2013-01-01

Automatic Elucidation of GPI Molecular Structures with Grid Computing Technology

Juan Clemente Aguilar Bonavides
University of Texas at El Paso, clemente.aguilar@gmail.com

AUTOMATIC ELUCIDATION OF GPI MOLECULAR STRUCTURES WITH GRID COMPUTING TECHNOLOGY

JUAN CLEMENTE AGUILAR BONAVIDES

COMPUTATIONAL SCIENCE PROGRAM

APPROVED:

_____
Ming-Ying Leung, Ph.D., Chair

_____
Igor C. Almeida, Ph.D., Co-Chair

_____
Tunna Baruah, Ph.D.

_____
Benjamin C. Flores, Ph.D.
Dean of the Graduate School

Dedication

I dedicate this dissertation to my wife, Ruth, for her unconditional support and love. Also to Clara, Oliver and Willy, who were always willing to participate in my study breaks.

AUTOMATIC ELUCIDATION OF GPI MOLECULAR STRUCTURES WITH GRID COMPUTING TECHNOLOGY

by

JUAN CLEMENTE AGUILAR BONAVIDES, M. Sc.

DISSERTATION

Presented To the Faculty of the Graduate School Of

The University Of Texas at El Paso

in Partial Fulfillment

of The Requirements

for The Degree Of

DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTATIONAL SCIENCE

THE UNIVERSITY OF TEXAS AT EL PASO

May 2013

Acknowledgements

I owe my deepest gratitude to my mentors Dr. Ming-Ying Leung and Dr. Igor C. Almeida for their encouragement, patience and guidance.

I am heartily thankful to my collaborator and friend, Dr. Ernesto Nakayasu, whose enthusiasm and support enabled me to develop an understanding of the subject.

I also want to acknowledge Dr. Tunna Baruah for kindly accepting being a part of my dissertation committee; Dr. Emma Arigi, Mr. Leonel Saldivar, Mr. Felipe G. Lopes, Mr. Gerardo Cárdenas, Mr. Luis Basurto and Mr. Julio Olaya for their productive comments and sharing their knowledge.

This work was supported by NIH grants R01AI070655, 3R01AI070655-04S1, 2G12RR008124-16A1, and 2G12RR008124-16A1S1; NHARP grant 003661-0013-2007; and NSF grant DMS0800272.

Abstract

Glycosylphosphatidylinositol (GPI)-anchored proteins are involved in many biological processes and are of medical importance. The identification and analysis of the entire collection of free and protein-linked GPIs within an organism (i.e., GPIomics) requires highly sensitive instruments. At present, liquid chromatography-tandem mass spectrometry (LC-MS/MS or -MS^n) is the most efficient laboratory technique for these tasks. As a typical MS^n experiment produces hundreds of thousands of spectra, the data analysis creates a major bottleneck in high-throughput GPIomic projects. Yet, no computational tool for characterizing the chemical structures of GPI is available to date. We propose a library-search algorithm to identify GPIs by matching fragment peaks in the spectra with molecular masses derived from a collection of theoretical GPI structures constructed based on properties of currently known GPIs. A theoretically possible GPI structure is assessed by a scoring scheme that incorporates its fitness values for individual observed spectra as well as its frequency of being considered as a good fit. The algorithm has been tested on a set of experimentally confirmed GPIs for the protozoan parasite *Trypanosoma cruzi*. The final list of predicted GPI candidates contains 76 out of the 78 known structures in the test set. Three different versions of the proposed algorithm have been developed. Firstly, we ran the algorithm on a single computer completing the predictions in approximately 10 days. A second version uses HTCondor; with an average of 16 processors, it took 3 days, 19 hours, 38 minutes to complete the job. A third version was implemented in MPI; with 72 processors it completed in 22 hours and 38 minutes. Finally, a probability is assessed by logistic regression model that can incorporate expert opinion. This computational tool is expected to quicken the discovery and characterization of GPI molecules.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Biomedical and agricultural significance

The cell membrane of living organisms contain embedded proteins, which are involved in a variety of cellular processes. Some of these proteins are anchored to the cell membrane by a glycolipid. The molecule glycosylphosphatidylinositol (abbreviated GPI-anchor or GPI), is a glycolipid composed of a polysaccharide (glycan) group and a lipid group. In mammalian cells and several protozoa some free GPIs are found at the cell surface. Free GPIs are sometimes referred to as GIPLs and share a common structure to protein-linked GPIs. The GPI-anchor is not just an "anchor" that binds proteins to the cell membrane, but also acts as a functional molecule (Fujita *et al*., 2012).

GPIs have a broad presence in living organisms; they have been identified in many eukaryotes, including humans, and are particularly abundant in protozoa. GPI-anchored proteins are involved in a number of functions such as enzymatic catalysis, adhesion, and in some cases they can mediate signal transduction across the plasma membrane (Ikezawa, 2002).

GPIs are contributing factors in medical and veterinary diseases. For instance, the dengue virus uses the cellular machinery of the host to express a GPI-anchored protein on the surface of infected cells, which is a target of human antibody responses to dengue virus infection (Jacobs *et al*., 2000). Inhibitors of GPI biosynthesis could be drug targets for diseases caused by pathogenic protozoa (Nosjean *et al*., 1997). It has been suggested that GPIs may play a role in the pathogenesis of prion diseases such as Alzheimer's and mad cow (Chesebro *et al*., 2005). In hematopoietic stem cells, a mutation that encodes the first enzyme of GPI biosynthesis results in paroxysmal nocturnal hemoglobinuria (PNH), an acquired hemolytic disease (Bessler *et al*., 2001).

Development of subunit vaccines could prove useful for the control of bovine babesiosis. Surface antigens of *Babesia bovis*, a parasite that causes important economic losses and limits cattle production in tropical and subtropical areas of the world, constitute a family of GPI-

anchored proteins. The epitopes that these proteins contain are vaccine candidates against this disease (Dominguez *et al*., 2010).

Recent studies have demonstrated that protozoan GPIs can stimulate or inhibit different cells and functions from the host's immune system; in particular, these studies have shown the ability of different GPI-anchor structures to activate macrophages (Ropert *et al*., 2000; Campos *et al*., 2001).

GPI-anchored proteins have been associated with some types of cancer. Dangaj *et al*. (2011) found that the tumor antigen mesothelin is linked to GPI anchor. This antigen is capable of activate tumor-associated macrophages (TAMs) in ovarian cancer. Zhao *et al*. (2010) have shown that GPI-anchored proteins are elevated in breast carcinoma.

1.2 Molecular structure of GPI-anchored proteins

The biosynthesis of GPI-anchored proteins is carried out in the Endoplasmic Reticulum (ER) (Ferguson, 1999). Depending on the protein to which they are attached and the organism in which they are synthesized GPI structures are very heterogeneous. A typical core structure of GPI is composed of a lipid group attached to a glycan group via inositol-phosphate; this glycolipid is then bounded to a mature protein (Figure 1.1). The lipid group allows the entire molecule to anchor the protein to the cell membrane. Heterogeneity in GPI anchors is derived from various substitutions of this core structure.

GPI linking is a post-translational modification (PTM) or the chemical modification of a protein after its translation. The PTM of proteins extends the range of functions of the protein by attaching to it other biochemical functional groups.

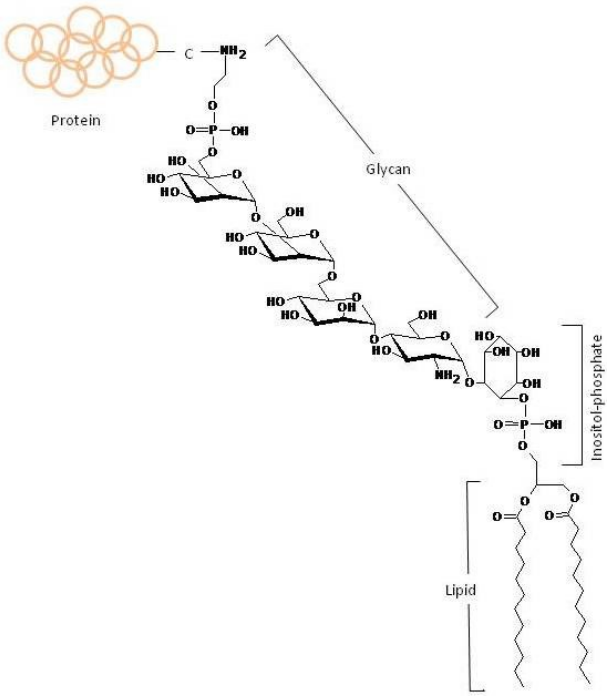


Figure 1.1 General structure of GPI anchors.

1.3 Mass spectrometry data analysis

The prediction and identification of PTMs in protein sequences is an important step in the annotation of proteomes since this can aid in developing the understanding of biological functions. Liquid chromatography-tandem mass spectrometry (LC-MS/MS) is a powerful tool for identifying these PTMs. In high-throughput experiments, mass spectrometry (MS) database search engines, such as SEQUEST provide a ranked list of peptide identifications based on thousands of MS/MS spectra obtained in a MS experiment. However, since the GPIs are complex structures, these search results are not in themselves sufficient for confident assignment of GPI sites as identification of typical mass differences requires time-consuming manual assessment of the spectra by an experienced analyst. This examination requires interpretation of the fragmentation pattern in line with the experience of the scientist, and comparison of multiple interpretations of the data.

Proteins destined to be linked with GPIs contain a short amino acid sequence, called the omega-site (ω-site), that serve as a signal to attach the protein to the GPI. This feature has been

used to develop computational methods based on Neural Networks (NNs) and Hidden Markov Models (HMMs) to predict the GPI-anchoring sequences with good accuracy (Fankhauser and Mäser, 2005). These prediction schemes are heavily dependent on training sets of experimentally confirmed protein sequences to be GPI-anchored. The main idea is to identify the ω-site of the protein and then classify it as a potential GPI linked protein. Omaetxebarria *et al*. (2007) used LC-MS/MS data for the identification of GPI-anchored proteins. They combined Bayesian networks with the use of the machine learning platform WEKA to make their predictions.

Recently, Nakayasu *et al*. (2009) identified 78 novel species of GPIs using manual interpretation of the fragmentation pattern with samples obtained from the parasite *T. cruzi* and analyzed by LC-MS/MS. Their method is the first large-scale analysis of the entire collection of free and protein-linked GPIs of a eukaryote and resulted very effectively in characterizing complete GPI molecules. Their interpretation and decision process has a series of steps that can be captured in a computational model which in turn can speed up the analysis and obtain high-quality results.

The interpretation of mass spectrometry data is an integral part of structure elucidation of chemical compounds. In essence, there are two ways to represent the outcome of an LC-MS/MS experiment: examine the LC-MS/MS data as if it was a spectrum, showing the peak masses and intensities in a graph; or use the peak table, called peak list, generated by the LC-MS/MS software. The spectrum is a visual summary of the results of a mass spectrometry experiment (figure 1.2). The Y axis is labeled relative abundance. This is the abundance relative to the tallest peak in the spectra with the tallest peak set to 100%. The X axis is mass divided by charge, *m/z*. For example, if the mass of a molecule is 2000 Da and the molecule posses two proton adducts (double charge or MS^2) its *m/z* value is equal to (2000+2)/ 2, the *m/z* value read on the spectrum is 1001. This is the peak with heavier *m/z* value in the spectra also known as the "parent ion" and accounts for the entire *m/z* of a molecule. Peaks with smaller *m/z* than the parent ion may represent the fragments of the molecule, or they may be just noise. The fragmentation pattern not only allows the determination of the mass of an unknown

compound but also allows guessing the molecular structure. For some time, interpretation of the fragmentation patterns for the identification of GPI structures requires manual assessment of the spectra. In order to recognize GPI structures after an LC-MS/MS experiment, an experienced analyst performs the following series of steps:

1. Prefilter. Since all known GPI anchors must have at least three mannose residues, the analyst filters the obtained data from an LC-MS/MS experiment by the presence of fragments correspondent to this monosaccharide (shift of 162.052823 Da).
2. Look for familiar peaks. Once the data has been prefiltered, the analyst performs a visual inspection of the spectra and looks for the parent ion and peaks that resemble the mass and fragmentation pattern of a GPI molecule and selects various candidate spectra.
3. Chemical structure drawing. For each peak that resembles a part of a GPI, the analyst draws a chemical representation of this fragment using software such as ACD/ChemSketch. Adding up all the fragments the software calculates the mass of the structure which should be equivalent to the parent ion *m/z* value. This step is performed multiple times in order to take into consideration the possible variations of the molecular structure.
4. Data interpretation. After considering the different possibilities of a structure, the analyst decides the best candidate (or candidates) according to his experience and knowledge.

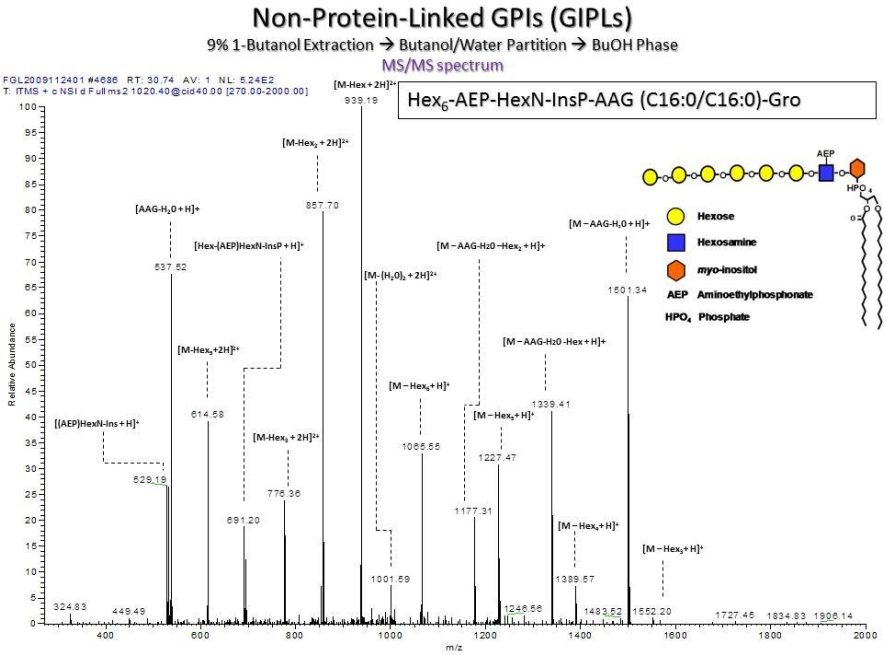


Figure 1.2 MS^2 spectra of GPI species Hex6-AEP-HexN-InsP-AAG(C16:0/C16:0)-Gro with annotation

A DTA file is a representation of the spectra that contains the parent mass, charge and observed fragmentation pattern in the form of peak lists. A DTA sample is provided in figure 1.3.

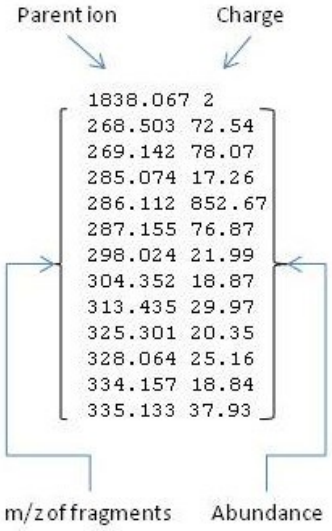


Figure 1.3 Peak list sample generated by the LC-MS/MS software

1.4 Statement of the Problems

The aim of the current dissertation is to provide a tool to automate the analysis and interpretation of the GPIomic MS data, attempting to address the following problems:

- **Problem 1.** How can we build a computational model to represent the analysis and decision process of an experienced scientist to substantially speed up and validate with robust analytical criteria a large scale GPIomic experiment?
- **Problem 2.** Can we make use of specialized parallel computer architectures alongside traditional serial computation, for accelerating the time to obtain the predictions?
- **Problem 3.** How can we incorporate expert opinion into the prediction?

The following summarizes my approach to these three problems.

Problem 1

I propose a library-search algorithm to identify GPIs by matching fragment peaks in the spectra with molecular masses derived from a collection of theoretical GPI structures. The algorithm begins with the reduction of the data. Afterwards, it finds the set of all candidate structures in a structure library that have a match with the observed parent ion mass. Then it scores and counts the number of times each candidate structure is observed. A fitness function scores every identified structure and provides a list of the most likely unique structures considering every DTA. Figure 1.4 is a schematic representation of the proposed algorithm.

The biosynthesis of GPI molecules produces a core structure consisting of a lipid attached to a glycan as shown in Figure 1.1. The glycan group can be produced with a combination of a small number of monosaccharides along with inositol and phosphate. The lipid part of the molecule can have one of the four possible types of lipid tails: ceramide, lyso-acyl or lyso-alkyl glycerol (lyso), alkylacylglycerol (AAG) and diacylalkylglycerol (DAG). A data table with all the possible combinations of these two groups can be constructed, making it possible to scan a given peak list for a match between a parent ion and a structure. The construction of the glycan and lipid tables corresponds to step 0 in the algorithm.

Our aim is to develop a tool that automates the processing of large numbers of spectra with high sensitivity and sufficiently good selectivity in the identification of GPI molecules. After each one of the candidate structures have gone through the process of finding a match in the peak list, they must be scored to determine the most probable structure (or structures). An explanation of the score system developed is presented in sections 3.2.4 and 3.2.5 (Steps 3 and 4 of the algorithm).

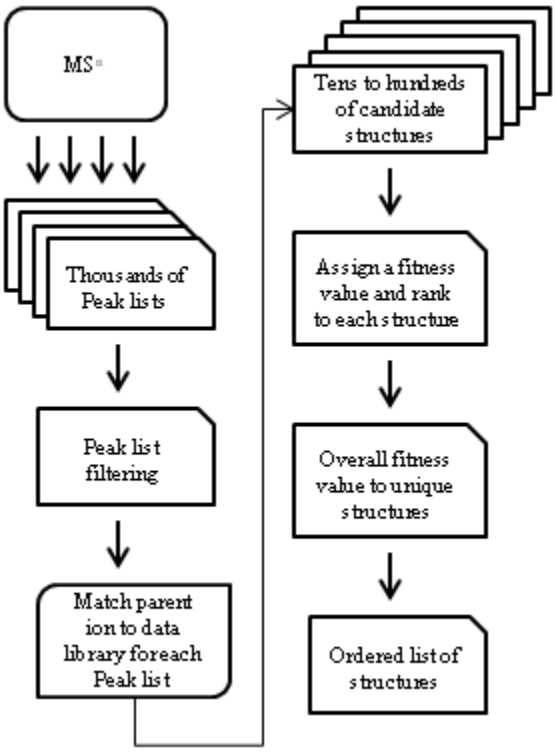


Figure 1.4. Algorithm flowchart for GPI structure identification

The computational assessment of a peak list over the manual interpretation of a spectra should accelerate a GPIomic analysis both quantitatively and qualitatively; quantitatively because a large volume of LC-MS/MS data can be analyzed in less time; and qualitatively because the manual assessment of a spectrum can be influenced by subjective judgment, and researchers with less experience in this type of analysis can benefit with the aid of an automated process.

Problem 2

The development of a computational model to characterize GPI molecules using a library-search approach can require a big amount of computer time and memory resources. In particular because the proposed algorithm needs to match thousands of peak lists against a data library. Ideally one would like a method that would be computationally efficient. I propose the use of HTCondor to distribute in a number of computers the tasks of peak list matching. HTCondor enables to effectively harness wasted CPU power from otherwise idle desktop workstations.

Problem 3

Expert opinion may be incorporated into the model in an iterative process between the expert and prediction model. Experts can provide input about the significance of the results, and then a logistic regression (LR) model can be employed to obtain predictive values for the scoring scheme and adjust the model. At the same time LR can supply a value for the predictions in the form of probabilities.

The dissertation is organized as follows. In chapter 2, we introduce the biochemistry, LC-MS/MS analysis and review of the current computational prediction methods of GPIs. In chapter 3, we will show our methodology for modeling large scale GPIomic experiments, including our algorithm and score system. In chapter 4 we present the results and discuss the effectiveness of the model. Finally in chapter 5 we talk about our conclusions and future work.

Chapter 2: Literature Review

2.1 GPI anchor molecules

All living cells have a membrane that encloses their contents and serves as a semi-porous barrier to the outside environment. The cell membrane is composed of a bilayer of phospholipids which have a hydrophilic head and two hydrophobic tails. Within the phospholipid bilayer of the cell membrane, many different proteins are either peripheral proteins or integral membrane proteins. Some of these proteins have carbohydrates attached to their external surfaces and are, therefore, referred to as glycoproteins. A number of these glycoproteins contain also a lipid group, which help anchor the entire molecule to the cell membrane. A glycan part attached to a protein faces the extracellular environment (Figure 2.1). GPIs are present in eukaryotes and take part in significant biological processes such as cell-cell interactions and antigenic presentation (McConville and Ferguson, 1993).

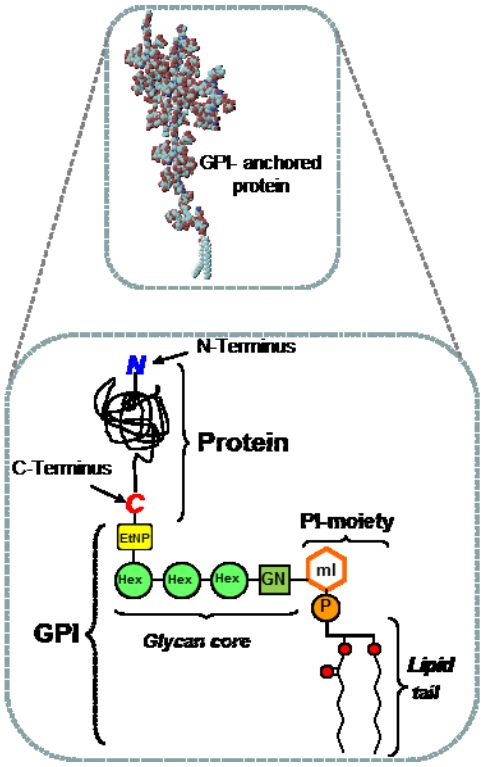


Figure 2.1 GPI anchored molecule.

2.1.1 Biochemistry

The biosynthesis of GPIs requires a variety of enzymes, all of them acting in a defined and consecutive way. These reactions are carried out in the endoplasmic reticulum (ER). The final product posses a common structure consisting of a lipid tail attached to a glycan core (Figure 2.2). Modifications of this general structure make a number of variations of the molecule; these include extra mannose (Man) , ethanolaminephosphate (EtNP), and/or aminoethylphosphonate (AEP) residues substituting the glycan core, and/or an extra fatty acid (acyl) group attached to the myo-inositol ring, increasing the complexity of the GPI structure (Ferguson, 1999; McConville and Ferguson, 1993). Because of their complex structure and amphiphilic nature, GPIs are difficult to be extracted, purified, and fully characterized.

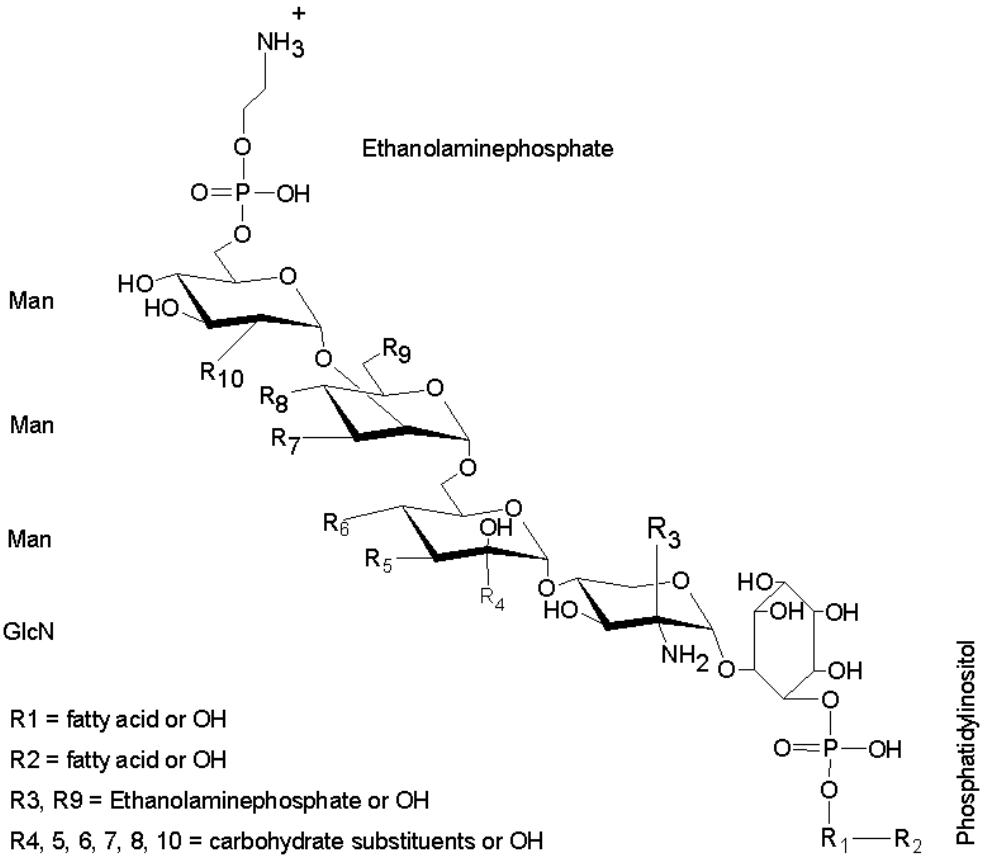


Figure 2.2 General structure of GPI anchors. Diversity of GPIs is derived from various substitutions of this core structure and is represented as R groups.

2.1.1.1 Glycan part

The term glycobiology joins the knowledge of carbohydrate chemistry and biochemistry with modern understanding of molecular biology of complex carbohydrates, which are often named glycans in this context. Glycans, therefore refer to the carbohydrate portion of a glycoconjugate, such as a glycoprotein, glycolipid or a proteoglycan.

In general, the glycan part of a GPI molecule consists of a phosphatidylinositol linked to a glucosamine (GlcN) residue followed by three Man residues. The terminal Man is connected to EtNP, which links in turn to the C-terminus of a protein.

The structural complexity of glycans, far greater than that of proteins and nucleic acids, allows them to encode information for specific molecular recognition and to determine protein folding, stability and pharmacokinetics (Morelle and Michalski, 2005). The glycan may be a single monosaccharide or an oligosaccharide. The attachment of glycans to a protein makes glycoproteins especially diverse. Glycans are linked to other biomolecules, such as lipids or amino acids, through glycosidic linkages to form glycoconjugates. In general, glycoproteins compose an assorted population of glycoconjugates containing between one and several dozen different glycans. The position and amount of glycans within a molecule makes it possible to form as many as 10^{12} distinct structures from as few as six different monosaccharide units (Morelle and Michalski, 2005). Therefore, in order to elucidate the structure of a particular glycan it is necessary to determine the sequence, composition and branching of its monosaccharide units as well as its glycosidic linkages and anomeric configuration. For that reason, the challenges of analytical glycobiology are much greater than those encountered in genomics and proteomics.

2.1.1.2 Lipid part

Lipids are a diverse class of biological molecules that play a central role as structural components of biological membranes, energy reserves, and signaling molecules (Yetukuri, et al., 2007). Lipids are structurally highly diverse because of the many possible variations of the

lipid building blocks, how these blocks are linked and their variation of both chain length and degree of saturation. Yetukuri, et al. (2007) estimated that the theoretical number of lipids covering major lipid classes is close to 200,000.

There are four classes of lipid tails in a GPI molecule: ceramide, lyso-acyl or lyso-alkyl glycerol (lyso), alkylacylglycerol (AAG) and diacylalkylglycerol (DAG). In figure 2.3 we show some lipid tails that can be present in GPIs.

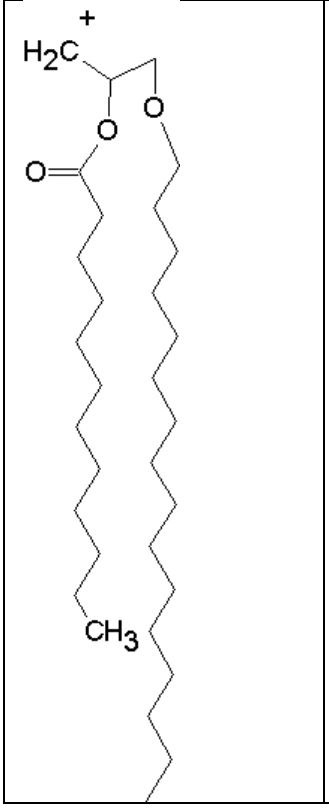
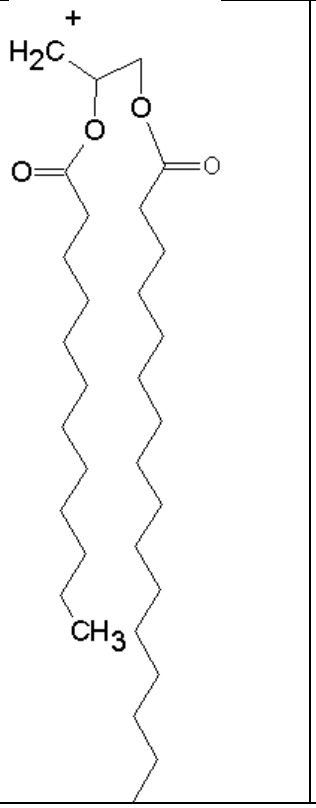
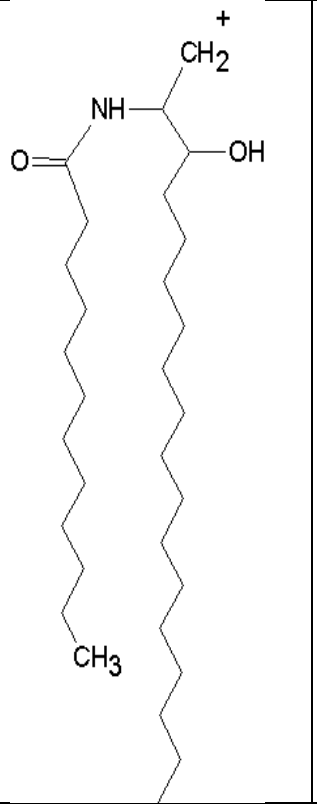
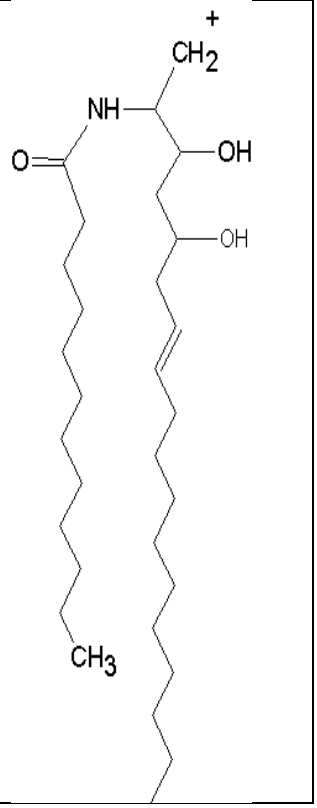
AAG-C16:0/C12:0	DAG-C16:0/C12:0	Ceramide-C12:0/d18:0	Ceramide-C12:0/t18:1
			

Figure 2.3 Possible arrangements of lipid tails in GPIs

2.1.1.3 GPI diversity

Isobaric ions are ions that have the same nominal mass but different exact mass. For example N_2, C_2H_4 and CO all have a nominal mass of 28 Da. Their exact masses are: N_2 = 28.00615 Da, C_2H_4 = 28.0313 Da and CO = 27.99491. Because of the variety of the structures, modification and branching GPIs are often isobaric (Figure 2.4).

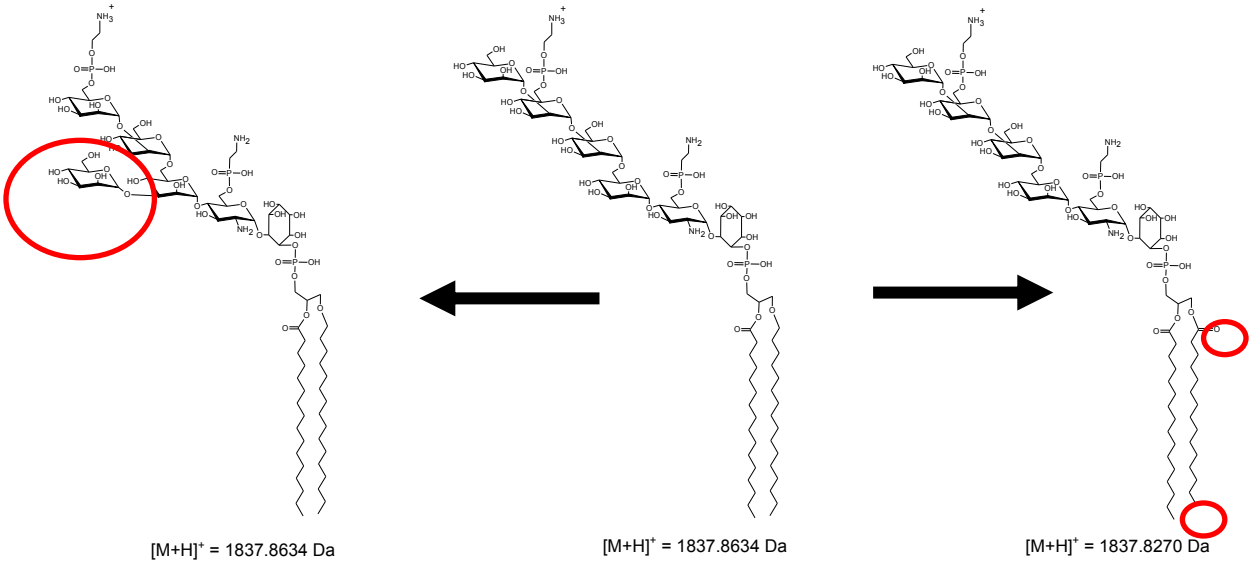


Figure 2.4 Isobaric GPIs

2.1.2 LC-MS/MS analysis

The study of glycoconjugates is a challenging task because these molecules exhibit complex structures that can differ in linkage and the level of branching. MS is one of the most powerful and versatile techniques for the structural analysis of glycoconjugates. The characterization of glycoproteins by mass spectrometry is typically more difficult than the mass spectrometric analysis of proteins, because glycoproteins exhibit extensive heterogeneity and because they are ionized less efficiently than proteins (Morelle and Michalski, 2005).

LC-MS^n has been successfully used for the characterization of GPI molecules (Nakayasu et al., 2009). The structure of GPIs can be elucidated using MS, exposing the types of glycans present and providing evidence of structures that are potentially important for biological function.

Monosaccharide sequences, branching, and linkages can be determined through fragmentation. The observed fragments depend on factors such as the type of ion ([M+H]+, [M+Na]+ etc.), its charge state, and the time available for the fragmentation (retention time). In general, glycans fragment to give two major types of ions. These ions are the result of two types of cleavage: glycosidic cleavages where bond rupture occurs between the sugar rings and involves a hydrogen migration and cross-ring cleavages that involve the rupture of two bonds on the same monosaccharide. The nomenclature generally used for describing these fragment ions is that proposed by Domon and Costello (1988) (Fig. 2.5). Glycosidic cleavages provide information on constituent monosaccharide sequence and branching. Crossring cleavages are usually weaker. The fragmentation of glycans through LC-MS^n is usually performed by MS^2.

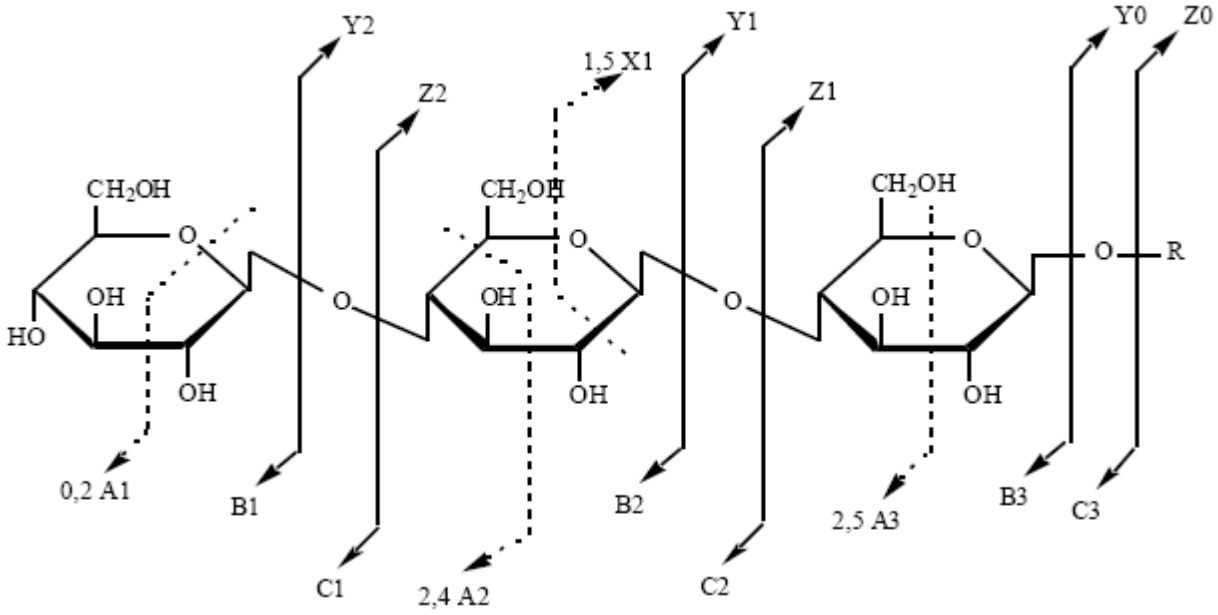


Figure 2.5 Nomenclature for describing the fragmentation of carbohydrates, after Domon and Costello (1988)

Although the lipids commonly produce specific fragmentation patterns by tandem MS, such information is not always available for each specie analyzed by lipidomics experiments (Niemela et al., 2009). Due to the structural characteristics of lipids their identification from fragment mass spectra requires the use of MS^3. We, however, receive data from MS^2 experiments in which the fragments of glycans can be present but the lipids are not fragmented, they only

appear as single peaks. Then we are aiming to interpret GPIs with a fragmented glycan and the entirety of the lipid part.

2.2 Computational Prediction

Bioinformatics resources for glycomics and lipidomics are very poor as compared with those for genomics and proteomics which are widely available for molecular biologists. Tools for the characterization of glycans and lipids have been built separately, but essentially they mirror the way in which proteomic characterization has been performed. There are two main classes of algorithms for the identification of all these compounds given LC-MS/MS data: database search algorithms and de novo sequencing algorithms. Here we will describe both.

2.2.1 Database search algorithms

In proteomics, various algorithms and computer programs have been developed for the identification of protein sequences using MS data to search over a database of known proteins. Corresponding *m/z* values within the database are scored and ranked to find the best match between a protein and the spectra. MS data are submitted to these programs in the form of peak lists. One of the first programs of this kind which is still widely used is SEQUEST (Eng et al., 1994). In SEQUEST the interpretation of a spectrum for an unknown protein sequence proceeds by identifying a consecutive series of fragment ions whose differences correspond to residue masses for amino acids. In general there are four steps in the identification of proteins using this software. On the basis of mass alone, SEQUEST searches a database for candidate peptides. A virtual spectrum is created for each of these peptides and then it checks whether it matches the observed spectrum. The final outcome is a list with the best peptides that matched the spectra; each peptide is shown with their corresponding score.

MASCOT (Perkins et al., 1999) is another database search algorithm. MASCOT's fundamental approach is to calculate the probability that the observed match between the experimental data set and each sequence database entry is a chance event. The match with the lowest

probability is reported as the best match. Intensity information is ignored because peak intensities depend on the physical and chemical properties of the samples.

Since glycan databases are very small when compared to those existing in proteins and genomes, theoretical databases have been created to mimic searching algorithms used in proteomics. For example, Joshi et al. (2004) generated a theoretical database of glycan fragments. Using the 1674 fully characterized carbohydrate structures from GlycoSuiteDB (Cooper et al., 2003) they systematically produced a database with 3×10^6 fragments taking into consideration the Domon and Costello notation. With this database a searching algorithm finds the set of all candidate structures that have a fragment mass within a tolerance for each observed mass. Then, the union of the sets of candidate structures accumulated is found counting the number of times each carbohydrate structure is observed. Finally the structures are sorted by the number of times each carbohydrate structure was observed in order of most number of hits to least.

Another method for annotating possible N-glycan structures synthesized by mammals is to generate libraries based on biosynthetic rules and patterns, also called cartoons (Goldberg, et al., 2005). A table of potential glycans is used to match the peaks with cartoons within a tolerance. The cartoons provide compositional information and predictions of probable topologies. When the set of cartoons is obtained, a confidence score is assigned to each cartoon, hence the number of matching structures is greatly reduced

Lipid identification strategies also include the generation of databases with theoretically possible lipids, with information on masses, isotope patterns and additional constraints such as retention time (Yetukuri et al., 2007).

2.2.2 De novo sequencing algorithms

The de novo peptide sequencing problem is the reconstruction of a peptide sequence from LC-MS/MS data without the aid of a database from which known peptides can be matched. In the process of collision-induced dissociation (CID), a peptide bond at a random position is broken

into complementary ions, typically N-terminal ions called b-ions (prefix) and C-terminal ions called y-ions (suffix). Intermediate bonds are also broken (Figure 2.6). These ions are complementary because joining them determines the original peptide sequence.

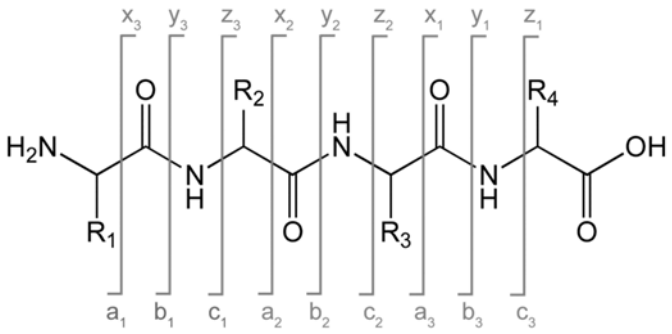


Figure 2.6 Nomenclature for describing the fragmentation of peptides

The interpretation of a spectrum basically deals with two factors (Chen et al. 2001): (1) it is unknown whether a mass peak corresponds to a prefix or a suffix subsequence; (2) some ions may be lost in the experiments and the corresponding peaks disappear in the spectra. In practice noise and other factors can affect a spectra. An ion may display two or three different peaks because of the distribution of isotopic carbons in the molecules (i.e., C^{12} and C^{13}). An ion may lose a water molecule or an ammonia molecule and display a different peak from its normal one. Also, the fragmentation may result in some other ion types such as a- and z-ions.

Dancik et al. (1999) approached the de novo sequencing problem by the following method. First, the spectral data is transformed to a directed acyclic graph, called a spectrum graph, where a node corresponds to a mass peak; an edge connects two nodes that differ by the total mass of possible amino acids in the nodes. Each node represents a possible prefix subsequence for the peak. Then, an algorithm is called to find the highest-scoring path in the graph or all paths with scores higher than some threshold. The concatenation of edge labels in a path gives one or multiple candidate peptide sequences. The longest path in the resulting directed acyclic graph is the path that best explains the spectra.

Glyco-Peakfinder (Maass et al., 2007) is a web-service for the de novo determination of the composition of glycans. The algorithm requires a set of mandatory masses including the possible occurring monosaccharides, the cross-ring fragments for each monosaccharide, modifications that can occur at the reducing end, and charged ions. The calculation basically is a consecutive addition of mass increments of monosaccharides and losses of other small molecules that lead to core structure and depends on user settings. The algorithm provides all possible results for each given m/z value from a mass list independently from other peaks.

As reviewed above, numerous software packages have been created for the assignment of free glycan compositions. Following the creation of these software packages, various research groups released software packages aimed at solving glycopeptide compositions. It is estimated that glycoproteins occur on more than half of all eukaryotic proteins (Apweiler et al., 1999). One of the most recent software tools designed to accelerate the process of accurately determining glycopeptide composition from MS/MS data is GlycoPep grader (GPG) (Woodin et al., 2012). GPG is a freely available software tool and it assigns glycans at sites known or predicted to be glycosylated.

2.2.3 Computational Prediction of GPI-anchored molecules

In many respects, a library search algorithm for the annotation of GPIs faces a different set of complexities from those of protein, glycan or lipid database-search algorithms alone. GPI structures are not isolated from the protein they are linked with, nor from the cell, tissue and developmental stage of the organism to which they belong. Also, glycans and particularly lipids often exist as isomers differing only in their degree of branching and/or linkage in the case of glycans, and in their degree of saturation and chain length in the case of lipids.

Some algorithms have been successful in predicting protein anchoring motifs (Poisson et al., 2007; Pierleoni et al., 2008). They identify GPI-anchored proteins on existing protein sequences, looking for characteristic signals of hydrophobicity and spacer regions at the N-terminus and the C-terminus of a given protein sequence to find the omega site (ω-site). This is the site of GPI anchor addition within the protein. In general, these algorithms include a two-step filtering

procedure a classification system based on support vector machine (SVM) (Pierleoni eta al., 2008) or Neural Networks (NN) (Poisson et al., 2007) that classifies the proteins as being GPI- and non-GPI-anchored; and a hidden Markov model (HMM) that predicts the cleavage site. In contrast, the distinctive feature of our method is that it indentifies the anchor part of the molecule using LC-MS/MS data.

Recently, the web server predictor ProFASTA (de Groot et al., 2012) has been developed to perform filtering of proteins with cell surface characteristics, including GPI-anchored proteins. This tool offers *in silico* analysis of protein sequence properties in large datasets and returns sequences in FASTA format. ProFASTA as such is not a new protein predictor but enables and facilitates custom analysis and extraction of data from commonly used cell surface protein predictors, such as SignalP, TMHMM and big-PI. In addition, it provides keyword, iso-electric point, composition and pattern scanning.

Chapter 3: Methods

3.1 Modeling Large Scale GPIomics Experiments

Because the biosynthesis of GPIs requires a variety of enzymes, all of them acting in a defined and sequential way, there are currently no methods available to amplify these molecules similar to DNA amplification using polymerase chain reaction (PCR) techniques. Consequently, highly sensitive analytical chemistry methods have to be applied. This is a difficult task and few GPIs have been completely characterized, although large-scale screening of samples is a realistic possibility (Nakayasu *et al*., 2009).

Many GPIs are very similar, or are isometric forms, and therefore difficult to separate and may be present in only low amounts. The peaks separated in LC-MS/MS profiles cannot generally be identified directly as several GPIs may have the same properties. For example, they might have the same mass. Hence, the identification of GPI structures requires detailed knowledge of the properties of the molecule and its fragmentation patterns. So far, this process has been performed manually by experienced analysts.

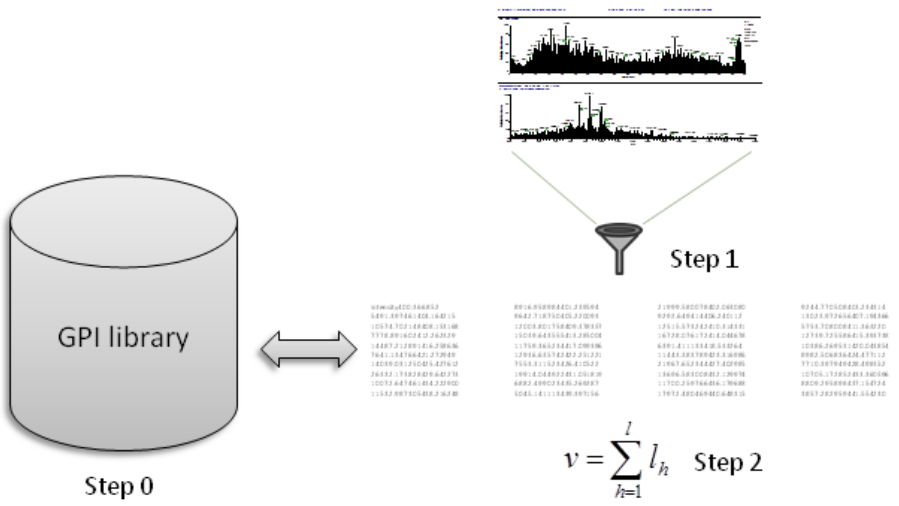
GPIomic analysis is comparable to proteomic analysis. However the development of an algorithm will differ due to the possible presence of branch points, linkages and anomericity in GPI molecules. In peptide sequencing, the fragmentation is across the peptide bond and a linear sequence can be matched to a database of known proteins. Similarly, in order to identify GPI structures, LC-MS/MS data can be matched against a theoretical database containing a set of all theoretically possible GPIs. The key to this methodology, as in protein sequencing, is the criteria established for ranking the potential GPI structures that are compatible with the observed spectra.

3.2 Serial Algorithm

Our algorithm for the identification of GPI structures, in general works in the following way: Given a set of LC-MS/MS data, match the spectra against a theoretical library of structures; then rank these structures and keep the best. The analysis strategy begins with the

development of a theoretical library of GPI structures (Step 0). The set of data from the spectrometer is reduced (Step 1). Then, the mass and charge state of parent ions are determined in order to search for corresponding structures in the GPI library, next the fragments for each structure are mapped against the peak list. Every candidate structure is evaluated by a fitness value function and ordered (Step 2). The structures are scored and ranked (Step 3). Finally, a logistic regression model is used to assess the probability of every structure.

The steps are explained below in more detail and an example is shown in Figure 3.1.



S	Mass	Structure	Avg d	Avg v	Frequency	DTA	Y	P	Rank
34686.40	1836.82	EtNP-Hex4-AEP-HexN-Ins-P-DAG-31:0	1.19	55.55	712	esn20080403026452645 2.2.dta	40.45	1.00	1
31858.19	1836.856	EtNP-Hex4-AEP-HexN-Ins-P-AAG-32:0	1.22	55.50	714	esn20080403026452645 2.2.dta	40.42	1.00	2
24056.13	1837.799	EtNP-Hex4-AEP-HexN-Ins-P-36:6d	2.10	53.87	598	esn20071026014868.4868.2.dta	37.54	1.00	3
21954.67	1837.855	EtNP-Hex4-AEP-HexN-Ins-P-34:0t	2.14	53.86	594	esn20071026014868.4868.2.dta	37.44	1.00	4
10721.74	1820.861	AEP-Hex4-AEP-HexN-Ins-P-AAG-32:0	1.34	53.18	314	esn20071026014958.4958.2.dta	32.58	1.00	5
11599.30	1820.825	AEP-Hex4-AEP-HexN-Ins-P-DAG-31:0	1.35	53.18	314	esn20071026014958.4958.2.dta	32.57	1.00	6
7590.42	1982.878	EtNP-Hex5-AEP-HexN-Ins-P-AAG-31:1	2.93	58.02	218	esn20080513185507.5507.2.dta	28.77	1.00	7
8106.26	1982.841	EtNP-Hex5-AEP-HexN-Ins-P-DAG-30:1	2.87	58.09	214	esn20080513185507.5507.2.dta	28.69	1.00	8
7770.44	1981.896	EtNP-Hex5-AEP-HexN-Ins-P-34:1d	2.58	58.59	194	esn20080513184611.4611.2.dta	28.25	1.00	9
7170.59	1981.86	EtNP-Hex5-AEP-HexN-Ins-P-33:2t	2.64	58.56	195	esn20080513184611.4611.2.dta	28.23	1.00	10
6969.21	1983.912	EtNP-Hex5-AEP-HexN-Ins-P-34:0d	3.34	57.44	206	esn20080513184611.4611.2.dta	27.70	1.00	11
6514.81	1983.875	EtNP-Hex5-AEP-HexN-Ins-P-33:1t	3.38	57.42	207	esn20080513184611.4611.2.dta	27.70	1.00	12
6207.56	2497.175	Hex8-AEP-HexN-Ins-P-44:3t	4.76	55.08	248	esn20080513254576.4576.2.dta	27.26	1.00	13
6305.96	2020.904	Hex6-AEP-HexN-Ins-P-33:2t	3.08	57.29	187	esn20080513183964.3964.2.dta	27.16	1.00	14
6609.76	2020.94	Hex6-AEP-HexN-Ins-P-34:1d	3.10	57.22	185	esn20080513186654.6654.2.dta	26.97	1.00	15

Step 3

Figure 3.1 example of the steps required to obtain a GPI structure

3.2.1 Step 0: GPI library.

The biosynthesis of glycoproteins and glycolipids is not under direct genetic control, and different enzymes are involved in the synthesis of sugar chains attached to proteins or lipids. This can lead the cell in the production of several different glycan and lipid chains, many of them unknown. The availability of a comprehensive GPI database is a prerequisite for successfully developing a computational tool aimed at deciphering new, so far unknown GPI structures. We developed our own library of theoretical GPIs by combining the building blocks of the molecule. Our library consists of 1.9 million possible GPI structures.

In the first place we considered the basic glycan core structure of GPI molecules (Figure 1) (Ferguson, 1999). This is composed of three Hex residues, and one hexosamine, followed by InsP. By making substitutions using the residues that are observed in known GPIs, we obtained a number of potential glycans. Other building blocks of the glycan part are: deoxyhexose (dHex), ethanolaminephosphate (EtNP), aminoethylphosphonate (AEP), N-acetylhexosamine (HexNAc), and N-acetylneuraminic acid (NANA). Table 3.1 shows the building blocks of the glycan part plus Inositol and Phosphate.

Table 3.1. Constant mass values for glycan residues, inositol and phosphate.

Residue	Abbreviation	Mass (Da)
hexose	Hex	162.053
deoxyhexose	dHex	146.058
ethanolaminephosphate	EtNP	123.008
aminoethylphosphonate	AEP	107.014
hexosamine	HexN	161.069
N-acetylhexosamine	HexNAc	203.080
N-acetylneuraminic acid	NANA	291.095
myo-inositol	Ins	162.053
phosphate	P	97.977
Pentose	Pent	132.04

The substitutions were made adding residues in different positions of the glycan core considering the following rules:

The number of Hex residues can vary from 3 to 8; the number of deoxyhexoses (dHex) can be 1 or 2; and ethanolamine phosphate (EtNP) residues can be 1, 2 or 3.

We developed the following formula to represent the different permutations that the glycan residues can form:

X=NANA(0:1)-HexNAc(0:1)-dHex(0:2)-Hex(3:8)

X-HexN-InsP

X-AEP-InsP

AEP-X-InsP

AEP-X-AEP-InsP

EtNP(1:3)-X-InsP

EtNP-X-AEP-HexN-InsP

The myo-inositol ring is the linker between the lipid and glycan part of the molecule and is followed by phosphate (inositolphosphate or InsP). InsP can take five different forms. It can be present in the molecule as the ring itself (Ins) or the following acylated forms: Ins(C16:0), Ins(C18:0), Ins(C18:1) or Ins(C18:2). The first number indicates the chain length and the second, the number of unsaturations (double bonds) in the chain. We obtained 2880 potential glycans.

We combined the 72 rows and 60 columns and obtained 2880 potential glycans.

While modifications of the glycan part include additional fragments arranged in different way to the core structure, lipids can differ in chain length and saturation. There are five classes of lipid

tails in a GPI molecule [Ferguson]: ceramide (Cer), lyso-alkylglycerol (LAkG), lyso-alcylglycerol (LAcG,) alkylacylglycerol (AAG), and diacylglycerol (DAG).

Each organism and each tissue may synthesize Cer moieties in which there are a variety of di- (d-Cer) and trihydroxy (t-Cer) bases linked to the fatty acids. To construct our library we considered a chain length in a range of 30 to 44 carbons and 0 to 6 unsaturations in the ceramide (Cer) moiety.

LAkG and LAcG are phospholipids missing one of its two acyl chains; we considered chains with 12 to 26 carbons, and 0 to 4 unsaturations for these fatty-acids. AAG consists of one alkyl chain at C-1 and one long fatty-acid chain at C-2 position of the glycerol backbone. On the other hand, DAG consists of two fatty acid chains at C-1 and C-2 positions. We considered 28-52 carbons and 0-5 unsaturations in the fatty acid and alkyl chains to build our library.

In total we obtained 660 different lipid structures. Our theoretical GPI library is the combination of the glycan and lipid arrangements. Table 3.2 shows a sample of these GPI structures.

AEP has been found so far in some species of Trypanosomids and no other organism (Varki A, et al., 2009), then, when we perform a search we can use any of two versions of the library; one version contains all the structures and a shorter version contains only those structures that do not have AEP. This second version is used to search structures of those organisms different than Trypanosomids, accelerating the search process in these species.

Table 3.2. Sample of theoretical GPI structures included in the GPI library.

Structure	Mass (Da)
Hex3-HexN-Ins-P-12:4/lysoakylglycerol	1142.4263
Hex3-HexN-Ins-P-12:3/lysoakylglycerol	1144.4419
EtNP3-HexNAc-Hex4-HexN-Ins-P-40:6dCer	2073.8950
AEP-dHex-Hex4-AEP-HexN-Ins-P-DAG-40:4	2084.9614
HexNAc-Hex6-AEP-HexN-Ins-P-AAG-29:5	2188.9073

EtNP-NANA-Hex5-HexN-Ins-P-AAG-32:5	2188.9075
AEP-NANA-Hex5-HexN-Ins-P-DAG-32:4	2188.9075
AEP-Hex7-HexN-Ins-(C18:0)-P-35:2tCer	2639.3035
NANA-Hex7-HexN-Ins-P-AAG-52:1	2678.3809
dHex-Hex8-HexN-Ins-(C18:0)-P-36:1dCer	2678.3841
dHex2-Hex7-HexN-Ins-(C18:0)-P-36:1tCer	2678.3850

3.2.2 Step 1: Peak list filtering.

High-throughput profiling of GPIs generates very large amounts of data. All these data, however, can be used meaningfully only if we can eliminate most of the noise and extract the relevant information pertaining to the fragmentation patterns of GPI molecules. Peaks in a spectrum differ in their intensities, which depend on the physical and chemical properties of the samples. Our criteria to select the strong and clean peaks corresponding to GPIs are based on examining the set of spectra from the confirmed structures (Nakayasu et al., 2009).

a) High-confidence peak assignments are typically among the most intense 200 peaks (Goldberg et al., 2005). We observed that the fragments corresponding to confirmed GPIs are all above 15 percent of the maximum peak intensity and the corresponding spectra has at least 130 peaks; we therefore use peaks above 15 percent intensity and eliminate those peak lists below 130 rows.
b) The presence of Hex residues is an indication that the data can be used to characterize entire GPIs. The algorithm identifies peak lists by the presence of fragments corresponding to this monosaccharide (shift of 162.053 Da). This is done by taking the difference between two fragments observing if the difference is within the tolerance from the mass of Hexose. Since two consecutive fragments can have minor variations in mass due to the fragmentation process, we took the difference in Hex shift between the current peak and the next ten peaks, starting from the first peak; we call this difference an r-difference where $r = 1,2, \dots, 10$. If one of those differences matches this criterion, we consider the peak list a candidate for searching against our GPI library.

$$162.053 - \mathrm{t} \leq \mathrm{m_{j+r}} - \mathrm{m_j} \leq 162.053 + \mathrm{t}$$

Here, *m* is the mass of the current fragment above intensity cutoff; $\mathrm{j} = 1,2,\ldots,\mathrm{n}$ is the number of peaks above intensity cutoff in peak list; s = shift due to Hex residue = 162.053; and *t* = tolerance. Our tolerance here is 1 Da.

3.2.3 Step 2: Search for structures and fragment identification.

For each filtered peak list, the algorithm finds the parent ion and charge. This information reveals the whole electrically charged molecule that dissociates to form the fragments, whose information occupies the remaining rows of the file.

To match a spectrum to a structure in the data table, the *m/z* of the parent ion is scanned throughout the data table to find all possible structures with a tolerance of ±2Da of the *m/z* value. The *m/z* value is a quantity formed by dividing the mass number of an ion by its charge number, for example, for the ion $C_7H_7^{+2}$, *m/z* equals 45.5. The *m/z* calculation also provides information of the possible fragmentation pattern of the parent ion ($[F+H]^+$, $[F+Na]^+$, $[F+2H]^{+2}$, etc).

For each candidate structure the algorithm iterates through the fragments to find an occurrence within the peak list with a tolerance of ±0.5Da of the value. A lookup table of theoretical fragments is used to acquire the mass value of the current fragment.

3.2.4 Step 3: Scoring and ranking.

For each GPI candidate structure that matches the *m/z* of the parent ion, the algorithm iterates through the fragments to find an occurrence within the peak list with a tolerance of ±0.5Da of the fragment mass value. For simplicity, we represent a GPI structure as a sequence of blocks. Each block is a glycan residue, InsP or a lipid.

Let m_i = mass of the *i*th block, *i*= 1, 2,…, *l*, where *l* is the number of blocks in the candidate structure. So, we can write

$$M = \sum_{i=1}^{l} m_i$$

where M = parent ion *m/z*. Furthermore, we define the *h*th, *h* = 1,…, *l*, fragment mass as

$$f_h = \sum_{i=1}^{h} m_i$$

Each candidate structure is evaluated by counting the number of fragment matches within the peak list. Suppose there are *k* observed fragment masses, denoted by $s_1, s_2, \ldots, s_k$ listed in the peak list. Let *t* denote the tolerance. For *h* = 1,…,*l*, define

$$I_h = \begin{cases} 1 & \text{if } f_h - t \le s_j \le f_h + t \text{ for some } 1 \le j \le k \\ 0 & \text{otherwise} \end{cases}$$

Then

$$v = \sum_{h=1}^{l} I_h$$

represents the fitness value for the candidate structure.

For every fragment match, the fitness value (*v*) is also increased by the relative intensity of the fragment. It is widely accepted that a valid primary peak tends to have high intensity and is accompanied by derivative peaks, including isotopic peaks, neutral loss peaks, and complementary peaks (Zhang *et al*., 2011). The largest peak in the spectrum (100% relative intensity) is called the base peak. We increased v by adding to it the ratio between the base peak and the intensity value of the current peak.

Diagnostic fragment-ions are also considered. A diagnostic ion is a fragment ion that is sufficient to identify a molecular species. Fragment-ions corresponding to the neutral loss of

Hex residue(s), and AEP or EtNP are highly abundant in *T. cruzi*. For instance, one of the most abundant ions is the fragment corresponding to the InsP attached to AEP-HexN (AEP-HexN-InsP) (*m/z* 529.3) (Nakayasu *et al*., 2009). EtNP linked to the first Hex residue of the molecule is another major feature of the GPI structure (Vainauskas and Menon, 2006). If a diagnostic ion is present in the structure, *v* is incremented by 25 points. The neutral losses of water are also considered and if a fragment – $(H_2O)_2$ is found, *v* is incremented by one point.

For every peak list we obtain a list of structures with a corresponding *v*; then these structures are ordered. We use a constant ordering function assigning the same order to those structures with the same fitness value, causing the structures with higher v to be listed first. We call this function the fitness order function and represent it as *d*.

Every structure obtained can be associated with a number of peak lists; therefore, to obtain a list of sorted unique structures, a scoring function was developed. In general, our scoring function is based on the following criteria: the more frequent the structure, the more likely it is; the higher the fitted value and fitted order assigned to the structure, the more likely it is. The scoring function is represented as:

$$S = e^{-\bar{d}} \sum_i v_i$$

Where $\bar{d}$ is the average of the fitness order function *d.* We select only the top 10 ranks for every peak list. We considered a number of different summarization methods but chose this approach because it was most highly associated with GPI identification.

A Perl script was written to implement the algorithm. The computer used to run the script was a Fujitsu server with two quad-core Xeon (8 cores total) CPU, 8GB RAM memory and RedHat Enterprise Linux 5.

3.3 Parallel Algorithm

In order to reduce the waiting time for the GPI predictions, we also implemented the algorithm using the HTCondor high throughput distributed computing environment (Thain *et al.* 2005), which harnesses the idle CPU cycles in the grid of computers we have in our bioinformatics computing lab on campus (Figure 3.2). HTCondor's file transfer mechanisms distribute the workload to the available processors in the grid, which consists of 54 processors in 64-bit machines running CentOS Linux. The processor speeds range from 1 to 3.3 GHz HTCondor handles all the details of sending executable and data files to computing resources and retrieving the computation results. Also, HTCondor provides checkpointing: if the application is interrupted, checkpointing saves the computation's state so it can be resumed later (instead of starting from scratch).

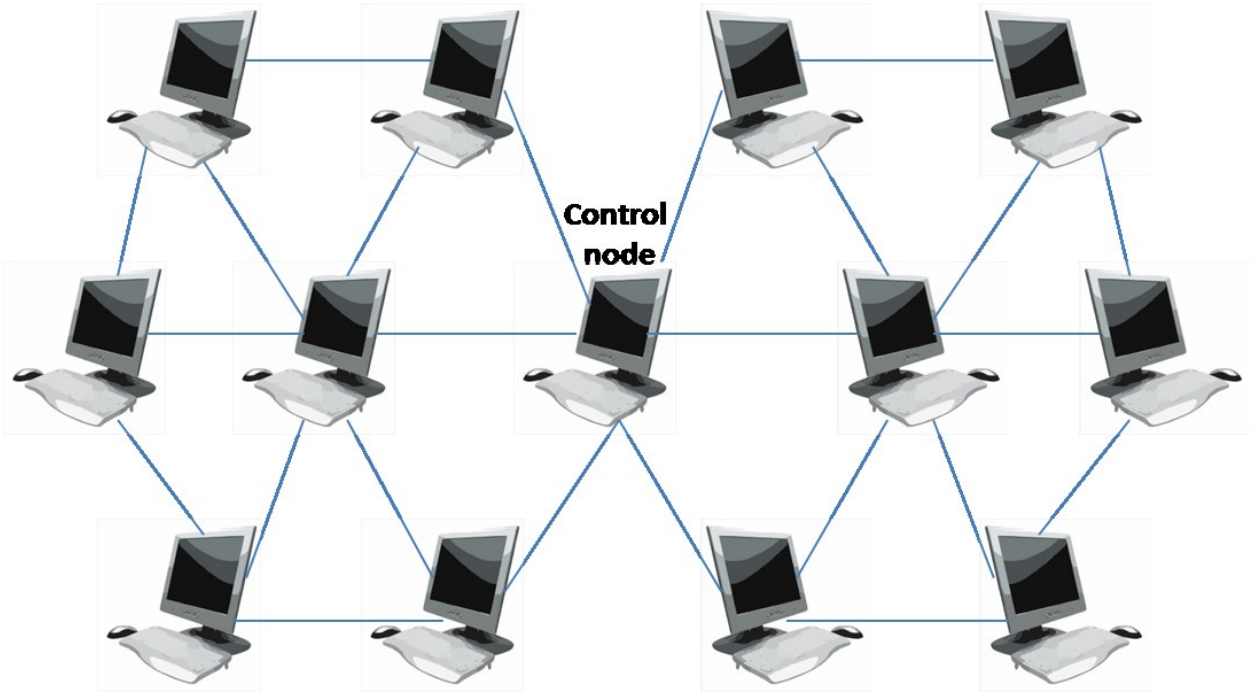


Figure 3.2 Bioinformatics grid computing environment

Individual peak lists, the data library and the serial algorithm are submitted to HTCondor. The primary argument to `condor_submit` is the name of a HTCondor submit description file. The

file contains commands and keywords that direct job queuing, setup, execution and wrapup. Below is a sample of our `condor_submit` file.

```
# GPI Project Condor Submit File
executable = /home/caguilar6/GPI2/gpi.pl
universe   = vanilla

output     = Output/out.$(Process)
#error      = Error/err.$(Process)
#log        = Log/log.$(Process)

#initialdir = Data

requirements = ( Arch=="X86_64") && ( OpSys=="LINUX" )
should_transfer_files = YES
when_to_transfer_output  = ON_EXIT
notification = never

arguments  = esn2008051325.1003.1003.2.dta
transfer_input_files  =
esn2008051325.1003.1003.2.dta,DB5.data,DB5_index.data,lipids5.data
queue

arguments  = esn2008040703.7230.7230.2.dta
transfer_input_files  =
esn2008040703.7230.7230.2.dta,DB5.data,DB5_index.data,lipids5.data
queue
```

3.4 Incorporation of expert opinion.

Although the only information provided by the spectra is molecular weight, specific monosaccharides and lipids can be assigned for portions of the structure by an expert. Based on expert opinion and our algorithm we developed an iterative process to assess the validity of our method. First, we took the top 20 structures according to our scoring function *S;* and also we took 20 structures with the lowest *S.* This set of structures denotes the two possible outcomes in the binary response, which can be seen as successfully predicted structures versus non-successfully predicted structures. We called *S'* this response and used the logistic regression model in R statistical package to perform the calculations. There are three predictor variables: The mean of the fitness order function ($\bar{d}$), the mean of the fitness value ($\bar{v}$) and the frequency of each structure:

$$S' = \bar{d} + \log(\bar{v}) + \log(Frequency)$$

Once that we obtain the coefficients of this model we acquire the suitable probabilities of all the predicted structures by using a logit link function as follows:

$$Y = \beta_0 + \bar{d} * \beta_1 + \log(\bar{v}) * \beta_2 + \log(frequency) * \beta_3$$

$$P = \frac{\exp(Y)}{1 + \exp(Y)}$$

With these results we take 40 top *Y* structures and 40 low *Y* structures and repeat the process until we cover the entire set of structures obtained by our algorithm.

3.5 User defined parameters

Our tool was developed using the data and confirmed GPI structures by Nakayasu *et al*. (2009). The results obtained are a good fit for these data, however, other organisms and experiments will have different characteristics, therefore, we have identified a set of user defined parameters that can increase the chance to obtain good results when different types of data are used. When no expert knowledge about the expected structures is available, we recommend using least restrictive parameter values. However, less restrictive criteria increases the chances of getting false positives.

Here is the part of the Perl code in which these parameters are set:

```
my $organism = 0; ## 0 For T. cruzi, 1 for other
my $max_rank = 10;
my $parent_ion_tolerance = 3;
my $fragment_tolerance = 0.5;
my $min_length_DTA = 130;

if ($organism == 0){
   $DB = "Tcruzi_DB.data"; #Structure library
   $DB_index = "Tcruzi_DB_index.data";## Index of GPI mass values
   $min_score = 50;
   $distinctive_ion = 25;
```

```
}
if ($organism == 1){
   $DB = "Other_DB.data";#Structure library
   $DB_index = "Other_DB_index.data";##Index of GPI mass values
   $min_score = 30;
}

my $lipid_DB = "lipids5.data";
```

organism. Because our collection of data is from the parasite *T. cruzi* we have a set of parameters for this parasite and a set for any other organism.

max_rank. This value corresponds to the fitness order function (*d*). A value of 1 indicates a higher ranked structure, whereas a value of 10 indicates a lower ranked structure. A less restrictive value will be any number greater than 10.

parent_ion_tolerance and fragment_tolerance. The choice of the mass tolerance parameters for parent ion and fragments depends on the mass accuracy and mass resolution of the *instrument* collecting the data. With low mass accuracy instruments, such as ion traps, one should specify a fairly large mass tolerance. Whit TOF-based mass spectrometers it is possible to achieve a mass accuracy of less than 0.1 Da, and even better mass accuracy of less than 0.05 Da with Fourier transform MS instruments.

min_length_DTA. This constraint is specified considering high-confidence peak assignments and observation from our experimental data set. High-confidence peak assignments are relatively large (typically among the highest 200 peaks) (Goldberg, et al., 2005). With our set of experimental data, we observed that the peak lists corresponding to the identified structures were always greater than 150 peaks. Then, to provide a tolerance we use those peak lists greater than or equal to 130 peaks.

DB. Our data library of theoretical structures is a work in progress. As more structural knowledge of GPI molecules is been obtained, the library can be enriched. For instance, it has been found that only *T. cruzi* and a few other parasites belonging to the class Kinetoplastida can add AEP to their GPIs (Ferguson *et al.*, 1982; Previato *et al.*, 1992; and Routier *et al.*, 1995). We

can use this information to perform a search for *T. cruzi* or other organisms, accelerating the search process for the second group.

min_score. This is the minimum value accepted for the fitness value (*v*) and is based on the assumption of our set of data. Values lower than 50 resulted in large number of false identifications. When analyzing data from species different than *T. cruzi* we recommend to be less restrictive.

distinctive_ion. As described before, a set of diagnostic ions have been identified in *T. cruzi*; when these ions are found, *v* is incremented. These ions can vary between organisms and in many species these ions are unknown. We have set to zero the initial value of this parameter when organisms different than *T. cruzi* are analyzed. However with more recognized diagnostic ions, this value can be changed to reduce the chances for false identifications.

lipid_DB. The current library of lipids contains 660 different structures. If the library is updated with more structures, this parameter should be modified with the name of the new library.

3.6 Running the computational tool

We have parallelized the algorithm in three different systems, HTCondor, UTEP's High Performance Computing Virtual Research Lab (HPCVCRL) or HPC cloud, and the Lonestar system ath the Texas Advanced Computing Center (TACC).

The Research Cloud @ UTEP provides UTEP researchers with access to computational resources which allow them to reserve a customized remote computer with a desired operating system and set of applications; and remotely access it over the Internet and remotely submit a job to a specified number of computers for execution. HPC @ UTEP is available 24 hours a day, 7 days a week. The hardware of this system in general is composed of 3 IBM BladeCenters with a total of 36 blades (432 processor cores) and a 20.4TB RAID storage system connected via a 10Gb network. For more information visit:

http://research.utep.edu/Default.aspx?alias=research.utep.edu/rc

The Texas Advanced Computing Center (TACC) serves thousands of researchers each year and provides comprehensive advanced computing resources in the following areas: High performance computing (HPC) systems of a variety of architectures; advanced scientific visualization resources; data storage/archival; networking; and software and tools to assist scientists and technical practitioners in using advanced computing, data hardware, and remote visualization resources. For this project I use TACC's Lonestar system, which contains 22,656 cores within 1,888 Dell PowerEdgeM610 compute blades (nodes), 16 PowerEdge R610 compute-I/Oserver-nodes, and 2 PowerEdge M610 (3.3GHz) login nodes. Each compute node has 24GB of memory, and the login/development nodes have 16GB. For more information visit: http://www.tacc.utexas.edu/

The following is a list of instructions to run the computational tool on the different systems. Some of these instructions were adapted from http://quantum.utep.edu/?q=node/16

3.6.1 Converting RAW data to peak lists with Sorcerer

1. In a web browser connect to Sorcerer at http://129.108.48.2
2. Select input files. Select RAW files to be analized and check box "Search each checked file or directory separately"
3. Choose a search profile", choose profile name "(3974) GPIomics DTA Generator"
4. Submit search
5. Sorcerer will switch to Queue screen
6. On the directory "/home/sorcerer/output" find the directories corresponding to the search, usually the most recent.
7. Underneath those directories there must be a subdirectory named "original" and underneath a subdirectory with the name of a RAW data file such as "EAA20120717-02". Inside that directory are the DTA files.

3.6.2 Running GPIomics on HTCondor

1. Connect with ssh at the main node of HTCondor (biolinux20.bioinformatics.utep.edu)

2. Create a directory on HTCondor underneath /export/home/caguilar/GPI2/DTA/ where the new DTAs will be placed.
3. Transfer all the peak list files for every directory in Sorecer to the recently created directory in HTCondor.
4. On /export/home/caguilar/GPI2 run script “submit-condor.pl” using as arguments the organism identifier (0 for T cruzi, 1 for other) and name of the DTA subdirectory to filter peak list candidates.

   ```
   perl submit-mpi.pl 0 DTA/DTA_subdirectory
   ```

   This script will generate the file gpi.submit
5. Submit the job to the HTCondor pool:

   ```
   condor_submit gpi.submit
   ```

   To monitor the job, use `condor_q` and/or `condor_status`
   To kill the job use `condor_rm –all`
6. When the job has finished running, the location /export/home/caguilar6/GPI2/Output will contain all the output files with the predicted molecules
7. On /export/home/caguilar/GPI2 run script “final-report-condor.pl” using as arguments the organism identifier (0 for T cruzi, 1 for other) and name of the file (with csv extension) which will have the results.

   ```
   perl final-report-condor.pl 0 out.csv
   ```

   The results will be placed underneath /export/home/caguilar/GPI2/Results

3.6.3 Running GPIomics on HPC

1. Browse to http://hpcvcl00.utep.edu/vcl (the browser may give you a warning that this page poses a security risk) this will get you into the Virtual Computing Lab; here you need to provide your login name and password.
2. Once you login, click on the "New Reservation" tab.

3. Make sure you select the cenots6mpi environment from the drop down list, as this environment is the only one that provides a Linux image with the MPI environment.
4. Select the hours you need to work with this environment (8hr maximum limit).
5. When the reservation is ready, the page will give you the ip address of your assigned node, your username and password on the node (the password is different that the VCL password and is valid only for this reservation.
6. You can now login to this node with the provided information (in Windows you can use putty or ssh).
7. Compile your job with mpicc:

   ```
   mpicc –o gpi-hpc gpi-hpc.c
   ```

8. Once the code compiles you need to login to the HPC server at 10.91.31.5 (or just ssh to hpc.utep.edu) to submit your job, with the username and password from the VCL login, (the main password for the site, not the reservation password)
9. Here is a sample job.mpi script:

   #BSUB -n 72

   #BSUB -W 24:00

   #BSUB -q high_priority

   #BSUB -e error.$J

   #BSUB -o output.$J

   #BSUB -J test1

   /shared/mpi/bin/mpirun -np 72 ./gpi-hpc > log.txt

   Where the options are:

   -n number of processors on which to execute (must be a multiple of 12).

   -W maximum execution time (in hours).

   -q queue to submit job.

   -e file to print error information (if you job executes correctly, this file should be empty).

   -o file to print output information (prints mostly cluster usage).

   -J job name (small text to identify calculation).

   The last line is the MPI execution of the gpi binary file called gpi-hpc on 72 processors

(this number must be the same as the one in the -n option), all the STDIN output will be redirected to a file called. log.txt.

The queues you can use are:

normal_priority up to 32 cores.

medium_priority up to 48 cores.

high_priority up to 72 cores.

10. Submit your job with: bsub < job.mpi

    You can monitor your jobs with: bjobs

    You can kill jobs with: bkill #jobnumber

    You can also submit jobs through the HPC server website, goto http://10.91.31.5 (or http://hpc.utep.edu) and login, you can create and submit jobs from there.

11. When the job has finished running, the location /home/caguilar/GPI will contain all the output files with the predicted molecules

12. On /home/caguilar/GPI run script "final-report-hpc.pl" using as arguments the organism identifier (0 for T cruzi, 1 for other) and name of the file (with csv extension) which will have the results.

    ```
    perl final-report-hpc.pl 0 out.csv
    ```

    The results will be placed underneath /home/caguilar/GPI/Results

3.6.4 Running GPIomics on TACC

1. Login to Lonestar with ssh (lonestar.tacc.utexas.edu).
2. Compile your job with mpicc:

   ```
   mpicc –o gpi-tacc gpi-tacc.c
   ```

3. Once the code compiles submit your job, with qsub

   ```
   qsub job.mpi
   ```

4. Here is a sample job.mpi script:

   #!/bin/bash

```
#$ -V                          # Inherit the submission environment
#$ -cwd                        # Start job in  submission directory
#$ -N GPI                      # Job Name
#$ -j y                        # combine stderr & stdout into stdout
#$ -o $JOB_NAME.o$JOB_ID       # Name of the output file (eg. GPI.oJobID)
#$ -pe 12way 72                # Requests 12 cores/node, 72 cores total
(multiple of 12)
#$ -q normal                   # Queue name
#$ -l h_rt=4:00:00             # Run time (hh:mm:ss) - 4 hours
ibrun ./gpi-tacc               # Run the MPI executable named "gpi-tacc"
```

The last line is the MPI execution of the gpi binary file called gpi-tacc, in this example it will run on 72 processors (-pe option, always use multiples of 12), all the STDIN output will be redirected to a file called GPI.oJobID.

5. You can monitor your jobs with `qstat`
6. When the job has finished running, the location /home1/01424/caguila6/GPI/Output will contain all the output files with the predicted molecules
7. On /home1/01424/caguila6/GPI run script “final-report-tacc.pl” using as arguments the organism identifier (0 for T cruzi, 1 for other) and name of the file (with csv extension) which will have the results.

   ```
   perl final-report-tacc.pl 0 out.csv
   ```

   The results will be placed underneath /home1/01424/caguila6/GPI /Results

Chapter 4: Results and Discussion

LC-MS/MS is the most efficient tool today for GPI profiling. The amount of data produced in each MS experiment is a major bottleneck in high-throughput GPIomic projects. Efficient computational tools can significantly reduce the amount of time in the analysis of MS data; however, at present the automatic interpretation of these data to annotate GPI structures is absent.

We have developed a tool to automate the analysis and interpretation of the GPIomic MS data. We have addressed a set of problems stated in chapter 1; we will discuss each problem and show the results obtained.

4.1 Problem 1. Building a computational model for large scale GPIomic experiments

The field of proteomics has produced successful algorithms that calculate a "matching score" and report how closely a given peptide sequence matches the masses identified in the spectra. Examples of these scores were discussed in Chapter 2, but in essence they attempt to solve the Spectrum Matching Problem (Fenyo and Beavis 2003), which can be stated as follows: Given a spectrum *S* and a score threshold *T* for a spectrum-peptide scoring function, find the probability that a random peptide matches the spectrum *S* with score equal to or larger than *T*.

Our method for annotating GPI structures in essence is a Spectrum Matching Problem. However instead of peptides we have monosaccharide fragments followed by a lipid block. The algorithm finds the set of all candidate structures in a structure library that have a match with the observed parent ion mass. Then it scores and counts the number of times each candidate structure is observed. The unique structures are sorted according to our fitness function. This function allows us to score every identified structure and provide us with a list of the most likely unique structures considering every peak list.

When compared to genomics and proteomics, glycan databases are very poor, in particular, experimentally annotated GPIs account for only a few hundreds. Annotation of the spectra and assignment of GPI structures to the mass peaks is typically done manually by an expert.

However, using a data library of GPI structures increases the efficiency of annotation. We developed our own data library of theoretical GPIs for *T. cruzi*. To date this is the first library of GPIs.

The overall effectiveness of our algorithm was tested with a set of 78 GPI structures that had been previously manually annotated (Nakayasu et al., 2009). Using the set of spectra from the same experiments we obtained a total of 181051 peak list files, from which 40911 files were filtered as potential candidates for GPI structures. We were able to correctly match MS/MS spectra with their proposed GPI structure using our testing set of structures.

We predicted 5,039,589 structures. Selecting the top 10 ranks ordered according to our fitness order function (*d*) we obtained a total of 4686 unique GPIs. Over 70% of the testing structures are among the top 647. Figure 2 shows the number of top ranking predicted GPIs required to contain the given percentages of structures in the testing set. Table 4.1 shows the results for the annotated structures by Nakayasu et al. (2009). The results table contains the following columns: Species (just to correlate with previously annotated structures), mass (Da), Structure, Average *d*, Average *v*, Frequency, peak list, S, Y, P, Rank.

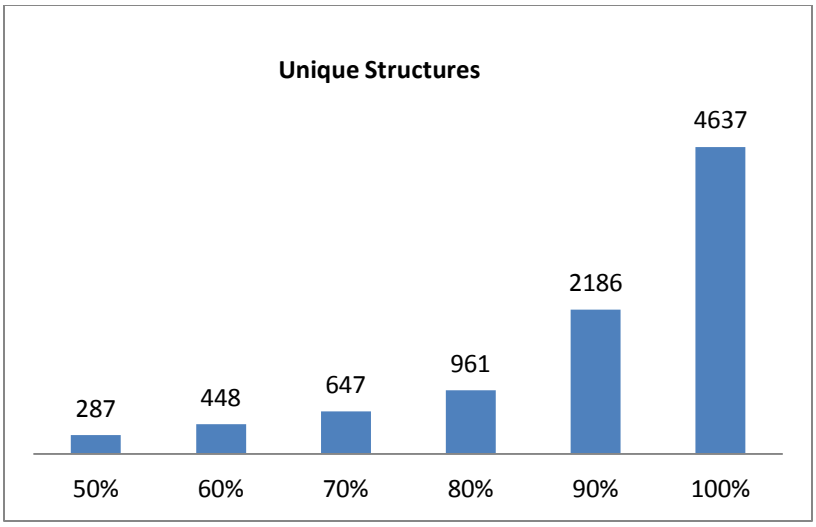


Figure 4.1 Number of predicted GPIs required containing the given percentages of testing structures.

Table 4.1 Results obtained for the testing structures

Species	Mass	Structure	*Avg d*	*Avg v*	Freq	peak list	*S*	Y	P	Rank
1	1999.908	EtNP-Hex5-AEP-HexN-Ins-P-34:0t	3.00	57.11	88	esn2008051318.3394.3394.2.dta	3052.13	20.57	1.00	51
2	2038.952	Hex6-AEP-HexN-Ins-P-34:0t	4.46	56.55	72	esn2008051318.6266.6266.2.dta	1863.30	17.12	1.00	73
3	1981.896	EtNP-Hex5-AEP-HexN-Ins-P-34:1d	2.58	58.59	194	esn2008051318.4611.4611.2.dta	7770.44	28.25	1.00	9
5	2020.94	Hex6-AEP-HexN-Ins-P-34:1d	3.10	57.22	185	esn2008051318.6654.6654.2.dta	6609.76	26.97	1.00	15
6	1965.901	AEP-Hex5-AEP-HexN-Ins-P-34:1d	1.93	58.08	134	esn2008051318.3436.3436.2.dta	4685.69	25.61	1.00	19
7	2182.993	Hex7-AEP-HexN-Ins-P-34:1d	4.15	60.14	13	esn2008051318.3459.3459.2.dta	389.44	3.28	0.96	293
8	2127.954	AEP-Hex6-AEP-HexN-Ins-P-34:1d	2.60	58.75	5	esn2008051324.5168.5168.2.dta	109.93	-3.68	0.02	670
9	1696.835	Hex4-AEP-HexN-Ins-P-34:1d	1.80	54.08	5	esn2008051318.3580.3580.2.dta	176.32	-3.85	0.02	681
11	1983.912	EtNP-Hex5-AEP-HexN-Ins-P-34:0d	3.34	57.44	206	esn2008051318.4611.4611.2.dta	6969.21	27.70	1.00	11
12	2185.009	Hex7-AEP-HexN-Ins-P-34:0d	5.30	60.10	10	esn2008051318.3459.3459.2.dta	248.78	-0.27	0.43	458
13	2254.102	AEP-Hex6-AEP-HexN-Ins-P-43:1d	3.33	56.69	3	esn2008051318.5192.5192.2.dta	70.96	-9.38	0.00	1213
14	2145.928	EtNP-Hex6-AEP-HexN-Ins-P-33:1t	5.43	57.35	7	esn2008051318.3600.3600.2.dta	115.51	-4.11	0.02	698
15	2022.956	Hex6-AEP-HexN-Ins-P-34:0d	4.67	56.40	135	esn2008051318.5724.5724.2.dta	3575.77	22.34	1.00	35
16	1967.917	AEP-Hex5-AEP-HexN-Ins-P-34:0d	3.04	56.38	160	esn2008051324.3608.3608.2.dta	4210.33	25.58	1.00	20
17	1698.85	Hex4-AEP-HexN-Ins-P-34:0d	4.50	53.41	6	esn2008040302.6317.6317.2.dta	133.35	-5.35	0.00	801
18	2037.953	Hex6-AEP-HexN-Ins-P-AAG-32:0	3.03	56.92	75	esn2008051318.6506.6506.2.dta	2519.20	19.11	1.00	58
19	1998.909	EtNP-Hex5-AEP-HexN-Ins-P-AAG-32:0	2.62	57.18	71	esn2008051318.4136.4136.2.dta	2594.10	19.13	1.00	57
20	2362.059	Hex8-AEP-HexN-Ins-P-AAG-32:0	3.75	54.16	28	esn2008051318.4082.4082.2.dta	729.65	9.09	1.00	156
21	1982.914	AEP-Hex5-AEP-HexN-Ins-P-AAG-32:0	5.93	55.91	82	esn2008051318.4172.4172.2.dta	1141.28	16.52	1.00	80
22	1836.856	EtNP-Hex4-AEP-HexN-Ins-P-AAG-32:0	1.22	55.50	714	esn2008040302.6452.6452.2.dta	31858.19	40.42	1.00	2
23	1713.848	Hex4-AEP-HexN-Ins-P-AAG-32:0	5.00	54.14	7	esn2008051318.4337.4337.2.dta	128.50	-4.37	0.01	726
24	1808.967	Hex4-AEP-HexN-Ins-P-42:1d	4.00	53.76	2	esn2008051318.5326.5326.2.dta	32.09	-14.31	0.00	1979
25	2063.969	Hex6-AEP-HexN-Ins-P-AAG-34:1	5.44	57.90	16	esn2008051318.4756.4756.2.dta	324.44	3.22	0.96	297
26	1737.848	Hex4-AEP-HexN-Ins-P-AAG-34:2	6.33	53.50	3	esn2008051318.4335.4335.2.dta	33.91	-13.36	0.00	1809
27	2065.985	Hex6-AEP-HexN-Ins-P-AAG-34:0	4.00	57.83	12	esn2008051318.4756.4756.2.dta	292.03	2.25	0.90	343
28	2105.039	Hex6-AEP-HexN-Ins-P-40:1d	8.00	56.58	1	esn2008051324.4409.4409.2.dta	0.13	-24.06	0.00	4041
29	2052.016	AEP-Hex5-AEP-HexN-Ins-P-40:0d	9.00	56.96	1	esn2008051324.4423.4423.2.dta	0.05	-25.05	0.00	4234
30	2068.011	EtNP-Hex5-AEP-HexN-Ins-P-40:0d	6.50	54.88	2	esn2008051318.4769.4769.2.dta	35.23	-16.76	0.00	2467
31	2080.012	AEP-Hex5-AEP-HexN-Ins-P-41:1t	3.40	57.36	100	esn2008051318.5628.5628.2.dta	2129.22	21.31	1.00	45
32	1739.863	Hex4-AEP-HexN-Ins-P-AAG-34:1	4.20	52.34	5	esn2008051318.4269.4269.2.dta	104.77	-6.86	0.00	943
33	1973.035	Hex5-AEP-HexN-Ins-P-42:0d	3.67	57.01	3	esn2008051318.5259.5259.2.dta	85.43	-9.67	0.00	1246
34	2131.057	Hex6-AEP-HexN-Ins-P-42:2d	5.41	55.83	22	esn2008051318.4778.4778.2.dta	483.64	5.57	1.00	219
36	2064.017	AEP-Hex5-AEP-HexN-Ins-P-41:1d	3.00	58.11	3	esn2008051324.4496.4496.2.dta	81.37	-8.71	0.00	1148
37	2096.045	AEP-Hex5-AEP-HexN-Ins-P-42:0t	5.20	55.19	60	esn2008051318.5357.5357.2.dta	991.61	14.41	1.00	102
39	2151.084	Hex6-AEP-HexN-Ins-P-42:0t	6.11	58.33	18	esn2008051318.5627.5627.2.dta	293.44	3.61	0.97	272
40	2121.072	Hex6-AEP-HexN-Ins-P-41:0d	3.00	59.13	2	esn2008051324.4489.4489.2.dta	44.41	-12.03	0.00	1578

41	2112.04	EtNP-Hex5-AEP-HexN-Ins-P-42:0t	4.33	56.95	9	esn2008051318.4834.4834.2.dta	236.29	-0.82	0.31	487
42	2076.018	AEP-Hex5-AEP-HexN-Ins-P-42:2d	5.38	56.03	34	esn2008051318.5830.5830.2.dta	384.36	9.45	1.00	152
43	2078.033	AEP-Hex5-AEP-HexN-Ins-P-42:1d	2.45	58.39	100	esn2008051318.6004.6004.2.dta	3222.55	22.56	1.00	32
44	2457.178	Hex8-AEP-HexN-Ins-P-42:1d	3.00	56.04	14	esn2008051324.4634.4634.2.dta	490.82	4.28	0.99	245
45	2110.024	EtNP-Hex5-AEP-HexN-Ins-P-42:1t	3.10	55.08	10	esn2008051318.4856.4856.2.dta	320.15	1.02	0.74	405
46	2133.072	Hex6-AEP-HexN-Ins-P-42:1d	3.33	56.83	135	esn2008051318.7289.7289.2.dta	4689.90	23.89	1.00	29
47	2066.032	AEP-Hex5-AEP-HexN-Ins-P-41:0d	4.86	53.77	7	esn2008051318.4863.4863.2.dta	78.54	-4.30	0.01	716
48	2295.125	Hex7-AEP-HexN-Ins-P-42:1d	3.08	58.70	12	esn2008051324.4746.4746.2.dta	447.72	3.43	0.97	284
49	2240.086	AEP-Hex6-AEP-HexN-Ins-P-42:1d	1.29	59.33	7	esn2008051318.4976.4976.2.dta	268.02	0.81	0.69	413
50	2135.088	Hex6-AEP-HexN-Ins-P-42:0d	5.18	55.86	89	esn2008051318.6586.6586.2.dta	1968.68	18.03	1.00	68
51	2297.141	Hex7-AEP-HexN-Ins-P-42:0d	4.50	59.81	6	esn2008051318.4990.4990.2.dta	141.54	-3.92	0.02	691
53	2080.049	AEP-Hex5-AEP-HexN-Ins-P-42:0d	3.30	57.46	100	esn2008051318.6004.6004.2.dta	2406.16	21.44	1.00	44
54	2459.194	Hex8-AEP-HexN-Ins-P-42:0d	5.50	57.73	6	esn2008051324.4634.4634.2.dta	120.55	-5.45	0.00	814
55	2242.102	AEP-Hex6-AEP-HexN-Ins-P-42:0d	3.43	57.18	7	esn2008051318.4976.4976.2.dta	150.90	-1.98	0.12	573
56	2096.044	EtNP-Hex5-AEP-HexN-Ins-P-42:0d	3.56	56.56	79	esn2008051318.6560.6560.2.dta	2547.33	18.90	1.00	62
57	1971.02	Hex5-AEP-HexN-Ins-P-42:1d	4.33	52.75	6	esn2008051318.5145.5145.2.dta	112.94	-5.32	0.00	799
58	2108.045	EtNP-Hex5-AEP-HexN-Ins-P-43:1d	2.44	55.85	16	esn2008051324.4993.4993.2.dta	569.20	6.02	1.00	212
59	1810.946	Hex4-AEP-HexN-Ins-P-41:1t	7.00	52.81	1	esn2008051318.7317.7317.2.dta	0.20	-23.84	0.00	3995
60	2147.089	Hex6-AEP-HexN-Ins-P-43:1d	4.25	57.51	8	esn2008051318.5183.5183.2.dta	171.77	-1.63	0.16	546
61	2107.055	Hex6-AEP-HexN-Ins-P-40:0d	7.00	61.16	2	esn2008051318.4624.4624.2.dta	11.15	-15.94	0.00	2299
62	2489.206	Hex8-AEP-HexN-Ins-P-43:0t	5.00	50.87	1	esn2008040703.1997.1997.2.dta	0.54	-22.14	0.00	3680
63	2149.105	Hex6-AEP-HexN-Ins-P-43:0d	3.20	58.45	5	esn2008051318.5318.5318.2.dta	149.46	-4.39	0.01	731
64	2311.121	Hex7-AEP-HexN-Ins-P-42:1t	4.00	61.71	2	esn2008051318.5246.5246.2.dta	22.00	-12.58	0.00	1671
65	2065.995	EtNP-Hex5-AEP-HexN-Ins-P-40:1d	7.00	53.01	2	esn2008051318.6984.6984.2.dta	8.36	-17.74	0.00	2740
66	2092.05	AEP-Hex5-AEP-HexN-Ins-P-43:1d	3.05	56.68	22	esn2008051324.5007.5007.2.dta	610.43	8.32	1.00	169
67	2110.06	EtNP-Hex5-AEP-HexN-Ins-P-43:0d	4.00	54.98	13	esn2008051318.4856.4856.2.dta	384.10	2.31	0.91	341
68	2094.065	AEP-Hex5-AEP-HexN-Ins-P-43:0d	4.53	54.79	32	esn2008051324.4617.4617.2.dta	619.99	9.56	1.00	151
69	2161.105	Hex6-AEP-HexN-Ins-P-44:1d	3.86	55.73	14	esn2008051324.5189.5189.2.dta	394.28	3.29	0.96	292
70	2106.066	AEP-Hex5-AEP-HexN-Ins-P-44:1d	2.53	56.65	30	esn2008051318.5380.5380.2.dta	902.70	11.58	1.00	125
71	2122.061	EtNP-Hex5-AEP-HexN-Ins-P-44:1d	2.40	56.49	20	esn2008051318.5415.5415.2.dta	755.10	8.15	1.00	175
72	2163.121	Hex6-AEP-HexN-Ins-P-44:0d	5.67	58.03	9	esn2008051324.5171.5171.2.dta	200.95	-2.02	0.12	575
73	2124.077	EtNP-Hex5-AEP-HexN-Ins-P-44:0d	3.92	57.32	13	esn2008051324.5213.5213.2.dta	369.39	2.92	0.95	310
74	2108.082	AEP-Hex5-AEP-HexN-Ins-P-44:0d	3.56	55.41	27	esn2008051318.5458.5458.2.dta	517.16	9.27	1.00	153
75	2150.079	Hex6-AEP-HexN-Ins-P-AAG-40:0	6.22	58.97	9	esn2008051318.5627.5627.2.dta	180.06	-2.42	0.08	596
76	2256.081	EtNP-Hex6-AEP-HexN-Ins-P-42:1d	1.00	56.63	1	esn2008040730.5855.5855.2.dta	21.74	-16.46	0.00	2387
77	2149.068	Hex6-AEP-HexN-Ins-P-42:1t	5.33	55.12	6	esn2008051318.5318.5318.2.dta	139.43	-5.85	0.00	847
78	2092.012	EtNP-Hex5-AEP-HexN-Ins-P-42:2d	3.71	55.03	34	esn2008051324.4617.4617.2.dta	1004.12	11.04	1.00	129
35,38	2094.028	EtNP-Hex5-AEP-HexN-Ins-P-42:1d	2.57	57.49	93	esn2008051318.5135.5135.2.dta	3618.34	21.60	1.00	43
4,10	2345.046	Hex8-AEP-HexN-Ins-P-34:1d	3.54	58.06	13	esn2008051318.3483.3483.2.dta	411.40	3.50	0.97	278

For every peak list there is a number of structures related to it. In an attempt to eliminate false positives we took the top 1, 2, …, 5 and all unique structures for every peak list according to our function *S* (Table 4.2).

Table 4.2 Unique structures

	Correctly interpreted	Missiniterpreted	Missinterpreted lipid	Non-identified	Numer of structures
DTA1	45	7	24	2	1465
DTA2	57	4	15	2	2592
DTA3	69	1	6	2	3184
DTA4	70	0	6	2	3638
DTA5	71	0	5	2	3945
All	76	0	0	2	4686

4.2 Problem 2. Increased efficiency by grid computing

With the serial algorithm the waiting time to obtain the GPI predictions was 10 days. In order to benchmark the waiting time we performed a proteomics analysis using the same testing data with Proteome Discoverer (PD) Software from Thermo Scientific. SEQUEST was used for our timing comparison. The database of protein sequences was constructed downloading from NCBI the known proteins of all the trypanosomids, *Leishmania spp*, human keratin and trypsin. In total we obtained a set of 230,363 non-redundant protein sequences. PD took 3 days, 19 hours, 38 minutes to complete the search. The computer used for this task has an Intel Xeon 2.00 GHz processor with 2 cores x86_64, 48 GB of RAM and running Windows 7. In comparison, the computer running our GPIomics algorithm has 2 Intel Xeon 2.83 GHz processors with 4 cores x86_64, 24 GB of RAM and running CentOS Linux. In general, the computing power to obtain the GPI predictions was twice as much as the computing power to obtain the protein predictions, however, the number of protein sequences in the protein database was just a fraction (12%) compared to the number of GPI structures within the GPI library (1.9 million).

Although the current HTCondor pool of machines consists of 54 processors, an average of 16 processors was claimed during the predictions because only those with idle CPU cycles can be

used. With an average of 16 processors, the waiting time to obtain the predictions was reduced to 3 days, 10 hours, 21 minutes.

To compare the time spent in completing the experiment between HTCondor, HPC and TACC, we used a set of 1000 random peak lists. With HTCondor we observed that the completion time decreases, roughly inversely proportional to the number of processors when this number is up to 16 (Figure 4.2). However, no further substantial time reduction is achieved when the number of processors increases beyond 16. We are currently investigating the cause for such limitation.

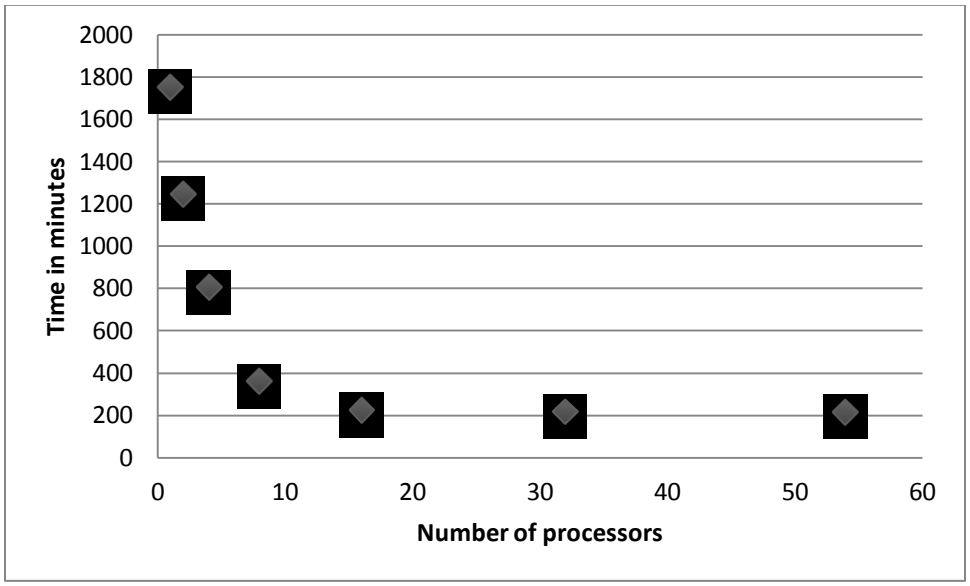


Figure 4.2 Number of processors and completion time on HTCondor

HPC and TACC allow the use of processors in multiples of 12; for our experiments we started with 12 processors and increased the number up to 72. The results are shown in Figure 4.3. Compared to HTCondor, there is an improvement on performing the experiment with MPI. The time spent to analyze 1000 random peak lists with 64 processors in HTCondor was 3 hours and 26 minutes, whereas the time spent with 60 processors in HPC and TACC were 48 and 43 minutes respectively. This is expected given the limitation mentioned above. Moreover, HTCondor assumes a variable availability of resources, while the processing availability of MPI is constant. TACC revealed to be slightly faster than HPC.

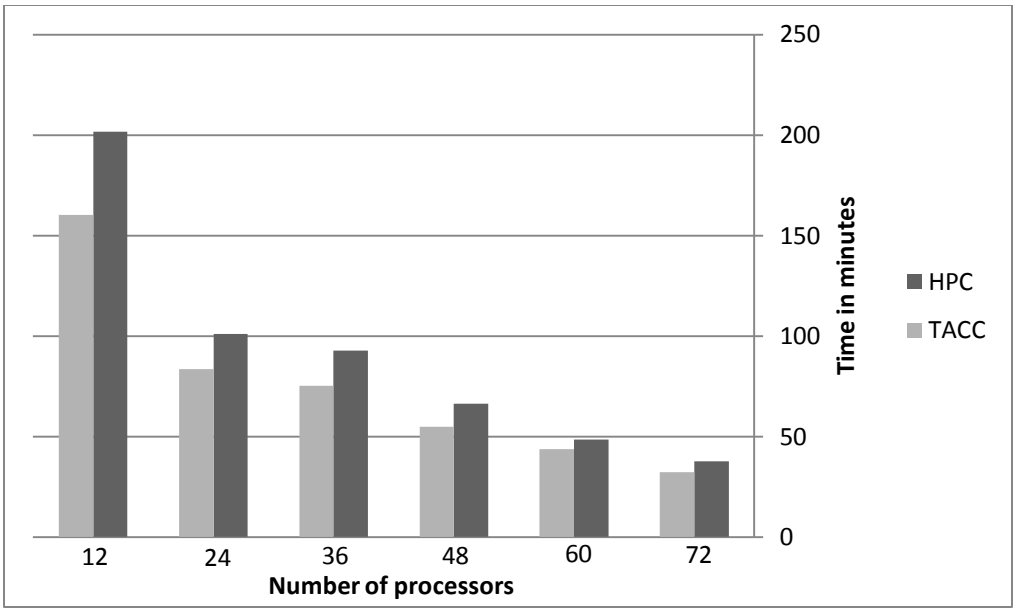


Figure 4.3 Number of processors and computing time on HPC and TACC

HTCondor has a couple of commands that allow viewing the jobs submitted (condor_q) and to display the status of jobs (condor_statuts), below is an example of both commands.

> condor_q

```
-- Submitter: rnavlab.utep.edu : <129.108.156.117:9613> : rnavlab.utep.edu
 ID      OWNER            SUBMITTED     RUN_TIME ST PRI SIZE CMD
3100.13968 caguilar6     12/5  18:09   9+14:27:32 R  0   732.4 gpi.pl esn20080403
3100.13979 caguilar6     12/5  18:09   9+14:26:17 R  0   732.4 gpi.pl esn20080513
3140.319 caguilar6     12/16 17:05   0+00:50:29 R  0   732.4 gpi.pl esn20080513
3140.381 caguilar6     12/16 17:05   0+00:40:17 R  0   732.4 gpi.pl esn20080513
3140.603 caguilar6     12/16 17:05   0+00:04:52 R  0   0.0  gpi.pl esn20080403
3140.611 caguilar6     12/16 17:05   0+00:03:08 R  0   0.0  gpi.pl esn20080403
3140.612 caguilar6     12/16 17:05   0+00:03:00 R  0   0.0  gpi.pl esn20080513
3140.615 caguilar6     12/16 17:05   0+00:02:46 R  0   0.0  gpi.pl esn20080407
3140.619 caguilar6     12/16 17:05   0+00:01:48 R  0   0.0  gpi.pl esn20080403
3140.620 caguilar6     12/16 17:05   0+00:01:47 R  0   0.0  gpi.pl esn20080403
3140.622 caguilar6     12/16 17:05   0+00:01:42 R  0   0.0  gpi.pl esn20080513
3140.623 caguilar6     12/16 17:05   0+00:01:25 R  0   0.0  gpi.pl esn20080403
3140.624 caguilar6     12/16 17:05   0+00:01:17 R  0   0.0  gpi.pl esn20080513
3140.626 caguilar6     12/16 17:05   0+00:01:13 R  0   0.0  gpi.pl esn20080403
3140.628 caguilar6     12/16 17:05   0+00:01:01 R  0   0.0  gpi.pl esn20080403
3140.629 caguilar6     12/16 17:05   0+00:00:34 R  0   0.0  gpi.pl esn20080403
3140.630 caguilar6     12/16 17:05   0+00:00:29 R  0   0.0  gpi.pl esn20080407
3140.631 caguilar6     12/16 17:05   0+00:00:26 R  0   0.0  gpi.pl esn20080407
```

```
3140.632 caguilar6     12/16 17:05   0+00:00:06 R  0   0.0  gpi.pl esn20080407
3140.633 caguilar6     12/16 17:05   0+00:00:01 I  0   0.0  gpi.pl esn20080407
3140.634 caguilar6     12/16 17:05   0+00:00:00 I  0   0.0  gpi.pl esn20080407
3140.635 caguilar6     12/16 17:05   0+00:00:00 I  0   0.0  gpi.pl esn20080403
…
3140.40906 caguilar6     12/16 17:05   0+00:00:00 I  0   0.0  gpi.pl esn20080513
3140.40907 caguilar6     12/16 17:05   0+00:00:00 I  0   0.0  gpi.pl esn20080403
3140.40908 caguilar6     12/16 17:05   0+00:00:00 I  0   0.0  gpi.pl esn20080403
3140.40909 caguilar6     12/16 17:05   0+00:00:00 I  0   0.0  gpi.pl esn20080407
3140.40910 caguilar6     12/16 17:05   0+00:00:00 I  0   0.0  gpi.pl esn20071027
3140.40911 caguilar6     12/16 17:05   0+00:00:00 I  0   0.0  gpi.pl esn20071027

40297 jobs; 40278 idle, 19 running, 0 held
> condor_status

Name            OpSys      Arch   State     Activity LoadAv Mem   ActvtyTime

slot1@biolinux01.b LINUX      X86_64 Claimed   Busy     0.500  2861  0+00:00:41
slot2@biolinux01.b LINUX      X86_64 Unclaimed Idle     0.350  2861  0+00:01:50
slot1@biolinux02.b LINUX      X86_64 Claimed   Busy     1.380  2861  0+00:00:04
slot2@biolinux02.b LINUX      X86_64 Claimed   Busy     0.790  2861  0+00:00:19
slot1@biolinux03.b LINUX      X86_64 Claimed   Busy     1.190  2861  0+00:00:03
slot2@biolinux03.b LINUX      X86_64 Claimed   Busy     2.840  2861  0+00:07:17
slot1@biolinux05.b LINUX      X86_64 Unclaimed Idle     0.000  3871  0+00:01:46
slot2@biolinux05.b LINUX      X86_64 Unclaimed Idle     0.000  3871  0+00:01:48
slot1@biolinux06.b LINUX      X86_64 Unclaimed Idle     0.000   938  0+00:00:03
slot2@biolinux06.b LINUX      X86_64 Unclaimed Idle     0.410   938  0+00:00:03
slot1@biolinux07.b LINUX      X86_64 Claimed   Busy     0.880  3871  0+00:00:04
slot2@biolinux07.b LINUX      X86_64 Unclaimed Idle     0.000  3871  0+00:00:02
slot1@biolinux08.b LINUX      X86_64 Claimed   Busy     0.910  2857  0+00:00:04
slot2@biolinux08.b LINUX      X86_64 Unclaimed Idle     0.210  2857  0+00:00:02
slot1@biolinux09.b LINUX      X86_64 Unclaimed Idle     0.000   424  0+00:08:21
slot2@biolinux09.b LINUX      X86_64 Claimed   Busy     0.820   424  0+00:00:05
slot3@biolinux09.b LINUX      X86_64 Unclaimed Idle     0.000   424  0+00:08:31
slot4@biolinux09.b LINUX      X86_64 Unclaimed Idle     0.060   424  0+01:54:23
slot1@biolinux10.b LINUX      X86_64 Unclaimed Idle     0.000   424  0+00:00:36
slot2@biolinux10.b LINUX      X86_64 Claimed   Busy     0.830   424  0+00:00:04
slot3@biolinux10.b LINUX      X86_64 Unclaimed Idle     0.000   424  0+00:00:39
slot4@biolinux10.b LINUX      X86_64 Unclaimed Idle     0.000   424  0+00:00:40
slot1@biolinux11.b LINUX      X86_64 Claimed   Busy     0.000   424  0+00:00:04
slot2@biolinux11.b LINUX      X86_64 Claimed   Busy     0.920   424  0+00:00:04
slot3@biolinux11.b LINUX      X86_64 Unclaimed Idle     0.000   424  0+00:00:04
slot4@biolinux11.b LINUX      X86_64 Unclaimed Idle     0.100   424  0+00:09:04
slot1@biolinux13.b LINUX      X86_64 Claimed   Busy     0.920   424  0+00:00:04
slot2@biolinux13.b LINUX      X86_64 Unclaimed Idle     0.120   424  0+00:00:03
slot3@biolinux13.b LINUX      X86_64 Unclaimed Idle     0.330   424  0+00:00:06
slot4@biolinux13.b LINUX      X86_64 Unclaimed Idle     0.000   424  0+00:08:16
slot1@biolinux14.b LINUX      X86_64 Unclaimed Idle     0.000   424  0+00:07:42
```

```
slot2@biolinux14.b LINUX     X86_64 Unclaimed Idle    0.000   424  0+00:11:24
slot3@biolinux14.b LINUX     X86_64 Unclaimed Idle    0.000   424  0+00:11:24
slot4@biolinux14.b LINUX     X86_64 Unclaimed Idle    0.000   424  0+00:11:32
slot1@biolinux15.b LINUX     X86_64 Unclaimed Idle    0.000   424  0+00:11:19
slot2@biolinux15.b LINUX     X86_64 Unclaimed Idle    0.010   424  0+00:11:37
slot3@biolinux15.b LINUX     X86_64 Unclaimed Idle    0.000   424  0+00:11:19
slot4@biolinux15.b LINUX     X86_64 Unclaimed Idle    0.000   424  0+01:55:48
slot1@biolinux16.b LINUX     X86_64 Unclaimed Idle    0.000   424  0+00:07:49
slot2@biolinux16.b LINUX     X86_64 Unclaimed Idle    0.000   424  0+01:40:58
slot3@biolinux16.b LINUX     X86_64 Unclaimed Idle    0.000   424  0+01:53:55
slot4@biolinux16.b LINUX     X86_64 Unclaimed Idle    0.000   424  0+01:53:10
slot1@biolinux17.b LINUX     X86_64 Unclaimed Idle    0.890   424  0+00:00:16
slot2@biolinux17.b LINUX     X86_64 Claimed   Busy    0.000   424  0+00:00:04
slot3@biolinux17.b LINUX     X86_64 Claimed   Busy    0.420   424  0+00:00:06
slot4@biolinux17.b LINUX     X86_64 Unclaimed Idle    1.000   424  0+00:03:37
slot1@biolinux18.b LINUX     X86_64 Claimed   Busy    0.260   424  0+00:00:02
slot2@biolinux18.b LINUX     X86_64 Unclaimed Idle    0.000   424  0+00:07:54
slot3@biolinux18.b LINUX     X86_64 Unclaimed Idle    0.190   424  0+00:08:12
slot4@biolinux18.b LINUX     X86_64 Unclaimed Idle    0.000   424  0+00:07:42
slot1@biolinux19.b LINUX     X86_64 Unclaimed Idle    0.000   424  0+00:08:33
slot2@biolinux19.b LINUX     X86_64 Unclaimed Idle    0.000   424  0+00:08:41
slot3@biolinux19.b LINUX     X86_64 Unclaimed Idle    0.000   424  0+00:08:41
slot4@biolinux19.b LINUX     X86_64 Unclaimed Idle    0.000   424  0+00:08:36
slot1@biolinux20.b LINUX     X86_64 Claimed   Busy    0.830   424  0+00:00:04
slot2@biolinux20.b LINUX     X86_64 Unclaimed Idle    0.000   424  0+00:00:02
slot3@biolinux20.b LINUX     X86_64 Unclaimed Idle    0.290   424  0+00:00:03
slot4@biolinux20.b LINUX     X86_64 Unclaimed Idle    0.000   424  0+00:09:19
rnavlab.utep.edu   LINUX     X86_64 Unclaimed Idle    1.310  8036  0+00:00:04
slot1@CRBL401-1318 WINNT51   INTEL  Unclaimed Idle    0.150  1759  0+00:00:04
slot2@CRBL401-1318 WINNT51   INTEL  Owner     Idle    1.000  1759  1+04:10:49
```

	Total	Owner	Claimed	Unclaimed	Matched	Preempting	Backfill
INTEL/WINNT51	2	1	0	1	0	0	0
X86_64/LINUX	59	0	16	43	0	0	0
Total	61	1	16	44	0	0	0

4.3 Problem 3. Incorporating expert opinion into the prediction

The possibility of including expert knowledge was an important part in the construction of the model since it allowed us to assess our results. We incorporated an iterative process of structure identification and validation of these structures by an expert. First, we obtained the

initial “guess” structures, where we used the top 20 structures according to our scoring function *S*, and 20 structures with the lowest *S*. Next, we applied LR and obtained the coefficients of this model, with the suitable probabilities of all the predicted structures by using a logit link function. The expert then validated these results and given his feedback we choose the top 40 correct structures and 40 structures with low *S*. We proceeded in the same way until 100 top correct structures were validated by the expert. These iterative algorithm–expert cycles offered us the results shown in Figure 4.4. In general, both LR and our scoring function *S* converged in similar results. However, LR provided us a value for the predictions in the form of probabilities, hence, giving a probabilistic basis to judge whether a GPI structure was likely or not.

Below is a sample of an R session used for the application of LR.

```
> binomial <- read.csv("C:/Users/Clem/Desktop/GPI/Logistic/iterative/2012-07-15-binomial-100.csv")
> attach(binomial)
> f2<- glm(Binomial~Avg_d+log(Avg_v)+(log(log(Frequency)+.1)), family=binomial(link="logit"),
na.action=na.pass)
Warning message:
glm.fit: algorithm did not converge
> b.vector <- as.vector(f2$coefficients)
> newdata <- data.frame(Binomial,S,Mass,Structure,Avg_d,Avg_v,Frequency,DTA,fitted(f2))
> gpidata.full <- read.csv("C:/Users/Clem/Desktop/GPI/Logistic/iterative/2012-07-01-Estr.csv")
> attach(gpidata.full)
The following object(s) are masked from 'binomial':

   Avg_d, Avg_v, DTA, Frequency, Mass, S, Structure
> Y = (b.vector[1]+(Avg_d*b.vector[2]))+(log(Avg_v)*b.vector[3])+(log((log(Frequency)+.1))*b.vector[4])
> P = exp(Y)/(1+exp(Y))
> gpidata.full.predictions = data.frame(S,Mass,Structure,Avg_d,Avg_v,Frequency,DTA,Y,P)
>
> file="C:/Users/Clem/Desktop/GPI/Logistic/iterative/dis-round-5.csv"
> sink(file, append=FALSE, split=FALSE)
> gpidata.full.predictions
> sink()
> b.vector
[1] -42.8098470   0.7852826  11.5152076  13.0973917
>
```

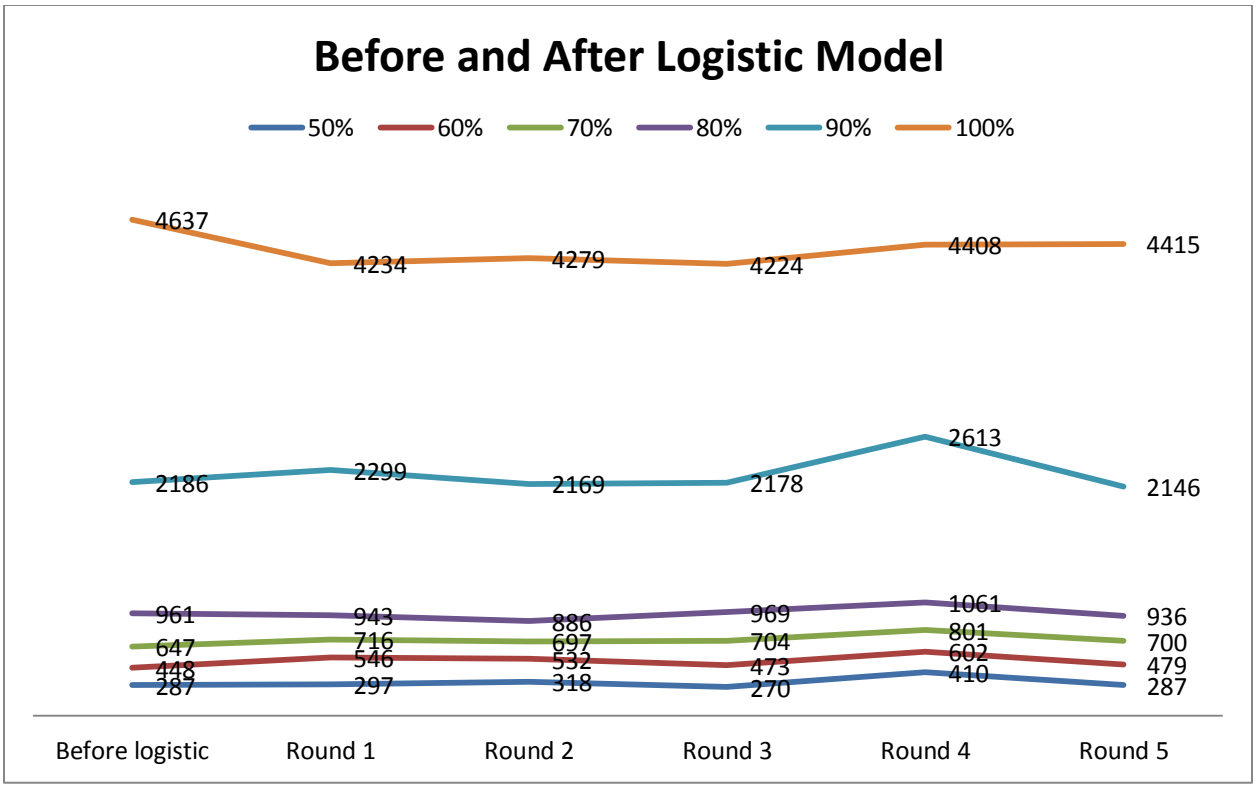


Figure 4.4 Comparative graph of score function *S* and five algorithm–expert iterations using logistic regression

While our algorithm did identify all the known GPI structures in *T. cruzi*, it shows low selectivity (78 real structures out of 4637 predicted overall, and 39 real structures out of the top ranking 287 predicted). The main reason why this occurred is the poor availability of information for predicting the lipid part of the molecule. Lipid structures are highly diverse because of the many possible variations of the lipid building blocks, how these blocks are linked, and their variations in both chain lengths and degree of saturation. They have multiple isobaric forms corresponding to the same mass. Although the interpretation of MS^2 data, which is what we have in our testing data set, can be used to designate glycan structures with a degree of precision, an accurate prediction of the lipid part of the GPIs would require MS^3 spectra. Because our set of data consists of MS^2 spectra, it is possible to obtain information of a detailed glycan part; however, the lipid part is revealed as a condensed mass, allowing way too many possible matches with the structures in the theoretical GPI library, thus contributing to the low selectivity in the resulting predictions.

Moreover, the choice of the mass tolerance parameters for parent ion and fragments depends on the mass accuracy and mass resolution of the instrument collecting the data. With low mass accuracy instruments, such as ion traps, one should specify a fairly large mass tolerance. With TOF-based mass spectrometers it is possible to achieve a mass accuracy of less than 0.1 Da, and even better mass accuracy of within 0.05 Da with Fourier transform MS instruments. We used a high tolerance because of the low mass accuracy of our instrument.

Chapter 5: Conclusions and Future Work

5.1 Conclusions

The structural roles of GPIs are important in understanding the interactions between cells and extracellular environment. In recent years, many structures and functions of GPIs have become clear. In this work, the identification of GPI structures was achieved by comparing observed fragment masses with theoretical fragment masses. We have used our own theoretical data library that was built starting from a basic set of glycan and lipid components. We were able to correctly match MS/MS spectra with their proposed GPI structure using our testing set of structures.

We developed three different versions of the proposed algorithm. We used a single computer, and two parallelized versions: HTCondor with 64 processors and MPI with 72 processors. Using the testing set of data, the single computer took approximately 10 days to complete the predictions. Taking advantage of the HTCondor high throughput distributing environment, we ran the algorithm using and average of 16 processors; this implementation took 3 days, 19 hours, 38 minutes to complete the predictions. A third version was implemented in MPI on the HPC resource at our institution; with 72 processors it completed in 22 hours and 38 minutes. Out of the two parallelized versions presented and examined, the HPC implementation proved to be more efficient.

5.2 Future work

Through the work done to date, I have identified several specific issues in GPI structure determination that still need to be investigated.

Our data library of theoretical structures is a work in progress. As more structural knowledge of GPI molecules is been obtained, the library can be enriched. Recently it has been discovered that GPI molecules can contain one or two pentose sugars in their backbone (unpublished results). Then, a new version of the library will consist of 5.7 million theoretical structures as pentoses are added. Also, as mention previously, it has been found that only *T. cruzi* and a few

other parasites belonging to the class Kinetoplastida can add AEP to their GPIs, then, to reduce the search time when an organism different from those containing AEP are investigated, a shorter library without AEP can be built. Such library will include pentose and therefore consist of 2.8 million theoretical structures.

It is necessary to collect and classify confirmed GPI structures in a database. Such database should be updated regularly as new GPI structures are confirmed. Future work also involves integration and cross-referencing of GPIs with GPI-associated proteins and genes. Association of GPIs with metabolic pathways should be also considered.

Ideally, one would like to construct organism specific data libraries (i.e., we will have one specific library for *T. cruzi*, one for human, etc.). Currently, as the dataset of its GPI structures is most accessible, we have constructed the specific data library for *T. cruzi only*. As new structures for different organisms are confirmed, specific data libraries for different organisms can be assembled.

We are working on extending the functionality of the algorithm to identify two additional common elements, called adducts, that GPI structures often contain: the sodium and potassium ions, Na+ and K+. We expect this extension will improve our selectivity, since a spectrum containing this adducts can be misinterpreted with our current algorithm.

More computers are being added to our HTCondor pool in the near future, we expect that the increased processing capacity will further reduce the runtime. We will need to investigate how to most efficiently handle the idle CPU cycles in our HTCondor pool. Our goal is to speed up the computations so that the entire GPI annotation process can be completed within one day on this system. We will also develop a user interface via web that would make the algorithm accessible to other research groups worldwide.

References

Apweiler R, Hermjakob H, Sharon N (1999) On the frequency of protein glycosylation, as deduced from analysis of the SWISS-PROT database. *Biochim Biophys Acta*. 1473:4–8.

Bessler M, Schaefer A, Keller P (2001) Paroxysmal nocturnal hemoglobinuria: Insights from recent advances in molecular biology. *Transfusion Medicine Reviews.* **15(**4):255-267.

Campos MA, Almeida IC, Takeuchi O, Akira S, Valente EP, Procópio DO, Travassos LR, Smith JA, Golenbock DT, Gazzinelli RT (2001). Activation of Toll-like receptor-2 by glycosylphosphatidylinositol anchors from a protozoan parasite. *J Immunol*. **167(1):**416-23.

Chen T, Kao MY, Tepel M, Rush J, Church GM (2001) A dynamic programming approach to de novo peptide sequencing via tandem mass spectrometry. *Journal of Computational Biology*, **8:** 325-337.

Chesebro B, Trifilo M, Race R, Meade-White K, Teng C, LaCasse R, Raymond L, Favara C, Baron G, Priola S, Caughey B, Masliah E, Oldstone M (2005) Anchorless Prion Protein Results in Infectious Amyloid Disease Without Clinical Scrapie. *Science* **308**:1435-1439 .

Cooper CA, Joshi HJ, Harrison M J, Wilkins MR, Packer NH (2003) GlycoSuiteDB: a curated relational database of glycoprotein glycan structures and their biological sources. 2003 update. *Nucleic Acids Res.* 2003, **31:**511–513.

Dancik V, Addona T, Clauser K, Vath J, Pevzner P (1999) De novo protein sequencing via tandem mass-spectrometry. *Journal of Computational Biology* **6**:327-341.

Dangaj D, Abbott KL, Mookerjee A, Zhao A, Kirby PS, Sandaltzopoulos R, Powell Jr. DJ , Lamazie`re A, Siegel DL, Wolf C, Scholler N (2011) Mannose Receptor (MR) Engagement by Mesothelin GPI Anchor Polarizes Tumor-Associated Macrophages and Is Blocked by Anti-MR Human Recombinant Antibody. *PLoS ONE* **6(12):** e28386.

de Groot PW, Brandt BW (2012) ProFASTA: a pipeline web server for fungal protein scanning with integration of cell surface prediction software. *Fungal Genet Biol* **49(2)**:173-179.

Domiguez M, Echaide I, Torioni de Ecahide S, Mosqueda J, Cetrá B. Suarez CE, Florin-Christensen (2010) *In silico* predicted conserved B-cell epitopes in the merozoite surface antigen-2 family of B. bovis are neutralization sensitive. *Veterinary Parasitology* **167**:216 – 226.

Domon B, Costello CE (1988) A systematic nomenclature for carbohydrate fragmentations in FAB-MS/MS spectra of glycoconjugates. *Glycoconjugate* **5**: 397-409.

Eng JK, Fischer B, Grossmann J, MacCoss MJ (2008) A Fast SEQUEST Cross Correlation Algorithm. *Journal of Proteome Research* **7**:4598–4602

Fankhauser N, Mäser P (2005) Identification of GPI anchor attachment signals by a Kohonen self-organizing map. *Bioinformatics* **21**: 1846–1852.

Ferguson MA (1999) The structure, biosynthesis and functions of glycosylphosphatidylinositol anchors, and the contribution of trypanosome research. *Journal of Cell Science* **112**:2799-2809.

Ferguson MAJ, Allen AK, Snary D. (1982) The detection of phosphonolipids in the protozoan Trypanosoma cruzi. *Biochem J* **207:**171-174.

Fenyo D, Beavis R (2003) A method for assessing the statistical significance of mass spectrometry-based protein identifications using general scoring schemes. *Analytical Chemistry* **75**:768–774.

Fujita M, Kinoshita T. (2012) GPI-anchor remodeling: potential functions of GPI-anchors in intracellular trafficking and membrane dynamics. *Biochim Biophys Acta*. **1821(8):**1050-8.

Goldberg D, Sutton-Smith M, Paulson J, Dell A (2005) Automatic annotation of matrix-assisted laser desorption/ionization N-glycan spectra. *Proteomics* **5:**865–875.

Ikezawa H (2002) Glycosylphosphatidylinositol (GPI)-anchored proteins. *Biol Pharm Bull* **25:**409-417.

Jacobs MG, Robinson PJ, Bletchly C, Mackenzie JM, Young PR (2000) Dengue virus nonstructural protein 1 is expressed in a glycosyl-phosphatidylinositol-linked form that is capable of signal transduction. *The FASEB Journal*. **14**:1603-1610.

Joshi HJ, Harrison MJ, Schulz BL, Cooper CA, Packer NH, Karlsson NG (2004) Development of a mass fingerprinting tool for automated interpretation of oligosaccharide fragmentation data. *Proteomics* **4:**1650–1664.

Kim S, Gupta N, Pevzner PA (2008) Spectral probabilities and generating functions of tandem mass spectra: a strike against decoy databases. *J Proteome Res*. **7(8):**3354–3363.

Ma B, Zhang K, Liang C (2005) An effective algorithm for the peptide de novo sequencing from MS/MS spectrum. *Journal of Computer and System Sciences* **70:**418-430.

Maass K, Ranzinger R, Geyer H, Lieth CW, Geyer R (2007) "Glyco-peakfinder" – de novo composition analysis of glycoconjugates. *Proteomics* **7:**4435–4444.

McConville MJ, Ferguson MA (1993) The structure, biosynthesis and function of glycosylated phosphatidylinositols in the parasitic protozoa and higher eukaryotes. *Biochem J* **294**: 305–324

Morelle W, Michalski JC (2005) The mass spectrometric analysis of glycoproteins and their glycan structures. *Current Analytical Chemistry*, **1**:29-57

Nakayasu ES, Yashunsky DV, Nohara LL, Torrecilhas AC, Nikolaev AV, Almeida IC (2009) GPIomics: global analysis of glycosylphosphatidylinositol-anchored molecules of Trypanosoma cruzi. *Mol Syst Biol* **5**:261.

Niemela PS, Castillo S, Sysi-Aho M, Oresic M (2009) Bioinformatics and computational methods for lipidomics. *Journal of Chromatography* **877**:2855‑2862

Nosjean O, Briolay A, Roux B (1997) Mammalian GPI proteins: sorting, membrane residence and functions. *Biochim Biophys Acta*. **1331**:153-86.

Omaetxebarria MJ, Elortza F, Rodriguez-Suarez E, Aloria K, Arizmendi JM, Jensen O N, Matthiesen R (2007) Computational approach for identification and characterization of GPI-anchored peptides in proteomics experiments. *Proteomics* **7**: 1951-1960.

Perkins DN, Pappin DJC, Creasy DM, Cottrell JS (1999) Probability-based protein identification by searching database using mass spectrometry data. *Electrophoresis* **20:**3551-3567.

Previato JO, Medonça-Previato L, Jones C, Wait R, Fournet B. (1992) Structural characterization of a novel class of glycophosphosphingolipids from the protozoan Leptomonas samueli. *J Biol Chem* **267:**24279-24286

Ropert C, Gazzinelli RT. (2000) Signaling of immune system cells by glycosylphosphatidylinositol (GPI) anchor and related structures derived from parasitic protozoa. *Curr Opin Microbiol*. **3(4):**395-403.

Routier FH, da Silveira EX, Wait R, Jones C, Previato JO, Medonça-Previato L. (1995) Chemical characterization of glycosylinositolphospholipids of Herpetomonas samuelpessoai. *Mol Biochem Parasitol* **69:**81-92.

Vainauskas S, Menon AK (2006) Ethanolamine phosphate linked to the first mannose residue of glycosylphosphatidylinositol (GPI) lipids is a major feature of the GPI structure that is recognized by human GPI transamidase. *The journal of biological chemistry* **281:**38358–38364.

Varki A, Cummings RD, Esko JD, et al., editors.Essentials of Glycobiology. 2nd edition.Cold Spring Harbor (NY): Cold Spring Harbor Laboratory Press; 2009.

Woodin CL, Hua D, Maxon M, et al. (2012) GlycoPep Grader: A web-based utility for assigning the composition of N-linked glycopeptides. *Anal Chem* **84**:4821–4829.

Yetukuri, L, Katajamaa M, Medina-Gomez G, Seppanen-Laakso T, Vidal- Puig A, Orešič M (2007) Bioinformatics strategies for lipidomics analysis: characterization of obesity related hepatic steatosis *BMC Systems Biology* **1:**12.

Zhao P, Nairn AV, Hester S, Moremen KW, O'Regan RM, Oprea G, Wells L, Pierce M, Abbott KL (2012) Proteomic identification of glycosylphosphatidylinositol anchor-dependent membrane proteins elevated in breast carcinoma. *The Journal of Biological Chemistry*, **287(30):**25230–25240.

List of Abbreviations

AAG – alkylacyl glycerol

AEP – aminoethylphosphanate

Cer – ceramide

CID – collision-induced dissociation

Da – dalton

DTA – Peak list

ER – endoplasmic reticulum

EtNP – ethanolamine phosphate

FDR – false discovery rate

GIPL – glycoinositolphospholipid

GLcN – glucosamine

GPI – glycosylphosphatidylinositol

Gro - glycerol

Hex – hexose

HexN – hexosamine

HMM – hidden Markov model

HPCVCRL - Performance Computing Virtual Research Lab

InsP – inositol phosphate

LC-MS – liquid chromatography tandem mass spectrometry

Man – mannose

MM – Markov model

MS – mass spectrometry

MS/MS – tandem mass spectrometry

m/z – mass-to-charge ratio

NANA – N-acetyl neuraminic acid

NN – neural network

PI – phosphatidylinositol

PTM – post-translational modification

TACC - Texas Advanced Computing Center

Xcorr – cross-correlation score

LC-MS/MS – liquid chromatography tandem mass spectrometry

Vita

Clemente Aguilar received his degree in Veterinary Medicine from the Autonomous University of Ciudad Juarez (UACJ) in Ciudad Juarez, Chihuahua Mexico. In 2008 he completed his Master of Science degree in Bioinformatics from the University of Texas at El Paso and joined the Doctoral Program in Computational Science in 2009.

He has attended national and international conferences, and presented posters and short talks at meetings, such as, the Annual Biomedical Research Conference for Minority Students (2009), SACNAS National Conference (2010), RECOMB Satellite Conference on Computational Proteomics (2010), Rio Grande Branch Annual Meeting of the American Society for Microbiology (2010, 2011), ACM Conference on Bioinformatics, Computational Biology and Biomedicine (2012), International Conference on Bioinformatics and Computational Biology (2013).

He authored and co-authored per-reviewed articles published in scientific journals, most of them in collaboration with other research groups from Brazil, United States and Singapore. He also coordinated the NSF-founded Undergraduate Participation in Bioinformatics Training Program (UPBiT) at UTEP under the direction of Dr. Ming-Ying Leung. This group consists of about 15 students, including undergraduates at UTEP and some early-college high school students, many of whom are underrepresented minorities. He hopes to continue working with undergraduates and high school students on research problems in the future.

The focus of his research is to develop computational methods for mass spectrometry data analysis, particularly on the molecular characterization of glycolipids and protein posttranslational modifications of pathogen parasites including *Trypanosoma cruzi*, etiological agent of Chagas disease. His dissertation, Automatic Elucidation of GPI Molecular Structures with Grid Computing, was supervised by Dr. Ming-Ying Leung and Dr. Igor C. Almeida.

After defending his dissertation he will start working on his post-doctoral training at the National Institute for Mathematical and Biological Synthesis (NIMBioS).