B.Sc. in Computer Science and Engineering Thesis

Combining Bayesian and Meta-heuristic approaches for the Protein Inference Problem

Submitted by

Sanjida Nasreen Tumpa
ID: 201114016

Md. Bhaktear Uddin Ahmed
ID: 2011141042

Ibtasham Afrin
ID: 201114052

Supervised by

Dr. M. Sohel Rahman

Professor

Department of Computer Science and Engineering

Bangladesh University of Engineering and Technology (BUET)

Dhaka-1000, Bangladesh

Department of Computer Science and Engineering
Military Institute of Science and Technology

December 2014

CERTIFICATION

This thesis paper titled **"Combining Bayesian and Meta-heuristic approaches for the Protein Inference Problem"**, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering in December 2014.

Group Members:

Sanjida Nasreen Tumpa

Md. Bhaktear Uddin Ahmed

Ibtasham Afrin

Supervisor:

Dr. M. Sohel Rahman
Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology (BUET)
Dhaka-1000, Bangladesh

CANDIDATES' DECLARATION

This is to certify that the work presented in this thesis paper, titled, "Combining Bayesian and Meta-heuristic approaches for the Protein Inference Problem", is the outcome of the investigation and research carried out by the following students under the supervision of Dr. M. Sohel Rahman, Professor, Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology (BUET), Dhaka-1000, Bangladesh.

It is also declared that neither this thesis paper nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

Sanjida Nasreen Tumpa
ID: 201114016

Md. Bhaktear Uddin Ahmed
ID: 2011141042

Ibtasham Afrin
ID: 201114052

ACKNOWLEDGEMENT

We are thankful to Almighty Allah for his blessings for the successful completion of our thesis. Our heartiest gratitude, profound indebtedness and deep respect go to our supervisor, Dr. M. Sohel Rahman, Professor, Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology (BUET), Dhaka-1000, Bangladesh, for his constant supervision, affectionate guidance and great encouragement and motivation. His keen interest on the topic and valuable advices throughout the study was of great help in completing thesis.

We also like to thank and express our sincere gratitude to Group Captain Md. Afzal Hossain, psc, the Head of the department of Computer Science and Engineering (CSE) of Military Institute of Science and Technology (MIST) for providing us great support during the thesis work.

We are especially grateful to the Department of Computer Science and Engineering (CSE) of Military Institute of Science and Technology (MIST) for providing their all out support during the thesis work.

We are also thankful to Saikat Chakraborty, Lecturer, Department of Computer Science and Engineering (CSE), Ahsanullah University of Science and Technology, for his help and guidance.

Finally, we would like to thank our families and our course mates for their appreciable assistance, patience and suggestions during the course of our thesis.

Dhaka
December 2014

Sanjida Nasreen Tumpa

Md. Bhaktear Uddin Ahmed

Ibtasham Afrin

ABSTRACT

Protein inference refers assembling peptides identified from tandem mass spectra into a list of proteins. Due to the existence of degenerate peptides, it is very difficult to determine which proteins are present in the sample. This problem is called protein inference problem and it represents a major challenge in shotgun proteomics as well as in proteomics research. Many approaches have been introduced for solving protein inference problem. In this paper, we have combined Bayesian and Meta-heuristic approaches for solving protein inference problem. Meta-heuristic approaches provide a very fast and efficient heuristic search strategy to infer proteins with reasonable accuracy and precision. It provides the flexibility to infer proteins either parsimoniously or optimistically or somewhere between the two by taking some tuning parameters. On the other hand Bayesian model provides a probabilistic model that incorporates the predicted peptide detectabilities as the prior probabilities of peptide identification. We propose a combination of these two approaches. We showed it by combining MAgPI (A Memetic Algorithm Based Approach in Protein Inference Problem) as Meta-heuristics approach and Gibbs Sampler for protein inferencing as Bayesian approach. In our system, Gibbs Sampler is processing the input of MAgPI and finally MAgPI is refining the input. As, our input is going through two refining processes, the final output is better than others in several aspects. Another important fact is that in our system computation time of MAgPI is less than before as after first stage of refining, the size of candidate solution becomes very short. We used Sigma49 dataset to test our method and got good result in several aspects.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

List of Algorithms

LIST OF ABBREVIATION

PissGA	: The Protein Inference using Steady State Genetic Algorithm
MAgPI	: Memetic Algorithm approach for Protein Inference
GA	: Genetic Algorithm
MMP	: Minimum Missed Peptide approach
PP	: ProteinProphet
BB	: Basic Bayesian model
BBA	: Basic Bayesian model with detectability Adjustment
ABP	: Advanced Bayesian model using raw PeptideProphet probabilities
ABL	: Advanced Bayesian model using converted Probability scores
ABLAP	: ABLA with estimated protein prior probabilities
CBM	: Combining Bayesian and Metaheuristics approaches for the Protein Inference Problem
TP	: True Positive
FP	: False Positive
FN	: False Negative
Pr	: Precision
Rc	: Recall
MA	: Memetic Algorithm
FDR	: False Discovery Rate

CHAPTER 1
INTRODUCTION

Proteomics, based on mass spectrometry is a large-scale study of proteins, particularly their structures and functions. It provides information that is not readily available from genomic sequence or RNA expression data. An explicit goal of proteomics is the identification of all proteins expressed in a cell or tissue as they are the vital parts of living organism and main components of the physiological metabolic pathways or cells.

1.1 Overview

The concept of "Shotgun proteomics" arose to overcome the difficulties of using previous technologies to separate complex mixtures. In 1975, two-dimensional polyacrylamide gel electrophoresis was described by OFarrell and Klose with the ability to resolve complex protein mixtures. The development of matrix-assisted laser desorption ionization, electrospray ionization, and database searching continued to grow the field of proteomics. However these methods still had difficulty identifying and separating low-abundance proteins, aberrant proteins, and membrane proteins. Shotgun proteomics emerged as a method that could resolve even these proteins. It refers to the use of bottom-up proteomics techniques of identifying proteins in complex mixtures using a combination of high performance liquid chromatography combined with mass spectrometry [6].

The comprehensive and quantitative analysis of proteins expressed in a given organ, tissue or cell line, provides additional valuable information about biological systems to complement the knowledge gained by genomics or transcriptomics approaches. Being able to identify and quantify proteins is of main importance in molecular and systems biology, since these macromolecules, as well as the interactions between them, play an essential role in cell functions.

It is important to know which proteins are present in a sample, but the abundance of these molecules is also of major interest. For instance, one would like to be able to figure out which are the most/least abundant proteins in a sample, or to compare the abundance of the same protein in two samples under different biological conditions. In medical sciences, for example, biomarkers can be used to monitor the efficiency of a treatment by comparing the molecule's concentration before and after a therapy [7].

The most common method of shotgun proteomics starts with the enzymatically digested proteins in the mixture which are optionally fractionated from their biological source. The resulting peptide mixtures are then ionized and scanned by tandem mass spectrometry (MS/MS) to obtain a set of MS/MS spectra. These spectra are subsequently searched against a protein database to identify peptides present in the sample. After the peptides have been identified, it is necessary to validate the identification process so that further steps do not suffer from noisy inputs. Many peptide search engines have been developed, among which Sequest [8], Mascot [9] and X!Tandem Sequest [10] are commonly used. Finally, peptides and proteins are identified by computational analysis [1]. Figure 1.1 shows the whole procedure as a pipeline diagram.

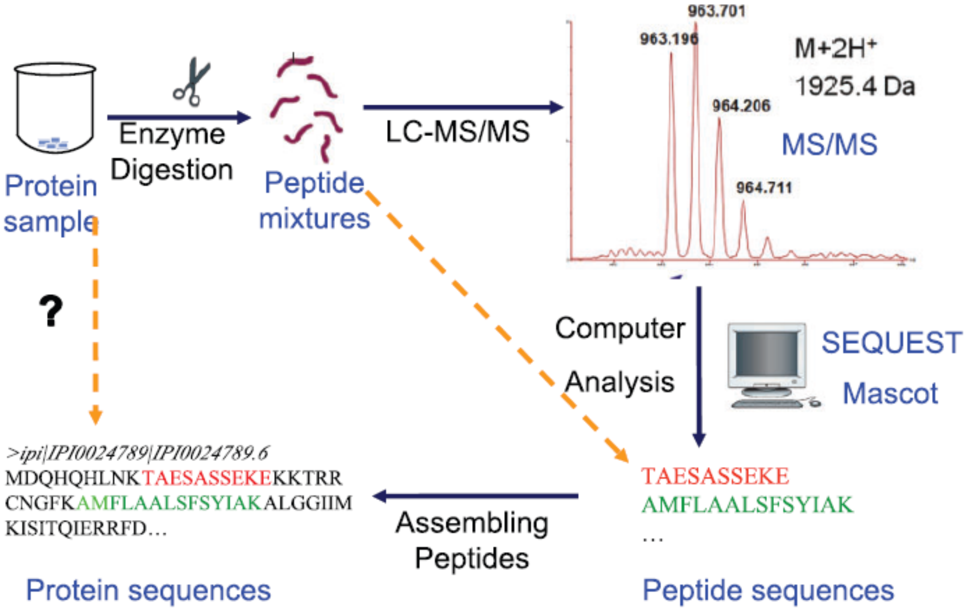


Figure 1.1: A general pipeline for the identification of proteins in shotgun proteomics. Figure has been borrowed from [1].

Most algorithms have set their input by modelling the relationship between the identified peptides and the proteins in the database as a bipartite graph. It is necessary to make a one to one mapping between the proteins and the peptides but if there were a one to one mapping from the proteins to peptides or vice-versa, the solution would be trivial, but the actual situation is not that simple. Even if the identified set of peptides is reliable, it does not ensure that a reliable list of proteins can be assembled from these peptides. Figure 1.2 shows that, protein P_1 is an *one hit wonder* protein as it generates only one peptide after digestion. We can identify that protein very easily from its generated peptide by searching the database. But protein P_2 and protein P_3 both generates same peptide 5. So, it is not straightforward to say from which protein peptide 5 generated. These type of peptides are called *degenerated peptide*. One of the most challenging problems is the peptide degeneracy issue, which arises when a single peptide can be mapped to multiple proteins. Thus, in the presence of a degenerate peptide, it is always difficult to deduce with protein or protein family actually generated the peptide. As a result, the protein inference problem, often has multiple solutions and can be computationally intractable. Three types of model are used in the protein inference domain -

- Statistical model (ensures protein inferencing with high accuracy)
- Parsimonious model (assumes only a small subset of proteins should be sufficient to explain all identified peptides)
- Optimistic model (returns all the protein or protein families that has some potential) [3]

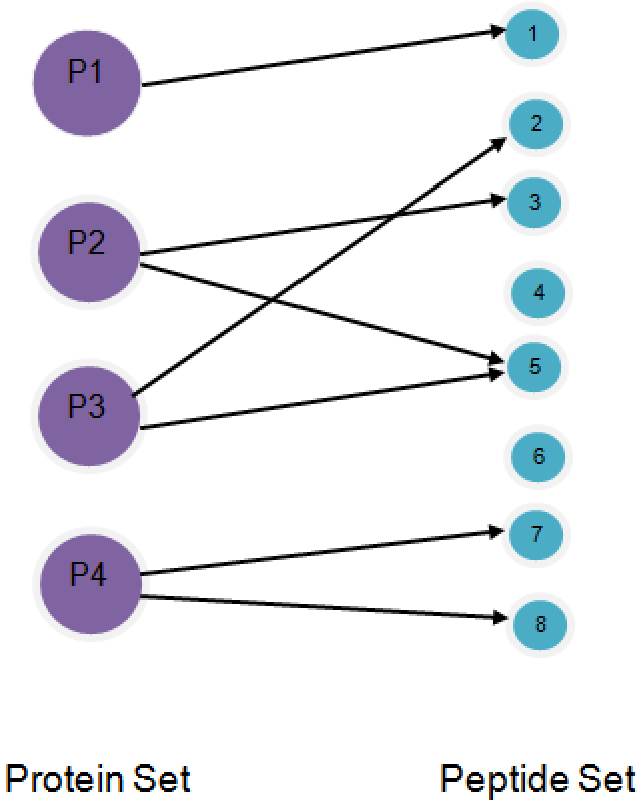


Figure 1.2: A bipartite graph represents the protein-peptide relation

The protein identification procedure therefore can be divided into three major steps

1. Peptide Identification
2. Protein Inference

3. Result Evaluation

The main difficulty lies in the intermediate step *Protein inference*. So, the problem of determining which of the proteins is present in the sample is known as *Protein inference problem* [3].

1.2 Motivation

The main motivation behind combining these two approaches is to reach globally optimal solution by incorporating predicted peptide detectability as prior probability to yield a better result. We have used prior probability to identify those peptides which are not being identified by peptide search engine but have a significant impact on the work. We were intended to map protein inference problem as evolutionary search problem and use Memetic Algorithm as a synergy of evolutionary with separate individual learning procedure from its surrounding. We have used different technique to maintain the diversity of solution. The Meta-heuristic part of our approach does not work with the best individuals only, it gives chance to the less fit ones too.

1.3 Objectives

It is necessary to fix some objectives or aim before starting any kind of research work. Following objectives have been decided to solve protein inference problem with a high performance by combining Bayesian and Meta-heuristic approaches:

- Incorporating peptide detectability as prior probability in peptide identification
- Implementing Gibbs Sampling algorithm for the output as MAP probability
- Reducing search area for protein identification which decreases the searching time
- Incorporating Memetic Algorithm with diversity maintenance mechanism
- Achieving a better result in the combined process than the individual ones

1.4 Organization of the thesis

This thesis is organized as follows: chapter 2 describes some basic but important definitions and the related works have been done to solve protein inference problem. Chapter 3 discusses our combination procedure in detail. Chapter 4 shows the experimental setup, results using comparison with other works and gives possible future directions. Finally, Chapter 5 draws the conclusion.

CHAPTER 2
PRELIMINARIES

In this chapter, we have discussed about some definitions of common technical terms and related works that are intended to solve protein inference problem.

2.1 Basic Definitions

To have a clear knowledge on the working procedure of our approach, one has to be familiar with some basic but very important definitions. The related definitions have been provided below:

- **Protein**

  Protein can be defines as, Any of a class of nitrogenous organic compounds that consist of large molecules composed of one or more long chains of amino acids and is an essential part of all living organisms, especially as structural components of body tissues such as muscle, hair, collagen, etc., and as enzymes and antibodies [11].

  In other words, proteins are large biological molecules, or macromolecules, consisting of one or more long chains of amino acid residues [12].

- **Peptide**

  Peptide can be defined as, a compound consisting of two or more amino acids linked in a chain, the carboxyl group of each acid being joined to the amino group of the next by a bond of the type -OC-NH- [11].

- **Peptide Detectability**

  Peptide detectability is defined as the probability that a peptide is identified in an LC-MS/MS experiment and has been useful in providing solutions to protein inference and label-free quantification [11].

- **Prior Probability**

  In Bayesian statistical inference, a prior probability distribution, often called simply the prior, of an uncertain quantity p is the probability distribution that would express one's uncertainty about p before some evidence is taken into account [13].

- **Posterior Probability**

  In statistical terms, the posterior probability is the probability of event A occurring given that event B has occurred [11].

- **Gibbs Sampling**

  Gibbs Sampling is a commonly used strategy to approximate a high-dimensional joint distribution that is not explicitly known [14, 15]. This algorithm is used to achieve the optimal protein configuration with the MAP probability.

- **True Positive (TP)**

  True positive denotes the amount of proteins that are correctly identified by an approach. Increase in TP is positive for an approach.

- **False Positive (FP)**

  False positive denotes the amount of proteins that are identified by an approach but they are not in the true class. Decrease in FP is considered as positive.

- **False Negative (FN)**

  False negative denotes the amount of proteins that are not identified by an approach but they are in the true class. Decrease in FN is considered as positive.

- **Precision (Pr)**

  Ratio of correctly inferred proteins within the test peptide set that is inferred to be present. Precision of an algorithm is considered as best if it is 1.

  $$Precision = \frac{TP}{(TP + FP)} \tag{2.1}$$

- **Recall (Rc)**

  Recall can defined as the rate of true positive. Increase in recall is better for an approach.

  $$Recall = \frac{TP}{P} \tag{2.2}$$

- **F Measure**

  The harmonic mean of precision and recall is called F measure. It is better to have a better f measure.

  $$FMeasure = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \tag{2.3}$$

2.2 Related Works on Protein Inference Problem

Over the past few years, various research takes place on proteomics. Various method has been used for proteomics. Nesvizhskii and his colleagues first addressed this protein inferencing challenge using a probabilistic model [16], After that different problem formulations and new solutions have been proposed as well [17–19]. A new concept of peptide detectability with the goal of finding the set of proteins with the minimal number of missed peptides is discussed in [18]. Again greedy or graph-pruning strategies [18] address the protein inference problem without performance guarantee. In other approach [2], the protein inference is addressed by proposing two novel Bayesian models that take as input a set of identified peptides from any peptide search engine, and attempt to find a most likely set of proteins from which those identified peptides originated. The basic model assumes that all identified peptides are correct, whereas the advanced model also accepts the probability of each peptide to be present in the sample. Few classes of cooperative Meta-heuristics like the Island model, Spatially embedded models and genetic programming have been used in protein identification. The evolutionary identification approach [20] tries to find the entire sequence of a protein, even in the case of variants or unknown proteins. To accomplish that, different peptides that composes a given protein must be identified.

As we worked on the target of combining Bayesian approach and Meta-heuristic approach, we have gathered knowledge on these topics deeply. In the following sub sections these two approaches have been discussed by inspired from [2–4].

2.2.1 Bayesian Approach

Bayesian approach solves protein inference by proposing two Bayesian models where a set of identified peptides from any peptide search engines used as input and then attempt to find the set of proteins from which those identified peptides originated which is actual protein.

Representation of protein and peptide

A bipartite graph can be defined as a protein configuration graph. In Figure 2.1,

- Two disjoint sets of vertices represent the proteins in the database and the peptides from these proteins, respectively.
- Each edge indicates that the peptide belongs to the protein.
- The protein configuration graph is independent of the proteomics experiment, and thus can be built from a set of protein sequences.
- In this model, identified peptides and non identified peptides both are considered.

- A protein configuration graph is partitioned into connected components, each representing a group of proteins sharing one or more (degenerate) peptides. If there are no degenerate peptides in the database, each connected component will contain exactly one protein and its peptides.

- The protein configuration graph can be interpreted as a Bayesian network with edges pointing from proteins into peptides,

- It is cleared that protein inference can be addressed separately for each individual connected component.

- The peptide identification results are first mapped to the protein configuration graph.A vector of indicator variables $(y_1, ..., y_j, ..., y_n)$, are referred to as the peptide configuration, to denote a set of identified peptides.

Now, the protein configuration graph can be simplified by removing proteins containing no identified peptides. After the simplification, one connected component in the original protein configuration graph may be partitioned into several small components. Identify trivial and non-trivial connected component so that the protein inference problem can be reduced to finding the protein configuration $(x_1, ..., x_i, ..., x_m)$ by analysing non-trivial components only [2].

Basic Bayesian Model

This model assumes that all identified peptides are correct.It can be considered as a special case of the advanced model, where the probabilities $r_{i,j}$ for different peptides j are limited to 0 as non-identified peptides or 1 as identified peptides. Practically, the basic model can be used when the probabilities rj are not provided and while while the identified peptides are obtained at a stringent false discovery rate (FDR), e.g., 0.01, by either a heuristic target-decoy search strategy [19, 21, 22] or by probabilistic modeling of random peptide identification scores [23,24]. In the next section, basic model is extended to a more realistic model in which different probabilities are incorporated for different identified peptides that are estimated based on the peptide identification scores. When the probabilities of identified peptides are available, it is expected that the advanced model should perform better than the basic model [2].

Now considering m proteins and n peptides from these proteins within a non-trivial connected component of the protein configuration graph. Each protein i is either present in the sample or absent from it, which can be represented by an indicator variable x_i. Therefore, any solution of the protein inference problem corresponds to a vector of indicator

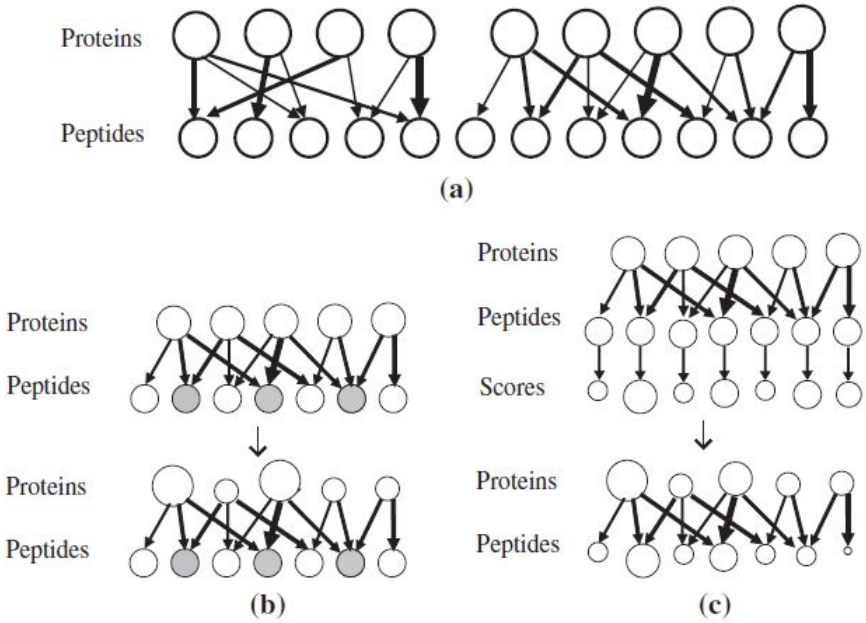


Figure 2.1: (a) A protein configuration graph consisting of two connected components. (b) Basic Bayesian model for protein inference. In which peptides are represented as a vector of indicator variables: 1 (gray) for identified peptides and 0 (white) for non-identified peptides. (c) Advanced Bayesian model for protein inference. In which each peptide is associated to an identification score (0 for non-identified peptides). Sizes of circles reflect prior/posterior probabilities. Figure has been borrowed from [2].

variables,$(x_1, ..., x_m)$, referred to as a protein configuration. Given the set of identified peptides from peptide search engines (peptide configuration $(y_1, ..., y_n)$), the goal is to find the maximum a posteriori (MAP) protein configuration, that is the configuration that maximizes the posterior probability $P(x_1, ..., x_m|y_1, ..., y_n)$,

$$(x_1, ..., x_m)_{MAP} = argmax_{(x_1,...,x_m)} P(x_1, ..., x_m|y_1, ..., y_n) \tag{2.4}$$

Using Bayes rule, the posterior probability can be expressed as:

$$P(x_1, ..., x_m|y_1, ..., y_n) = \frac{P(x_1, ..., x_m)P(y_1, ..., y_m|x_1, ..., x_n)}{\sum_{(x_1,...,x_m)}[P(x_1, ..., x_m)P(y_1, ..., y_m|x_1, ..., x_n)]}$$
$$= \frac{P(x_1, ...x_m)\prod_j [1 - P_r(y_j = 1|x_1, ..., x_m)]^{1-y_j} P_r(y_j = 1|x_1, ..., x_m)^{y_j}}{\sum_{(x_1,...,x_m)} P(x_1, ..., x_m)\prod_j [1 - P_r(y_j = 1|x_1, ..., x_m)]^{1-y_j} P_r(y_j = 1|x_1, ..., x_m)^{y_j}} \tag{2.5}$$

where $P(x_1, ..., x_m)$ is the prior probability of the protein configuration. Assuming the presence of each protein i is independent of other proteins, this prior probability can be computed as:

$$P(x_1, ..., x_m) = \prod_i P(x_i) \tag{2.6}$$

$P_r(y_j = 1|x_1, ..., x_m)$ is the probability of peptide j to be identified by shotgun proteomics given the protein configuration $(x_1, ..., x_m)$. Assuming that different proteins contribute independently to the identification of a peptide, we can compute it as:

$$P_r(y_j = 1|x_1, ..., x_m) = 1 - \prod_i [1 - x_i P_r(y_j = 1|x_i = 1, x_1 = ... = x_{i-1} = x_{i+1} = ... = x_m = 0)] \tag{2.7}$$

where $P_r(y_j = 1|x_i = 1, x_1 = ... =\mathrm{x}_{i-1} = x_{i+1} = ... = \mathrm{x}_m = 0)$ is the probability of peptide j to be identified if only protein i is present in the sample. This probability, referred to as the standard peptide detectability d_{ij}, is an intrinsic property of the peptide (within its parent protein), and can be predicted from the peptide and protein sequence prior to a proteomics experiment [25]. Combining equations (2.5) and (2.7), we can compute the posterior probabilities for protein configurations as:

$$P(x_1, ..., x_m|y_1, ..., y_n) = \frac{\prod\limits_i P(x_i) \prod\limits_j \{[\prod\limits_i (1 - x_i d_{ij}]^{1-y_j} [1 - \prod\limits_i (1 - x_i d_{ij})]^{y_j}\}}{\sum_{(x'_1, ..., x'_m)} \prod\limits_i P(x'_i) \{[\prod\limits_i (1 - x'_i d_{ij}]^{1-y_j} [1 - \prod\limits_i (1 - x'_i d_{ij})]^{y_j}\}} \tag{2.8}$$

It is also possible to compute marginal posterior probability of a specific protein i to be present in the sample, which can be expressed as:

$$p(x_i|y_1, ..., y_n) = \sum_{(x_1, ...x_{i-1}, x_{i+1}, ..., x_m)} P(x_1, ..., x_m|y_1, ..., y_n) \tag{2.9}$$

Advanced Bayesian Model

The advanced model accepts the probability of each peptide to be present in the sample. Where in basic Bayesian model, all identified peptide has equal probabilities of being correctly identified. Here peptide identification score s_j for each peptide j is introduced, which is the output of the peptide search engines. It is assumed the peptide identification score is highly correlated with the probability of a peptide being correctly identified and their relationship can be approximately modelled using probabilistic methods. The goal is to compute $P(x_1, ..., x_m|s_1, ..., s_n)$ by enumerating all potential peptide configurations:

$$P(x_1, ..., x_m|s_1, ..., s_n) = \sum_{(y_1, ..., y_n)} [P(x_1, ..., x_m|y_1, ..., y_n) P(x_1, ..., x_m|s_1, ..., s_n)] \tag{2.10}$$

It is assumed that s_j is independent of the presences of the other peptides (i.e., $(y_1, ...,$y$_{j-1}$, y_{j+1}, . . ., y$_n$)) for each peptide j, so

$$P(s_1, ..., s_n|y_1, ..., y_n) = \prod_j P(s_j|y_j) \tag{2.11}$$

By applying Bayes rule,

$$P(s_1, ..., s_n|y_1, ..., y_n) = \prod_j \frac{P(y_j|s_j)P(s_j)}{P(y_j)} = \prod_j \frac{(1-r_j)^{1-y_j} r_j^{y_j} P(s_j)}{P(y_j)} \tag{2.12}$$

where the marginal probability of a peptide j can be computed as:

$$P(y_j) = \sum_{x_1,...,x_m)} [P(y_i|x_1...x_m)] = [1-\prod_i(1-P_r(x_i = 1)d_{ij}]^{y_j}[1-\prod_i(1-P_r(x_i = 1)d_{ij}]^{1-y_j} \tag{2.13}$$

Combining these equations, it is possible compute the posterior probability of protein configurations as:

$$P(x_1, ..., x_m|s_1, ..., s_n) = \frac{\sum_{(y_1,...,y_n)}\{\prod_i P(x_i)\{[\prod_i(1-x_i d_{ij})]^{1-y_j}[\prod_i(1-x_i d_{ij})]^{y_j}\frac{(1-r_j)^{1-y_j} r_j^{y_j}}{P(y_j)}\}\}}{\sum_{(x'_1,....x'_m)(y_1,....y_n)}\{\prod_i\{P(x'_i)\{[\prod_i(1-x'_i d_{ij})]^{1-y_j}[1-[\prod_i(1-x'_i d_{ij})]^{y_j}]\frac{(1-r_j)^{1-y_j} r_j^{y_j}}{P(y_j)}\}\}} \tag{2.14}$$

It is also possible compute the posterior probability of a specific protein i present in the sample as:

$$P(x_i|s_1, ..., s_n) = \sum_{(x_1,....x_{i-1},x_{i+1},.......,x_m)} P(x_1, ..., x_m|s_1, ..., s_n) \tag{2.15}$$

The marginal posterior probability of a peptide j as:

$$P(y_j|s_1, ..., s_n) = \sum_{(x_1,....,x_m,y_1,.....y_{j-1},y_{j+1}.....y_n)} [P(x_1, ..., x_m|y_1, ..., y_n)P(y_1, ..., y_n|s_1, ..., s_n)] \tag{2.16}$$

Gibbs Sampling Algorithm

Gibbs Sampling is a commonly used strategy to approximate a high-dimensional joint distribution that is not explicitly known [14, 15]. It is adopted to achieve the optimal protein configuration with the MAP probability. The original Gibbs Sampling algorithm considers one individual variable at a time in the multi-dimensional distribution. It, however, often converges slowly and is easily trapped by local maxima for long time. Several techniques have been proposed to improve the search efficiency of Gibbs Sampling algorithm, such as random sweeping, blocking, and collapsing [15]. Because in this each variable x_i to be sampled has small search space (i.e., 0,1) and the block sampling technique is applied in this Gibbs Sampler algorithm. Without increasing the computational complexity, a novel memorizing strategy is adopted that keeps a record of all posterior probabilities among all configurations are evaluated during the sampling procedure, and report the maximum solution in the end [2].

2.2.2 Meta-heuristic Approach

Two Meta-heuristic approaches for solving the protein inference problem are proposed. They have been discussed in the following sub-section.

1. PlssGA
2. MAgPI - It is the continuation work of PIssGA.

The first attempt to solve protein inference problem using Meta-heuristic approach is PIssGA. In PissGA, a steady state genetic algorithm was used to solve the protein inference problem. The reason behind choosing the steady state version of genetic algorithm is due to the exploitative nature of the protein inference problem. PIssGA iterates through the fitness evaluation,parent selection, breeding and survival selection procedures after initialization. The contribution of this algorithm is,

- To provide a very fast and efficient heuristic search strategy to infer proteins with reasonable accuracy and precision
- To provide the flexibility to infer proteins either parsimoniously or optimistically or somewhere between the two, based on some tuning parameter [3]

In MAgPI the main target is to reach globally optimal solution avoiding the local optima as in steady state GA approach, the solution can be stuck in local optima. MAgPI does not neglect the less fit individuals completely and It gives some chance to the less fit individuals to reproduce and even to survive. The contribution of this algorithm is,

- It is based on Memetic Algorithm and explored more utilization of evolutionary computation in this research. The algorithm 1 has been included in the algorithm section.
- It utilizes different technique in maintaining diversity of solutions from the traditional distance based diversity maintenance

- In MAgPI protein inference problem has been mapped as an evolutionary search. In this strategy, selected individuals are allowed to learn from the surrounding and to propagate the improvement to the next generations [4].

Following these two algorithms will be discussed described:

Candidate solution Representation

For representing candidate solution in both algorithm,

- If the protein is present, the corresponding gene has value 1 and

- If the protein is absent, the corresponding gene has value 0

Thus a string of 0 and 1s will form an individual. Figure 2.2 gives an example representation which assumes five potential proteins. In Figure 2.2, the candidate solution represents that the testing biological sample contains proteins Pr_2, Pr_3 and Pr_5. Main database contain more protein which is not potential with respect to peptide sequences obtained from MS and MS/MS spectrum analysis. So, they can be discarded from candidate solution representation as they never can appear [3,4].

Pr_1	Pr_2	Pr_3	Pr_4	Pr_5	Pr_6
0	1	1	0	1	0

Figure 2.2: Representation of the candidate solution.

Generating Initial Population

In many protein databases, peptide sequences of proteins are well known and available. For the initial population generation, consider following steps,

- **Step 1: Mapping**

  The mapping of protein to its peptide sequence. An hypothetical mapping is demonstrated in Figure 2.3, where Pr_n is a protein and the symbols at the right of the arrow sign actually represents its hypothetical peptide sequence.

- **Step 2: Potential Protein Identification**

  If protein Pr_1 generates a peptide sequence which contains the peptide G, then the former is obviously a potential parent of the latter. In this way, identify all the potential proteins in the sample, form a set of potential proteins removing any repetitions and

$Pr_1 \rightarrow$ GKEUTR
$Pr_2 \rightarrow$ LY ARE
$Pr_3 \rightarrow$ LV GARTHNB
....
....
$Pr_n \rightarrow$ WHBAFGSTHJSY B

Figure 2.3: An hypothetical mapping of protein and peptide

discard the rest of the proteins as they are of no interest for this particular test case. As the number of proteins present in the test sample cannot exceed the cardinality of the potential protein set, number of genes in an individual is kept equal to the cardinality of the latter. This helps to reduce the size of the candidate solution and reduce the search space.

- **Step 3: Generation of Initial Population**

  After the number of potential proteins in an individual and also the individual potential proteins have been identified,it is ready to generate the initial population [3,4]. Figure 2.4 shows a sample initial population,

Pr_3	Pr_4	Pr_9	Pr_{11}	Pr_{14}
0	0	1	0	0
0	1	0	1	1
0	0	1	1	0
1	1	0	0	1

Figure 2.4: A sample initial population

Recombination and Mutation as Breeding Operators

In PlssGA uniform crossover was used as recombination, because candidate solution is encoded in Boolean valued vector and sequence gene has not any special significance. This means every gene has equal probability of swapping and this probability is denoted by γ which is user defined. Figure 2.5 demonstrates uniform crossover process which swaps the genes corresponding to Pr_1, Pr_3 and Pr_6.

For mutation, bit-flip mutation operator is applied. Here, a gene is randomly randomly selected from an individual and with a very small probability μ. Flip is done to change value of gene. Figure 2.6 demonstrates this process of mutation operator flips the value of Pr_3 from 1 to 0 [3,4].

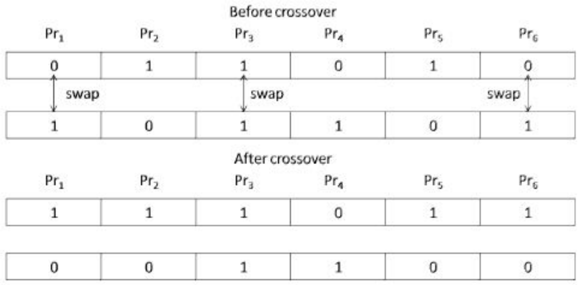


Figure 2.5: Uniform Crossover. Figure has been borrowed from [3,4].

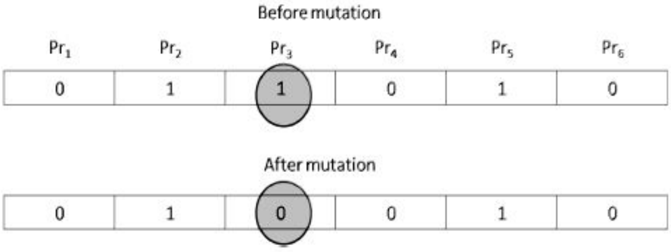


Figure 2.6: Bit flip mutation operator. Figure has been taken from [3,4].

Procedure of Offspring Education in MAgPI

To accommodate the essence of Memetic Algorithm, offspring education procedure is used on selected individuals. m

Parent Selection Procedure for Breeding

In PIssGA, it is performed using the roulette wheel selection mechanism, also known as proportionate reproduction. It selects individuals for crossover and mutation operations.

The survival selection follows the exploitative approach. PIssGA only replaces a parent individual only when the child has a greater fitness [3].

On the other hand in MAgPI, The survivor or parent selection of MAgPI is a combination of both elitist and Fitness Uniform Selection. Kindividuals for next generation are selected in elitist approach, the rest are selected uniformly over fitness landscape(FUSS) [4].

Evaluating Fitness

In PlssGA the fitness function should consider the following issues:

- First issue is the choice of whether to consider the minimum or maximum number of proteins that cover the test peptide sequence or whether to keep both the options and make it adaptive.

- Percentage of test peptides covered by the proteins inferred by a candidate solution.
- Another issue is how much the hypothetical peptide set constructed from the inferred proteins of an individual is similar to the test peptide sequence. The more similar they are, the higher is the probability of the inferred protein sets to be correct. Also important is the issue of how many redundant peptides does this hypothetical peptide set contains.
- The protein inference procedure is not deterministic due to the presence of both one-hit wonders and degenerate peptides. So, an exact solution may not be available and approximate solutions may be only possible option.

On the other hand, in MAgPI while evolving the candidate solutions of protein inference problem, the fitness function should consider the following issues:

- The most unique features of MAgPI which gives user the control over whether to prefer the minimum or maximum number of inferred proteins or take a mid way around.
- Percentage of peptides covered within the test peptide sequence by the inferred proteins of a candidate solution.
- The amount at which the hypothetical peptide set constructed from the inferred proteins of an individual is similar to the test peptide sequence. The more they are similar, the higher is the probability of the inferred protein set of being correct. How many redundant peptides does this hypothetical peptide set contains.
- Procedure is not deterministic due to the presence of both one-hit wonders and degenerate peptides. So, an exact solution may not be available and approximate solutions may be only possible option.

So, the following functions are used in both algorithms:

- N(c): It returns the number of 1s in an individual c.
- C(c): This is the coverage function which returns how much the hypothetical peptide set constructed from the inferred proteins of individual c cover the test peptide sequence. of individual c cover the test peptide sequence.

- R(c): This is the redundancy function which returns how much the hypothetical peptide set resulting from inferred proteins of individual c contains redundant proteins with respect to the test peptide sequence [3].

- S(c): This is the shield function which returns number of test peptides not covered by the hypothetical peptide set resulting from inferred protein set in MAgPI [4].

Consider Figure 2.7 for a better understanding of the Coverage and the Redundancy measure,Here, Coverage means the intersection of the inferred hypothetical peptide set (A) and the test peptide set (B) and Redundancy contains the peptides in the inferred hypothetical peptide set that are not part of the test peptide set [3].

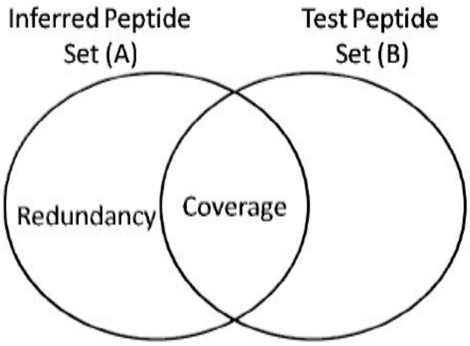


Figure 2.7: Demonstration of coverage and redundancy. Figure has been borrowed from [3].

Now In PlssGA the fitness equation can be expressed as:

$$F(c) = [(C(c))^{\alpha} - (R(c))^{\beta}] * (N(c))^{1-\epsilon} * (\frac{1}{N(c)})^{\epsilon} \tag{2.17}$$

Here, the fitness function defined by equation (2.17) which tries to maximize the coverage and minimize the redundancy simultaneously.Here, α and β are two weighting exponent fractions. These two constants are varied for 0 to 1 and adjusted by experiment. ϵ is a user defined parameter, which indicates the user preference for whether to take maximum possible number of proteins or minimum possible number of proteins as the desired output or something in between the two [3].

In MAgPI fitness function can be expressed in the following manner:

$$F(c) = \frac{1}{\Psi_c^i * \frac{1}{f} + (1 - \Psi_c^i) * \frac{1}{\eta}} * N(c)^{\epsilon} * (\frac{1}{N(c)})^{1-\epsilon} \tag{2.18}$$

Where f is Fidelity and ϵ is Exposure. Fidelity signifies how trustworthy the individual is in inferring the protein according to our proposed heuristic. And Exposure signifies how much expressive an individual in expressing the test peptide set,

$$f = \frac{C(c)}{C(c) + R(c)} \tag{2.19}$$

$$\eta = \frac{C(c)}{C(c) + R(c)} \tag{2.20}$$

CHAPTER 3
THEORETICAL FRAMEWORK OF OUR APPROACH

The theoretical description of our approach will be discussed in this chapter. Here we discussed about selection of algorithms for the improvement process of output quality and combining Bayesian and Meta-heuristic approaches. Particularly, initial generation of input, using Gibbs Sampling algorithm, the process of filtering the input, mapping the protein and peptide, apply filtered input in MAgPI and at last the overall work flow will be discussed here.

3.1 Selection of Algorithm

We had to select two algorithm based on Meta-heuristic approach and Bayesian approach. In the previous chapter we have discussed about various algorithm of this two approaches.

- From Bayesian approach, we are taking Gibbs Sampler for protein inferencing using the advanced model. Because here peptide detectability is included.
- From Meta-heuristic approach, we are taking MAgPI as it the latest work and continuation of PlssGA.

3.2 Improvement Process of Output Quality

- If we want a better output, one of the possible strategies is providing a better quality input set to the algorithm. In MAgPI in the second phase “Initial Population Generation” peptide detectability and prior probability have not considered while generating the candidate solution. By using Gibbs Sampler we can take account of peptide detectability and prior probability of our input.
- Generally MAgPI takes a generated peptide and then includes all the potential parent proteins that might have generated the peptide in the protein set by searching the protein-peptide database. In this way, all the potential proteins in the sample have been included to form a set of potential proteins. The size of these protein set is large

and it has a major effect on the output of the approach. But by using Gibbs Sampler as a pre processing unit of input refining, database size will reduce.

- As database where all protein-peptide relation exists in MAgPI is large, the searching procedure takes a long time which increases the running time of the algorithm. But in our approach searching time will definitely reduce as database is short.

So, the most important challenge of our approach is increasing the input quality of MAgPI by using Gibbs Sampler so that it can results in a better quality output.

3.3 Our Approach

In our approach we are refining the initial input into two phase. The output of first phase is going to as input to the second phase. So there is Gibbs Sampler which is working as a pre-processing unit of input and MAgPI as a final refiner. Note that, we did not implement these algorithms. The implementation of MAgPI was provided by our supervisor and Gibbs Sampler by [26].

3.3.1 Initial Input Generation

In initial input, we have incorporated the peptide detectabilities as the prior probabilities of peptide identification. Prior probability is important as all the peptides belong to the same protein are not observed. Some of them can be found but the others are not. The peptides that are not identified can have significant impact on the work [25].

We have collected the input files from the website [26]:

- Input file with a list of sequences and confidence score for candidate peptide identifications
- Input file with detectability for all tryptic peptides from all candidate proteins
- Input file with prior probability for proteins

3.3.2 Gibbs Sampling

Gibbs Sampling is adopted to achieve the optimal protein configuration with the MAP probability. It often converges slowly and is easily trapped by local maxima for long time. We have used MsBayesPro.exe which is a proof-of-principle implementation of the Bayesian protein inference algorithm. It gives output including the posterior probabilities for the proteins, posterior probabilities for the peptides and estimated even prior for proteins and also uses Gibbs Algorithm 2

The generated output files are:

- An output file with an extension of .bayes53 which contains the full inference result including the posterior probabilities for the proteins
- An output file with an extension of .peppost which contains the posterior probabilities for the peptides. The peptide posterior probabilities are better measures of correctly identifying peptides than the original probabilities.
- An output file with an extension of .prior which contains estimated even prior for proteins, used for iterative protein inference procedure [26]

By filtering the output, we got a set of most probable protein list which may present in the sample. This protein list is going to be our candidate solution.

3.3.3 Filtering

We analysed the generated output file. In the output file, following variables are contained, protein quantity, MAP state by memorizing, posterior decode state by memorizing, positive probability by memorizing, number of identified peptides of this protein, total peptides, number of proteins, probability of at least one protein exist. All these information is generated each time memorized approach converges. Actually we didnt need all this information.

We decided to take only those proteins whose MAP state by memorizing field is 1. The reason behind is that it is the MAP (maximum a posterior) solution; it provides the best combination of protein existence states to explain the data. In other words, it is telling us weather this protein exists or not. To extract these proteins we wrote a program which checks every field and generate the protein list as well as its corresponding peptides. The algorithm of filtering is illustrated in Figure 3.1.

3.3.4 Mapping

In the generated protein list, proteins name are given in the form of accession number used in the Swiss-Prot Database as entry. This accession number is the combination of letters and numbers, for example P10363. Peptides are represented as a sequence of amino acid for example, 'ABDFRGS..'.

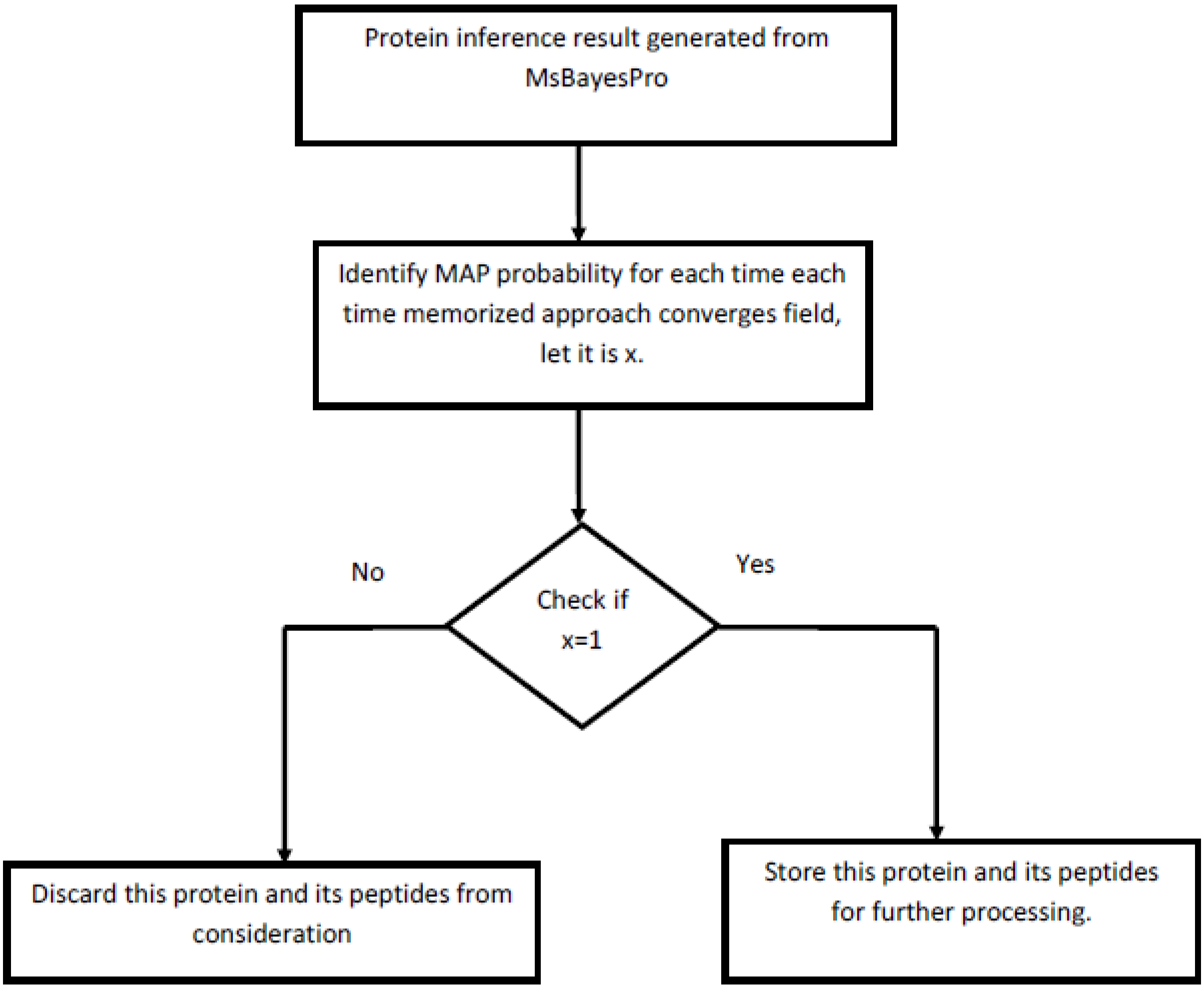

Protein inference result generated from MsBayesPro
Identify MAP probability for each time each time memorized approach converges field, let it is x.
No
Check if x=1
Yes
Discard this protein and its peptides from consideration
Store this protein and its peptides for further processing.

Figure 3.1: Process of filtering the full inference result of MsBayesPro

Now using this proteins and peptides name in this format is difficult. So we have mapped them. All proteins and peptides are given a unique integer number. Then protein and peptides are mapped by taking account of their relation. In order to map them, we had to write several programs. Language we used was C++ and Java.

3.3.5 Procedure of Providing Filtered Input in MAgPI

Implementation of MAgPI is done according to the Algorithm 1. The filtered and mapped protein-peptide list is used as database here. The test peptides as sample are given as input. MAgPI is generating candidate solution from our updated database which is now short in size compare with before and taking less time in searching process. Generation of candidate solution is done by the following steps,

1. From the test peptides list, peptides are taken.

2. Each peptide is searched in the database which we developed by the filtering and

mapping process.

3. The corresponding proteins are taken from database for each peptide.
4. By this, all peptides in the sample is searched and corresponding proteins are taken. This list of proteins is our desired candidate solution.

The rest of the part in MAgPI remains same as before. Here, we only made the change in the generation process of candidate proteins which is shown in Figure 3.2.

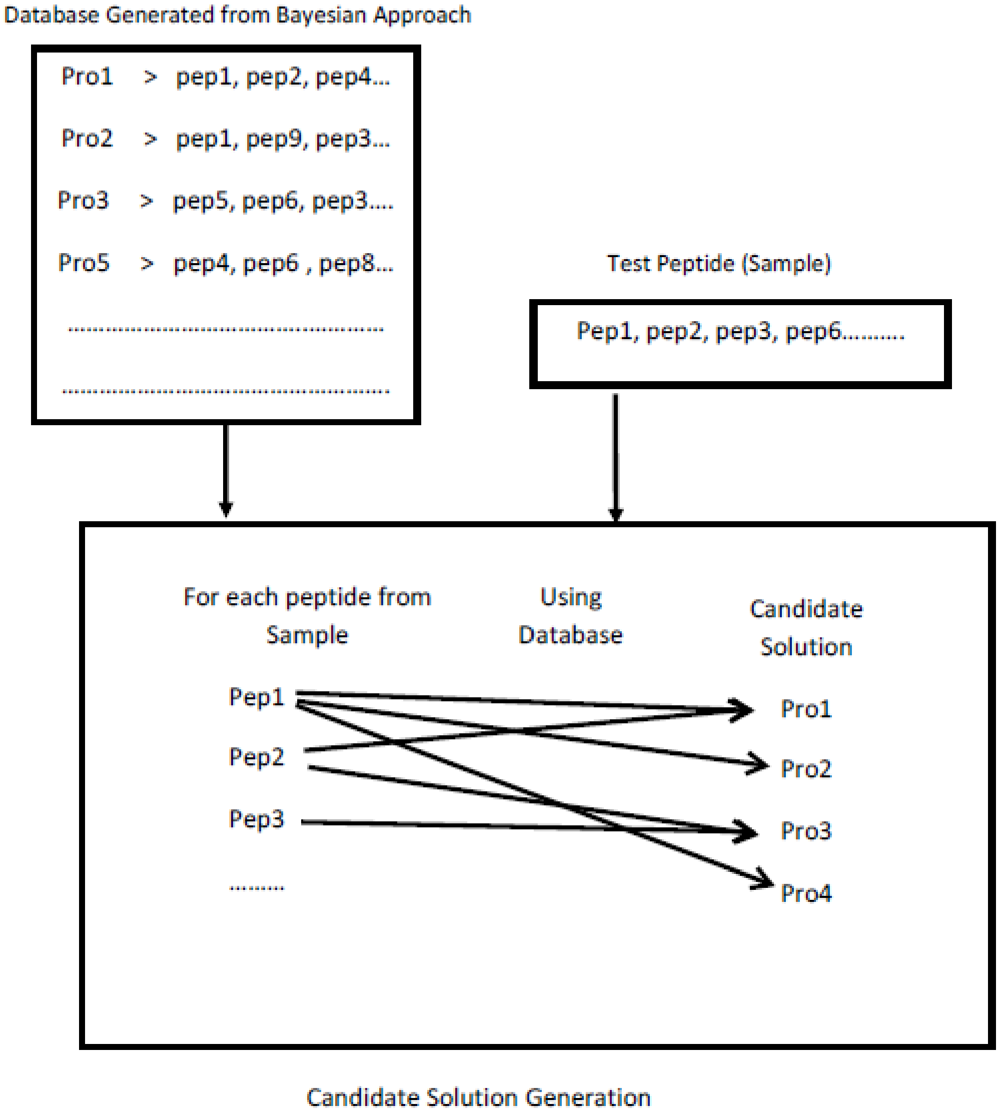
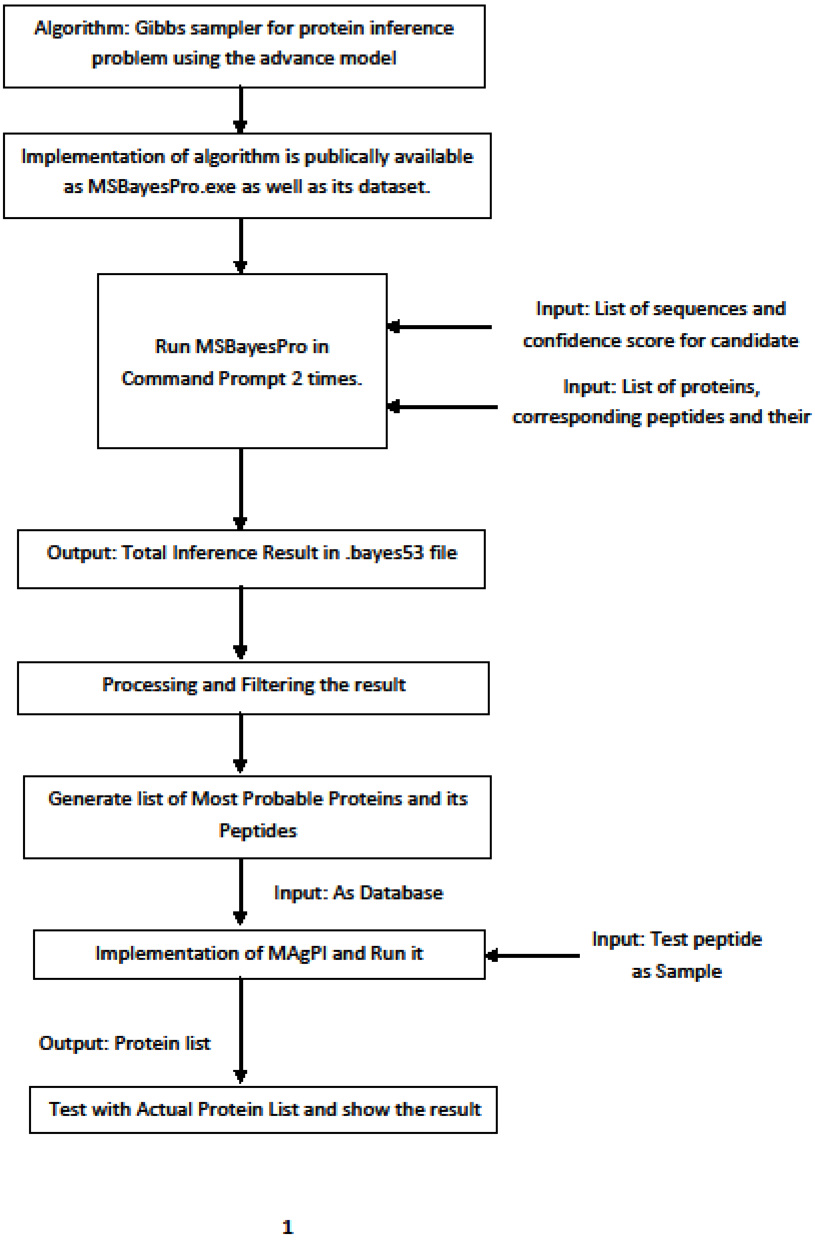

Database Generated from Bayesian Approach
Pro1 > pep1, pep2, pep4...
Pro2 > pep1, pep9, pep3...
Pro3 > pep5, pep6, pep3....
Pro5 > pep4, pep6 , pep8...
Test Peptide (Sample)
Pep1, pep2, pep3, pep6.........
For each peptide from Sample
Using Database
Candidate Solution
Pep1
Pep2
Pep3
Pro1
Pro2
Pro3
Pro4
Candidate Solution Generation

Figure 3.2: Generation of Candidate Solution

3.4 Overall Work Flow

The whole discussion of the previous section is represented by a flow chart in Figure 3.3.

:



Algorithm: Gibbs sampler for protein inference problem using the advance model
Implementation of algorithm is publically available as MSBayesPro.exe as well as its dataset.
Run MSBayesPro in Command Prompt 2 times.
Input: List of sequences and confidence score for candidate
Input: List of proteins, corresponding peptides and their
Output: Total Inference Result in .bayes53 file
Processing and Filtering the result
Generate list of Most Probable Proteins and its Peptides
Input: As Database
Implementation of MAgPI and Run it
Input: Test peptide as Sample
Output: Protein list
Test with Actual Protein List and show the result
1

Figure 3.3: Flowchart of the working procedure of CBM

CHAPTER 4
EXPERIMENT AND RESULT

After developing an approach, it is necessary to implement it practically so that we can get the experimental value to compare the approach with the other protein identification approaches.

4.1 Dataset

Dataset is same as used in Bayesian approach that is Sigma49 dataset which is a synthetic mixture of 49 standard human proteins. It was made available by Sigma Corporation for the assessment of protein analysis protocols. Among 49 proteins, 44 proteins contain at least one peptide that can be identified by shotgun proteomics. In addition, 9 keratin proteins and 4 other proteins are categorized as the keratin contamination and bonus proteins, and are believed to be present in the sample due to contamination. It also contains the detectability of the peptides to be present in that protein. The reason of using this dataset is mainly due to the fact that the currently available peptide detectability predictors can only handle tryptic peptides.

4.2 Experimental Setup

We ran our experiment with our filtered data on a core i-3 Intel micro-processor, 4 GB RAM machine using JAVA. We tried to infer 44 proteins of Sigma49 as the rest five proteins dont have a single peptide that can be identified by the Peptide Prophet search. For test peptide (sample) we have taken peptides of 44 sigma proteins which we got from MSBayesPro.

4.3 Performance Measures

After an approach has been developed, assessing its performance is still a problem. The most straightforward strategy for assessing the performance of different protein identification methods is to compare all the methods using same parameters.

The performance measures are mainly based on the following parameters:

1. True Positive (TP)

2. False Positive (FP)

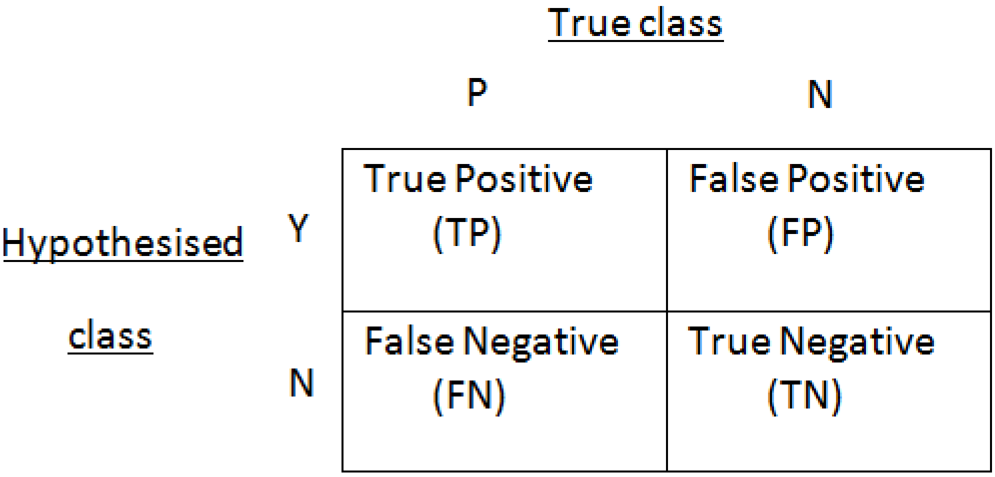
3. False Negative (FN)

4. Precision (Pr)

5. Recall (Rc) and

6. F-measure (F)

TP, FP, FN and TN are represented by a confusion matrix in Figure 4.1.



		True class	
		P	N
Hypothesised class	Y	True Positive (TP)	False Positive (FP)
	N	False Negative (FN)	True Negative (TN)

Figure 4.1: Confusion matrix. Figure has been borrowed from [5].

4.4 Comparison of Results

Results are compared among following approaches: Minimum Missed Peptide approach (MMP), ProteinProphet (PP), Basic Bayesian model (BB), Basic Bayesian model with detectability Adjustment (BBA), Advanced Bayesian model using raw PeptideProphet probabilities (ABP), ABP after detectability Adjustment (ABPA), Advanced Bayesian model using converted Probability scores (ABL), ABL after detectability adjustment (ABLA), ABLA with estimated protein prior probabilities (ABLAP) and Meta-heuristic approach (PIssGA), A Memetic Algorithm Based Approach in Protein Inference Problem (MAgPI) and our approach, Combining Bayesian and Meta-heuristic Approaches for the Protein Inference Problem(CBM).

Table 4.1 is showing the summary of our results with comparisons.

Table 4.1: Comparison of Results

	MMP	PP	BB	BBA	ABP	ABPA	ABL	ABLA	ABLAP	PIssGA	MAgPI	CBM
TP	39	41.5	39	37	35	43	37	44	43	40.5	39.62	40.61
FP	6	7.5	16	6	4	22	4	9	6	8	0.3	0.99
FN	5	2.5	5	7	9	1	7	0	1	3.5	4.38	3.39
Pr	0.87	0.85	0.71	0.86	0.9	0.66	0.9	0.83	0.88	0.84	0.99	0.98
Rc	0.89	0.94	0.89	0.84	0.8	0.98	0.84	1.0	0.98	0.92	0.90	0.92
F	0.88	0.89	0.79	0.85	0.84	0.79	0.87	0.91	0.92	0.88	0.94	0.95

4.5 Discussion

Our approach performed good among all the approaches in terms of F-measure. Also, in terms of precision our approach is better than all other approaches except MAgPI. True proteins number is not that good as ABPA, ABLA or ABLAP. The reason for this may be attributed to the errors introduced by the Gibbs Sampler. Recall that we are using the output of the Gibbs Sampler as the input of our second stage. As we are taking directly the output, some error is included in our system automatically. For example, after examine the filtering stage we found that in the most probable protein list which we are using as our candidate solutions, is missing 2 proteins of sigma49 proteins among 44 proteins. So actually we are giving 42 proteins as input in the MAgPI instead of 44. But we are evaluating our result out of 44 proteins. That is the reason of less true proteins. Our system beats the MAgPI in 4 aspects out of 6, where precision is very near with MAgPI. As, precision=TP/(FP+TP); only decreasing FP will solve the rest two. Our pre-processing of candidate solution, reduce the number of candidate solution m than MagPI. This set of candidate solution is working as database in our system and it is needed to search proteins very frequently. So reduction in size of candidate solution is reducing the computational time of MAgPI.

CHAPTER 5
CONCLUSION

In this paper, we have introduced a way of combining Meta-heuristic approach and Bayesian approach and measured its performance on Sigma49 dataset. The experimental results show that, our approach outperforms in several parameters among all of the existing protein inference methods. MAgPI computational time is also reduces because of filtered small size of candidate solutions. So far the result is promising. Now, as we said before Gibbs Sampler contains 42 true proteins (2 proteins are missing) and our system is evaluated for 44 proteins. We think this is the possible reason of higher FP or extra captured proteins. We did not implement Gibbs Sampler manually by ourselves, so we could not solve this error. Thats why our next target will be to implement the Gibbs Sampler manually to extract error free protein list from an intermediate stage of processing which will lead us to increase TP and decrease FP.

The idea of this two phase input refining system is new and we got good results from our approach. We hope that it will have a good use in the further research of solving protein inference problem.

REFERENCES

[1] T. Huang, J. Wang, W. Yu, and Z. He, "Protein inference: a review," *Briefings in bioinformatics*, 2012.

[2] Y. F. Li, R. J. Arnold, Y. Li, P. Radivojac, Q. Sheng, and H. Tang, "A bayesian approach to protein inference problem in shotgun proteomics," *Journal of Computational Biology*, vol. 16, no. 8, pp. 1183–1193, 2009.

[3] S. K. K. Santu, S. Rahman, S. Chakraborty, and M. S. Rahman, "Pissga: An ultra fast meta-heuristic approach to solve protein inference problem," in *International Conference on Computer and Information Technology (ICCIT), 2013*, 2013.

[4] S. Chakraborty and M. S. Rahman, "Magpi: A memetic algorithm based approach in protein inference problem." Provided by my thesis supervisor.

[5] T. Fawcett, "An introduction to roc analysis," *Pattern Recogn. Lett.*, vol. 27, pp. 861–874, June 2006.

[6] Wikipedia, "Shotgun proteomics — wikipedia, the free encyclopedia," 2014. [Online; accessed 15-December-2014].

[7] F. Wang, *Biomarker Methods in Drug Discovery and Development.* Springer, 2008.

[8] J. R. Yates, J. K. Eng, A. L. McCormack, and D. Schieltz, "Method to correlate tandem mass spectra of modified peptides to amino acid sequences in the protein database," *Anal. Chem.*, vol. 67, pp. 1426–1436, apr 1995.

[9] D. N. Perkins, D. J. C. Pappin, D. M. Creasy, and J. S. Cottrell, "Probability-based protein identification by searching sequence databases using mass spectrometry data," *Electrophoresis*, vol. 20, p. 35513567, dec 1999.

[10] R. Craig and R. C. Beavis, "Tandem: matching proteins with tandem mass spectra," *Bioinformatics*, vol. 20, no. 9, pp. 1466–1467, 2004.

[11] O. U. Press, *The Oxford American College Dictionary.* G.P. Putnam's Sons, 2002.

[12] Wikipedia, "Protein — wikipedia, the free encyclopedia," 2014. [Online; accessed 24-December-2014].

[13] Wikipedia, "Prior probability — wikipedia, the free encyclopedia," 2014. [Online; accessed 24-December-2014].

[14] S. Geman and D. Geman, “Stochastic relaxation, gibbs distributions, and the bayesian restoration of images,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 6, pp. 721–741, 1984.

[15] J. S. Liu, *Monte Carlo strategies in scientific computing*. springer, 2008.

[16] K. E. Nesvizhskii AI, Keller A, “A statistical model for identifying proteins by tandem mass spectrometry,” *Anal Chem*, 2003.

[17] A. R. Nesvizhskii AI, Vitek O, “Interpretation of shotgun proteomic data: the protein inference problem,” *Mole Cell Proteomics*, 2005.

[18] P. Alves, R. J. Arnold, M. V. Novotny, P. Radivojac, J. P. Reilly, and H. Tang, “Advancement in protein inference from shotgun proteomics using peptide detectability.,” in *Pacific Symposium on Biocomputing*, vol. 12, pp. 409–420, 2007.

[19] B. Zhang, M. C. Chambers, and D. L. Tabb, “Proteomic parsimony through bipartite graph analysis improves accuracy and transparency,” *Journal of proteome research*, vol. 6, no. 9, pp. 3549–3557, 2007.

[20] J. charles Boisson, L. Jourdan, E. ghazali Talbi, B. M. cit Scientifique, and C. Rolando, “A preliminary work on evolutionary identification,” in *of Protein Variants and New Proteins on Grids, in ”20th International Conference on Advance Information Networking and Application (AINA*, pp. 18–20.

[21] J. E. Elias and S. P. Gygi, “Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry,” *Nature methods*, vol. 4, no. 3, pp. 207–214, 2007.

[22] J. E. Elias, W. Haas, B. K. Faherty, and S. P. Gygi, “Comparative evaluation of mass spectrometry platforms used in large-scale proteomics investigations,” *Nature methods*, vol. 2, no. 9, pp. 667–675, 2005.

[23] A. Keller, A. I. Nesvizhskii, E. Kolker, and R. Aebersold, “Empirical statistical model to estimate the accuracy of peptide identifications made by ms/ms and database search,” *Analytical chemistry*, vol. 74, no. 20, pp. 5383–5392, 2002.

[24] M. Bern and D. Goldberg, “Improved ranking functions for protein and modification-site identifications,” *Journal of Computational Biology*, vol. 15, no. 7, pp. 705–719, 2008.

[25] H. Tang, R. J. Arnold, P. Alves, Z. Xun, D. E. Clemmer, M. V. Novotny, J. P. Reilly, and P. Radivojac, “A computational approach toward label-free protein quantification using predicted peptide detectability,” *Bioinformatics*, vol. 22, no. 14, pp. e481–e488, 2006.

[26] Y. F. Li, “Msbayespro @ONLINE,” dec 2014.

APPENDIX A
ALGORITHMS

A.1 MAgPI Algorithm

In Algorithm 1 shows about Memetic Algorithm Based Approach in Protein Inference (MAgPI).

Algorithm 1 Memetic Algorithm Based Approach in Protein Inference (MAgPI)

```
  Initialize population P(0) with $\lambda$ solutions
  $k \leftarrow 0$
  while $k < G_{max}$ do
    $Q(k) \leftarrow \phi$
    while $|Q(k)| < \mu$ do
      $a_1 \leftarrow RWSS(P(k))$
      $a_2 \leftarrow FUSS(P(k))$
      $[o_1, o_2] \leftarrow CROSSOVER(a_1, a_2)$
      $MUTATE(\mathrm{o}_1)$ with probability $\upsilon$
      $MUTATE(\mathrm{o}_2)$ with probability $\upsilon$
      $Q(k) \leftarrow Q(k) \cup \{o_1, o_2\}$
    end while
    L $\leftarrow$ Set of candidates for Offspring Education from P(k) $\cup$ Q(k)
    for $\forall l \in L$ do
      educate l by offspring education procedure
    end for
    $P(k+1) \leftarrow \lambda$ survivors from P(k) $\cup Q(k)$
    $k \leftarrow k+1$
  end while
  RWSS- Roulette Wheel Selection
  FUSS - Fitness Uniform Selection Scheme
```

A.2 Gibbs Sampling Algorithm

Algorithm 2 Gibbs sampler for protein inferencing using the advanced model

```
Input : Probabilities of correct peptide identification $(r_1, ..., r_n)$ and peptide detectabilities $d_{ij}$
Output : MAP protein configuration $(x_1, ..., x_m)$
Initialize $(x_1, ..., x_m)$ and $(y_1, ..., y_n)$ randomly ;
MaxPr ← 0 ;
while {
not convergence} do
    c ← a random number between 0 and t ;
    $(v_1, ..., v_c)$ ← a random c-block from $(1, ...., m)$ ;
    d ← t-c ;
    $(w_1, ..., w_d)$ ← a random d-block from $(1, ...., n)$ ;
    Compute normalizing factor T ← $(\frac{Value(x_1,...,x_m;y_1,...,y_n)}{F(x_{v_1},...,x_{v_c},y_{w_1},...,y_{w_d})})$ ;
    for all $(x_{v_1}, ..., x_{v_c})$ and $(y_{w_1}, ..., y_{w_d})$ do
        Compute $(x_{v_1}, ..., x_{v_c}; y_{w_1}, ..., y_{w_d})$ ;
        memorising : Value $(x_1, ..., x_m, y_1, ..., y_n)$ ← F*T ;
        if Value$(x_1, ..., x_m, y_1, ..., y_n)$ >MaxPr then
            MaxPr Value ← $(x_1, ..., x_m, y_1, ..., y_n)$ ;
            $(x_1^{Max}, ..., x_m^{Max})$ ← $(x_1, ..., x_m)$ ;
            $(x_{v_1}^{Max}, ...., x_{v_c}^{Max})$ ← $(x_{v_1}, ......x_{v_c})$ ;
            $(y_1^{Max}, ..., y_n^{Max})$ ← $(y_1, ..., y_n)$ ;
            $(y_{w_1}^{Max}, ...., y_{w_c}^{Max})$ ← $(y_{w_1}, ......y_{w_c})$ ;
        end if
    end for
    Sample $(x'_{v_1}, ......x'_{v_c}; y'_{w_1}, ......y'_{w_d})$ from normalized $F(x_{v_1}, ......x_{v_c}; y_{w_1}, ......y_{w_d})$ ;
    $(x_{v_1}, ......x_{v_c})$ ← $(x'_{v_1}, ......x'_{v_c})$ ;
    $(y_{w_1}, ......y_{w_d})$ ← $(y'_{w_1}, ......y'_{w_d})$ ;
end while
Report MaxPr, $(x_1^{Max}, ..., x_m^{Max})$ and compute marginal probabilities ;
```