brendan S Crabb, and Paul R Gilson. Protein export in malaria parasites: an update. *Cellular microbiology*, 16(3):355–63, March 2014.

[6] Matthias Marti and Tobias Spielmann. Protein export in malaria parasites: many membranes to cross. *Current opinion in microbiology*, 16(4):445–51, August 2013.

[7] Esther Mundwiler-Pachlatko and Hans-Peter Beck. Maurer's clefts, the enigma of Plasmodium falciparum. *Proceedings of the National Academy*

of Sciences of the United States of America, 110(50):19987–94, December 2013.

[8] Susan M Kraemer and Joseph D Smith. A family affair: var genes, PfEMP1 binding, and malaria disease. *Current opinion in microbiology*, 9(4):374–80, August 2006.

[9] Justin a Boddey, Teresa G Carvalho, Anthony N Hodder, Tobias J Sargeant, Brad E Sleebs, Danushka Marapana, Sash Lopaticki, Thomas Nebl, and Alan F Cowman. Role of plasmepsin V in export of diverse protein families from the Plasmodium falciparum exportome. *Traffic (Copenhagen, Denmark)*, 14(5):532–50, May 2013.

[10] Malcolm J Gardner, Neil Hall, Eula Fung, Owen White, Matthew Berriman, Richard W Hyman, Jane M Carlton, Arnab Pain, Karen E Nelson, Sharen Bowman, Ian T Paulsen, Keith James, Jonathan A Eisen, Kim Rutherford, Steven L Salzberg, Alister Craig, Sue Kyes, Man-Suen Chan, Vishvanath Nene, Shamira J Shallom, Bernard Suh, Jeremy Peterson, Sam Angiuoli, Mihaela Pertea, Jonathan Allen, Jeremy Selengut, Daniel Haft, Michael W Mather, Akhil B Vaidya, David M A Martin, Alan H Fairlamb, Martin J Fraunholz, David S Roos, Stuart A Ralph, Geoffrey I McFadden, Leda M Cummings, G Mani Subramanian, Chris Mungall, J Craig Venter, Daniel J Carucci, Stephen L Hoffman, Chris Newbold, Ronald W Davis, Claire M Fraser, and Bart Barrell. Genome sequence of the human malaria parasite Plasmodium falciparum. *Nature*, 419(6906):498–511, October 2002.

[11] Tobias J Sargeant, Matthias Marti, Elisabet Caler, Jane M Carlton, Ken Simpson, Terence P Speed, and Alan F Cowman. Lineage-specific expansion of proteins exported to erythrocytes in malaria parasites. *Genome biology*, 7(2):R12, January 2006.

[12] Daniel E Goldberg and Alan F Cowman. Moving in and renovating: exporting proteins from Plasmodium into host erythrocytes. *Nature reviews. Microbiology*, 8(9):617–21, September 2010.

[13] Olivier Dietz. Studies on interactions of exported" plasmodium falciparum" membrane proteins. *PhD Thesis*, 2014.

[14] Justin a Boddey and Alan F Cowman. Plasmodium nesting: remaking the erythrocyte from the inside out. *Annual review of microbiology*, 67:243–69, January 2013.

[15] Arlett Heiber, Florian Kruse, Christian Pick, Christof Grüring, Sven Flemming, Alexander Oberli, Hanno Schoeler, Silke Retzlaff, Paolo Mesén-Ramírez, Jan a Hiss, Madhusudan Kadekoppala, Leonie Hecht, Anthony a Holder, Tim-Wolf Gilberger, and Tobias Spielmann. Identification of new PNEPs indicates a substantial non-PEXEL exportome and underpins common features in Plasmodium falciparum protein export. *PLoS pathogens*, 9(8):e1003546, January 2013.

[16] Christof Grüring, Arlett Heiber, Florian Kruse, Sven Flemming, Gianluigi Franci, Sara F Colombo, Elisa Fasana, Hanno Schoeler, Nica Borgese, Hendrik G Stunnenberg, Jude M Przyborski, Tim-Wolf Gilberger, and Tobias Spielmann. Uncovering common principles in protein export of malaria parasites. *Cell host & microbe*, 12(5):717–29, November 2012.

[17] Douglas J LaCount, Marissa Vignali, Rakesh Chettier, Amit Phansalkar, Russell Bell, Jay R Hesselberth, Lori W Schoenfeld, Irene Ota, Sudhir Sahasrabudhe, Cornelia Kurschner, Stanley Fields, and Robert E Hughes. A protein interaction network of the malaria parasite Plasmodium falciparum. *Nature*, 438(7064):103–7, November 2005.

[18] Stefan Wuchty and Jonathan J Ipsaro. A draft of protein interactions in the malaria parasite P. falciparum. *Journal of proteome research*, 6(4):1461–70, April 2007.

[19] Shailesh V Date and Christian J Stoeckert. Computational modeling of the Plasmodium falciparum interactome reveals protein function on a genome-wide scale. *Genome research*, 16(4):542–9, April 2006.

[20] Narayanan Eswar, Ben Webb, Marc A Marti-Renom, M S Madhusudhan, David Eramian, Min-Yi Shen, Ursula Pieper, and Andrej Sali. Comparative protein structure modeling using MODELLER. *Current protocols in protein science / editorial board, John E. Coligan ... [et al.]*, Chapter 2:Unit 2.9, November 2007.

[21] Takashi Ishida and Kengo Kinoshita. Prediction of disordered regions in proteins based on the meta approach. *Bioinformatics (Oxford, England)*, 24(11):1344–8, June 2008.

[22] a Lupas, M Van Dyke, and J Stock. Predicting coiled coils from protein sequences. *Science (New York, N.Y.)*, 252(5009):1162–4, May 1991.

[23] a Krogh, B Larsson, G von Heijne, and E L Sonnhammer. Predicting transmembrane protein topology with a hidden Markov model: application

to complete genomes. *Journal of molecular biology*, 305(3):567–80, January 2001.

[24] Nicola J Mulder, Rolf Apweiler, Terri K Attwood, Amos Bairoch, Alex Bateman, David Binns, Margaret Biswas, Paul Bradley, Peer Bork, Phillip Bucher, Richard Copley, Emmanuel Courcelle, Richard Durbin, Laurent Falquet, Wolfgang Fleischmann, Jerome Gouzy, Sam Griffith-Jones, Daniel Haft, Henning Hermjakob, Nicolas Hulo, Daniel Kahn, Alexander Kanapin, Maria Krestyaninova, Rodrigo Lopez, Ivica Letunic, Sandra Orchard, Marco Pagni, David Peyruc, Chris P Ponting, Florence Servant, and Christian J a Sigrist. InterPro: an integrated documentation resource for protein families, domains and functional sites. *Briefings in bioinformatics*, 3(3):225–35, September 2002.

[25] Lawrence a Kelley and Michael J E Sternberg. Protein structure prediction on the Web: a case study using the Phyre server. *Nature protocols*, 4(3):363–71, January 2009.

[26] Cristina Aurrecoechea, John Brestelli, Brian P Brunk, Jennifer Dommer, Steve Fischer, Bindu Gajria, Xin Gao, Alan Gingle, Greg Grant, Omar S Harb, Mark Heiges, Frank Innamorato, John Iodice, Jessica C Kissinger, Eileen Kraemer, Wei Li, John A Miller, Vishal Nayak, Cary Pennington, Deborah F Pinney, David S Roos, Chris Ross, Christian J Stoeckert, Charles Treatman, and Haiming Wang. PlasmoDB: a functional genomic database for malaria parasites. *Nucleic acids research*, 37(Database issue):D539–43, January 2009.

[27] Peter J A Cock, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, and Michiel J L de Hoon. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics (Oxford, England)*, 25(11):1422–3, June 2009.

[28] Stefan van der Walt, S Chris Colbert, and Gael Varoquaux. The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science & Engineering*, 13(2):22–30, March 2011.

[29] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D: Data-Driven Documents. *IEEE transactions on visualization and computer graphics*, 17(12):2301–9, December 2011.