Methods for analysing 2D electrophoretic gel images

Cand. Scient. Thesis
University of Bergen
Department of Informatics

Kristian Flikka

16th December 2002

Acknowledgements

The writing of this thesis has been guided by my adviser Ingvar Eidhammer, and many interesting discussions have contributed to the final result. For help on biological questions, Harald B. Jensen and Frode S. Berven have been of invaluable help.

I would also like to thank the study group from my early years at the University, which has contributed with numerous more or less meaningful discussions.

Finally, I direct my thanks to my family and my wife, Silje, for showing both encouragement and complete indifference, both crucial factors. Silje has also been of great help with regards to the English spelling and grammar.

Contents

1 Introduction **1**
1.1 Bioinformatics 1
1.2 Biological background 3
1.3 Thesis overview 7

2 2D gel matching **8**
2.1 Notation 9
2.2 Reproducibility 9
2.3 Normalisation 9
2.4 Reasons for errors and uncertainties 10
2.5 Feature extraction 11
2.6 Image matching, or image registration 11
2.7 Point-set matching 12
2.8 Pair-wise gel matching 13
2.9 Multiple gel matching 13

3 Algorithms for pair-wise image matching **15**
3.1 Using the raw image data to match gels 15
3.2 Using spot lists 15

4 Towards a multiple matching of 2D gels **24**
4.1 The goal 24
4.2 Definitions 25
4.3 Existing multiple gel matching methods 25
4.4 Inputs to the multiple matching 26
4.5 Output from the matching 28
4.6 Organising the data 29
4.7 Preparing the pair-wise match data 30
4.8 Extending the pair-wise algorithm 31
4.9 Time complexity of pair-wise comparing all gels 32

5 Progressive solutions to the multiple matching problem **34**
5.1 A progressive approach to multiple alignment 34
5.2 A method for including a new gel to a match-set 38

6 Using a graph model to perform multiple matching **40**
6.1 Defining inconsistency 40
6.2 A graph representation of the multiple matching 40
6.3 Formal definitions 41
6.4 Agenda 42
6.5 Constructing the SRG 43
6.6 Approach: Removing edges from the graph 43
6.7 Approach: Adding edges to an initially empty graph 56
6.8 Comparing the two main approaches 59
6.9 Algorithms 59

7 From sequence to 2D gel pseudo-image **61**
7.1 Introduction 61
7.2 Important aspects when going from a sequence to a protein spot 62
7.3 Automatic detection of modifications in gel images 64
7.4 Predicting the isoelectric focusing point from an amino acid sequence 65
7.5 Protein expression of predicted genes in a genome 67

8 Methodology for identification of proteins **73**
8.1 Calculating a complete "theoretical" 2D gel image 73
8.2 Using biological information to reduce / expand the theoretical 2D gel image 75
8.3 Using the theoretical image 76
8.4 Empirical testing of the synthetic 2D image 77
8.5 Synthetic gel in automatic matching 85
8.6 Improvements 85

9 Implementation **86**
9.1 File formats 86
9.2 Data abstractions 86
9.3 Algorithms 87
9.4 Loading a FASTA format sequence file 89

10 Conclusion **90**
10.1 Further work 90

Chapter 1

Introduction

An important group of biological macro-molecules are the proteins, often referred to as the workers of the cell. The proteins have many functions, they help in digestion (stomach enzymes), aid in movement (muscles), and play a part in our ability to see (the lens of our eyes is pure crytalline protein) [1]. Proteins are produced in the cells, but which proteins that are produced and at which amount, varies over time and with the changing conditions the cell finds itself under. The process of monitoring the produced protein amounts is called expression level monitoring.

In this thesis we will try develop methods for the biologist who wants to explore protein expression levels and identify proteins. The biological method explored for this purpose is 2D gel electrophoresis.

The writing of this thesis was initiated from a project called GABI (Gas and Biotechnology), which aims at studying a bacterium, *Methyloccus Capsulatus*. The proteins produced by this bacterium have been analysed with 2D gel electrophoresis and by different sequencing methods.

This chapter contains a short introduction to the field of bioinformatics and gives an overview of the biological principals most relevant for the thesis.

1.1 Bioinformatics

Bioinformatics is a relatively new branch in computer science, and can be described as a merge between biology and computer science.

The ultimate goal of the field is to enable the discovery of new biological insights as well as to create a global perspective from which unifying principles in biology can be acknowledged. There are three important sub-disciplines within bioinformatics [1] :

- The development of new algorithms and statistics with which to assess relationships among members of large data sets.
- The analysis and interpretation of various types of data including nucleotide and amino acid sequences, protein domains, and protein structures.

[1] http://pkr.sdsc.edu/html/pk_education/kerney/protein.html

- The development and implementation of tools that enable efficient access and management of different types of information.

Some reasons why bioinformatics is such a rapidly growing area in computer science as well as in biology are [1]:

- An explosive growth in the amount of biological information which necessitates the use of computers for information cataloguing and retrieval.
- A more global perspective in experimental design. As we move from the one scientist-one gene/protein/disease paradigm of the past to a consideration of whole organisms, we gain opportunities for new insights into health and disease.
- Data-mining - the process by which testable hypotheses are generated regarding the function or structure of a gene or protein of interest by identifying similar sequences in better characterised organisms. For example, new insight into the molecular basis of a disease may come from investigating the function of homologous of the disease gene in model organisms. Equally exciting is the potential for uncovering phylogenetic relationships and evolutionary patterns.

1.1.1 The complexity of bioinformatics problems

When considering the running time of any algorithm, one often speaks about the running time as a function of the size of the input. The size or the length of the input is often referred to as n. If an algorithm has a running time that is polynomial in n, it may in many cases be classified as a "fast" algorithm. More formally we can define a class of problems, P, which is the collection of problems that *can* be solved in polynomial time.

Another group of problems is called NP (**N**ondeterministic **P**olynomial, can be solved in polynomial time with non-deterministic polynomial time Turing Machine). Given a solution to such a problem, this solution can be verified in polynomial time. It is not known whether $P \subseteq NP$ or if $P = NP$.

We may then look at problems in NP that are the most "difficult" to solve. These are called $NP - complete$, and if such a problem should turn out to be solvable in polynomial time, then $P = NP$.

Many problems that one runs into are, in fact $NP - complete$. These problems have no known polynomial time solution, and takes thus (too) long time to solve precisely. Examples of such problems are:

- The prediction of the three dimensional folding of a protein, given its amino acid sequence.
- Multiple string matching.

As a consequence, people seeking solutions to these problems, tend to develop *heuristic* methods. In short terms, a heuristic method is a method that calculates a solution based on some "expert" knowledge. Such a method is not guaranteed to produce the right answer. It may not be proved to perform with

[1] NCBI Education, http://www.ncbi.nlm.nih.gov/Education/

a certain probability of giving the correct answer either. Nevertheless, it may perform adequately good to be interesting. And most important, it "solves" the problem in a reasonable amount of time.

1.2 Biological background

Molecular biology is the study of gene structure and function at the molecular level. This section reviews the most central concepts of molecular biology.

1.2.1 The macromolecules

The basis of molecular biology is the biological macromolecules:

- **D**eoxyribo**N**ucleic **A**cid - DNA
- **R**ibo**N**ucleic **A**cid - RNA
- Proteins

DNA molecules are the largest covalent molecules in organisms. The DNA molecule consists of four different deoxyribonucleotides, **Adenine, Guanine, Thymine and Cytosine**. The nucleotide bases form a chain, covalently joined together by phosphodiester linkages, in a double-stranded helix. Long chains of nucleotides are called polynucleotides, and short chains of nucleotides are called oligonucleotides. The bases are complementary, i.e. if you know the base on one of the strands, then the other is known. Adenine (A) is complementary to Thymine (T), and Guanine (G) is complementary to Cytosine (C).

RNA molecules are built up by four different ribonucleotides. They are the same as for the DNA, except that the Thymine is replaced by Uracil. The RNA molecules have a single stranded structure, and are much shorter than the DNA molecules. There are three main classes of RNA, messenger RNA (m-RNA), transfer RNA (t-RNA), and ribosomal RNA (r-RNA). They are involved in converting DNA sequence information into proteins.

Both RNA and DNA has a polarity and the potential to form base pairs with a complementary strand in an anti-parallel orientation. This is called hybridisation, and is used when the DNA sequence is "read", see section 1.2.2.

Proteins are linear chains of amino acids. There are 20 standard amino acids, and each has a specific side group. Peptide bonds link amino acids together to form polypeptides or proteins. Polypeptides have a defined sequence; that is, the arrangement of the amino acids in a given protein is in a specific order along the polypeptide chain. Proteins are the "workers" of the cell. Some catalyse reactions, some provide structural integrity to the cell, some control gene expression by binding DNA or RNA and some are involved in synthesis of other proteins.

1.2.2 The central dogma of molecular biology

This dogma explains that the sequence of a strand of DNA corresponds to the amino acid sequence of a protein. The DNA sequence consists of many genes. A gene is a part of the DNA sequence that codes for one or several proteins. A "gene-part" of the sequence is recognised by a specific promotor, followed by

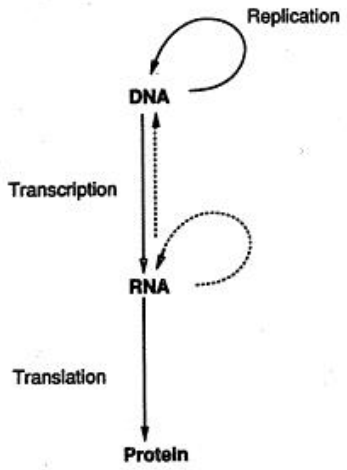


Figure 1.1: The central dogma of molecular biology. The dogma explains the main processes of molecular biology.

several exons and introns. The exons are the expressed parts of the gene, and the introns are unexpressed parts.

The transcription process in Figure 1.1 is the transcription of a DNA sequence into RNA. During this process, the introns are kept, but they are removed before the translation process. This is called splicing. The final RNA product, after the transcription, therefore only consists of exons. This is the mRNA.

The translation process translates the mRNA strand into protein. This is the protein synthesis. Three ribonucleids are read at a time, and are translated into an amino acid. This coding system is explained in Figure 1.2. As one can see, the system is redundant. This decreases the probability that a mutation in the DNA sequence affects the protein product. Because of the splicing process of the RNA and post-translational modifications to the proteins, the step from RNA to protein may produce different proteins from one gene. So one gene does not necessarily result in just one particular polypeptide [1], which used to be the theory in the early years of functional genomics.

1.2.3 Functional genomics and proteomics

The word *genome* is used to describe the complete genetic information contained in an organism. The word proteome is a similar term, and refers to the *PROTE*ins expressed by the gen*OME* [1]. That means that the proteome of an organism is the complete expressed protein result in a (part of the) cell, at any given time. That leads to the most significant difference between the genome and the proteome: The proteome is not a fixed feature of an organism, but changes with the state of development, the tissue, or even the environmental conditions under which an organism finds itself [1]. This leads to the situation that most molecular biologists acknowledge, namely that an organism has one genome, but several proteomes.

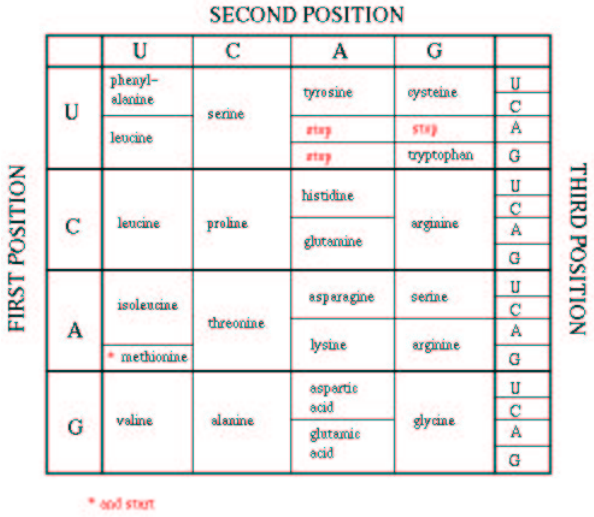


Figure 1.2: The genetic code. This system is used to decode nucleotide sequences into proteins.

Proteomics is the study of an organism's proteome, i.e. a global analysis of the protein activities. In practise this includes identification of proteins, and the study of expression levels and post-translational modifications. Post - translational modifications is the situation where a protein is modified *after* the translation from RNA to protein. As the different genome projects finishes, the proteomics area emerges as a necessary continuation of these projects. Proteomics is necessary in order to understand the function of the genes and their coherence.

It is important to emphasise that the relation between a gene and the resulting protein is quite complex. Both the up and down regulation of mRNA vs. the levels of protein, and the actual protein results are quite complex.

To perform global analysis of the protein activities, one needs to be able to separate the proteins involved, and ideally be able to monitor the expression levels over a period of time.

1.2.3.1 2D gel electrophoresis

An important technique to explore the proteome, is two dimensional electrophoresis. This technique separates the proteins of a prepared sample into two dimensions. 2D gel electrophoresis was first introduced in the early 1970's, and is considered to be the only method available which is capable of simultaneously separating thousands of proteins [1]. Recently, the micro-array technology has been used to identify expressed proteins [2]. This will be discussed in Section 1.2.3.2.

As mentioned above, the proteins are separated in two dimensions. The first dimension is the isoelectric focusing (IEF). Here, the proteins are separated in a pH gradient until they reach a stationary position where their net charge is zero. The pH at which a protein has zero net charge is called its

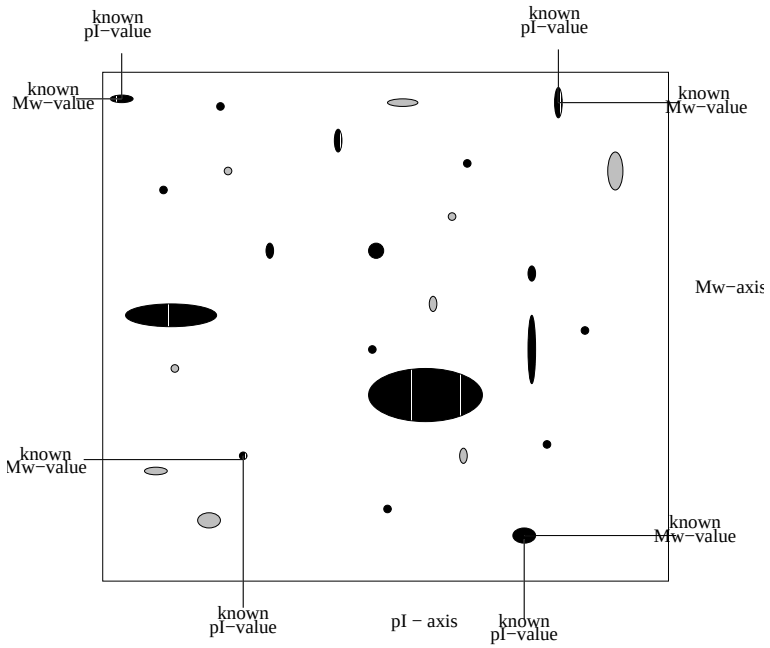


Figure 1.3: Schematic example of a 2D gel.

isoelectric point (pI). In the second dimension the proteins separated by IEF are separated orthogonally by electrophoresis in the presence of sodium dodecyl sulphate (SDS-PAGE). When the SDS-coated proteins migrate in a sieving gel, they separate based on their molecular mass (Mw).

When 2D gel electrophoresis is performed, a visual image is not obtained immediately. The proteins on the gel must be stained in order to be visually perceivable. The most common way of staining the proteins is silver staining. There are disadvantages with this approach, and these will be discussed in Chapter 2.4.

When the gel is stained, one obtains a visual image of how the proteins have been separated. The gel may now be digitalised via a scanner, and represented in a computer as a bitmap image. It is this image that is processed later in the process. The scanning of the gel should be of high enough resolution to be able to pick up all the useful information on the gel.

If a set of proteins with known pI and Mw values are added to the sample before the electrophoresis, their values can be used to interpolate the values of the other proteins, see Figure 1.3. These interpolated values are often called apparent values.

Each of the spots on the gel represent one, or possibly several proteins. Commercial software designed for this purpose can separate, quantify and mark the spots with coordinates. This is sometimes referred to as feature extraction, and the feature is usually the centre-point coordinate of each spot, and a spot intensity value.

1.2.3.2 Using micro-array technology in proteomics

Traditional methods for monitoring gene products generally works on one gene per experiment. A new technology called DNA micro-array has made it possible to monitor the whole genome on a single chip. Base-pairing, or hybridisation is the underlining principle of DNA micro-array. The idea is to probe the array with different gene products. Usually c-DNA or oligonucleotides are used. The chip is now ready to be used. To explore the expressed genes in a sample, the sample is flushed over the chip. Because of the hybridisation property of the c-DNA, one can measure on which probes there have been reactions. If there is a reaction on a probe, that means that the gene represented by the probe is expressed in the sample.

A scheme for conducting micro-array experiments with proteins is described in [2]. In the same way as for the gene arrays, proteins are probed on the chips. This was done by cloning 5800 open reading frames, and over-expressing their corresponding proteins. Proteins do not have the hybridisation feature, so one has to find a way to make the proteins on the probes react with the ones in the sample (target), a special binding motif. The common way of doing this, is to produce antibodies for each protein, which is an expensive and time consuming process if many proteins are used as probes. This problem makes the protein array quite different from the elegant DNA micro-array technology, using the base-paring ability of the nucleotides. Another issue is whether the protein array is capable of detecting post-translationally modified proteins or not.

The fact that producing antibodies for the probes on the array is extremely expensive, together with the difficulty of detecting modifications, have resulted in a modest use of protein arrays up to this point. 2D gel electrophoresis is still the main method for monitoring protein expression levels.

1.3 Thesis overview

This chapter contains a short introduction to the thesis, and gives an introduction to bioinformatics and relevant biological theory and methods. In Chapter 2, general information and motivation regarding 2D gel matching is given. In addition, theory concerning general feature matching is also presented. Existing matching methods are revised in Chapter 3. Important factors when extending the pair-wise matching towards multiple matching is addressed in Chapter 4. In Chapter 5 progressive solutions to multiple gel matching are suggested, while Chapter 6 suggests a graph model approach. A different perspective on the 2D gel issues is explored in Chapter 7 and 8, where the gene sequences of an organism are used to create a synthetic 2D gel image. Chapter 9 explains some implementation details, while some conclusions are drawn in Chapter 10.

Chapter 2

2D gel matching

To study the functions of different proteins, and their connection to each other, it is interesting to compare different images of electrophoretic gels. These images can for example be of several gels showing different states of an organism. These different states may be caused by different growth conditions, temperature variations etc. In the GABI project it is in particular the effect of changes in Cu^+ concentrations that are interesting.

After performing 2D gel analysis on a protein sample, one often wishes to compare this gel with one or several other gels to look for changes in expression. Typical is that a protein either is monitored for on/off switching, or one can monitor a selection of the spots for more global changes in expression levels. In order to compare different protein expression levels from one gel to another, the spot/protein quantities must be normalised. This will be discussed in Section 2.3. The reasons why normalisation is needed together with sources for other errors will be discussed in Section 2.4.

Usually, we want to either match two gels, or as in most cases, many gels. To do this automatically, efficient algorithms are needed. These algorithms can be based either on the raw image, this is probably the best solution because the raw image contains most information, or on a list of spots with one or several features. These features can for example be the centre-coordinate of a spot. The latter approach is easier and less computationally complex. But in [7] a procedure for using the raw image to match gels, is proposed. This procedure is explained more in Chapter 3.1.

The main difference between 2D gel matching, and other bioinformatics matching problems, like string or structure matching, is that when matching 2D gels there is a uniquely correct answer. For string and structure matching, a "good" alignment is often the aim for the matching. For example, mismatching characters in a string comparison may get a positive score according to a scoring matrix. The spots on the 2D gel represent proteins, and should ideally be uniquely matched to a corresponding protein on another gel. This makes the problem of matching 2D gels a yes/no problem (two spot match, or not), while string matching is a problem involving approximative matches.

2.1 Notation

To keep track of the different parts involved in the different matching procedures, the following notation is introduced.

- The proteins that we wish to find in the gels are denoted $P_1, P_2, ..., P_m$, where P_i is protein number i in our list. This list is not predefined, but may be generated from a main experiment or as the matching goes along.
- The lists of spots, corresponding to gels, will be denoted $G_1, G_2, ..., G_n$, where G_i is the spot list for gel number i.
- Spot number j inside G_i will be denoted g_i^j
- When two or more gels are matched, we search for corresponding spots on the different gels. The result of a matching between these gels is called a *match set*, and will be denoted by M.

2.2 Reproducibility

The main problem with expression monitoring by using electrophoresis, is the failing reproducibility. When running several equal experiments, and monitoring a protein's spot characteristics, it is not always for certain that the protein appears at the same location and with the same intensity each time.

In [11] an extensive experiment was performed to explore the reproducibility of 2D gel experiments. 49 gels were cast, run, and stained in parallel. The average number of spots on the gels were 2170. These gels should ideally be identical. The gel matching program that was used, was based on using one gel as a master gel, and comparing all other gels to this one. The match score between two arbitrary gels is defined as the percent of the spots on the gels that have been assigned a corresponding spot on the other gel. The average match score in the pair-wise matching between the master gel and all other gels turned out to be $89 \pm 4\%$ for these 49 gels.

Afterwards a multiple matching was performed. The results presented is a measure of how many spots that were recognised on all gels. Looking at all 49 gels, only 8 spots were recognised on all gels. If looking at number of spots matched in at least 40 gels, this number was 192, or 8.9%. For spots matched in at least 25 gels, the number was around 700, and still far less then half of the spots were matched.

If the results described here are representative for the programs available, large scale protein expression profiling is currently very difficult to perform automatically. The amount of data will probably be too small. These observations indicate of that it should be possible to improve the performance of the methods concerning multiple gel matching. In one way or another, it should be possible to move away from the "all gels toward a master gel" approach. The individual equalities between *all* the gels should be utilised better.

2.3 Normalisation

When comparing the intensity of a protein spot in different gels, one needs to normalise the values. This is so because even if two corresponding spots have dif-

ferent intensities, it does not mean that they are expressed differently. In other words, equally expressed proteins may appear differently. This phenomenon has several reasons, and these reasons will be discussed in Section 2.4.

To compensate for the non-expression related variations, each spot on a gel may be assigned a new intensity value. This new value will depend on the intensities of the other spots on the gel. The actual new value may be calculated in many different ways, but variations over this formula will be the basis:

$$\text{New Intensity} = \text{Old Intensity}/\text{Total Intensity} \tag{2.1}$$

So the normalised spot intensity is a value relative to the intensities of the other spots on a gel. In this way, a difference in the intensity level for one protein spot in one gel to another gel is more likely to be caused by actual changes in level of expression.

2.4 Reasons for errors and uncertainties

One of the main problems with 2D gel matching is the fact that up to date one has not been able to propose an assay that creates reproducible results with a quantification method that linearly corresponds to protein amount.

2.4.1 Errors concerning protein quantification

As mentioned earlier, to be able to visually see and quantify a protein spot, it has to be stained. There are several aspects of the silver staining method that limit its use when aiming at overall protein quantification. Some of the problems concerning silver staining are [1]:

- Non-linear relationship between protein concentration and intensity. The silver staining method gives some linearity but not over large differences in protein amount.
- The relationship between silver staining density and protein concentration is characteristic for each protein, meaning that while some proteins may be stained linearly with the concentration, others may not.

Other methods than silver staining also exist. SYPRO Ruby protein gel stain shows a greater linear range than silver staining. Coomassie Brilliant Blue is another staining method, perhaps the one most frequently used. Staining proteins fluorescently is also possible.

Silver staining may in some cases detect DNA, lipopolysaccharides and polysaccharides [17]. This means that a stain on the gel may not necessarily indicate a protein.

2.4.2 Errors leading to geometric distortions

Different characteristics of the gels, for example unevenly distributed perforations, tears and cracks, may all lead to geometric distortions on a gel.

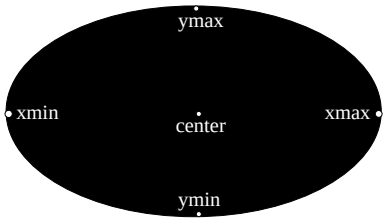


Figure 2.1: Suggestion to feature extraction. By not only extracting the centre-coordinate, but four other values, the characteristics of each spot may be better represented.

2.5 Feature extraction

One or several features are usually extracted from the images. Perhaps a solution with more than one coordinate per spot as feature can be successful. For instance five points per protein; xmax, xmin, ymax, ymin and centre-point. See Figure 2.1.

By using the values of the five points, one can conserve some of the information from the raw image. This information can be used as a parameter when deciding whether two spots match or not. It will of course only be a factor in the matching, a protein may very well change in shape from one gel to another. It will probably be convenient to store the four extreme points as deviations from the centre coordinate, in stead of the real coordinates of the four points.

To obtain the coordinate values one has to look at the raw image data. This will not be done in this thesis, but may very well be a possible approach, laying between the traditional methods, and the one described in Chapter 3.1.

2.6 Image matching, or image registration

As stated in the introduction to this chapter, there are two main approaches to the problem of matching possibly distorted computer images. The most computer intensive is array processing techniques, i.e. processing of the raw image data, converted to a colour/intensity array. Arrays from different images are compared directly, which on large images is a slow process. When the number of "interesting" elements in an image is low, relative to the image size, it is often more sensible to extract the interesting features from the image, and then match the feature lists. The elements, or features of an image may for example be buildings, protein spots, micro-array spots, people etc., all depending on the context.

Thus there are two main tasks:

- Recognise the spots in a computer image. This consists of detection of areas where the pixel intensities indicate spot locations.
- Match corresponding spots to each other.

These tasks may be performed sequentially, or simultaneously. The sequential approach first apply image processing techniques to retrieve a list of spots, and then performs a matching of the lists. The simultaneous approach uses

the raw image intensity values to match different images together, while also detecting spot coordinates.

2.7 Point-set matching

The basics of almost any point-based registration scheme involves matching feature points extracted from a sensed image toward their corresponding points in another image. The extraction of features may in some cases decrease the information amount available for the matching. This is the case if the points to be matched may have characteristic shapes, colours etc. In these cases it will be difficult to preserve all the information if the image is converted to a list of coordinates. To increase the information preserved, one can, in addition to the point coordinates, store information about for example shape, colour or intensity.

2.7.1 Point patterns

A name that is often used for the extracted coordinates from an image is a point pattern. A point pattern may be defined as a non-empty, finite set of points on a two dimensional plane. Each point in a point pattern P is called p_i. Each point has either only coordinate values (x_i, y_i), or in addition a label, so that $p_i = (x_i, y_i, l_i)$. The label may be information about size, shape, colour, intensity, etc.

2.7.2 Point pattern matching

There exists a variety of methods for point pattern matching (PPM). Usually the different techniques vary depending on the applications that inspired their development.

2.7.2.1 Traditional approaches to point pattern matching [12]

This section briefly explains some of the main approaches to general point pattern matching.

- Clustering
- Inter point distance algorithms
- Relaxation methods
- Other methods not falling into any above category

Clustering methods assume that two point patterns differ by one or more transformations. The transformation parameters for each possible pair of points are calculated. Then, for each such combination of pairs, the transformation that best aligns all the pairs is recorded. The transformations (or their parameters) are clustered, and the strongest cluster is chosen. Most often some combinations of pairs are disregarded, because if there are many points, it will be too ineffective to try all different pairs. Pre-knowledge about the maximal discrepancies may be helpful in limiting the number of different pairs.

Inter point distance algorithms utilise the distance between the points in an image in different ways. The inter point distances in one image are compared to the inter point distances in the other image. The distances may be represented as vectors, or as elements in a graph model of the point pattern. In addition to inter point distances one may also examine the differences in inter point vector angles.

Relaxation methods generally assign values to similar objects from different images in an iterative way. Let for example an object be a pair of points from two different images. The values of the objects (points) p_i, q_k may for example reflect how well the other points match when p_i is matched onto q_k. The idea is that over a series of iterations the values of the objects will converge. In each iteration a criterion is used to re-consider the assignment of spot pairs, and after the iteration, the results are examined, and if the results converge, the iterations are terminated. Converging will in this case mean that the global score of the matching is not improved by altering the objects, for example by letting p_i match q_l instead of last iteration's assignment of p_i to q_k.

2.8 Pair-wise gel matching

In order to compare protein expression levels from different gels, it is crucial to correctly identify corresponding spots on the various gels, i.e. recognise a specific protein on different gels. The corresponding spots may not exist, if the production of this protein has stopped for some reason, or an error has occurred. See Section 2.4.

Because of experimental errors and geometric distortion, it is not trivial to find the corresponding spots from one gel to another. The obvious solution of simply putting one gel on top of the other, and look for overlapping spots, is far from realistic.

Computer procedures to automatically perform this matching have been developed, and some of them are reviewed in Chapter 3. The first idea which was used, was to perform one linear global transformation on one of the gels, and look for close or overlapping spots. This is still a common way of aligning images in other applications. The global transformation may be found on the basis of several user-specified landmarks. Landmark setting implies that the user selects pairs with one spot from one of the gels and the other spot from the other gel. This pair is then assumed to be the same, corresponding protein in both gels.

The 2D electropherograms are full of geometric distortion and local skews. A single global transformation is therefore not enough to assign the corresponding spots to each other. This is the reason why the most successful approaches compare local descriptions of the gel, looking for local, conserved patterns on the gels. These descriptions come in many variants, and will be described in Chapter 3.

2.9 Multiple gel matching

In the proteomics research area, one wishes to examine the protein expression levels over a *series* of gels. Thus, we ideally seek a "complete" multiple alignment

of the gels. A popular approach to the multiple alignment, is to choose one gel as a reference gel, and then pair-wise compare all the others to this one. This is a reasonable approach, but it has its traps and disadvantages. The main disadvantage is that the result of the "pseudo-multiple" matching relies on the choice of reference gel. If the reference gel is missing some of the protein-spots that exist on other gels, these spots may never be detected.

A similar approach generates a reference gel from all the gels, a kind of consensus gel, and then pair-wise compares the involved gels to this.

Chapter 3

Algorithms for pair-wise image matching

A variety of different methods have been explored to perform matching of 2D images, and in our case images of electrophoretic gels. Common for the majority of these methods, is that they focus only on a *pair-wise* comparison. These methods use one gel as the master gel, and compare the other(s) to this master. The gels that are compared to the master gel are called *matched* gels. Mainly methods for gel matching are considered here, but as a comparison, a method for matching star lists is included as well.

All the methods described in this chapter are solutions to the problem of matching two gels.

3.1 Using the raw image data to match gels

In [7] a new approach is introduced. The main difference between this method and the traditional ones, is that the basis of the matching procedure is the raw image, and not a list of extracted centre-coordinates. It is argued in [7] that the main reason why most registration procedures for 2D gel images fail, is that they are based on a too small amount of information. That means that the common way of registering images, neglects a lot of useful information. The work-flow of the traditional approach is shown in Figure 3.1.

Further, the approach is based on a general observation that the confidence of a match is improved if the algorithm is using a spot region by spot region comparison, instead of the spot by spot comparison. It is worth noticing that this observation is also used in some of the feature matching procedures, for example the one described in Section 3.2.1.1.

3.2 Using spot lists

Using lists of spot-coordinates is a common way of comparing two gel images. In this section some of the algorithms used for matching spot-lists are explained. A method that is applied to another problem, the problem of matching coordinate list of stars, is also explained briefly.

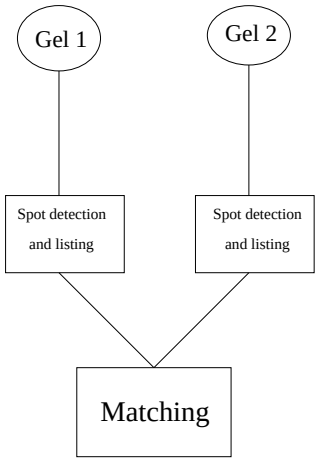


Figure 3.1: The traditional work-flow. The spots are first detected, and then extracted as a list of spot coordinates.

It is a common observation that matching local areas of the gels to each other increases the reliability of the overall matching. This is sometimes realised by defining the neighbourhood of each spot. The neighbourhood of a spot may be defined by for example the geometric configuration of a selection of the closest spots. It is often sufficient to use the 10 closest spots to create a unique neighbourhood.

3.2.1 Matching point-patterns using local neighbourhood similarities.

A common approach when using local point-patterns to match two images, is to first manually assign some landmark positions on both images. Landmarks are points that with a high probability are "real" matches. Based on these landmarks, a global transformation is performed. The purpose of this transformation is to roughly align the two images.

The next step is then, for each point on the master image, to find the corresponding point on the matched image. There are several ways to approach this. A successful approach is to define the spot by its neighbourhood, usually the closest spots, and then for each spot in the master image compare its neighbourhood to the neighbourhood of some of, or all of the points in the matched image.

It is worth noticing that when matching a spot in the master image against the matched image, there might be multiple matches in the matched image, and a choice is made between the different possible matches, based on a score-procedure. A better approach is too keep possible match candidates for later consideration. Figure 3.2 gives an intuitive impression of this.

3.2.1.1 An algorithm using local neighbourhood matching

This section explains the implementation of the local neighbourhood matching described in [4]. This method uses a list of spots with the corresponding

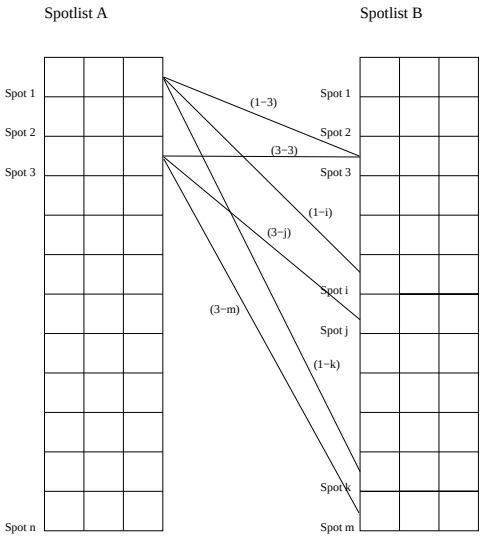


Figure 3.2: Shows the idea of keeping multiple matches. Each row corresponds to a spot. An edge from one entry to another symbolises a match. The different edges have weights corresponding to the goodness of the match of the edges' spots.

coordinates and intensity. The coordinate of a spot is the spot's centre-point.

The first step is to acquire a global transformation that transforms the matched gel onto the master gel. This is done to eliminate global distortion. By choosing landmarks on both gels and searching for a transformation aligning these, a global transformation is found. The polynomial transformation is used, because it is sufficiently accurate.

Now the gels are roughly aligned, but the first global transformation is not accurate enough to match the spots by simply looking at overlapping spots with equal coordinates.

The next step is to find, for a spot C_m in the master gel, the corresponding spot C_c in the matched gel. For one master spot, C_m, one needs to consider the closest spot in euclidean distance, this spot is called C_c^*. Thereafter, one compares C_m to C_c^*. In addition, a limited number of C_c^*'s neighbours are chosen to act as possible C_c. After choosing a large enough number of spots to act as possible C_cs, each of the candidate's matching score against the spot C_m on the master gel is evaluated. A decision is then made about which spot to choose.

The scoring of a match is found by using a composite criterion. The first criterion is measured by constructing a vector V between the match candidate, C_c, and the master spot, C_m. The length and angle is then compared to the length and angle of the vector between the closest landmarks in the two gels. See Figure 3.3.

A second criterion is used to compare the neighbourhood of each candidate C_c to the master spot's neighbourhood.

To construct and compare the neighbourhood of a spot, the area around the spot is sliced up into different smaller areas. These areas are all assigned a

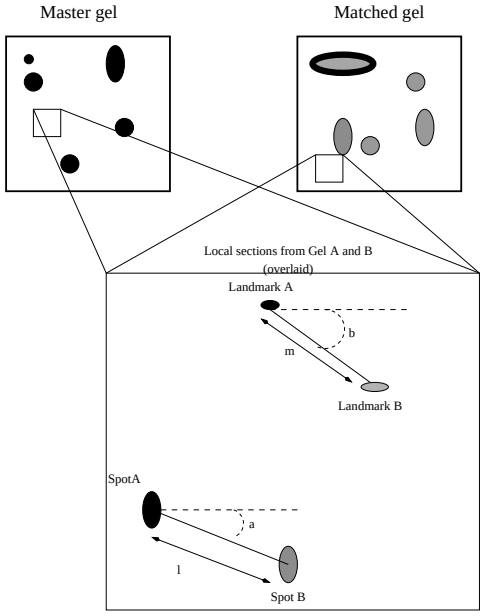


Figure 3.3: Shows the comparison of the vector between the candidates, and the vector between the closest landmarks. The angles a and b are compared, so are the lengths l and k.

unique area-code. The area-codes are defined with the current candidate spot as centre. The area-codes differ in at least one character, and are used to describe the neighbourhood of the spot. This is done in the following way:

To construct the so-called neighbourhood descriptor (ND) of a spot, C, a matrix that represents the neighbourhood is created. This matrix is of size $(n+1)*(n+1)$, where n is the number of points in the neighbourhood.

$$\begin{pmatrix} X & C \rightarrow 1 & ... & C \rightarrow n \\ 1 \rightarrow C & X & ... & 1 \rightarrow n \\ . & . & . & . \\ . & . & . & . \\ n \rightarrow C & n \rightarrow 1 & ... & X \end{pmatrix}$$

The matrix is calculated, and an entry, C $\rightarrow$ 2 is calculated by using spot C as centre, and find the area-code of 2 relative to C. The lower triangular matrix contains equivalent information to the upper, for example entry $C \rightarrow 1$ bears equivalent information to $1 \rightarrow C$. The matrix is constructed for all the C_c candidates.

Next, a score matrix is constructed. The aim of this matrix is to find which of the points in C_m's neighbourhood that fall into the same area as C_c's neighbour points. The two neighbourhoods may have different size, so the maximum number of corresponding spots, N, is limited by the smallest neighbourhood.

The scoring matrix is now computed.

$$\begin{pmatrix} (C_m \rightarrow 1_m) + (C_c \rightarrow 1_c) & (C_m \rightarrow 1_m) + (C_c \rightarrow 2_c) & ... & (C_m \rightarrow 1_m) + (C_c \rightarrow n_c) \\ . & . & . & . \\ . & . & . & . \\ (C_m \rightarrow n_m) + (C_c \rightarrow 1_c) & (C_m \rightarrow n_m) + (C_c \rightarrow 2_c) & ... & (C_m \rightarrow n_m) + (C_c \rightarrow n_c) \end{pmatrix}$$

The + denotes the similarity between the two arguments. The result of the + operation is the number of equal symbols in their area code. The term $(C_m \rightarrow 1_m) + (C_c \rightarrow 1_c)$ thus results in the number of equal symbols in their area-code. For example, if $(C_c \rightarrow 1_c) + (C_m \rightarrow 3_m) = 3$, spot 1_c and 1_m have three common characters in their area-code, relative to their respective centre-spots. The following is an example of a scoring matrix with 4 spots in the master gel's neighbourhood, and 3 spots in the matched gels neighbourhood:

$$\begin{pmatrix} 0 & 4 & 1 \\ 4 & 1 & 2 \\ 1 & 0 & 3 \\ 2 & 0 & 1 \end{pmatrix}$$

Since the numbering of the spots is not necessarily equal in the two gels, the N highest scores are chosen to be the corresponding spots. These N scores are summed up to give a score between the neighbourhood of C_m and C_c. In the example matrix, N is equal to 3, because this is the size of the smallest neighbourhood. The corresponding spots are in this case found by examining the columns (because this is the smallest dimension, and thus limits the number of corresponding spots). The results are that with the selected spots C_m and C_c as neighbourhood centres, 1_m corresponds to 2_c, 2_m corresponds to 1_c, and 3_m corresponds to 3_c. Spot 4_m has apparently no corresponding spot in the matched gel's neighbourhood.

To decide whether a spot C_c matches C_m the last score is combined with the goodness score defined by vector lengths and angles described earlier. It may very well happen that several points in the test gel is matched to the same point in the master gel, and vice versa. The best match is then chosen.

The overall algorithm is set up in Algorithm 1, with subroutine 2.

Algorithm 1 MATCHING(M,T), Pair-wise matching algorithm [4]

M and T are the master and matched gel's lists of spots
Subroutine: ***MATCH(M,T,L)***

L =set landmarks
LT =transformation based on L
T = transform(T,LT)
repeat
  M_s =**MATCH(M,T,L)**
  L = highest confidence matches in M_s
  Tr = transformation based on L
  T = transform(T,Tr)
until no further change in L

3.2.1.2 Comparing line segments

The algorithms described in [5], describes both spot detection, and matching of gels. This section will only focus on the part concerning the gel matching.

This method does not explore the neighbourhood of the spots in the same way as the method described in Section 3.2.1.1. One does indeed compare local

Algorithm 2 MATCH(M,T,L), Subroutine

M and T are the master and matched matrices, L is a set of landmarks. Initially L is the set of manually selected landmarks, in the other iterations of the main algorithm, high confidence matches are chosen to act as landmarks.

match M and T using the composite criterion described in the preceding text

patterns, but line segments from the local patterns to match spots are used. The gel image is thus divided into smaller parts, and these parts are compared as a unit, toward the opposite gel. The line segments consist of two spots in the local pattern. A line segment from the master gel local pattern is compared to the line segments in the matching gel. All lines between all pairs of spots in a local pattern are line segments that can be considered.

The line segments are rated by how similar they are by using a score that takes into account the length and angle of the line segments. The angles are relative to a horizontal axis. This is a similar method as one of the two criteria described in Section 3.2.1.1. But where the method in Section 3.2.1.1 uses the candidate spots' position relative to the landmarks, this method does not use such a criterion. The key idea of the algorithm is taken from [6].

The construction of the line segments is strongly related to the Delaunay triangulation. The triangulation is performed in a way that classifies the edges of the matched image into a category called "intense" edges. For an edge in the master image, it is now searched among only the "intense" edges of the matched image. This classification of edges reduces the search - space from potentially n^2 to, according to [5], 12n.

The algorithm considers one local area of the master gel at a time. To decide where the best scoring pattern is in the matched gel, a grid is laid over the matched gel image. The grid consists of nodes. Each node may receive a positive score. A node N's score is increased if there is a translation vector that ends inside of a square that N is a part of, see Figure 3.4.

For a given edge e in the master gel, and a sufficiently similar edge e' in the matched gel, the translation vector is the vector that translates the midpoint of e onto the midpoint of e'. The translation vectors are calculated for all edges where e and e' are similar enough to meet a given tolerance bound.

The nodes with a score higher than a threshold value are considered possible matching locations. The spots in these locations are matched against the pattern in the master gel. An example of a distribution of the scores of the nodes in the grid, and thus possible matching locations are depicted in Figure 3.5.

The method described uses the intensity of the spots to choose possible matches. This is done in several parts of the algorithm, and may be a weakness. This is so because, at least in the GABI project, we wish to study differences in spot intensities from one gel to another.

3.2.1.3 The star map problem

The star map problem, and a solution to it is presented in [3]. It considers the problem of matching a given star list against another star list, or against catalogue information. To match a star against another, the star' neighbourhoods

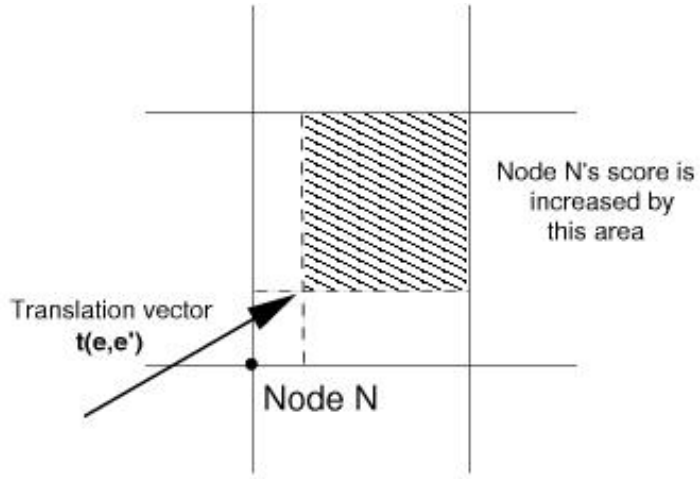


Figure 3.4: Illustrates the increased score of node N. The translation vector **t(e,e')** increases the score with an amount depending on the vector's endpoint distance from N. (Figure is from [5])

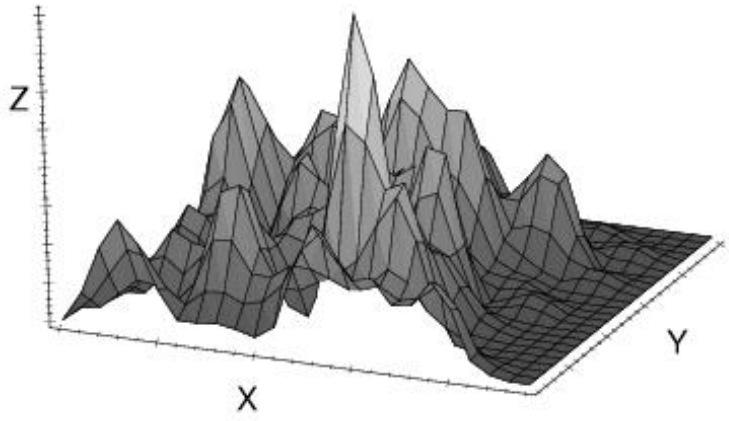


Figure 3.5: Shows the distribution of scores along the grid of nodes. X and Y are the node positions on the matched gel, and Z is the score. There are several possible matching locations, but one main peak. (Figure is from [5])

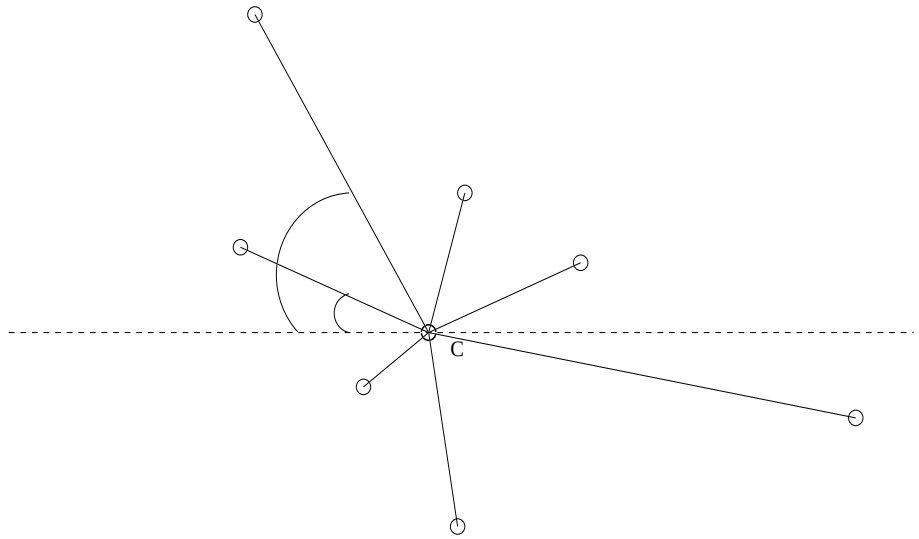


Figure 3.6: Illustrates the concept of a point's world view. The world view of point C is defined by the angles between the vectors from C to the other points, and the horizontal axis. The matched images should be roughly aligned, so that the horizontal axis is corresponding in the different images.

are compared. This is an example of an application of using the local neighbourhood information to match points. Although the solution to this problem is not directly applicable to 2D gel matching, it is included as an illustration of a different approach to solve a problem similar to 2D gel matching.

The reason why the star maps have to be matched is that two images of the same "area" does not coincide. The images are captured by different cameras, etc. so they are not completely identical, there may be distortions in the images caused by for example optical errors.

The proposed solution is based on a "world-view" for each of the n stars in the list. This is defined as a vector representing the angles subtended by the n-1 other points, relative to a horizontal axis. See Figure 3.6. The Euclidean distance between the points i and j, is used as a measure of the "effect" i has on j.

Matching of a point i, against a point j, is done by comparing the "world-views" of the two points, and check for sufficient equality.

The solution in [3] aims at finding a linear transformation between two spot lists. A linear transformation may contain rotation, translation and scaling. Such a simple transformation is not possible to apply to 2D gel matching, because of the geometric distortion.

3.2.2 Other methods

The article [4] also gives a short overview of other methods.

Early methods did not use the point pattern comparison approach to find the correct matching spots. These methods were based on a number of landmarks, and found a global transformation that transformed the landmarks of one gel

onto the other. This is what is done in the main method of [4] as well. But then, to match the rest of the gel, for each spot on the master gel, the best one on the transformed matched gel, is chosen with a composite criterion.

Other approaches have converted a spot point pattern to a specific graph. Others again have been based on the propagation of a landmark spot, which served as an initial match. Then the match was extended by looking in the neighbourhood of this spot for the next landmark. This looped until no further match was found.

The ELSIE/MELANIE [1] system used a comparison of spot clusters, instead of a simplified point pattern.

[1] http://www.genebio.com/interne.asp?pagename=melanie

Chapter 4

Towards a multiple matching of 2D gels

As a basis for the methods to perform multiple matching on 2D gels, some goals and definitions will be described in this chapter. Possible inputs and outputs from a matching procedure are discussed, together with a general pair-wise point-pattern based matching algorithm that is well suited for use as a part of a multiple gel alignment process, because it keeps several matches for a spot. The standard methods described in Chapter 3, only keeps one match for each spot.

In Chapter 2.9, it was explained why multiple alignment of 2D gels is interesting. In Section 4.3 a short description of the current approaches to perform multiple matching of gels is given. The attempt to improve the reliability and efficiency of multiple matching stems from the fact that the current methods neglect the direct relationships between the different pairs of gels, and only focus on the concept of choosing/creating a gel that has the role of a master gel.

The basis for the approaches described in this thesis for trying to improve the match-rate and reliability of multiple experiments, is the pair-wise alignments between some of, or all of the gels. The reason why it seems sensible to use the pair-wise matching information is that comparing a pair of gels is relatively computationally inexpensive. Performing for example 100 pair-wise matchings with a decent matching procedure, will typically only take minutes. On the other hand, using the same method to match for example 5 gels simultaneously may take in the order of 10 million times longer. These rough numbers are based on the following assumptions and calculations: The gels have $n = 1000$ spots, the matching procedure has time complexity of $O(n^k)$, where k is the number of gels. This gives a complexity of $O(n^2)$ for a pair-wise comparison etc. So performing 100 pair-wise matchings will take in the order of $100 * 1000^2 = 10^8$ operations. Performing a multiple matching of 5 gels will take $1000^5 = 10^{15}$ operations. $\frac{10^{15}}{10^8} = 10^7 = 10$ million.

4.1 The goal

The goal in multiple alignment is to find the corresponding protein spots on each of the gels, if such a correspondence exists. If such a correspondence does not

exist over all gels, the goal is to find corresponding spots in a subset of all gels. It is natural to think of the situation as a collection of different proteins whose representatives are situated on the gels. So the goal is to find *all* representatives for each protein. The highest possible number of representatives for a protein is the total number of gels. Finding all representatives will only be possible when a protein is present on every gel. But even if a protein is occurring on every gel, some of the occurrences may not be detected because of low abundance. It is also possible that a protein does not occur on a gel simply because it does not exist in the sample that the gel is run on.

A problem when defining the goal of a multiple alignment of gels, is that one in practise may experience that a single protein may appear as several spots on the gel. Should such occurrences be recorded as different proteins? The reasons for multiple occurrences of the same protein are debated, but in [19] it is claimed that an unsuccessful denaturation process, i.e. the process that unfolds a protein prior to a gel experiment, is a main reason for multiple occurrences of the same protein. Earlier it was assumed that mainly post-translational modifications caused these situations.

4.2 Definitions

The results of the pair-wise matching of gels will be labelled in the following way: The score between two different spots from two different gels, say g_k^i and g_l^j will be called $ss(g_k^i, g_l^j)$. The value of this spot score will be defined by the chosen pairwise gel matching procedure. The score between two gels G_i, G_j, will be denoted $gs(G_i, G_j)$.

4.3 Existing multiple gel matching methods

Most of the methods concerning the matching of 2D gels focus on the problem of matching pairs of gels, and it seems like the methods for comparing more than two gels may be improved. None of the commercial systems have applied methods similar to the common methods for multiple alignment of sequences or structures. For these situations one usually try different strategies that are more advanced than simply selecting one structure/sequence, and comparing the other objects to this. This latter approach is the one commonly used when comparing more than two gels.

Several commercial programs which allow multiple alignment of 2D gels exist. In the program PDQuest, this is done in the following way [10]:

The user chooses a reference gel G_m. This gel should ideally have a large number of spots. If necessary, the user may add spots to this gel. The resulting gel is then a modified copy of G_m, call it G_T, a template gel. Then all the other gels $G_1, ..., G_n$ are *pair-wise* compared to G_T. In this way, the relations from all the gels, including the reference gel, to the template, is established. This means that if spot g_3^4 matches spot g_T^6, and spot g_5^{45} also matches g_T^6, then g_3^4 is "defined" to match g_5^{45}. So the spots are linked together via their pair-wise correspondence to the spots $\in G_T$.

If there exists a spot on one of the gels that does not exist on the reference gel, this spot will perhaps not be registered. This is because this spot does not

match well with any spot on the reference gel. If this spot exists on several other gels, but all the time mismatches with the reference gel, then we have a protein expressed in different gels that is not registered.

In the program Melanie II [1], the multiple alignment is done in a similar way, but here a master gel is *constructed* from the other gels. The master gel contains all observed spots in all of the gels, and thus the problem of missing spots on the master gel is eliminated. Nevertheless, this method also ignores the pair-wise relations between the other gels.

4.4 Inputs to the multiple matching

Even though the ideal situation when matching 2D gels is a completely automatic procedure, in most cases it is accepted that the user must supply some information prior to the matching process. The aim is however to keep the amount of manual input at a minimum, while maintaining a good accuracy and efficiency of the matching.

4.4.1 Manual land-marking

An effective way to guide the matching algorithm towards a good solution, is to manually select a few corresponding spots on all the gels. These are called landmarks. Giving landmarks is the most common input to both multiple- and pair-wise matching procedures. The number of landmarks that is sufficient to perform a good matching is empirically determined, but usually 10 well distributed landmarks are sufficient.

4.4.2 Sequence information

It seems interesting to investigate if the genome sequence can supply useful information to the matching. In the case of the GABI project, the complete bacteria genome is sequenced. One possible usage of sequence information is to use the genes to predict the positions of all the proteins on the gel, and then use this information as a standard for how the experimental gel should look. The theoretical information may then be used to guide the matching of the experimental gels. In a real sample, not all genes are expressed, so a filtering of the genes is necessary.

When 2D gel experiments are performed, often only parts of the cells are examined at a time. One example of such a part is the outer-membrane. The proteins found in the outer-membrane are called outer-membrane proteins, abbreviated OMP. If the user supplies information about something that is common for the sequences coding for an OMP, we may be able to find all of the open reading frames (ORF) that may be coding for such proteins. In addition, information about post-translational modification sites can be supplied. This is quite important, because one gene may have different protein products because the proteins may be modified after the translation. We can then calculate the theoretical pI and Mw values of all these potential proteins (for details, see Chapter 7). These theoretical values may be calculated by using information

[1] http://www.genebio.com/interne.asp?pagename=melanie

about the molecular weights, and isoelectric focusing points of the single amino-acids contained in the ORFs. This will give us a theoretical image that may be helpful in the matching of other gels. The theoretical image of for example outer membrane proteins can be examined for patterns that may be used in the matching of the real, experimental 2D gels. In cases where there are different sub-patterns that match the same target pattern, one may choose the one that is most similar to the theoretical pattern.

Methods for calculating post-translational modifications already exists, but in practise they are difficult to use in a genome-wide experiment, because there are too many different possible modifications.

Another usage of sequence information, is to calculate a complete theoretical 2D gel image, and then use this image to identify proteins on a real gel. A thorough description of this kind of sequence usage is given in Chapter 7 and Chapter 8.

4.4.3 Pre-knowledge of post-translational modifications

If the user supplies knowledge of modifications, this may be taken into consideration in the matching process. The knowledge may be on the following form; spot g_j^i is likely to be modified in a certain way in gels $G_{j+1}, ..., G_n$. By including such information, one may be able to bias the search-space, so that when searching for the equivalent spot of g_j^i in other gels one may give a high score for spots that are positioned in a way that corresponds to the supplied modification information.

4.4.4 Micro-array analysis

Micro-array analysis gives information about expression levels of known genes. The expression levels of genes can give useful information to the process described in Section 4.4.2, and to the approach using the theoretical image to identify spots on a real gel, described in Chapter 7. We now have information about the expression of some of the possible, for example, outer membrane proteins. This information can be used to draw the attention to theoretical spots that are highly expressed in micro-array analysis, i.e. use the information as a filter to remove genes that are not likely to appear on a gel. Drawing the attention to these theoretical spots makes sense because there is often a correlation between micro-array measured gene levels and protein expression levels, and highly expressed proteins are the ones that most likely occur on a 2D gel.

Results from micro-array analysis may also be compared to the results of the 2D gel matching and analysis, to observe any differences in gene/mRNA level vs. protein level.

4.4.5 Questions

Some fundamental questions arise in the two sections, 4.4.2 and 4.4.4.

- Is it possible to predict the sub-cellular location of a protein based on only the gene sequence?
- Is it possible to predict the focusing point from the sequence of a gene, by possibly taking post-translational modifications into account?

The answer to the first question is that it is possible to *predict* the subcellular location of a gene's corresponding protein. This is not a trivial task, but programs to perform this task have been developed, see [20] and [23].

The answer to the second question is that it indeed is possible to predict the expected focusing position of a gene based on the sequence. This is explained in Chapter 7. Taking possible modifications into account may be more difficult, since there is a too large amount of possible modifications. However, if we strictly limit the possible modifications, they may also be taken into consideration when predicting the focusing position of a protein.

4.5 Output from the matching

Several results from the matching procedure may be presented:

- Obviously, the list of proteins $P_1, ..., P_n$, with their corresponding spots in the gels. This will be equivalent to Table 4.1, perhaps with multiple matches in same gel removed, or with alternative spot suggestions for each protein, and a statistical significance for each of the possible spots.
- The matching procedure may reveal possible post-translational modified proteins, by looking for a collection of matching spots, representing the protein P_i, where there is a significant deviation of values, both in pI and Mw values. The deviations may represent a pattern typical for certain known modifications, for example by showing a given deviation in pI value.
- If the ORFs of the organism involved are analysed, the theoretical pI and Mw values of the potential proteins (the ORFs) may be computed, and compared to the ones of the different spots on the gels. If the experimental gels are first matched to each other, to produce a list of proteins P_1 ... P_i with the corresponding spots in the gels, and then are matched with the theoretical values of the ORFs, for each protein in the list, we may suggest one or several corresponding ORFs.

  These suggestions may be based on the following calculations:

  The pI and Mw values of the spots in the gel are obtained by interpolating from other, known proteins on the gel. These interpolated values are considered to be "apparent" values. Consider a protein P_i, which has been assigned a series of spots from different gels. If we compare the average pI and Mw values of this series of spots to the theoretical ones, extracted from the genome sequence, we may get a match between the theoretical values from ORFs in the genome, and the apparent ones on the gel. The reliability of the match will of course depend on how good the methods for predicting the gene products are (in this context proteins), and also on how accurate the apparent pI and Mw values are. If the reliability is high, and we only have one match between a protein in the list, P_i and an ORF, we may have a clear indication of what is the originating gene of the protein. This would be of great help in trying to identify proteins. If a relatively high number of gels have been matched, then the average pI/Mw value of the spots corresponding P_i will probably be a good approximation of the real ones. This methodology is further discussed in Chapter 7 and Chapter 8.

	Gel 1	Gel 2	...	...	Gel k
Protein 1	(2)	(26,23)			(111)
.					
.					
.					
.					
.					
.					
.					
.					
.					
.					
.					
.					
.					
Protein n	(Ø)				(392, 187)

Table 4.1: A table showing a setup for keeping track of the proteins in the different gels. Each gel has one column, and there is one row for each protein that is expected to be found. The number in a cell tells which spot in that column's gel that is assigned to the protein on that row. Several spot numbers mean that there are multiple options, for example if two spots are very close, and it is not clear which spot is correct.

4.6 Organising the data

It is important to represent the data in an efficient and intuitive way. The main goal is that we would like to obtain a list of proteins. For each of these proteins we would ideally have identified their corresponding spot in all gels. Without having any predefined information, the proteins in our list will just be defined by some unique label. A visualisation of this is shown in Table 4.1. We may call this table MT, for Match Table. Notice that for each cell in the table, a list of possible matches are shown. These must be ranked by some match-score.

4.6.1 Representing the pair-wise information

As stated in the beginning of this chapter, it will be sensible to initially perform a pair-wise matching of all gels. If we store this information in a clever way, and in addition allows for later modifications, we have a powerful, yet simple and dynamic representation of the relations between spots on gels. This representation can for example be a set of arrays, with one array for each pairing. Other representations may be different graph representations. What is important is that the representation allows quick access to matches, and the possibility to change matches.

4.6.2 Representing a match-set consisting of several gels

A match-set is a collection of gels whose correspondence to each other is defined. One can think of at least two alternatives on how to represent this:

- Represent a match-set as a table of proteins. For each protein there is a column for each gel, where possible matching spots are listed. As the matching proceeds, entries are added into the table. This would be similar to Table 4.1.
- Simply define a match-set as a list of gels. Their correspondence to each other may be found in the pair-wise matching structure explained in Section 4.6.1. When the final match-set is constructed, one could then generate a protein list with corresponding spots. This method will probably be difficult to use in practise, because the matching itself will then define where to look in the pair-wise system, when presenting the final result.
- Use a graph representation. The matching procedure in Chapter 6 actually gives a nice representation of a set of gels matched together.

The first alternative seems convenient, because it gives a systematic way of filtering useful information from the pair-wise comparisons. When relations between gels need updating, perhaps because of newly discovered knowledge about the correspondence between specific spots, one could simply make this change by changing some values in the MT. Using the first alternative also makes it easy to visualise the correspondences between the gels.

4.6.3 Evaluating a final match-set

To be able to compare different methods for multiple alignment, and for general interest, a score of the results of such an alignment is defined, with a representation similar to the one in Table 4.1. It will be desirable that each row, i.e. each "discovered" protein, contains as few open slots as possible. Let the percentage coverage,

$$pc(row_{nr}) = \frac{\text{number of cells with data}}{\text{total number of cells}} * 100\% \tag{4.1}$$

be (a part of) the score of one protein/row.

Now we may calculate the total percentage coverage for all the rows, average coverage, standard deviation etc.

Another factor that may be incorporated in the scoring, may be the pair-wise scoring between the spots in each row.

4.7 Preparing the pair-wise match data

The scoring results from the pair-wise matching of the gels that are going to be used further, should be normalised, so that differences in the nature of the compared gels do not affect the global ranking between pairs. Whether this is necessary or not, depends of course on how the pair-wise matching algorithm scores the pairs.

An example of a situation where it may be necessary to normalise the scores between two gels, is a situation where two matched gels both have an overall

high intensity staining. These may be called gels "intensive" gels. If the scoring procedure that scores pairs of spots gives higher scores to more intensive spots, a spot pair from an "intensive" set of gels will achieve higher scores than a spot pair from a less intensive gel pair.

4.8 Extending the pair-wise algorithm

A general version of the algorithm described in Chapter 3.2.1.1 can be modified to preserve alternative matches for each spot in the matched gels. One of the alternative matches for each spot in the master gel may be considered to be the "active" match. Algorithm 3 is a suggestion to an algorithm that preserves several possible matches.

Generally, when performing multiple alignment of gels, one may reveal useful information about the pair-wise correspondence between gels. Useful information may concern common distortions and systematic deviations that appear when the gels are matched. More specific knowledge about matches may be the occurrence of inconsistent situations like the ones described in Chapter 6.1. When inconsistent configurations are discovered, it may be convenient to consider if other match candidates for the spots than the "active" ones solve the problem of inconsistency. For these reasons it may be sensible to maintain a set of match edges in the matching between two gels, instead of having a static set of matches between spots in two gels. By storing alternative matchings for a spot, one may go back and change the pair-wise matchings when new information is revealed. If only single possible matches are stored early in the matching, the rest of the matching will be affected in a *possibly* negative way. This is why it is meaningful to maintain a limited list of matches for each point. This list may be manipulated as the matching progresses. Matches may be removed or added.

Algorithm 3 PAIR-WISE-MATCH-EXTENDED(A,B)

A and B are the master and matched gel's lists of spots

M = empty match-set
specify guiding landmarks
perform coarse transformation based on landmarks
for i $\rightarrow$ 1 to length(A) **do**
  find the set $K \subseteq B$, with the k closest points to i $\in A$
  for all j $\in$ K **do**
    compare j's neighbourhood to i's
    if j has good enough match to i **then**
      add match (i $\rightarrow$ j) to M
    end if
  end for
end for
choose a current unambiguous matching
between A and B as a subset from M,
or simply keep all, possible ambiguous
matches.

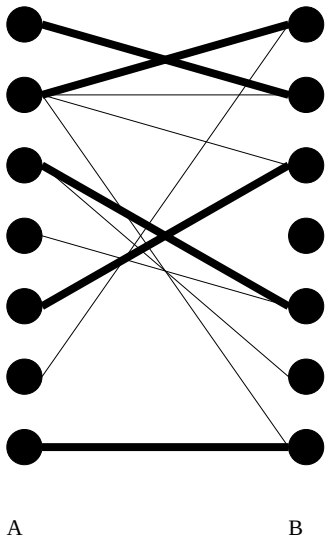


Figure 4.1: The forest representing the match-set, M, between *two* gels. Thick lines represents a possible choice of "active" match-edges.

The algorithm gives the opportunity to match a spot to several other spots. If all matches are kept, they may be removed at a later stage, when one wishes to remove ambiguities.

If one chooses to find an unambiguous subset, this may be viewed as choosing an optimal edge-set from a forest, where a node can only be a part of one edge. An example of a subset of M is shown by the bold lines in Figure 4.1. The question is how to choose edges. The weight of the edges should ensure that an edge is not chosen if it does not correspond to a match-score better than a certain value. If one has a *strict* limit on how good a match between two spots must be in order to be in the list of possible matches, one can be sure that adding an edge that is not in conflict with other edges, always produces a better overall match. This is important, because if *all* possible matches, good and bad, were given a positive score, it would always increase the total score when an edge was added, even if the added edge did not contribute to a more correct matching.

Choosing a subset of edges that gains an optimal total match-score is not trivial, because this problem does not have optimal sub-problems. If an edge that currently appears to be the highest scoring one is added, this may inhibit other edges that will be in a real optimal solution, since each node (spot) only can be a part of one edge.

4.9 Time complexity of pair-wise comparing all gels

The basis of the multiple matching methods described in this thesis is the pair-wise comparison between all pairs of gels. Describing the time complexity gives an impression of the running time of this initial step.

If the pair-wise matching procedure has a time complexity of $O(n * m)$, where n and m are the number of spots in each of the two gels, then a pair-wise matching between all pairs of gels will have a total time complexity of $O(\binom{k}{2}(n*m))$, with k being the number of different gels. This unproven assertion is a bit imprecise, since the number of spots vary from gel to gel. So if we rather say that the maximum number of spots on all the gels is N, the pair-wise matching would take $O(N^2)$ time, and a pair-wise matching between all gels would take $O(k^2 * N^2)$ time.

Time analysis: There are k different gels, with maximum N spots on each. We wish to calculate all the different possible pairs. This corresponds to choosing two elements from k, unordered and with repetitions of elements allowed, which are $\binom{k}{2}$ different pairs. For each of these $\binom{k}{2}$ number of pairs, the matching procedure takes N^2 time. The total number of operations will then be $\binom{k}{2} * N^2 \rightarrow O(k^2 * N^2)$, since $\binom{k}{2}$ is $O(k^2)$.

In practise, the pair-wise matching will not take as much as N^2 time. This is due to the fact that for each spot in one gel, only a limited number $<$ N will be considered in the other gel.

The total time complexity of $O(k^2 * N^2)$ is not too bad compared to for example an exponential running time, which is the case if several gels are matched using a "full" alignment. Such an alignment may be performed by using the same algorithmic ideas as for pair-wise matching, but with more than two gels. In the introduction to this chapter an example that illustrates the problems of an algorithm with exponential time complexity is given.

Chapter 5

Progressive solutions to the multiple matching problem

This first approach is based on the idea to use results from pair-wise gel matching to progressively make a multiple alignment of a set of 2D gels. Progressive methods for performing multiple matching have been used in for example sequence [8] and structure matching [9].

5.1 A progressive approach to multiple alignment

A progressive approach implies that one has different sets, that are combined in some way, until all gels are in one match-set. For convenience, different sets may be referred to as $M_1 ... M_i$, where each set may contain one or several gels. The structure of the procedure to combine sets can be divided into two main categories:

- Linear progressive. One by one the gels are added to a current match-set. In this case we only work on one match-set, which finally will contain all gels.
- Tree progressive. Based on some criterion several match-sets are created. The progression is based on the merging of match-sets, either consisting of several gels, or only a single gel.

The main difference between the two categories is shown in Figure 5.1. Situation **A** shows the tree progression. Which two match-sets should be combined to form a new match-set? Situation **B** shows the linear progression. Which of the single gels should be added to the current match-set, and form a new match-set?

The most straight-forward progression seems to be a linear progression. With this method there will intuitively be less information to keep track of, because the gels are considered one at a time.

Pseudo-code for the two approaches is shown in Algorithm 4 and Algorithm 5.

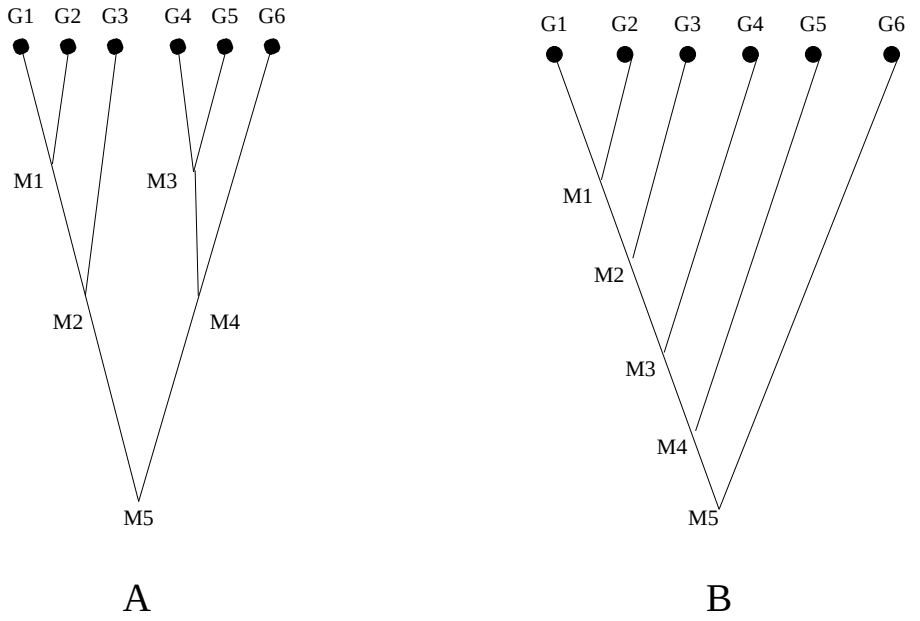


Figure 5.1: Situation A shows an example of tree progression. Situation B shows linear progression

Algorithm 4 Multiple matching using linear progression

Input: A set of gels, G

M = empty match-set
while G $\neq$ *empty* **do**
  choose best gel, G_i to add to M
  add G_i to M
  $G \rightarrow G - G_i$
end while
present M in a suitable manner

Algorithm 5 Multiple matching using tree progression

Input: A set of gels, G

for all gels $G_i \in G$ **do**
  initialise a match-set, consisting of G_i
end for
while there are more than one match-set **do**
  choose the best pair of match-sets to combine
  combine those into one
end while
present the resulting one match-set in a suitable manner

5.1.1 Expanding a set of matched gels

An important issue is how to choose the next gel in the merging of the match-sets. It seems sensible to have a measure of "how good" a matching is between complete gels or match-sets, and first combine the units with "best" corresponding information. The score between two gels is called **gs**(G_i, G_j), and should be an average of how good each spot in G_i corresponds to a spot in G_j, and vice versa. The spots that have no match should also be included, but with a score of zero, so if there in a pair of gels are many spots with no corresponding match, the pair receives a low average score. A simple suggestion to the score between two gels G_i and G_j is:

$$gs(G_i, G_j) = \frac{1}{|G_i| + |G_j|} \sum_{g^k \in (G_i \cup G_j)} ss(g^k, g^x) \tag{5.1}$$

The term $ss(g^k, g^x)$ symbolises the spot score between spot g^k, and its best match in the opposite gel, called spot g^x. How g^x is obtained, depends on the procedure that matches the gels G_i and G_j.

The score $gs(G_i, G_j)$ may be used to find a score between two match-sets M_i and M_j by combining the gs scores of the gels in M_i and M_j.

5.1.1.1 Maximising average gs score between gels in combined match-sets.

One way to choose which gels to combine is to maximise the average score between the combined sets:

- In the case of the linear progression method, one could choose the next gel to add to the match-set as the one $G_i \notin M$ that maximises the mean score. That means, choose G_i such that

  $$\frac{1}{N} \sum_{G_j \in M} gs(G_i, G_j) \tag{5.2}$$

  is maximised. By doing this, the gel that has the highest average **gs** score to the gels in M, is chosen as the next member of M. In this equation N is equal to the length of M.

- In the case of tree progression approach, one could choose to combine sets M_i and M_j such that

  $$\frac{1}{N} \sum_{G_k \in M_i} \sum_{G_l \in M_j} gs(G_k, G_l) \tag{5.3}$$

  is maximised. This formula summarises the **gs** scores between all pairs, with one gel from M_i and one gel from M_j, and divides this sum with the number of such pairs. In this equation N is equal to $|M_i| * |M_j|$.

A better way of describing the sums above, may be to introduce a generic score, $ms(M_i, M_j)$, which stands for **m**atch-set **s**core, and gives a total score between two match-sets. These match-sets can be of arbitrary size (including only a single gel). This is defined as:

$$ms(M_i, M_j) = \frac{1}{N} \sum_{G_k \in M_i} \sum_{G_l \in M_j} gs(G_k, G_l) \tag{5.4}$$

With this definition, $N = |M_i| * |M_j|$.

By using Equation 5.4 the general problem becomes to find two match-sets, M_i, M_j that gives the highest ms-score.

5.1.1.2 Maximising only a single gs score between the candidate match-sets.

One may choose the combination of M_i and M_j that gives the highest **gs** score between a gel in M_i and a gel in M_j.

In the linear progression case, the $G_i \notin M$ that has the highest **gs**-score against a gel in M is chosen. A variant of this method may be to choose $G_i \notin M$ as the one that has the highest **gs**-score against the gel that was latest added to the match-set. This is a biased method that expects that some gels show a lower degree of similarity to certain gels. This may be the case if the gels have changed in a systematic way, for example over a time period. One can expect that gels extracted from within a small time interval is more similar than gels extracted from different time periods. If there is no such systematic change of the gels, choosing the next gel to add as the one that is most similar to the gel latest added to the match-set, makes less sense.

For the tree progression, all pairs of match-sets must be considered, and for each of these pairs one must find the gel pair that is most similar. The pair of match-sets that gives the highest similarity between single gels is the pair that will be merged.

5.1.2 Consequences of combining two match-sets

When adding a new gel or match-set to another match-set, it may be necessary to reconsider some of the matchings done by the initial pair-wise matching. An intuitive example of this is shown in Figure 5.2. The figure shows that information from an initial pair-wise matching may be used to construct and modify a match-set between several gels. This observation emphasises the advantage of performing an initial pair-wise matching between all pairs of gels that may be modified later. An example of such a pair-wise matching is the algorithm suggested in Chapter 4.8.

5.1.2.1 Considerations regarding the influence of the new candidate

As Figure 5.2 illustrates, when a new spot from a new gel is introduced in the match-set, it will be interesting to see if the pair-wise matching between the different parts involved is consistent with each other. In this context, the matching is consistent if it has a transitive property. That means that if g_C^k matches g_A^l and g_B^m, then g_A^l and g_B^m should also match. If the spot from the new gel breaks with this property, it does not seem clever to reject the spot g_C^k, but rather try to look for a configuration that makes the matching of the new spot consistent with the rest of the match-set. This may include changing the "current" matching of spots in M by changing the pair-wise match-configuration. In the case of Figure 5.2, if the edge between l and m does not exist, and l is

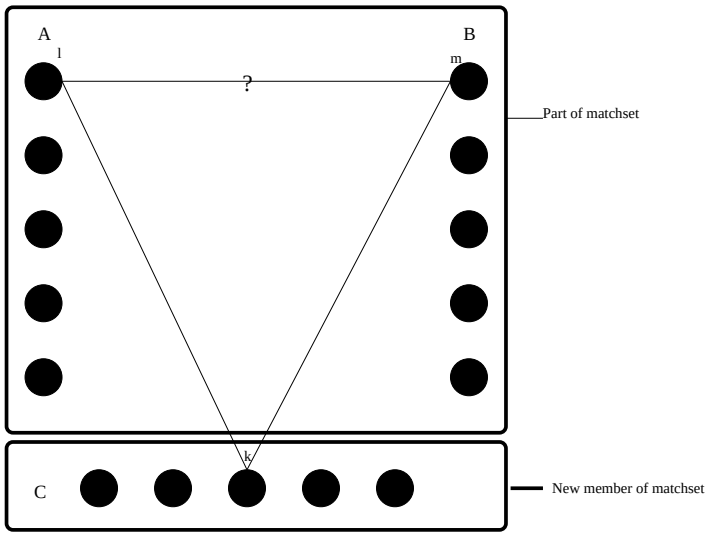


Figure 5.2: Matching information from pair-wise matching can be used when adding a new gel (C) to a match-set. If k is matched to l and m in the pair-wise matching between C and A and C and B, check if l and m is matched in "current" match-set. If not, does a better configuration exist?

matched to another spot in gel B, we have an inconsistent situation. If this match has a low score, one may consider to remove it, and replace it with a link between l and m. Chapter 6 thoroughly deals with the problem of inconsistency.

5.2 A method for including a new gel to a match-set

The different methods for choosing sets of gels when building a match-set that includes all gels, are discussed earlier in this chapter. This section suggests a method for including a new gel into the current match-set for the linear progression approach.

5.2.1 A simple max-score of gel-pairs guided procedure

Looking at the linear progression method, after choosing the next candidate, G_{next}, how can G_{next} be included in the match-set, M? The aim is to find the corresponding spots in G_{next} for all proteins in question. With for example a representation like the Match Table in Table 4.1 in mind, one wants to insert the spots of gel G_{next} in the correct rows of G_{next}'s column. But how may this be done? As explained earlier, a frequently used method is to compare G_{next} (and all of the other gels as well) to a master gel, and then place the spots of G_{next} in the same row as their corresponding spots in the master gel.

An alternative suggestion is not to have a fixed gel as a master gel, but choose among the gels that are already in the match-set. One may for example use the matching of G_{next} to the other gel already *in* M that gives the highest

gs score. This is similar to using a master gel, but the difference is that the "master" gel will vary.

When using a master gel, the spots on this gel define which proteins one searches for. When using the method suggested in this section, one may run into new proteins as the matching goes along. This is positive, because in this way one does not bias the matching towards any master gel.

Chapter 6

Using a graph model to perform multiple matching

In this chapter, some of the problems with traditional multiple matching are defined, and a new approach to perform the multiple matching of 2D gels by using a graph structure is suggested.

6.1 Defining inconsistency

During the procedure in Chapter 5.1, one may encounter that is seems sensible to insert into a cell of the match table (MT, see Table 4.1), spot X. It may very well be that this cell is already occupied.

An example is shown in Figure 6.1. Let gels A and B be in a match-set, i.e. a set of gels whose correspondence in a multiple matching is defined. Based on some criterion, gel C should be added. When adding C one wishes to insert C's spots in the correct positions in the MT. Looking at the pair-wise matching between B and C, assume that spot b_i pair-wise matches spot c_j. Then it is sensible that spot c_j should be in the same row as b_i, that means they are both assumed to be the same protein. Call this protein P_w.

Now, consider the pair-wise matches between gel A and C. A motivation for doing this is that there may be some spots in C that did not match any in B. By matching C to A these spots may also be included in the MT. Suppose that, during the processing of the pair-wise relation between A's spots in MT and C's spots, it is found that spot a_k matches spot c_j. Hopefully, this would place spot c_j in row P_w, but it may also not. In that case a_k matches c_j and b_i matches c_j, but a_k does not match b_j. This is called an inconsistency in the pair-wise matching.

6.2 A graph representation of the multiple matching

We may model the matching problem as a graph, call it the *S*pot *R*elation *G*raph, SRG. By manipulating and exploring this graph we wish to be able to present a result equivalent to a multiple matching of the gels.

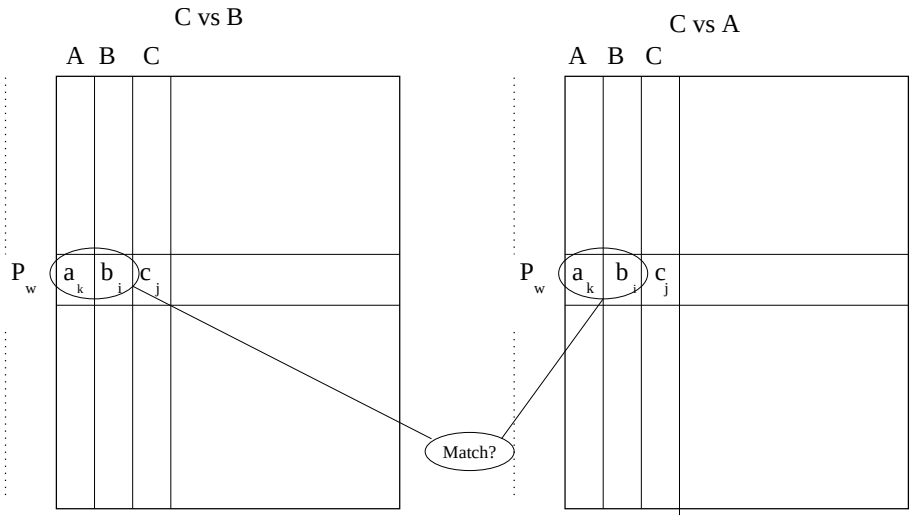


Figure 6.1: The figure illustrates the possible conflict when inserting a match into a match-table. A and B are in the match-set, and we wish to add gel C. First, the relation between C and B is considered, and spot c_j matches b_i. Then C vs. A is considered, and spot c_j matches spot a_k. If a_k and b_i are initially on the same row, then the three spots correspond to the same protein.

6.3 Formal definitions

A graph data structure and some new concepts are used in the graph approach described.

6.3.1 The graph idea

The central object is a graph, $G = (V, E)$, where all the spots are represented as vertices. Each **vertex** $v \in V$ has two integer attributes: One representing the originating gel, gelNumber(v), and another representing the spot's number inside the gel, spotNumber(v).

Edges: For each pair of vertices (u, v), an undirected edge between the two vertices means that these spots are matched to each other in some way. This means that there will be no edges between vertices with equal gelNumber. The edges representing a pair-wise match from a direct alignment of two gels, will be called "direct" edges. A match between a pair (u, v) which does not come directly from a pairwise alignment, may be represented in the graph as well, then as an "indirect" edge. As the name implies this is a match that has emerged between two vertices via some other vertices. An important question concerning these edges is whether ambiguous matches between spots should be allowed in the graph or not. This would lead to the possibility that a spot from G_i may have several (different) scoring edges/matches to spots in G_j. Only one of these may finally be valid.

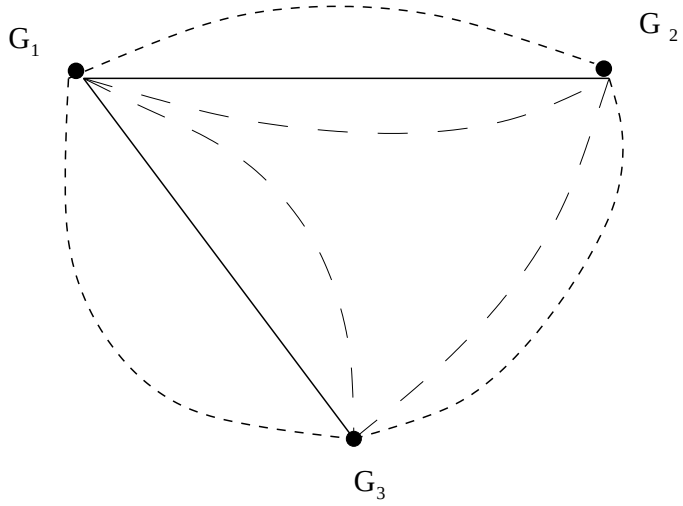


Figure 6.2: Figure showing the central concepts of the graph idea. Having different kinds of edges makes it easier to depict the different concepts of connection and correlation.

6.3.2 Some concepts

To describe the situations that may occur in the graph, it will be convenient to introduce some concepts:

- A *connection* $C(g_k^i, g_l^j) < g_k^i, ..., g_l^j >$ exists if there is a path $< g_k^i, .., g_l^j >$ in G between g_k^i and g_l^j. A connection may be represented as a path of edges (or vertices). In the proceeding text it is assumed that a path consists of edges. Such a path may be of length one (direct match between two spots) or more than one (indirect match between two spots). These connections will be represented in the graph either by a "direct" or an "indirect" edge. Not to be confused with directed and undirected edges.
- The *correlation* $Co(g_k^i, g_l^j)$ between g_k^i and g_l^j is the collection of all connections between them. This concept will in the graph be called a "sum" edge.

The concepts and different edge-types in the graph are shown in Figure 6.2.

6.4 Agenda

An overall view of tasks that should be performed during the process of matching the gels:

- Construct the graph.
- Recognise inconsistent configurations (Section 6.1).
- Suggest a subset $G^c = (V, E^c)$ that gives a consistent configuration of edges.
- Present this subset, G^c as a multiple alignment of gels.

The order of the three first tasks may vary, and may not necessarily appear as separate tasks, depending on the chosen approach to construct the SRG.

6.5 Constructing the SRG

Two approaches are considered for building the SRG:

- One alternative is to create a "complete" SRG, based on the pair-wise matching of all gels. This alternative gives the possibility to reveal *all* ambiguities, and the method will be explained in Section 6.6.
- Another alternative is to build the graph from only vertices, continuously adding edges while ensuring that the graph represents an unambiguous matching of the gels. This will be similar to a progressive matching scheme described in Section 5.1, and will be explained in Section 6.7.

Regarding the two presented approaches, it seems like the first alternative is the most promising. This approach has a graph consisting of all the known relations between the spots as starting point, and may probably detect inconsistencies more easily than the other approach. It may also make a more balanced choice when choosing one among several competing correlations. For these reasons, the method based on a removal of edges is the one that is implemented in the system, and most thoroughly described in this thesis.

In the sections describing the two approaches, the expressions connection, direct connection, and indirect connection are used when considering the adding and removal of edges in the graph. Direct and indirect edges are also used, serving the same purpose.

6.6 Approach: Removing edges from the graph

This method will be based on building the SRG from the pair-wise comparisons, and then remove edges from the graph until it is consistent. The order in which the edges are removed is important for the result, because removals early in the process may inhibit a good solution later.

As the name implies, the removal of edges is a central part of the approach. Connections between spots may be removed, because they are in conflict with other connections. But if several match-alternatives are kept from the initial pair-wise matching, one may replace connections with these alternative connections.

In this way we may say that we "correct" the results from the initial pair-wise matching.

6.6.1 Defining an inconsistent graph

The problems defined in Section 6.1 may be recognised by performing a modified transitive closure of the SRG. A transitive closure of a graph means to establish answer to the question: For all possible pairs of vertices (u,v), does there exist a path from u to v? In our case we will also be adding edges corresponding to these paths, such that if there exists a path from u to v, there should also be an edge (u,v). The modification in this case, is that there shall be no edge (u,v) if u and v has the same gelNumber label, and the path may not contain more than one spot from each gel. This means that the length of any path may not be any longer than the number of gels - 1. To be able to reveal, and more importantly, remove *all* inconsistencies, it will not be sufficient to simply find if there exists a path (u, v), but to find all such paths. In terms of the definitions in Section 6.3.2, we may say that there should be only one correlation between spot g_k^i and a spot in G_j, where $j = 1, ..., i-1, i+1, ..., n$, with n being the number of gels.

An example of an inconsistent graph constructed from pair-wise relations is shown in Figure 6.3.

Claim: After this modified transitive closure of $G \rightarrow G^* = (V, E^*)$, the graph is inconsistent if there, for any vertex u, exist edges (u, v) and (u, w) where gelNumber(v) equals gelNumber(w). In practise, this means that a vertex, u, should not have more than one edge connecting it with G_i, except for multiple edges.

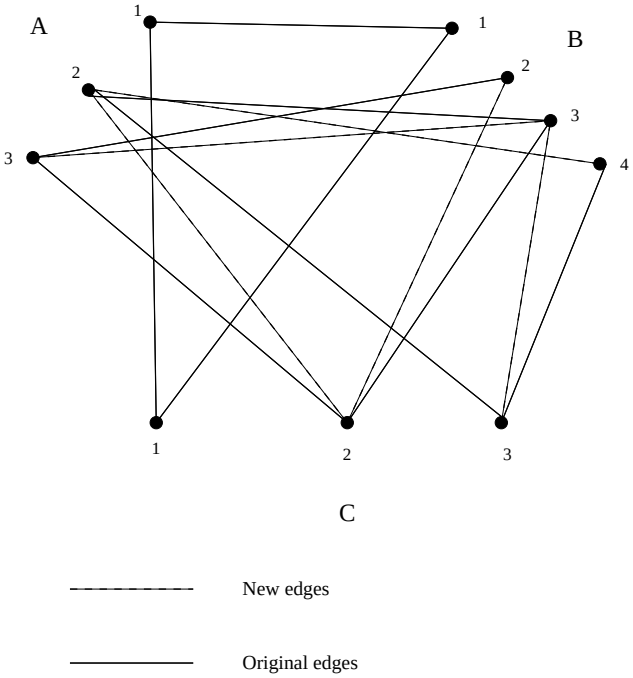
Proof: It is trivial to see that if there exists edges (u, v) and (u, w) where gelNumber(v) equals gelNumber(w), then spot u matches both spot v and w in the same gel, and the graph is inconsistent. There may be a slight exception here, in the case that spots v and w are identical. But this is a situation where the system can not know what to do, because if two spots are identical, they cannot be treated separately. □

6.6.2 Weights

To choose the set $E^c \subseteq E^*$, the edges $\in E^*$ should be weighted. The weight-component of an edge should be decided by two parameters:

- How long was the path that "created" this edge? Measured in number of edges.
  - For all edges, $e \in E$, path-length = 1.
  - For all edges, $e \in E^*$, path-length ≥ 1
- An edge $e \in E^*$ represents a match between two spots. The score of this match should be a parameter
  - This score should be defined as the average of the scores of the edges in the path of e.

If a somewhat higher perspective than edges and paths are used, one could give a score that indicates whether two spots should be connected or not. This could have been done by scoring the correlation between two points. An important component in this score will be the number of connections that are in the correlation.



Pairwise matchings:

A−B	A−C	B−C
A:	A:	B:
a1 − b1	a1 − c1	b1−c1
a2 − b3	a2 − c3	b2 − nil
a3−b2	a3 − c2	b3 − c2
		b4 − c3
B:	C:	C:
b1 − a1	c1 − a1	c1 − b1
b2 − a3	c2 − a3	c2 − b3
b3 − a2	c3 − a2	c3 − b4
b4 − nil		

Figure 6.3: The generated graph from the pair-wise matchings with the transitive closure edges (new edges). Clearly, there are inconsistencies. For instance a_2 is matched to both b_3 and b_4 (via the transitive closure edges).

Given the number of gels, k, and the length of the path represented by the edge E, the weight of E may be formulated as:

$$w(E) = \left(\frac{1}{length(path(E))} \sum_{e \in path(E)} score(e) \right) * F(length(E)) \tag{6.1}$$

where

$$F(x) = \frac{k - x}{k - 1} \tag{6.2}$$

The function F takes an input, and calculates a number greater than zero, and less than or equal to one. In practise, the average score of the edges on the path is multiplied by a function with values less than or equal to one. The output of the function is negatively correlated with the input (the length of the path), which makes sense because a short path, for example consisting of only one direct edge, is more reliable than a path consisting of a high number of direct edges. The score applies to all edges representing a single path, meaning all the direct and indirect edges.

For scoring a sum edge, use the average score of all the edges it represents. The score of a sum edge also contains a component that gives sum edges with a high number of contained direct/indirect edges higher credit.

6.6.3 The steps

The approach described in this section may be divided into several tasks that must be performed. The main steps are:

1. Discover the correlations that form inconsistencies. These correlations are divided into inconsistent sets, where one correlation may be a member of several sets. A set consists of different competing correlations, where the matching is ambiguous if all are kept.

2. Select single inconsistent sets to be made consistent, i.e. remove all but one member. Examples of inconsistent sets may be found in Figure 6.3. Here, $\{Co(a2, b3), Co(a2, b4)\}$ is one inconsistent set, and $\{Co(c2, a3), Co(c2, a2)\}$ another.

6.6.3.1 Discovering all the inconsistent sets

In this approach, we have to identify the sets of contradicting correlations, and for each set choose one valid correlation. To identify contradicting correlations, we can, for each spot u, find all the correlations that start or ends with u. In the graph, this will be represented by all the sum edges that start or end in u. If then, several of these edges have end vertices (opposite of u's gel) in the same gel, contradicting correlations exist. This is because one spot can only match one spot in each of the other gels. From the two or more contradicting correlations, only one must be chosen. The other correlations must be rejected, meaning there is a competition between the correlations.

6.6.3.2 Choosing the next inconsistent set to be made consistent

It is important to be conscious about the order that the contradicting sets are examined. The reason is that changes done to one set, may have consequences for the rest of the sets. One alternative is to process the sets in a decreasing order relative to the number of correlations present in the sets. This means that the next set to be processed is always the set with the highest number of correlations. By processing these large sets first, many of the smaller sets will shrink, because many of the correlations are common for different inconsistent sets. If the largest sets are processed first and the smaller one shrinks, on may shrink sets consisting of only one connection, and thus loose this match between two spots. This observation means that if the largest sets are processed first, the consequences of the changes on the smaller sets must be checked, to avoid changes that remove correlations with only one connection. We also risk that when processing the largest sets first, we may remove correlations and connections that also belong to other sets, and in that context are "good" elements.

Another alternative is to process the sets in an increasing order. Accordingly, the smallest sets will be processed first. The small sets, consisting only of two or three correlations may be simpler to make consistent. It is easier to choose one from two or three rather than choosing one from more than ten correlations. Starting with the smallest sets also has less consequences for the other correlations, simply because fewer correlations must be removed. Different alternatives are tested in Section 6.6.4

The order of the sets will be dynamic and not predefined, because the number of correlations in a set may decrease as a consequence of an edge removal done in the processing of another set. Alternatively, the processing order can be decided before any removal is performed. However, this is probably not a very good idea, as the size of the inconsistent sets will probably vary quite much during the removal process.

6.6.3.3 Looking at one inconsistent set: Choose one correlation to keep.

How can the "best" correlation be chosen from a set of contradicting correlations? If only one of the competing correlations contain a direct connection, then it is very likely that this is the best one too keep. Simply because the most reliable information we have, is the direct matching between two spots. As a general rule we choose the correlation containing a direct connection. There will in most cases not be several such correlations, since that would imply that a spot has a pair-wise matching with more than one spot in one single comparing gel. However, if we allow multiple matches for a spot, this might occur, and one of the correlations must be chosen. To summarise what to do when correlations in the inconsistent set contains direct connections:

- If only one correlation contains a direct connection, choose this to be the valid one.
- If there are several correlations containing a direct connection, choose one of them by using the general criteria described below.

The general criteria for choosing one correlation from a set of competing correlations are as follows:

- Look at the average scores of the connections in each correlation. The direct connections get their score from the pair-wise matching. The indirect edges are scored based on the length of the path they represent, and the score of the edges in the path, according to Equation 6.2.
- The number of connections that a correlation represents may be used. If a correlation contains a high number of connections, this indicates that the correlation is reliable.

The considerations described here, are relatively local. Ideally we should take into consideration how much impact the removal of a correlation (by removing direct connections), or the decision to keep a correlation has on the rest of the matching.

6.6.3.4 Removal of correlations

If a choice is made to keep a correlation, this means rejecting others, which again means rejecting connections. In which order should the rejected correlations be removed? It seems sensible to start removing the "worst" correlation, because this correlation is the one that we are most certain of is wrong, and thus will the consequences of its removal probably be more acceptable. The "worst" correlation may be the one represented as the sum edge with lowest score. Acceptable consequences mean that the removal does not affect other "good" correlations in a negative way. These theories of course depend on how the scoring of the edges is performed. It is crucial to have good scoring routines that reflects the real importance of the edges.

The rejection process has the potential of having a great affect on the other correlations present, because a rejection of a correlation implies removing the connections in the correlation, and thus, removing (at least) one direct edge. This direct edge may be part of another connection, and may therefore terminate the existence of other correlations than the one we intend to remove.

If we have chosen to remove a correlation, we must remove or "deactivate" all the connections in the correlation.

6.6.3.5 Removing a connection

If a connection in the correlation is direct, we must remove the corresponding direct edge in the graph, and in addition remove any indirect connections that the direct connection is a part of. This can be done because all the direct edges in the graph have pointers to the indirect edges they are a part of. All of these indirect edges must then be removed, while updating the sum edges.

If, on the other hand, a connection in the correlation that is to be removed is indirect, we have to remove one direct connection in the path of the indirect connection. How should the direct connection to be removed be chosen? This is a crucial choice, because if we choose to remove a direct connection, we reject something that the pair-wise matching has approved. Ideally it should not be necessary to remove any direct connections. The number of direct connections to be removed should be minimised, so we should be quite sure that

the connection we wish to remove is a "bad" one. This can be done by looking at the score of the connection (corresponding to the pair-wise score between the end-spots), and perhaps at the number of indirect connections the direct connection is a part of. If it is a part of *many* indirect connections, removing it will potentially cause large consequences, which may be both positive or negative. If a direct connection is a part of many connections in correlations that shall be removed, this is positive, because it helps keeping the number of removed direct connections down. On the other hand, if the direct connection is a part of many indirect connections whose status is unknown, removing the direct connection may remove connections that otherwise would be kept. These consequences must be considered when choosing which direct edge to break an indirect connection.

6.6.3.6 A global improvement of the connection removal

In order to minimise the number of direct connections that needs to be removed, it may perhaps be clever to find an order in which to remove the connections in a correlation that takes more global considerations into account. Consider the problem as: There is a collection of connections that must be removed. Call this collection C. The connections in this collection come from all the correlations that must be removed because of inconsistencies. When considering which direct connection it is clever to remove first, rank the direct connections by how many of the connections in C they are a part of. Consider the highest ranking direct connection: This is the most appropriate direct connection to remove, unless it is a part of many indirect connections outside of C. If this is the case, consider the second highest ranking direct connection etc.

The problem is to acquire all these numbers of occurrences of the direct connections in C. We have the total number of connections that a direct edge is a part of, but we do not know how many of these connections that are in C.

Let there be an initially empty list of direct connections L_d, where each direct connection shall eventually get a number corresponding to the number of connections in C it is a part of. To construct this list, we go through all the correlations that are in conflict with other correlations.

In more detail; look separately at each inconsistent set. In an inconsistent set, one correlation is chosen as the "valid" one. Examine all the other correlations, the rejected ones. Each of these correlations contain connections that are in the set C. The connections may be a direct one, C_i^d, or an indirect one, C_i^{id}. For each direct connection e either from a C_i^{id} or directly as a C_i^d, increase e's counter in L_d with one (if $e \notin L_d$, add e to L_d). See Figure 6.4.

When the list L_d is computed, we have an overview of all the direct connections that are involved in rejected connections, and thus rejected correlations. We may now choose the most appropriate direct connection to remove, based on a more global view.

When a direct connection is removed, we must remove all connections that contained this one, and update the correlations. We must also update L_d, because connections may (and probably will) disappear, and thus many items in L_d will get a lower number.

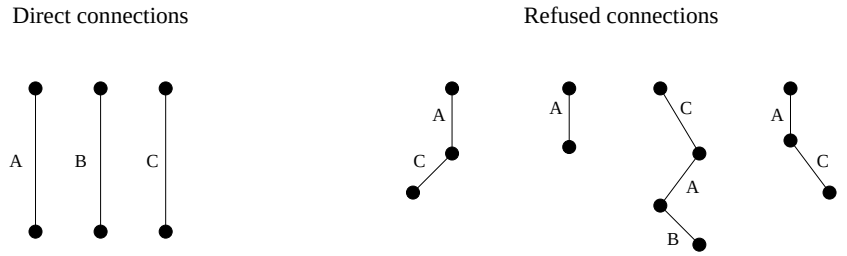


Figure 6.4: Figure showing the idea of removing direct connections with a more global view in mind. The constructed example shows that the edge A is contained in all four refused connections, while C is contained in three, and B is only contained in one. Removing A, will thus remove all the refused connections, while a removal of B only will remove one. A seems to be the most appropriate direct connection to remove, unless it also is a part of many connections that are not refused.

6.6.3.7 Advantage of ambiguous pair-wise matchings

At this point it seems quite clear that it will be an advantage to have alternative matchings from the pair-wise procedure. We may for example have a "preferred" match between two points, and some "backup" matches as well. In this way, when removing a direct edge, we may check if there are alternatives that may be introduced, without creating inconsistencies. In this way we may have a chance to avoid decreasing the matching efficiency when removing direct edges.

6.6.3.8 Actual removal of a direct edge/connection

The actual removal of a chosen direct connection is the same for the two possible scenarios: Removing a direct connection because it is a part of a rejected correlation, or removing it because it is a part of an indirect connection that is a part of a rejected correlation. As explained, when removing a direct connection, the indirect connection that the direct connection is a part of, must also be removed.

The direct connection removal makes us also remove all other connections the direct connection was a part of. When this "maintenance" is performed, obviously some connection(s) will disappear. This may lead to a change in, or the disappearance of a correlation. Theoretically, "valid" correlations may disappear as a consequence of the above mentioned. Figure 6.5 shows an example of the consequences of choosing a valid correlation from several (two) possible.

6.6.3.9 Locking chosen correlations

The fact that a removal of a direct connection has the potential of destroying other correlations, leads to the exploration of a way of "locking" correlations, which implies "locking" connections, direct as well as indirect ones.

- If a correlation or a connection for some reason should not be altered, or at least not removed, call it *locked*. This expression will also be used about the edges representing the different concepts.

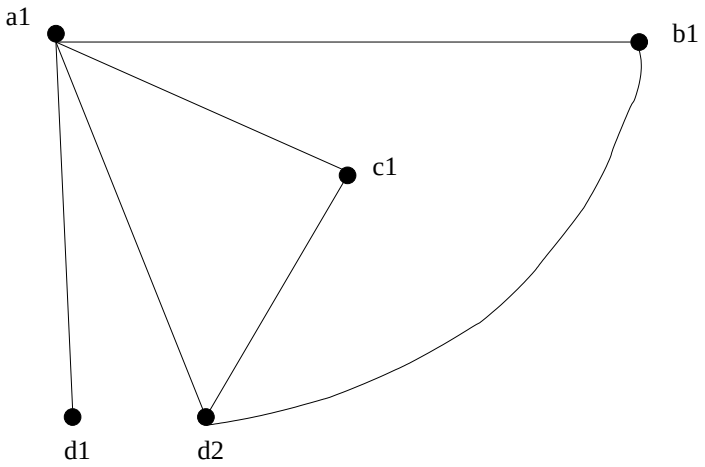


Figure 6.5: If the correlation $Co(a_1, d_1)$ is chosen as "valid", the correlation $Co(a_1, d_2)$ must be removed. $Co(a_1, d_2)$ contains $C(a_1, d_2) < a_1, b_1, d_2 >$, $C(a_1, d_2) < a_1, c_1, d_2 >$ and $C(a_1, d_2) < a_1, d_2 >$. Each of these three connections must be removed (in a suitable way). In practise, by removing edges.

This locking has consequences for the above explained procedure. This is because if an edge can be locked, one may risk that the correlation that at a certain point seems most appropriate to remove, is locked. Then we must, instead of removing this correlation, let it be the one to keep. In the case when we have an inconsistent set of correlations, we must remove all but one. If there exists several locked correlations in this set, it will be impossible to make it consistent, because we have earlier said that these edges should be kept.

The most straight forward way of choosing which correlations etc. that should be locked, is to simply at each stage lock the correlation that is chosen to be valid. The underlying connections of the valid correlation may not necessarily be locked, it is sufficient to be conscious about the fact that not all connections in the valid correlations can be removed.

6.6.4 Evaluating the result of the multiple matching

The graph algorithm described in this section is implemented, and some results are presented here. The starting point of the algorithm is the pair-wise matching between all pairs of gels. Ideally a pair-wise matching algorithm should have been implemented as a part of the system, but instead the gel matching program PDQuest is used.

6.6.4.1 Number of removed direct edges

It is interesting to observe some numerical results, since there are many choices that have to be made. One of the most crucial steps is the removal of direct edges, which represent a pair-wise match. The number of such removals should be minimised, and may be a good measure of the performance of the graph approach, when trying different variations of the algorithm. This assumption however, depends on another assumption, namely that the pair-wise matchings

actually are reliable. If this is not the case, it may be positive to remove direct edges, assuming we have the ability to recognise the truly bad ones.

6.6.4.2 Average percentage coverage on the proteins/spots

Another important performance measure is the percentage coverage (pc-score) described in Chapter 4.6.3. This score must be evaluated with caution, because a high average percentage coverage does not necessarily indicate a better result then a lower percentage coverage. To understand this, consider the following situation:

When a correlation is decided to be removed, because it is not consistent with another correlation, i.e. there is an inconsistency, the logical step is to remove all the connections in the correlation. This means removing all of the indirect and direct connections between the two spots the correlation is between. When removing an indirect connection (represented as an indirect edge), we must make the path of the connection non-existent, in practise by removing a direct edge. What if we simply remove the representation of the connection (the indirect edge), but leave the direct edges intact. By doing this, we would say that this connection should not be used, and further, the correlation it was a part of should not be used. So the connections that lead to inconsistencies are neglected, but the direct edges are intact. In practise, such a procedure would appear to be sensible, because the average coverage would be higher than when the direct edges were removed. The reason for this is that a direct edge may be "rejected" at one stage, but accepted at another. But logically, this would be wrong, because if a direct edge is rejected for one connection, it should not be a part of any other connections either. And from a biological view one would agree. If it at some stage is decided that there is no direct match between two spots, we should not fool ourselves by using this direct match simply to establish a connection between other spots in order to increase the total score of the matching.

6.6.5 Testing different settings

The testing performed in this section is based on the elements presented in Section 6.6.4.

6.6.5.1 Setup

To evaluate the multiple matching procedure, pair-wise matching information is needed. This was acquired from PDQuest, by importing five relatively similar images into the PDQuest program (plots are shown in Figure 6.6), and performing a pair-wise matching between all these. There was no scoring of the pair-wise matching between the gels available from PDQuest, so one was constructed from the values quality and peak level for each spot. The score between spot A and B was defined as:

$$score(A,B) = -abs\left(\frac{quality(A)}{peakvalue(A)} - \frac{quality(B)}{peakvalue(B)}\right) \tag{6.3}$$

The result of the subtraction inside the parenthesis should be close to 0 if the spots are similar. The argument for this is that if two spots are similar,

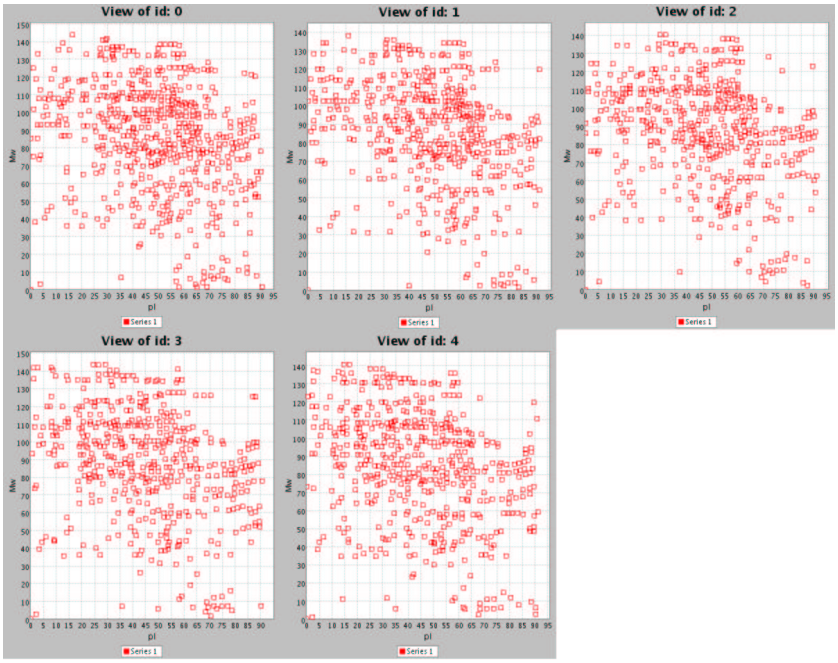


Figure 6.6: Plots of the gels used to test the graph-based multiple matching method.

the relation between the quality and peakvalue for each spot should be similar as well. This is perhaps not a valid assumption, but is merely an attempt to create a score between two matching spots. Taking the absolute value, gives values from 0 and up, where 0 is best. To make the highest numbers best, the negative value is chosen. Then the scores are from negative up to zero, where zero is best.

6.6.5.2 Different variables to test

There are several points in the algorithm where choices are made. Arguments may be given concerning which choices that are sensible, but it may be interesting to test different choices. The points where different choices are tested are:

- The discovered inconsistent sets must be removed. The order in which the sets are considered may be of importance. The different choices here are:
  - Should the sets be considered in an increasing or decreasing order with respect to the number of correlations in the sets.
  - Changing one set, may affect the size of other sets. Should the order of the sets be decided before any removal of connections, or should we choose on the basis of current size?
- When removing a correlation from an inconsistent set, connections must be removed. When a connection shall be removed, this is done by removing one of several possible direct connections. The direct connections are

Order of removing inconsistent sets	Data structure of incons. sets	Scoring of direct connections to decide removal	Resulting coverage (pc-score)	Removed direct connections
Increasing	Dynamic	Product	64.4%	152
Decreasing	Dynamic	Product	64.4%	149
Increasing	Dynamic	Num. part of	63.2%	152
Decreasing	Dynamic	Num. part of	63.0%	153
Increasing	Dynamic	Score	64.0%	147
Decreasing	Dynamic	Score	64.0%	140

Table 6.1: This table shows some results from several executions of the algorithm, with different settings. The definition of the pc-score is given in Chapter 4.6.3.

ranked in order to choose the most appropriate to remove. Which criterion should be used in the ranking? Alternatives are:

- The score of the direct connections. We should avoid removal of direct connections with high scores.
- The number of connections the direct connection is a part of. If it is a part of many indirect connections, removing it may have large consequences. Direct connections that are part of few other connections will be preferred.
- The product of the two above mentioned criteria. A low product gives a good candidate for removal.

6.6.5.3 Results

Table 6.1 shows the results from the experiment with five gels. Different settings are tried, and there are several aspects to comment upon here. A somewhat surprising result, is that a low number of removed connections, does not necessarily give a higher coverage than a high number of removed connections. This clearly indicates that some direct connections are more "important" than others.

The order in which the inconsistent sets are removed, does not seem to have too much impact on the final result. Only small changes occur when changing from an increasing to a decreasing order.

A factor that seems to have much impact, is the way the direct connections are scored, when one out of several shall be chosen. Choosing the product of the numbers of indirect connections containing the direct connection and the score of the direct connection, gives the highest coverage. Choosing the direct connections that are part of few other connections, gives a poor score, both concerning the coverage and the number of direct connections. Choosing on basis of the score of the direct connections gives a relatively high coverage, and a relatively few direct connections removed.

The combination of a decreasing order in processing the inconsistent sets and the use of the score of the direct connections, performs very well on the number of removed connections score.

A more sophisticated method for deciding the order in which the inconsistencies should be removed, like the one described in Section 6.6.3.6, would probably

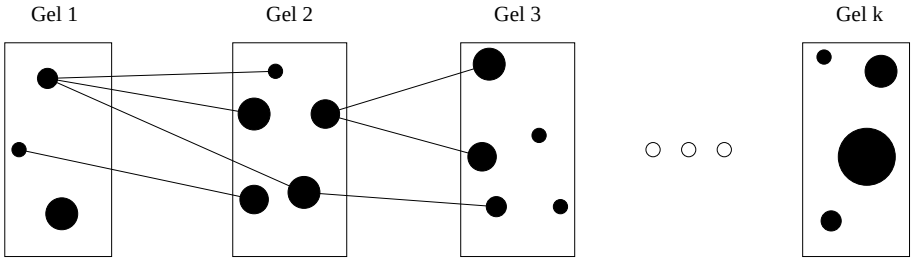


Figure 6.7: The "branching" factor (m) from each spot corresponds to the number of ambiguous matches between the spot and spots from another gel. In the figure m is three, there are k gels, and n is 5.

give better results, since the way the connections are removed obviously affects the results.

6.6.6 Problems

There are a few problems with the algorithm in Section 6.6. One major problem is that finding all paths between each spot in the different gels may in worst case be exponential in the number of gels. In Algorithm 6 on page 59, line 1, all correspondences shall be created. For two spots (u,v) this means that all paths between u and v should be found. In practise, this means that a very high number of paths will be explored, particularly if there are many, similar gels in question.

However, the running time relies on how many multiple matches each spot may have in a single matching gel. Let the maximum number of spots in a gels be n, and the total number of gels be k. If the maximum number of matches a spot in one gel may have against spots in another gel is m, the time complexity for finding all paths may be found as follows: There are a total of $O(n*k)$ spots. A search from a spot, finding all paths connecting this spot to other spots has time complexity of $O(m^k)$. This gives a total complexity of $O(n*k*m^k)$. See Figure 6.6.6 for a visualisation. If $m = n$, the running time is $O(n^k)$, but if m is small compared to n, the running time improves. This means that it is sensible to limit the number of ambiguous matches for a spot when the pair-wise matching is performed.

To speed up the algorithm at this point, we may limit the length of the paths to a fixed number. Call this maximum path length MPL. A slight modification must be done to Equation 6.2:

$$F(x) = \frac{MPL + 1 - x}{MPL} \tag{6.4}$$

An alternative to finding all paths between spot u and v, might be to only discover if there exists a path between u and v, that means to find only one of several possible paths. This makes the algorithm quite a lot faster, however, a lot of information is lost. For a spot pair (u, v), we do not know how many paths there are between the spots. As stated earlier, this number of paths gives an indication of how reliable a correlation is, and is used to weight the sum-edges, as explained in Section 6.6.2. This latter alternative makes the problem very similar to the transitive closure problem.

But even if only one path is enough, the complexity of $O(|V|^3)$(transitive closure) may quickly result in a very large problem as well.

If only one path between two vertices is found, then quite a lot will rely on this path. If there are inconsistencies, we have to choose between different matching possibilities. This means that the properties, i.e. score etc. of the path will decide if this path is chosen to be "valid". Therefore we should do more than just choosing a random path, but for example rather choose the shortest path. One may argue that the shortest path, (in number of edges in the path), is the most reliable. The best is of course to have all the paths between every vertex u and v.

The criterion for the shortest path between u and v may be the path with the fewest edges, or intermediate vertices. Such a path would intuitively be more reliable than a longer path. This may at least be a start. The "all pairs shortest path problem" can be solved in $O(|V|^3)$ time, |V| being the number of vertices in the graph, i.e. the total number of spots on all gels.

If the length of the path is simply the number of edges in the path, a breadth first search will reveal the shortest paths from u in $O(V + E)$ time. Running this on every vertex, gives quite a nice running time. Asymptotically $O(|V|^3)$, but due to the quite sparse construction of the graph, not that bad in practise.

Another serious problem occurs if two (or more) spots in the same gel have precisely the same coordinate value. This is a problem because it is a limitation with the 2D electrophoresis technology. The technology is based on separating proteins, and thus two proteins with the same coordinate value is not possible, because they will be registered as the same protein. So if such a situation occurs, where several proteins have the same coordinate value, they appear as only one spot.

6.7 Approach: Adding edges to an initially empty graph

This approach is also based on a graph structure and the same terminology as in Section 6.6 is used. The main difference is that the initial graph only consists of vertices, representing all the spots in all the gels, but no edges. The edges representing matches between spots are added one at a time, if and only if, the candidate edge does not lead to inconsistencies.

In this way the graph will be equivalent to a consistent matching all the time, and edges are added until no edge can be added without violating the consistency demand. Edges may be added independently of which gels they connect, meaning that this approach does not consider the gels in any specific order.

When an edge is added to the graph, it cannot be removed. This is a proposition that clearly differentiates this method from the one described in Section 6.6. As a consequence, when an edge is added, it does potentially inhibit other edges from being added. This means that we have to carefully decide which edge to try next. However, it may be an option in this method to allow the removal of already added edges, if there is an obvious improvement achieved by removing a certain edge.

6.7.1 Finding the next direct edge to add

We wish to find the "best" edge to add, the edge that does not inhibit us from eventually finding a good configuration of edges. An impression of what a good configuration implies is given in Chapter 4.1 and Chapter 4.6.3. If the pair-wise scores between all the gels are normalised, then an edge with a high score will be a good candidate to add.

When considering which direct edge to add next, we may simply choose the one with the highest score. To find the highest scoring direct edge candidate we can search through all the pairs each time. A better idea is to put all of the pairwise matches in a max-heap. In this way we can obtain the max scoring pair in $O(log(n))$ time, instead of $O(n)$ time.

6.7.2 Is this a "legal" edge to add?

When the "locally" best direct edge is found, based on for example the corresponding pair-wise score, we must update the graph, so that every new connection is found, leading to an update of the correlations.

When a candidate edge is presented, we must decide whether this new candidate contradicts any matching in the current graph. This can be done by checking different correlations, represented as sum-edges in the graph. Let the pair of spots that the new direct candidate edge connects be called (u,v). Let the gel that u belongs to be G_u and the gel that v belongs to be G_v. For the candidate direct edge (u,v) we must check that the following is satisfied: There exists no correlation between spot u and any spot in G_v, except for spot v, and there exists no correlation between spot v and any spot in G_u, except for spot u. If these two conditions are satisfied, we may check the next condition:

When a direct edge is added to the graph, new correlations may be created. This is because the new direct edge can create indirect paths between spots. Imagine that the direct edge that we consider is the "missing link" between two spots that have not earlier been linked. In such a case there may emerge inconsistencies somewhere else in the graph, not just locally, as the previous check revealed. There are thus two main checks that must be performed:

- The new edge (u,v) must not create local inconsistencies between the two gels it connects.
- The new edge (u,v) has the potential of creating new connections/correlations elsewhere in the graph than the one it creates between G_u and G_v, and these new connections/correlations must be checked so that they are not in conflict with any already existing correlations.

6.7.3 A more comprehensive choice of candidate direct edges

When choosing the next direct edge candidate, it may not be good enough to simply choose the edge with the best score. This edge may be a "bad" edge in a global view, i.e.. it inhibits us from achieving a good global solution. This may be illustrated using Figure 6.5, where we can see that the edge (a_1, d_1) and edge (a_1, d_2) are in conflict. At a certain point in this approach we may, or may not, see this conflict. Whether we become aware the conflict depends on whether

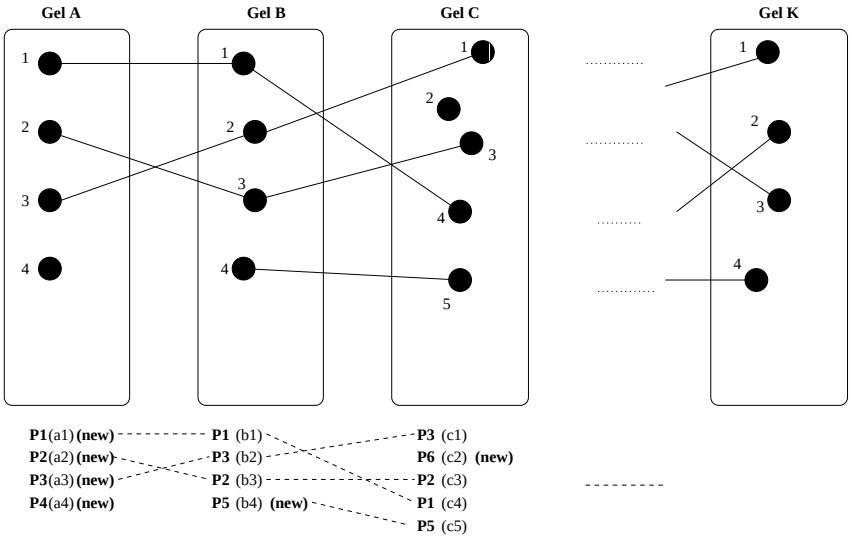


Figure 6.8: Showing the progress of starting with no edges, and adding one and one gel. Only the relations between the last added, and the new candidate are considered. Proteins, P_1... are added as unmatched spots occur.

one of the conflicting edges is already added when the other is considered, or not. If we become aware of the conflict, we do not have a choice, we must reject. But if non of the conflicting edges are added yet, we will not see the conflict when for instance considering edge (a_1, d_1), and thus perhaps choose this edge, without knowing that it (perhaps) inhibits a better configuration. This seems to be a clear disadvantage of this procedure compared to the one in Section 6.6.

6.7.4 A limited variant

Another variant of the approach may be to start with a pair of gels (as nodes in a graph). Next, we add all possible direct edges between this pair, corresponding to the pairs from the pair-wise matching. Then add one gel at a time and add new direct edges from the gels that are already added to the new one, constantly ensuring that the graph is consistent. This variant is similar to linear progression, as explained in Section 5.1.

This limited variant has the same problem as the previous one, it is not able to consider whether a new edge will create problems later.

To simplify this variant even more, we may only consider edges between the last added gel, and the new one. In this way we neglect many direct edges, and will not encounter the difficulties that are described earlier. This is because when only adding direct edges between one and one gel at a time, we will constantly extend a "path" from the proteins in the first gel and so on. This is shown in Figure 6.8. When adding a new gel, and there are unmatched spots, these will be assumed to be "new" proteins, not encountered earlier.

6.8 Comparing the two main approaches

The common feature of the two approaches is that the result is a subset of the full SRG. The main difference is that the first approach, the "remove edges" approach, may compare all the different possibilities, and make a qualified choice based on, among other things, the number of connections in each competing correlation. This is a clear advantage over the "add edges" approach, where we only see the correlations that have come so far. The advantage of this method though, is its simplicity. There is less information to keep track of, as we only look at one edge at a time.

6.9 Algorithms

Pseudo code algorithms for the two main approaches are shown in Algorithm 6, with subroutine Algorithm 7, and Algorithm 8.

Algorithm 6 Approach - Removing edges from the graph

Subroutine used: ***CLEAN(S)***
Input: The SRG / the pair-wise connections

1: Initialise all connections, and create all correspondences
2: Find all sets S of inconsistent correlations, Cos
3: **while** there are inconsistencies **do**
4: choose a set S_i to start validating *
5: **CLEAN**(S_i)
6: **end while**
{* = *The choice may be done by choosing the Co with the highest score*}

Algorithm 7 Subroutine **CLEAN**

Input: A (inconsistent) set of correlations (Co's)

choose best $Co_i \in S_i$
for all other $Co_j \neq Co_i \in S_j$ **do**
while Co_j is not empty) **do**
remove a connection from Co_j, by choosing and edge e **
update all connections containing e
{any Co may now shrink or disappear}
end while
end for
{** = *It may be better to look for an edge to remove directly, since an edge may be a part of several of the C's in* Co_j.}

Algorithm 8 Approach - Adding edges to an initially empty graph

Input: The SRG / the pair-wise connections

create graph G, with nodes of SRG
while G can be consistently enlarged **do**
  consider to add edge $e \in G$ (the "best")
  find new connections N
  if any $c \in N$ creates new correlation **then**
    check all for inconsistencies, if none, add **e**
  end if
end while

Chapter 7

From sequence to 2D gel pseudo-image

The aim of this chapter will be to explore the possibility of identifying proteins by analysing 2D gel images in combination with genome DNA sequence.

7.1 Introduction

When 2D electrophoresis is performed with the aim to identify proteins and protein modifications, the main property of 2D gels is used, namely the ability to separate proteins. In addition, visual inspection to suggest possible modified proteins is also enabled by the use of 2D gels. In order to identify proteins, meaning finding the sequence of the proteins and the corresponding sequence in the genome, proteins are usually removed from the gel, and sequenced in a laboratory.

The procedure of identifying proteins on a 2D gel should ideally have been possible without any lab work. Consider the following scenario: If some known proteins are added to the sample, before performing 2D electrophoresis, we may use these proteins to determine the pI and Mw of the other spots on the gel. These values are often called apparent focusing positions.

So for a random spot j in gel i, g_i^j we may find an apparent pI and Mw value. If we also know the gene sequences of the organism the sample originated from, we may calculate the theoretical pI and Mw values from these sequences. This prediction is quite accurate, and we can match the apparent values of spot g_i^j to all the theoretical values in the gene. We find a match, and the protein is identified.

Unfortunately, there are factors that complicates the prediction of a theoretical 2D image. Post-translational modifications and other modifications make it more difficult to accurately predict the mature proteins. In eukaryotes, where the genes contain exons and introns, we may have alternative splicing as well. This results in a situation where there are a large number of combinations of exons that actually results in a protein. In this study we are dealing with a prokaryotic organism, so fortunately this is not a problem.

7.2 Important aspects when going from a sequence to a protein spot

This section aims at describing some of the problems that may occur when trying to predict properties of a protein in an organism by the use of sequence information.

7.2.1 ORFans

When a genome is sequenced, in our case the M. Capsulatus, all the open reading frames (ORF) are discovered, and considered to be candidate genes. When the ORFs are compared to sequences in protein databases it is common to try to to discover sequence homologies between the ORFs and the proteins in the databases. But in practise, about 25% of the ORFs in the genome do not have any significant sequence similarity to proteins in the database. These ORFs are called *ORFans*.

This may of course mean that we have discovered new, unknown genes. However, according to [22] many of the ORFans may be old genes in decay. Examples of such decay is that stop-codons are inserted into the genes, and thus corrupts the synthesis of the proteins. So these ORFans are not expressed in the organism.

It should be considered whether these ORFans must be neglected when predicting the proteome of an organism. The pitfall is to neglect ORFans that actually represent new, unknown genes.

7.2.2 Protein folding

Protein folding makes some amino acids exposed, and other packed inside the protein. This may affect the properties of the proteins. When 2D gel electrophoresis is performed, the proteins are unfolded in a denaturation process. This is usually done by applying Urea in different concentrations. In [19] it is shown that the unfolding process is unsuccessful in some cases. This results in differently folded versions of the same protein on the gel. This may lead to "trains" of spots on the gel (see Figure 7.1), a phenomenon that earlier was claimed to be caused by post-translational modifications. This is still the case, but it is now realised that post-translational modifications are not the only source for "trains" in the gel. It is also experienced that one protein occurs several places on the gel, in no particular pattern.

7.2.3 Post-translational modifications

When an mRNA sequence is translated to a protein product, the same mRNA sequence may result in different proteins, depending on the state of the organism and perhaps also external factors. One of the reasons for this, is post-translational modifications. Such modifications to a protein occur after translation from mRNA to polypeptide. Modifications that occur simultaneously with the translation are called co-translational modifications.

A co- or post-translational modification to a protein most likely changes some of its properties. Changes may be in the protein's charge, mass, hydrofobicity etc. In the 2D gel context this means that a protein that is modified will move

on the gel according to the change in charge or mass. As a result, apparently different spots may originate from the same gene. An open question is if different modified variants of a protein should be assumed to be the "same" protein.

There are specific sequences in the polypeptide that tell the cell to add a certain post-translational modification. Many proteins are not functional without a specific modification.

Several hundred different modifications that can alter a protein after time exist, but some of them are more frequent than others.

7.2.4 Detection of post-translational modifications

An important part of any proteome study is to detect post-translationally modified proteins. This is because understanding such modifications are important in order to understand the processes that take place in an organism.

7.2.4.1 Inspection of 2D gels

The 2D electrophoretic separation of proteins is based on the isoelectric focusing point, and the mass of the proteins. As the proteins that are modified undergo a change in mass and/or isoelectric point, a modification will most likely be detectable in a 2D gel. To manually detect modified proteins in the 2D gel there are several approaches: Sequencing different spots that you suspect are modified versions of the same protein is an obvious option. Such a sequencing would perhaps reveal some of the modifications, but not all. Methods involving blotting of specific proteins may give a better overview of the modified versions of a certain protein.

As explained, modified proteins often occur as "trains" on the gel. See Figure 7.1. This may be used to automate the detection of modifications, and will be discussed in Section 7.3.

7.2.4.2 Sequence searching

Different modification sites, i.e. places in the protein sequence where a modification may occur, have different "signatures". Many of the most common co- and post-translational modification sites have motifs in the sequence that may allow for the prediction of modifications. This does of course assume that the amino acid sequence is available, and will only be an indication of where there may occur a modification.

Since a single protein may be modified in several different ways, one protein sequence may contain sequence motifs for several different modifications. So simply searching the sequence can only give a list of possible modifications. But in most practical situations there are only a minority of the modifications that are likely to occur.

Which modifications that are likely to occur in a specific context, must be decided by an expert.

7.2.4.3 Combining 2D electrophoresis with sequence searching

Since modifications to a protein in many cases will change the protein's isoelectric focusing point, and the mass, we may predict from the sequence how a protein will move on the 2D gel if it is affected by a modification.

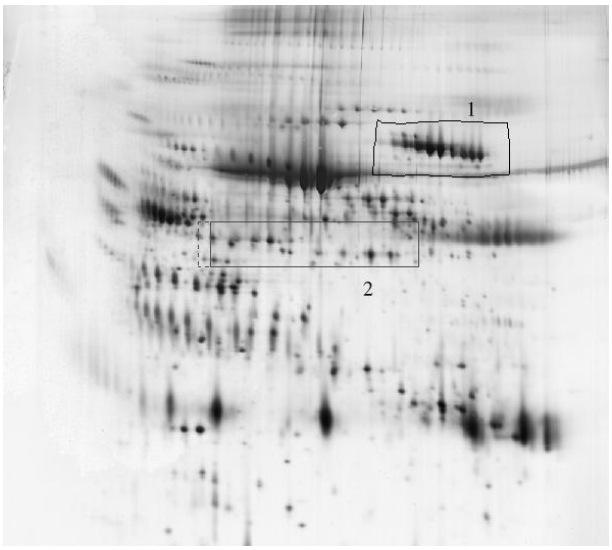


Figure 7.1: Example of 2D gel image. Area 1 showing a "train" of spots, area 2 showing more normal situation. Such trains are often a result of post-translational modifications to a protein. Another explanation may be an unsuccessful denaturation procedure.

When inspecting the 2D gel in search of modifications of a protein, the theoretical changes in pI and Mw for the different modifications may help to find modified proteins, and to state what kind of modification the protein has undergone. To use this methodology, we have to know the sequence and position of the specific protein.

7.2.5 Other modifications

There also exists other modifications such as the ones called post-tran*script*ional modifications. These are modifications that occur after transcription from DNA to RNA, but before translation from RNA to protein. There are three such modifications: 5' capping, addition of the poly A tail, and splicing. The process called 5' capping adds a "cap" to the 5' end of the mRNA molecule. This process is thought to improve the recognition of the mRNA by the ribosome. The adding of the poly A tail is an adding of adenine bases for an unknown reason. The splicing process splices the exons of the pre-mRNA into an mRNA molecule consisting of only exons. This process may combine the exons in different ways, and may thus lead to different proteins. The splicing process is not performed for bacterias as they have no introns.

7.3 Automatic detection of modifications in gel images

Detecting protein modifications is an important part of proteomics work. To find evidence for modifications, we must use sequencing or sequence recognition

methods on the proteins.

To find candidate proteins that may have been modified, we may try some automatic approaches.

7.3.1 Searching for "trains" in a single gel

Given a computer image of a 2D gel, we may search for constellations of spots that are similar to the ones in Area 1 in Figure 7.1. Looking at a single gel image, we may simply look for very close spots that form a straight line. This line of spots is often called a "train" of spots. The lines that may be formed are often horizontal, i.e. there has been a modification of the protein that induced a change in the proteins charge.

7.3.2 Detection through multiple matching

Looking at a series of gels, multiply aligned, we may have several situations with regard to proteins that exist in different variations.

- A protein may have marginally changed its position in the different gels due to a slight modification, so that it is still registered as the same protein. In this case we may search for a protein that show a tendency to deviate in its focusing position.
- Different modified versions of a protein may have been detected as different proteins. We may then search among the proteins, for a subset that forms a line, as described in 7.3.1.
- The multiple matching probably revealed ambiguities. Some of these may have been caused by a situation where one or several modifications have occurred.

7.3.3 Limitations when searching only on gel image(s)

If the modifications that have occurred, have caused shifts in both charge and mass, and the shifts do not form a line, we are not able to discover these simply by looking at the gel images. Such situations will look just like any other distribution of spots.

7.4 Predicting the isoelectric focusing point from an amino acid sequence

The isoelectric focusing point of a protein, is the pH where its net charge is zero. This focusing point is also called the pI value of the protein, and may be predicted from the amino acid sequence. The method used in this thesis is loosely based on the *protstat* program in the EMBOSS[1] package. Also see article [24].

To predict the isoelectric focusing point for a given protein sequence, we need to count the number of C, D, E, H, K, R, Y amino acids. These seven

[1] http://www.hgmp.mrc.ac.uk/Software/EMBOSS/

Amino acid	pKa value
Lysine (K)	10.5
Arginine (R)	12.0
Histidine (H)	6.5
Aspartic acid (D)	4.5
Glutamic acid (E)	4.5
Cysteine (C)	9.0
Tyrosine (Y)	10.5
N-terminal	7.5
C-terminal carbox. group	3.5

Table 7.1: Table showing some experimental pKa values for the electrically charged, hydrophilic amino acids

amino acids are the charged ones, and the charge decides the focusing points. The positively charged amino acids are H, K and R. The negatively charged amino acids are C, D, E, Y. It is interesting to notice that all of these amino acids are hydrophilic, meaning that these are not amino acids that typically place themselves on the inside of proteins.

In addition, the N-terminus and C-terminus are charged. The N-terminus is positive and the C-terminus is negative.

At a given pH, a certain percentage of a residue will be charged, while the remainder is neutral. With this in mind, we may describe the charge of a particular amino acid by its partial charge.

When the partial charge of an amino acid at a certain pH is calculated, this partial charge should be multiplied with the number of copies of this particular amino acid that occur in the protein. This assumes that a residue's impact on the total charge is invariant of the residue's position in the protein. This assumption may in some cases be incorrect.

The partial charge of the N- and C-terminus must also be added.

What makes us able to predict the net charge of a protein sequence, is the knowledge of the pKa values of the amino acids, the N-terminal amino acid, and the C-terminal carboxyl acid. If the simulated pH is equal to the pKa value of an amino acid, then this amino acids' net charge will be zero. This means that the amino acid may have positive and negative charges, but they equilibrate each other so that the net charge is zero. The pKa values are obtained experimentally, and are not absolutely correct, and may also vary depending on the obtaining procedure.

pKa values used are shown in Table 7.1.

To calculate the partial charge of one of the nine possible entries in Table 7.1, we calculate a property called the protonated fraction (PF). This value uses the pH currently simulated, and the pKa value of the current amino acid.

$$PF = \frac{10^{-pH}}{10^{-pH} + 10^{-pKa}} \tag{7.1}$$

Now, the net charge of the positively charged amino acid is simply the PF value, and for the negatively charged ones 1-PF.

This partial charge is calculated for each of the different charged residue types, and multiplied with the frequency of the corresponding residue in order

to get the total net charge of the protein.

Now we may simply simulate different pH values, and calculate the net charge of a protein. To find the isoelectric point, i.e. where the net charge is zero, we may proceed as shown in Algorithm 9.

One step of the algorithm worth noticing is line 7. When the net charge of a protein is calculated at a certain pH, this charge will be negative or positive. If the charge is negative, it means that the simulated pH is to high, and vice versa. To find the pH where the net charge is close enough to zero (the limit on line 2), a binary search is appropriate. For example if the initial pH is set to 7 and the resulting net charge is positive try to set the pH to 14. Then the net charge is probably negative, and the pH is set to 10.5 etc.

Algorithm 9 Calculate pI of amino acid sequence

Input: An amino acid sequence
Output: A pH value, the pI of the amino acid sequence

```
1: pH=set start pH (typically 7)
2: limit=set the allowed deviation from zero
3: find the frequency of the charged amino acids in the sequence
4: repeat
5:    nc = net charge of the sequence
6:    if (! nc inside limit) then
7:       pH=new simulation pH
8:    end if
9: until (nc inside limit)
10: output pH
```

7.5 Protein expression of predicted genes in a genome

An important question is if the expression levels of a protein can be predicted from its sequence.

In quantitative analysis, like micro-array and 2D gel experiments, the result dependents on the expression levels of the RNA/protein in the sample. If a gene is not translated and transcribed into a protein at all, it will certainly not appear on a 2D gel. In addition, as mentioned earlier, low abundant proteins tend to not appear on the 2D gel. The proteins that appear on the gel may have different intensities depending on their abundance.

The expression levels of the proteins in a cell change constantly, according to the cell cycle, the state of the organism, external factors etc.

7.5.1 Codon bias

An indication of the protein expression of a certain gene is its codon bias. The genetic code allows several different 3-tuples of nucleotide bases to be encoded into the same amino acid. But different organisms have a tendency to "prefer" a certain codon for each amino acid. By examining the sequences of highly expressed proteins, we can calculate the frequency of every codon in these proteins,

which gives us knowledge of which codons that are preferred, or biased.

It is often found ([15], [16]) for expressed genes, that the level of expression correlates with what kind of codons they use. If a gene mostly uses the preferred codons for each amino acid, it is more likely to be highly expressed than a gene not using the preferred codons. The reason why some genes are expressed higher is probably not a result of the fact that the genes show a codon bias. It is more likely that a codon bias is a result of the fact that a gene is highly expressed[16].

There are two main theories regarding biased codon usage [21]:

- Codon bias reflects a selection process that has lead to a more fit organism. This is the most common theory.
- Codon bias is a result of non-random mutations that occur independently of the genes. This means that there are certain types of mutations, for example TCC to TCT, that are more common than others, and thus leads to a biased codon usage. If this theory was correct, we would not expect to observe a difference in the degree of codon usage between highly expressed genes and other genes. We would not even expect to observe a difference in the degree of codon usage between introns and exons.

Experiments [21] suggest that perhaps both theories are correct. When using the index described in Section 7.5.1.1, a significant difference is revealed between mammals and other eukaryotic and procaryotic organisms. While there is a correlation between the mRNA levels and codon bias in procaryotic and most eukaryotic organisms, no such correlation is found in mammals. No difference between codon usage in introns and exons are found in mammals either.

Regarding the first theory, an important question is why some organisms prefer certain codons? As there is often a positive correlation between the codon bias of a gene and the expression level, we would assume that the reason for biased codon usage may lie in an organism's need for some genes to be expressed in a high amount. This is one of the alternative explanations for the first theory, namely that different factors that make an organism fit, have contributed to a codon bias. Another important theory, that also has roots in experimental results, is that the level of different types of tRNA in a cell has led to a codon bias. The tRNA is the molecule that is essential in the translation of mRNA to protein. The tRNA molecule has an anti-codon that is attached to a codon on the mRNA, as the translation goes along. The tRNA molecule also carries the corresponding amino acid (see Figure 7.2). If there in an organism is an over-representation of tRNA for a certain codon, say CTT for Leucin, the organism will have problems if all the Leucin amino acids are encoded as other codons than CTT. If this is the case, the natural selection will favour mutations of Leucin codons that results in a codon that is similar to CTT. It is worth noticing that the tRNAs are not exclusive in their binding to codons on the mRNA. This means that a given tRNA anti-codon can attach to slightly different codons. However, not all tRNAs can bind to all codons, so there is a need for the "right" tRNA.

According to [21], it is also experienced that there is a negative correlation between the degree of codon bias and the length of the protein. The reason for this phenomenon is unknown.

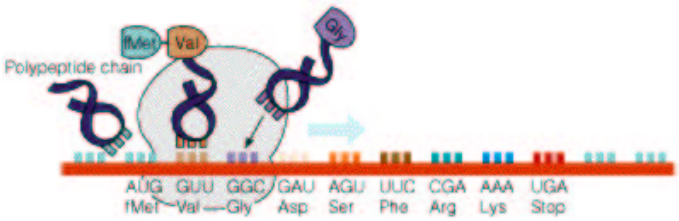


Figure 7.2: Figure showing the translation from mRNA to protein. The tRNA uses its anti-codon to attach to the current mRNA codon. The amino acids become attached to each other as the translation goes along.

7.5.1.1 Codon Adaptation Index

The codon bias can be measured by calculating the codon adaptation index (CAI), used in [16], a measure of to which extent a gene uses the most common codons for each amino acid. The index can have values between zero and one. A CAI value of one, means that a gene only uses the most common codons in the reference set. A CAI value of zero means that a gene only uses codons that are not used at all in the reference set.

To calculate the CAI, a preprocessing step is performed. A reference table is constructed from a set of highly expressed genes from the organism in question. This table shows the "relative synonymous codon usage" (RSCU). An RSCU value for a codon is simply the observed frequency of that codon divided by the frequency expected under the assumption of equal usage of the synonymous codons for an amino acid.

$$RSCU_{ij} = \frac{X_{ij}}{\frac{1}{n_i} * \sum_{j=1}^{n_i} X_{ij}} \tag{7.2}$$

In the formula, the 2-tuple $_{ij}$ means the j'th codon of the i'th amino acid. So X_{ij} is the number of occurrences of the j'th codon in the i'th amino acid, and n_i is the number of alternative codons for the i'th amino acid.

So there is one RSCU value calculated for each codon. If all codons for an amino acid are equally used, all RSCU values would be 1. A higher RSCU value for a codon means that this codon is used more than if all codons were used equally much.

Next, we define the relative adaptiveness of a codon, call it w_{ij}.

$$w_{ij} = \frac{RSCU_{ij}}{RSCU_{i<max>}} = \frac{X_{ij}}{X_{i<max>}} \tag{7.3}$$

The 2-tuple $_{i<max>}$ means amino acid i and codon max, where max is the index of the codon that is most used.

The Codon Adaptation Index for a gene is then calculated as the geometric mean of the RSCU values for each of the codons in the gene, divided by the geometric mean of the maximum RSCU values that each residue in the sequence could have.

$$CAI = \frac{CAI_{obs}}{CAI_{max}} \tag{7.4}$$

where

$$CAI_{obs} = (\prod_{k=1}^{L} RSCU_k)^{\frac{1}{L}} \tag{7.5}$$

and

$$CAI_{max} = (\prod_{k=1}^{L} RSCU_{k<max>})^{\frac{1}{L}} \tag{7.6}$$

L is the number of codons in the sequence. In CAI_{obs} the geometric mean of the RSCU values corresponding to the used codon is calculated. In CAI_{max} the geometric mean is also calculated, but if the codon is for example CTT, coding for Leucin, the RSCU value used in the formula is not the one corresponding to the CTT codon, but the highest of all RSCU values for the amino acid Leucin.

A few special rules are also applied. If a codon, *cod*, is never used in the reference set (when calculating the RSCU values), its X value is zero. If *cod* now occurs in a sequence for which the CAI is to be determined, this will result in a value of zero for the CAI of this sequence. To avoid this, if an X value is zero, it is assigned the value 0.5. The other special rule is that the number of ATG and TGG codons are subtracted from L, since their RSCU values are one, and thus do not contribute to the CAI.

An alternative formula for CAI, using the calculated w-values is:

$$CAI = (\prod_{k=1}^{L} w_k)^{\frac{1}{L}} \tag{7.7}$$

A problem with this equation is that when L is large, it is quite likely that there is a floating number overflow when the index is calculated using a computer. Multiplication of a series of numbers < 1 results in a small number that the computer may not be able to represent. This was experienced with the longest genes in M. Capsulatus (see Chapter 8.4.3).

An alternative to Equation 7.7 that eliminates the product by adding the natural logarithm of the value of every residue is obtained in the following way:

$$CAI = (\prod_{k=1}^{L} w_k)^{\frac{1}{L}} \tag{7.8}$$

$$ln(CAI) = ln((\prod_{k=1}^{L} w_k)^{\frac{1}{L}}) \tag{7.9}$$

$$ln(CAI) = \frac{1}{L} ln(\prod_{k=1}^{L} w_k) \tag{7.10}$$

$$ln(CAI) = \frac{1}{L} \sum_{k=1}^{L} ln(w_k) \tag{7.11}$$

$$e^{ln(CAI)} = e^{\frac{1}{L}\sum_{k=1}^{L} ln(w_k)} \tag{7.12}$$

$$CAI = e^{\frac{1}{L}\sum_{k=1}^{L} ln(w_k)} \tag{7.13}$$

The calculations are based on a set of reference sequences, and the choice of these sequences is important to the result because they set the standard for what the measurements are relative to. An illustrative example of this may be the following: If the reference CAI values are calculated for amino acid codon variants from a set of sequences from organism A, and then these values are used to calculate CAI scores for sequences in organism B, what does this tell us? If a gene in organism B gets a high score, it means that it uses codons that the reference organism(s) prefer(s). This may be useful knowledge, because if this "foreign" gene is used in the reference organism, proteins may be produced.

But this does not necessarily tell us anything interesting about organism B. This organism may have a completely different codon usage bias, so what is meaningful is to calculate the reference CAI values from the same organism that contains the gene sequences that are examined.

7.5.1.2 Codon Bias Index

Another way of calculating the codon bias, is the Codon Bias Index (CBI), used in [15] to evaluate the codon selection in yeast. The CBI has a max value of one, which indicates that the gene only uses codons that are "preferred" (explained later). A negative CBI can also occur, this would indicate that a gene non-randomly chooses codons that are not preferred. A value of zero indicates random codon use.

This method differentiates between codons by classifying them into preferred or non-preferred codons. The definition is that a codon is preferred if it is used more than 85% of the time when the corresponding amino acid is encoded. The actual index is a fraction where the numerator is the total number of times the preferred codons are used in the protein subtracted by the number of such usages expected if the code were read randomly. The denominator is the total number of amino acid residues in the protein (excluding Met, Trp and Asp) subtracted by the same random expected usage of preferred codons as in the numerator. The reason for excluding Met and Trp, is that there are only one codon for these amino acids. Asp was excluded because none of the codons for Asp were used more than 85% of the times. Using another organism will of course not necessarily exclude Asp for this reason.

In more detail:

$$CBI = \frac{TotNumPref - TotNumPrefExpect}{TotNumRes - TotNumPrefExpect} \tag{7.14}$$

where

$$TotNumPrefExpect = \sum_{a \in AA} Num(a) * fracPreferred(a) \tag{7.15}$$

The set AA is the set of all amino acids subtracted by Met, Trp and Asp. Num(a) is the number of occurrences of the amino acid a in the protein. fracPreffered(a) is the fraction of number of preferred codons divided by total number of codons for amino acid a.

7.5.1.3 Comparison of the two indexes

The two indexes measure the same thing, but in slightly different ways. The main difference is that the Codon Bias Index procedure separates highly used codons into one category, but there is no differentiation between codons within this category. With the Codon Adaptation Index there is a continuous rating of all codons. The latter seems more sensible because with the CBI method it is crucial whether a codon is used in 84.9% or in 85.0% of the times, while this is not so important in the CAI method. In reality the difference of 0.1% is neglectable. In Figure 8.6 and Figure 8.7 in Chapter 8.4.3 plots of the two indexes are shown.

Chapter 8

Methodology for identification of proteins

As an overall task for the GABI project is to identify the genome and the proteome(s) of M. Capsulatus, it may be interesting to create a link between the genome and the proteome. The representation of the proteome may be a 2D gel.

It seems clear that to come closer to the identification of proteins without actually sequencing proteins, we must search the genome. For the bacterium Methyloccus Capsulatus the whole genome is sequenced. There are about 4000 open reading frames, which may be coding for proteins.

A figure suggesting a work-flow for combining genome sequence with 2D gel images is shown in Figure 8.1.

8.1 Calculating a complete "theoretical" 2D gel image

To create a synthetic 2D gel image of M.Capsulatus, we may consider all the ORFs in our ORF database (ORFdb). For each of the entries in the ORFdb, calculate the molecular weight and the theoretical isoelectric focusing point.

This synthetic image will be equal to a situation where all genes are expressed, and none of them modified. Even though this is not a realistic situation, we may use this as a basis. It will be important to let the user interact, and be able to change the image.

An example of the total theoretical 2D gel image of M. Capsulatus is shown in Figure 8.2. To create this synthetic image, all the predicted genes in the genome of M. Capsulatus were translated to protein amino acid sequences. There are currently (2002 / 03 /21) 3857 predicted genes in M. Capsulatus. These genes are predicted by the program Glimmer.

Some of the proteins encoded in the bacterium have a signal sequence. This sequence is removed from the protein before it finds its position in the cell, so this should also be taken into consideration when predicting the theoretical image of M.Capsulatus. This is further discussed in Section 8.2.

To calculate the focusing points for these theoretical proteins on a 2D gel, the

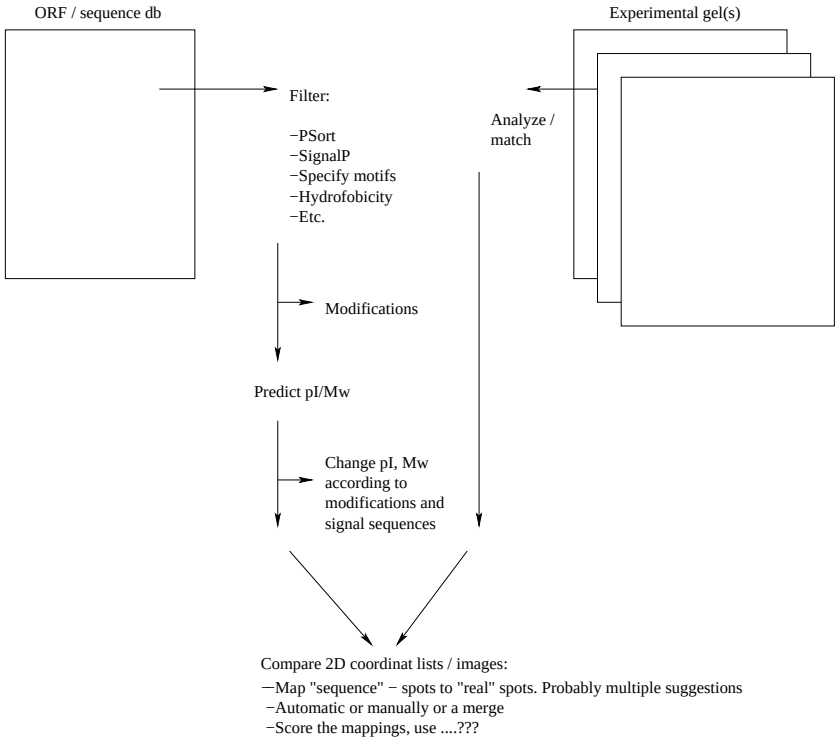


Figure 8.1: Work-flow for identification of proteins

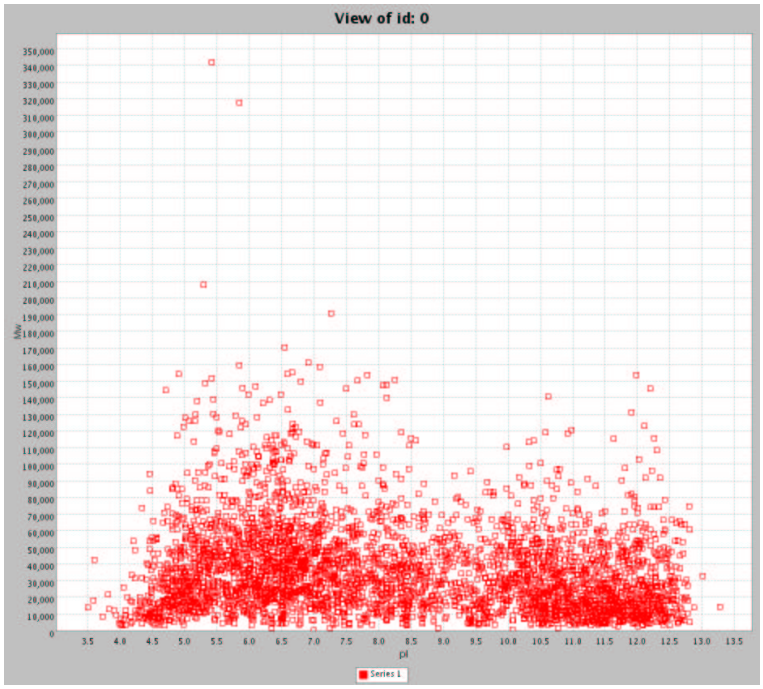


Figure 8.2: Theoretical 2D gel image of M.capsulatus

theoretical masses and isoelectric focusing points must be predicted. To predict the mass of an amino acid sequence, the molecular weights of all residues in the sequence are summarised. Predicting the isoelectric focusing point is explained in Chapter 7.4.

8.2 Using biological information to reduce / expand the theoretical 2D gel image

The theoretical 2D image may be reduced in order to for example only display a certain type of proteins, or expanded to for example show possible modifications on the proteins.

8.2.1 Reducing the complexity

As we learn more about typical characteristics of different proteins, we may use this knowledge to specialise the theoretical 2D image. This may for example be done if the user is able to find a profile or another property that is common for the proteins of interest. "The proteins of interest" may be different groups, sub-cellular locations are perhaps the most obvious grouping of the proteins. The reason for this is that 2D gel experiments are often performed on separate compartments of the cell.

A profile that is typical for a special group of proteins, for example outer membrane proteins, may be used to screen the ORFs, and then only display/calculate

the interesting theoretical proteins. The problem is of course whether such a profile is known, or even possible to acquire.

In the case of the outer membrane, inner membrane, and periplasmic proteins, a criterion that may be used to predict proteins that certainly is likely to be found in the outer membrane, is the presence of signal sequences. Signal sequences are sequences that tell the cell that a certain produced protein shall be transported to another area. For procaryotic organisms, the proteins belonging to the inner membrane, periplasma and outer-membrane contain signal sequences.

There exist programs to predict the signal sequences and/or the cellular locations of proteins. One example is PSort [13], which predicts signal sequence and cellular location. The predictions are done by evaluating characteristics of the amino acid sequence, and the program is formulated as a set of rules. In [20], it is claimed that PSort predicts the correct cellular location of 83% of the proteins in Gram-negative bacteria. In the designed program that calculates the synthetic 2D image, PSort is used to predict cellular locations of proteins.

SignalP is another example that predicts signal sequence[14].

Not only specific sequences or motifs may be used. The hydrofobicity of a protein may be predicted, and may give useful information about what kind of protein we are dealing with.

8.2.2 Expanding the theoretical image

So far, methods for reducing the complexity of a certain theoretical 2D gel image have been considered. Unfortunately, there are also factors that complicate the image that may be considered. If the "biologist" suspects a post-translational modification for a spot that he has on his gel, we may wish that this also appeared on the theoretical gel.

As mentioned, a wide collection of known post-translational modifications exist. The change that these modifications impose on the focusing position of protein is also known. The problem is then to decide if a certain modification will occur in a protein. There exist motifs for most modifications, but a protein may very well contain such a motif without necessarily being modified.

So what we can do is to suggest possible modifications for a protein. Hopefully this will help the biologist to better be able to decide if the things he sees on his gel may be caused by a modification.

8.3 Using the theoretical image

One, quite obvious utilisation of a theoretical 2D gel image calculated from the gene sequences, is to manually search for theoretical spots with the desired property. For example if a 2D gel experiment show a very interesting protein, of unknown sequence, the theoretical gel may be examined to search for this particular protein. Results from such a procedure is described in Section 8.4.

Another utilisation is to automatically match the theoretical gel with experimental gels, either pairwise, or in a multiple manner. In order to get meaningful results from an automatic approach several criteria have to be satisfied:

- The theoretical predictions must be relatively accurate. An indication of whether this is true is given in Section 8.4.

- The prediction settings, i.e. settings related to prediction of sub-cellular location, required level of the codon bias indexes, and the pI/Mw range on the theoretical gel, must be considered carefully in order to create a theoretical gel that is as similar to the experimental gel as possible.

8.4 Empirical testing of the synthetic 2D image

To test the accuracy of the calculated 2D image, we chose 5 gene sequences from the M.Capsulatus genome database as reference genes to calculate the Codon Adaptation Index and the Codon Bias Index. These values are used to filter away genes from the synthetic 2D image that most likely are not expressed highly enough to appear on a real 2D gel. The reference gene sequences were the ones coding for $MeDH_{\alpha}$, sMMO A_{γ}, MopB, MopF and MopG. These proteins were chosen on the basis of their expression levels, that by visual inspection of experimental 2D gels from M. Capsulatus seemed high.

From 2D gel runs of samples from different parts of the M.Capsulatus, we have partially annotated gel images [18]. The spots that are annotated are identified by mass spectrometry or N-terminal sequencing. The proteins were partitioned into soluble or outer-membrane fractions. The soluble fraction is supposed to contain the proteins that are in the cytoplasm or periplasma.

8.4.1 Test setup

To mimic a real situation, a selection of 20 spots from actual 2D gels with an apparent pI and Mw are chosen to be identified. The task is to find these spots in the synthetic gel, only using the information available from a real experiment, i.e. the apparent pI and Mw values and the compartment the spots occurred in. The task will be performed by tuning the settings according to the spots that are searched for. This will include limiting the pI and Mw range that is displayed, only displaying proteins predicted to belong to certain cellular locations, and only displaying proteins with a CBI and CAI value higher than certain limits.

In order to be able to verify the identification of the spots that are found, the 20 chosen spots have been sequenced. Information about the chosen spots is found in Table 8.1

8.4.2 Protein identification results

The overall settings on the synthetic gel that were applied to be able to identify spots are shown in Table 8.2.

$MeDH_{\alpha}$ was possible to identify with the quite strict settings depicted in Table 8.2. As Table 8.1 shows, the predicted focusing point deviated quite much from the experienced focusing point. By using a wide display range (observed pI $\pm$ 0.5 and observed Mw $\pm$ 10000) and the strict settings, $MeDH_{\alpha}$ was identified. However, in a real situation, the user could not have chosen such a strict limit on the CBI and the CAI values, and probably not restrict to only proteins predicted to be periplasmic.

$MeDH_{\beta}$ was more easily identified. Using not so strict CBI and CAI levels, range settings of observed pI $\pm$ 0.2 and observed Mw $\pm$ 5000, there were only 8

Name	Fraction	O. pI	O. Mw	N-term. seq	T. pI	T. Mw
$MeDH_{\alpha}$	Soluble	6.25	58	MQICKLAS	6.674	66.3
$MeDH_{\beta}$	Soluble	9.0	15	MMQKTSFV	8.969	12.5
sMMO A_{α}	Soluble	6.1	53	MALSTATK	6.651	61.5
sMMO A_{β}	Soluble	5.3	43	MSMLGERR	6.187	50.6
sMMO A_{γ}	Soluble	9.0	20	MAKLGIHS	9.324	28.3
sMMO B	Soluble	4.6	18	MSVNSNAY*	4.657	16.0
MspA	Soluble	5.2	38	MDGHGRLV	5.878	68.4
MspB	Soluble	5.9	26	MSVLVGKP	NF	NF
MspC	Soluble	5.0	18	MSDGKQDI	5.079	13.9
MspD	Soluble	5.3	18	MYESLLRF	9.618	19.7
MopB	O.membrane	4.75	35	MVKRTLMA	5.026	36.1
MopF	O.membrane	5.35	26	LLTASIAA	5.49	30.2
MopG	O.membrane	4.4	46	MMTQTAKL	4.821	49.0
MopE	O.membrane	5.4	61	MRDTMNEK	5.52	54.7
Gene 7593	O.membrane	4.75	50	LCGFCANV	6.253	58.4
Gene 5938	O.membrane	5.2	47	VLSFIRGE	5.411	56.8
Gene 6155	O.membrane	5.05	18	MSDGKQDI	5.079	13.9
Gene 6156	O.membrane	5.3	18	VPPSRPRL	9.618	19.7
Gene 5633	O.membrane	6.9	42	LLIFSTSG	9.283	51.2
Gene 5396	O.membrane	5.6	48	MRHRKAGR	10.938	13.6

Table 8.1: Table showing data about genes selected to be identified by using the synthetic gel image generated from the genome. The Mw values are in kDa. NF means not found, and may occur because the experimental 2D gels were annotated using an older sequence database than the one the synthetic gel was calculated from. Some of the names are not official annotation names. * The sMMO B was not found in the latest database, the one used in the creation of the synthetic gel. This is due to an erroneous gene prediction. The data is from a previous assembly.

Name	Settings (CAI, CBI, PSort)	Range(pI±, mW±)
$MeDH_{\alpha}$	CAI=0.7, CBI=0.4, Peri. fraction	0.5,10000
$MeDH_{\beta}$	CAI=0.7, CBI=0.4, Inner membrane	0.2,5000
sMMO A_{α}	CAI=0.7, CBI=0.4, Inner membrane	0.7,10000
sMMO A_{β}	CAI=0.6, CBI=0.3, Cytoplasmic fraction	0.9,10000
sMMO A_{γ}	CAI=0.6, CBI=0.3, Inner membrane	0.5, 10000
sMMO B	NA	NA
MspA	CAI=0.5, CBI=0.1, Inner membrane	0.7, >30000
MspB	NA	NA
MspC	CAI=0.6, CBI=0.3, Cytoplasmic fraction	0.1, 5000
MspD	CAI=0.5, CBI=0.4, Cytoplasmic fraction	>4,2000
MopB	CAI=0.6, CBI=0.4, Outer membrane	0.3,5000
MopF	CAI=0.6, CBI=0.4, Outer membrane	0.2,5000
MopG	CAI=0.6, CBI=0.3, Inner membrane	0.5,3000
MopE	CAI=0.7, CBI=0.5, Periplasmic fraction	0.2,7000
Gene 7593	CAI=0.7, CBI=0.5, Cytoplasmic fraction	1.6,9000
Gene 5938	CAI=0.6, CBI=0.3, Cytoplasmic fraction	0.3,10000
Gene 6155	CAI=0.6, CBI=0.3, Cytoplasmic fraction	0.1,5000
Gene 6156	CAI=0.5, CBI=0.4, Cytoplasmic fraction	5.0, 2000
Gene 5653	CAI=0.7, CBI=0.4, Inner membrane	3.0,10000
Gene 5396	CAI=0.5, CBI=0.2, Cytoplasmic fraction	>5.0,>40000

Table 8.2: Table showing results from manually searching for spots in the synthetic gel. Data on the different genes / proteins are found in Table 8.1. The range column specifies the range needed on the theoretical gel in order to find the desired protein. The range is given as the observed pI / Mw $\pm$ the sufficient deviation. The settings column specifies the CAI, CBI, and compartment prediction setting that was required to find the protein.

proteins that occurred. Worth noticing is that $MeDH_\beta$ was predicted by PSort as an inner-membrane protein, but the experiment it was identified in was only dealing with soluble proteins. This means that either the prediction was wrong, or that the protein should not have been in the soluble fraction. Most likely the prediction was wrong, since both periplasmic and inner-membrane proteins have a signal sequence. In these cases, we would have to show proteins in all locations.

sMMO A_α did need wide range settings to be displayed. In addition it was predicted to be inner-membrane, although it in the experiment came from the soluble fraction. In a realistic situation we would have to display all proteins to identify this protein. This protein is not likely to be found in the synthetic gel.

sMMO A_β was predicted to be cytoplasmic, which corresponded well with the source experiment. The CBI and CAI had to be lowered, and a quite wide range was applied. There were quite a lot of candidates, and this protein is not likely to be identified with help of the synthetic gel.

sMMO A_γ showed a relatively severe deviation in the observed mass value from the theoretical one. In addition, it was predicted to be inner-membrane protein. This contradicts the fact that the protein was found in the soluble fraction.

sMMO B was not found at all in the gene database, due to an error in the gene prediction.

MspA was predicted to be an inner-membrane protein. The theoretical Mw was far from the observed one. In addition, the gene for MspA did show a very low CBI and a quite low CAI.

MspB was not found in the gene database.

MspC was predicted to be cytoplasmic, which corresponded well with the experiment. The theoretical pI value was almost exactly the same as the observed one. The theoretical mass showed a larger deviation.

MspD was predicted to be cytoplasmic, corresponding well with experimental data. The theoretical pI value was more than four units higher than the observed one. The theoretical Mw was quite similar to the observed. The large deviation of pI may be caused by some experimental error, or feature.

MopB is an outer membrane protein, and was predicted to be this as well. With the settings shown in Table 8.2 the protein was uniquely identified on the theoretical 2D gel. The focusing point was quite accurate, and the protein shows a quite high degree of codon bias.

MopF showed approximately the same qualities as MopB, although there were two candidates in the theoretical gel.

MopG did not show the same degree of success. This protein was predicted to be inner-membrane, and had a deviation in the calculated focusing point, compared to the observed one.

MopE was predicted to be periplasmic, but was found in the outer membrane fraction.

Gene 7593 was predicted to be cytoplasmic, but was found in the outer membrane fraction. It had high deviations on the focusing points, so this protein is not likely to be identified by inspecting the theoretical 2D image.

Gene 5938 and Gene 6155 were predicted to be cytoplasmic, but were found in the outer membrane fraction. They both had focusing points with an acceptable accuracy.

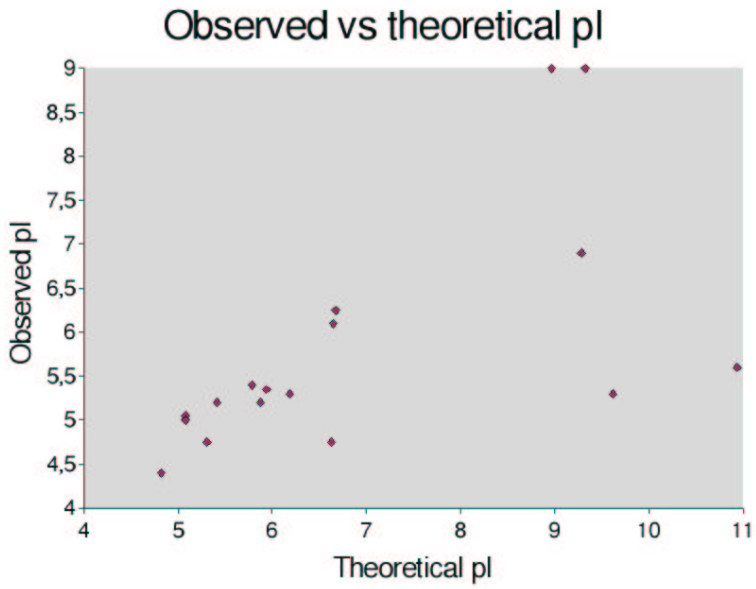


Figure 8.3: Relationship between observed and theoretical pI values for the proteins in Table 8.1

Gene 6156 was predicted to be cytoplasmic, but was found in the outer membrane fraction.

Gene 5653 was predicted to be inner-membrane, but was found in the outer membrane fraction.

Gene 5396 was predicted to be cytoplasmic, but was found in the outer membrane fraction.

The last three proteins all had severe deviations between the theoretic pI and the observed pI.

8.4.3 Discussion

A lot of the proteins had a higher theoretical Mw than the observed one. Reasons for this may be that the sequences in the database are too long, which have been observed, or that the obtained observed values are erroneous. A plot showing a relationship between observed Mw and theoretical Mw is shown in Figure 8.4, and for observed pI vs. theoretical pI in Figure 8.3. It may however look like the deviation between theoretical and observed Mw is quite systematic, and may be correlated simply by drawing a straight line through the data points.

The relationship between theoretical pI and observed pI does not seem to have the same systematic deviation as the Mw measurements had. They seem to be oriented around a line corresponding to equal values for theoretical and observed pI, with some outliers.

The calculation of the synthetic image is quite sensitive to errors in the genome database. Obviously, the prediction of pI and Mw can be wrong if a gene in the database is too long or too short. The theoretical pI values are more sensitive than the Mw values.

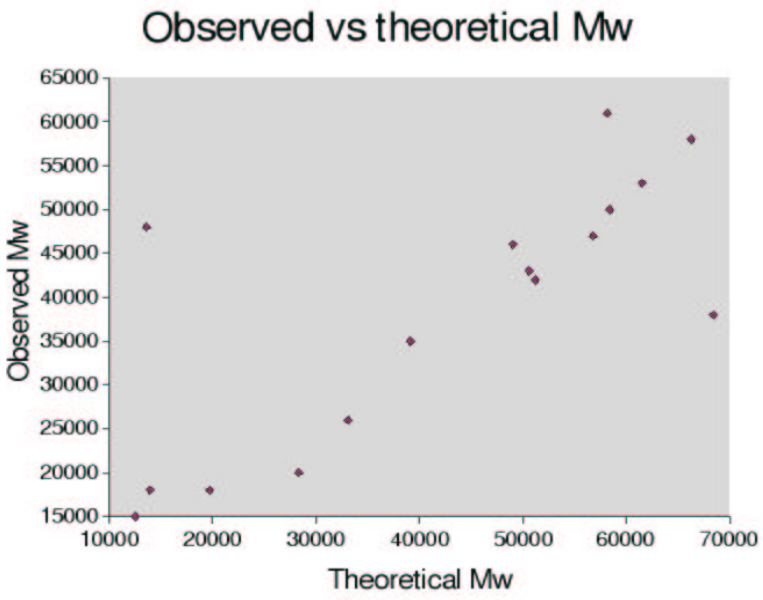


Figure 8.4: Relationship between observed and theoretical Mw values for the proteins in Table 8.1

But also the prediction of cellular location may be wrong. This is because the N-terminal and C-terminal ends of a protein are important when predicting cellular location. Several of the proteins that are found in the outer membrane fraction are predicted to be something else than outer membrane proteins. The most obscure predictions are the ones that predict outer membrane proteins to be cytoplasmic. This is because cytoplasmic proteins do not have a signal sequence at the N-terminal of the protein. On the other hand, outer membrane proteins that are predicted to be periplasmic or inner-membrane proteins are not that strange. These proteins share the property of outer membrane proteins with respect to signal sequences.

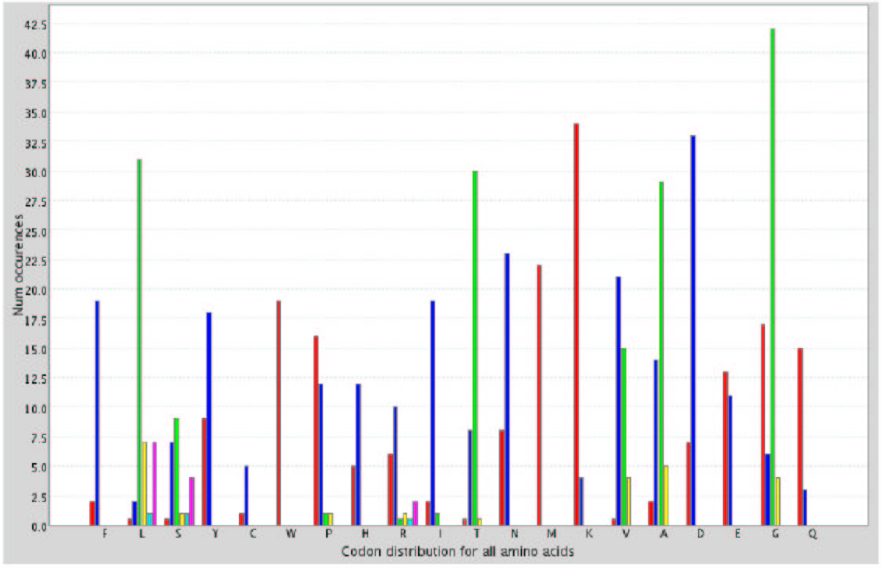
Overall, there were quite a lot of apparently wrong predictions of sub-cellular locations. It is of course possible that the biological experiments that set the standard for what is wrong and right, is erroneous. For example that a fraction that is supposed to only contain outer membrane proteins, may contain other proteins.

An important tool to create a useful theoretical gel image seems to be the CBI and the CAI. A lot of the proteins had relatively high values on these indexes, as shown by the limits in Table 8.2 that were used to filter away genes that were unlikely to be expressed.

Figure 8.5 shows the distribution of codons in the gene coding for $MeDH_{\alpha}$. Clearly, there is a non-random codon usage in this gene.

According to [21], there is a negative correlation between the degree of codon bias and protein length. In our case, this does not seem to be the case (see Figure 8.6 and Figure 8.7).

On the contrary, one can observe that the lowest values of both the CAI and CBI occur in genes of relatively short length.



Colour explanations:

F	TTT	TTC				
L	TTA	CTT	CTG	CTC	CTA	TTG
S	TCT	TCG	TCC	TCA	AGT	AGC
Y	TAT	TAC				
C	TGT	TGC				
W	TGG					
P	CCG	CCC	CCA	CCT		
H	CAT	CAC				
R	CCG	CGC	CGA	AGG	AGA	CGT
I	ATT	ATC	ATA			
T	ACT	ACG	ACC			
N	AAT	AAC				
M	ATG					
K	AAG	AAA				
V	GTT	GTG	GTC	GTA		
A	GCT	GCG	GCC	GCA		
D	GAT	GAC				
E	GAG	GAA				
G	GGT	GGG	GGC	GGA		
Q	CAG	CAA				

Figure 8.5: Example of codon usage for the gene coding for $MeDH_{\alpha}$ from M.Capsulatus. All the amino acids have bars corresponding to usage of synonymous codons. The amino acids have from one to six codons.

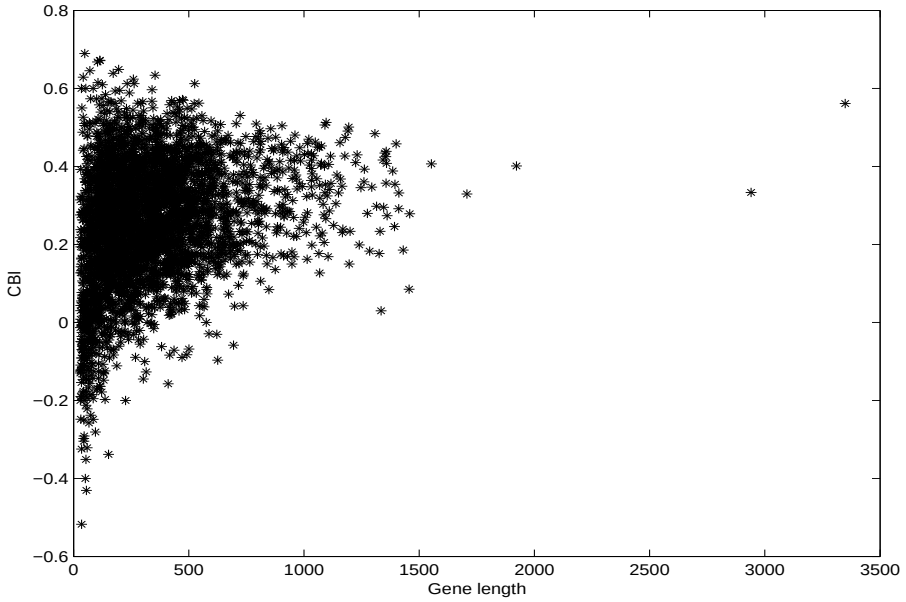


Figure 8.6: Relationship between gene length and codon usage bias, shown with the Codon Bias Index. The gene length is given in number of nucleotides. The figure shows the numbers for all predicted genes in M.Capsulatus.

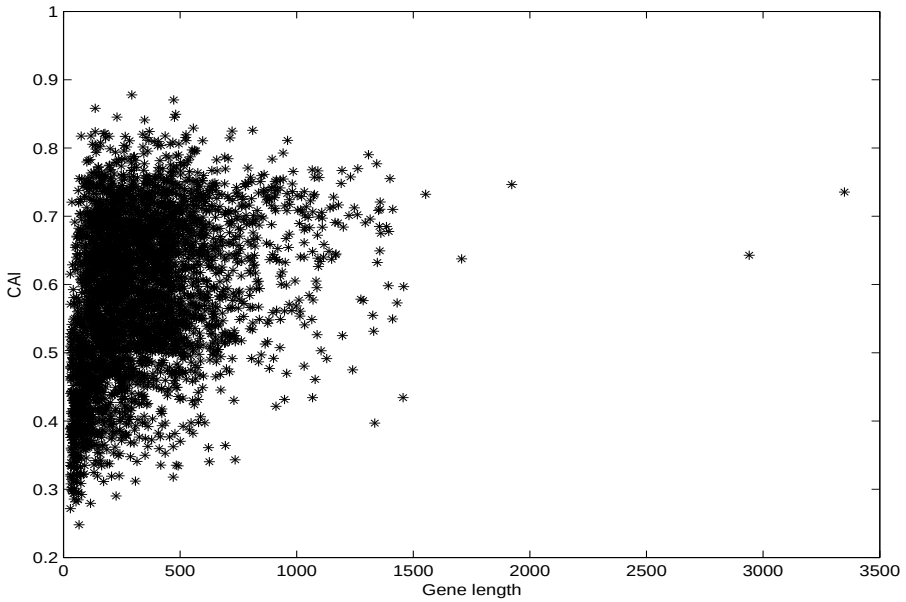


Figure 8.7: Relationship between gene length and codon usage bias, shown with the Codon Adaption Index. The gene length is given in number of nucleotides. The figure shows the numbers for all predicted genes in M.Capsulatus.

There is apparently a good correlation between the CBI and the CAI. It seems as if the CAI values have a higher variation than the CBI values. This is in compliance with the way the indexes are calculated, the method for calculating the CBI has a sharp cut-off when defining an optimal or non-optimal codon, whereas the method for calculating the CAI uses a continuous grading of codons.

8.5 Synthetic gel in automatic matching

The ideal situation would be to perform an automatic matching of an adapted synthetic gel with one or several experimental gels. The adaption of the synthetic gel could be done manually, by specifying pI/Mw range, setting CBI/CAI limits and by using for example PSort predictions to remove certain proteins etc. The testings done in Sections 8.4 indicate that an automatic matching would be problematic. Even if an automatic procedure is good, we will still be facing a situation where every experimental spot has at least 3-4 match-candidates. However, if the different predictions are more accurate, we may obtain better results.

8.6 Improvements

To improve the accuracy and usability of the theoretical 2D image, two factors appear to be important:

- The gene predictions must be more accurate. That means that the annotated gene actually starts with the correct start codon (ribosomal binding site). In addition, it is a problem that the database of predicted genes probably contains quite a lot of sequences that do not code for real proteins. This complicates the synthetic 2D image and makes it more difficult to find the real proteins.
- The predictions of a protein's cellular location must be improved. This may be done by using a newer, and more reliable prediction tool like SignalP.

Chapter 9

Implementation

The ideas described in Chapter 5 regarding linear progressive multiple matching, Chapter 6, Section 6.6 regarding multiple matching using a graph model, and Chapter 7 predicting a synthetic 2D image from a database of genes have all been implemented using the programming language Java [1]. A prototype version of a graphical user interface (GUI) is also incorporated. However, the design of the GUI has not had a high priority, as it is quite time consuming to design good GUIs. The program is developed using object oriented programming, where the main data structures and algorithms are divided into classes and methods.

9.1 File formats

The files for 2D images contain an enumerated list of spot coordinates, and in addition some features like spot intensity, spot size and shape, depending on the application that extracted the spot coordinates. There are no common standard for how these files should be constructed. The files that have been used in this program for testing and performance evaluations, are files from the gel analysis program PDQuest, and files downloaded from the 2D gel database at www.expasy.ch. The synthetic 2D images are stored in a simple format similar to the others.

The sequence data from M.Capsulatus was imported from a FASTA format file.

9.2 Data abstractions

The main data structures in the program are the list of coordinates from the 2D gel image and the sequences of the genes.

9.2.1 Representing a gel

A feature list extracted from a 2D gel is represented by a class called SpotList. This class contains a two-dimensional array to store the feature data in, with one row for each protein. The different columns contain the names, pI and

[1] The source for Java(TM) technology, http://java.sun.com

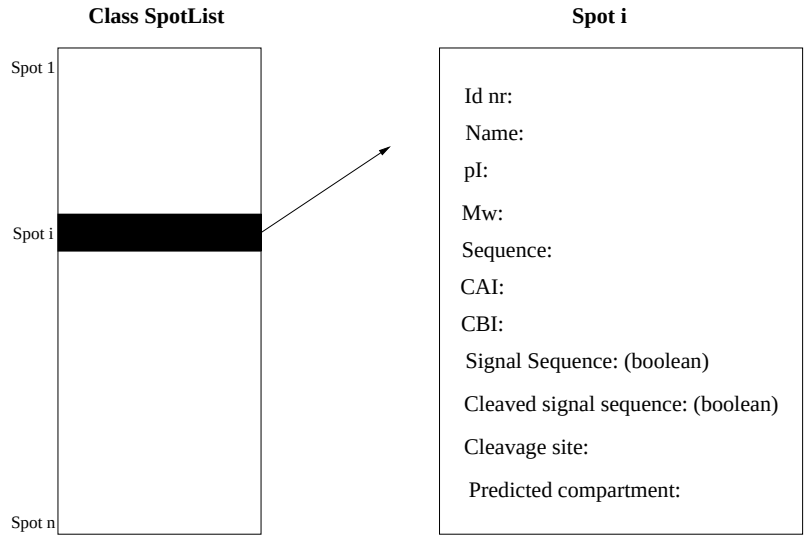


Figure 9.1: Representation of a 2D gel as a list of spots. The field Id nr, pI and Mw are compulsory, the other fields are optional.

Mw values etc.. Some of the features are mainly used when representing a *synthetic* gel in the same structure. The spots in a synthetic gel have features like sequence, predicted cellular location, etc. Figure 9.1 shows the possible features of a gel.

9.2.2 Representing a pair-wise match

As all gels are pair-wise matched towards each other, it is convenient to have a data structure storing this information. It is also important to be able to access, and modify this data in a simple way later.

A pair of gels is abstracted in a class called Pair. This class contains pointers to each of the SpotLists in question, and also information about the matching between them. It seems sensible to be able to quickly answer a question like: Does spot nr.22 in A match anything in B? With a representation similar to a neighbourhood matrix this could easily be achieved. But this requires much space, so a neighbour - list seems appropriate. This is made redundant, i.e. for each spot $s_i \in A$ there is a linked list of spots from B that matches s_i. To be able to answer queries about matches for spots in B in O(1) time as well, there is also a list for each of the spots in B.

Each match between two spots have information about the score, and which spots it is a match between. This information is encapsulated in a class called Match.

9.3 Algorithms

The main ideas of the algorithms are described in the different chapters of the thesis. This section only covers some implementation details.

9.3.1 Representing and implementing multiple matching

Two methods for multiple matching is implemented, one straight-forward with linear progression, and one more complex using a graph structure to represent the relationship between gels.

9.3.1.1 Linear progression solution

The linear progression solution is implemented using a linked list structure. Initially the linked list is empty. The first step is to add the first pair of gels to the linked list. Then, the rest of the gels are added to the linked list iteratively, one by one, as the matching proceeds.

The algorithm is found in the class proteinanalysator.tools.Match, method public static void multipleMatchingLinear.

9.3.1.2 Implementation of the graph solution

Graph implementation is done using the graph structure in the academic package JDSL [2]. The algorithm ideally finds all indirect connections between the spots, but this procedure has a time complexity of $O(n^k)$, where n is the number of spots, and k is the number of gels. In practise, if the number of gels is more than 3-4, it takes too long time. Finding only one indirect connection (similar to a transitive closure, $O(V^3)$, V is number of vertices) is a better solution regarding the time perspective, however, this approach has not got the power to discover all the inconsistencies. A heuristic approach that finds more than one, but less than all indirect connections is probably best, but not implemented.

A heuristic that limits the length of the indirect correlations, but uses the exponential method is implemented. This solution saves computation time, but with a large number of gels, it still takes long time.

The graph algorithms are found in the class proteinanalysator.tools.GraphTools.

9.3.1.3 The result of the multiple matching

The result of the multiple matching should be an enumeration of proteins, with the matching spots. For instance, an array with a finite number of "proteins", where each protein has a row in the array, and each column is a gel. Each cell contains none, one, or several entries representing the match(es) of the *row's* protein in the *column's* gel. The match-table is depicted in Table 4.1.

9.3.2 Calculating a theoretical 2D image

Creating a theoretical 2D image on the basis of a set of gene sequences is discussed in Chapter 7, and further discussed and tested in Chapter 8.

9.3.2.1 Predicting spot positions

The method for predicting the iso-electric focusing positions is described in Section 7.4. Predicting the other dimension, the total molecular weight of the protein is calculated by simply adding the single weights of all the amino acids.

[2] http://www.jdsl.org/

The prediction of spot positions may be found in the the class proteinanalysator.tools.ProtCalc.

9.3.2.2 Calculating codon bias indexes

The calculations of the codon bias indexes are done using the formula described in Chapter 7.5.1.

The implementation may be found in proteinanalysator.tools.struct.GenomeStat.

9.4 Loading a FASTA format sequence file

When loading a raw FASTA sequence file of nucleotides, in our case a complete genome or a part of one, a PSort-script is created together with a list of truncated sequences. A synthetic gel is then created by predicting the focusing positions based on the sequences. The PSort-script must be run manually, because java is not able to execute *nix scripts. When the PSort script is run, an output file is created, from which information may be extracted when the user selects "Load PSort info" in the program. The user may choose to display different subsets of the proteins based on the predicted sub-cellular location.

Chapter 10

Conclusion

The main goal of this thesis was to develop an algorithm for the purpose of multiple matching of 2D electrophoretic gels. The graph algorithm in Chapter 6.6 aimed at improving the efficiency of multiple matching compared to the common methods, and in addition validate the results to improve reliability of the results. Testing revealed that the algorithm discovered several inconsistencies in the pair-wise matching of gels that the standard algorithms would not have recognised. Regarding the efficiency, it was difficult to compare our results with other's, since the lack of an implementation of a pair-wise matching algorithm made the testing process rely on the output of an external program (PDQuest).

In the second part of the thesis, the aim was to explore the possibility of using the genome sequence information in connection with 2D gels by predicting a synthetic 2D gel from the gene sequences. The testing on the M.Capsulatus organism showed that it indeed is possible to predict 2D gels from sequence information, especially in conjunction with other tools, like protein sorting predictions. However, the predictions were not accurate enough to uniquely identify proteins on experimental gels just from the predicted gels.

It seems that in the future work with proteins, it will be important to develop methods that combine traditional protein analysis methods like 2D gel electrophoresis, mass spectrometry etc. with the use of acquired sequence information from genome projects.

10.1 Further work

To achieve a comprehensive system using the ideas described in this thesis some additional work needs to be done:

- To be able to use the graph algorithm in Chapter 6.6 in practise, an efficient and reliable algorithm for the pair-wise matching of 2D gels should be implemented in the same system as the graph algorithm. An example of such an algorithm is the one in [4].
- The predictions of signal sequence and sub-cellular locations should be combined with other tools, like prediction of start codon and prediction of transmembrane helices. This is also recognised out in the article concerning the SignalP system [23]. The sub-cellular prediction tool used in this

thesis, PSort [13], is a good and comprehensive tool, but new knowledge has been discovered since its development. This knowledge should be built into a new tool.

- Different post-translational modification sites and the effect of them should also be incorporated into the system in order to let the user recognise modifications on the experimental gels.

List of Figures

1.1 The central dogma of molecular biology. The dogma explains the main processes of molecular biology. 4
1.2 The genetic code. This system is used to decode nucleotide sequences into proteins. 5
1.3 Schematic example of a 2D gel. 6

2.1 Suggestion to feature extraction. By not only extracting the centre-coordinate, but four other values, the characteristics of each spot may be better represented. 11

3.1 The traditional work-flow. The spots are first detected, and then extracted as a list of spot coordinates. 16
3.2 Shows the idea of keeping multiple matches. Each row corresponds to a spot. An edge from one entry to another symbolises a match. The different edges have weights corresponding to the goodness of the match of the edges' spots. 17
3.3 Shows the comparison of the vector between the candidates, and the vector between the closest landmarks. The angles a and b are compared, so are the lengths l and k. 18
3.4 Illustrates the increased score of node N. The translation vector **t(e,e')** increases the score with an amount depending on the vector's endpoint distance from N. (Figure is from [5]) 21
3.5 Shows the distribution of scores along the grid of nodes. X and Y are the node positions on the matched gel, and Z is the score. There are several possible matching locations, but one main peak. (Figure is from [5]) 21
3.6 Illustrates the concept of a point's world view. The world view of point C is defined by the angles between the vectors from C to the other points, and the horizontal axis. The matched images should be roughly aligned, so that the horizontal axis is corresponding in the different images. 22

4.1 The forest representing the match-set, M, between *two* gels. Thick lines represents a possible choice of "active" match-edges. . 32

5.1 Situation A shows an example of tree progression. Situation B shows linear progression 35

5.2 Matching information from pair-wise matching can be used when adding a new gel (C) to a match-set. If k is matched to l and m in the pair-wise matching between C and A and C and B, check if l and m is matched in "current" match-set. If not, does a better configuration exist? 38

6.1 The figure illustrates the possible conflict when inserting a match into a match-table. A and B are in the match-set, and we wish to add gel C. First, the relation between C and B is considered, and spot c_j matches b_i. Then C vs. A is considered, and spot c_j matches spot a_k. If a_k and b_i are initially on the same row, then the three spots correspond to the same protein. 41
6.2 Figure showing the central concepts of the graph idea. Having different kinds of edges makes it easier to depict the different concepts of connection and correlation. 42
6.3 The generated graph from the pair-wise matchings with the transitive closure edges (new edges). Clearly, there are inconsistencies. For instance a_2 is matched to both b_3 and b_4 (via the transitive closure edges). 45
6.4 Figure showing the idea of removing direct connections with a more global view in mind. The constructed example shows that the edge A is contained in all four refused connections, while C is contained in three, and B is only contained in one. Removing A, will thus remove all the refused connections, while a removal of B only will remove one. A seems to be the most appropriate direct connection to remove, unless it also is a part of many connections that are not refused. 50
6.5 If the correlation $Co(a_1, d_1)$ is chosen as "valid", the correlation $Co(a_1, d_2)$ must be removed. $Co(a_1, d_2)$ contains $C(a_1, d_2) < a_1, b_1, d_2 >$, $C(a_1, d_2) < a_1, c_1, d_2 >$ and $C(a_1, d_2) < a_1, d_2 >$. Each of these three connections must be removed (in a suitable way). In practise, by removing edges. 51
6.6 Plots of the gels used to test the graph-based multiple matching method. 53
6.7 The "branching" factor (m) from each spot corresponds to the number of ambiguous matches between the spot and spots from another gel. In the figure m is three, there are k gels, and n is 5. 55
6.8 Showing the progress of starting with no edges, and adding one and one gel. Only the relations between the last added, and the new candidate are considered. Proteins, P_1... are added as unmatched spots occur. 58

7.1 Example of 2D gel image. Area 1 showing a "train" of spots, area 2 showing more normal situation. Such trains are often a result of post-translational modifications to a protein. Another explanation may be an unsuccessful denaturation procedure. . . . 64
7.2 Figure showing the translation from mRNA to protein. The tRNA uses its anti-codon to attach to the current mRNA codon. The amino acids become attached to each other as the translation goes along. 69

8.1 Work-flow for identification of proteins 74
8.2 Theoretical 2D gel image of M.capsulatus 75
8.3 Relationship between observed and theoretical pI values for the proteins in Table 8.1 81
8.4 Relationship between observed and theoretical Mw values for the proteins in Table 8.1 82
8.5 Example of codon usage for the gene coding for $MeDH_{\alpha}$ from M.Capsulatus. All the amino acids have bars corresponding to usage of synonymous codons. The amino acids have from one to six codons. 83
8.6 Relationship between gene length and codon usage bias, shown with the Codon Bias Index. The gene length is given in number of nucleotides. The figure shows the numbers for all predicted genes in M.Capsulatus. 84
8.7 Relationship between gene length and codon usage bias, shown with the Codon Adaption Index. The gene length is given in number of nucleotides. The figure shows the numbers for all predicted genes in M.Capsulatus. 84

9.1 Representation of a 2D gel as a list of spots. The field Id nr, pI and Mw are compulsory, the other fields are optional. 87

List of Tables

4.1 A table showing a setup for keeping track of the proteins in the different gels. Each gel has one column, and there is one row for each protein that is expected to be found. The number in a cell tells which spot in that column's gel that is assigned to the protein on that row. Several spot numbers mean that there are multiple options, for example if two spots are very close, and it is not clear which spot is correct. 29

6.1 This table shows some results from several executions of the algorithm, with different settings. The definition of the pc-score is given in Chapter 4.6.3. 54

7.1 Table showing some experimental pKa values for the electrically charged, hydrophilic amino acids 66

8.1 Table showing data about genes selected to be identified by using the synthetic gel image generated from the genome. The Mw values are in kDa. NF means not found, and may occur because the experimental 2D gels were annotated using an older sequence database than the one the synthetic gel was calculated from. Some of the names are not official annotation names. * The sMMO B was not found in the latest database, the one used in the creation of the synthetic gel. This is due to an erroneous gene prediction. The data is from a previous assembly. 78

8.2 Table showing results from manually searching for spots in the synthetic gel. Data on the different genes / proteins are found in Table 8.1. The range column specifies the range needed on the theoretical gel in order to find the desired protein. The range is given as the observed pI / Mw $\pm$ the sufficient deviation. The settings column specifies the CAI, CBI, and compartment prediction setting that was required to find the protein. 79

List of Algorithms

1 MATCHING(M,T), Pair-wise matching algorithm [4] 19
2 MATCH(M,T,L), Subroutine 20
3 PAIR-WISE-MATCH-EXTENDED(A,B) 31
4 Multiple matching using linear progression 35
5 Multiple matching using tree progression 35
6 Approach - Removing edges from the graph 59
7 Subroutine **CLEAN** 59
8 Approach - Adding edges to an initially empty graph 60
9 Calculate pI of amino acid sequence 67

Bibliography

[1] M.R.Wilkins, K.L.Williams, R.D.Appel, D.F.Hochstrasser *Proteome Research: New Frontiers in Functional Genomics*, 1997.

[2] H. Zhu et al. *Global Analysis of Protein Activities Using Proteome Chips*, Science, Vol.293, page 2101-2105, 14 September 2001.

[3] F. Murtagh *A New Approach to Point-Pattern Matching*, Astronomical Society of the Pacific, 1992 April.

[4] J. Panek, J. Vohradsky *Point pattern matching in the analysis of two-dimensional gel electropherograms*, Electrophoresis, 1999, 20, 3483-3491.

[5] K-P. Pleissner et al. *New algorithmic approaches to protein spot detection and pattern matching in two-dimensional electrophoresis databases*, Electrophoresis, 1999, 20, 755-765.

[6] G. Weber, L. Knipping, J. H. Alt, *Symbolic Computation*, 1994, 17, 321-340.

[7] Z. Smilansky *Automatic registration for images of two-dimensional protein gels*, Electrophoresis, 2001, 22, 1616-1626.

[8] I. Eidhammer, I. Jonassen, W.R. Taylor *Protein Sequence Analysis*

[9] I. Eidhammer, I Jonassen, W.R. Taylor *Protein Structure Comparison*

[10] *The User Manual for PDQuest*

[11] T. Voss, P. Haber *Observations on the reproducibility and matching efficiency of two-dimensional electrophoresis gels: Consequences for comprehensive data analysis*, Electrophoresis, 2000, 21, 3345-3350.

[12] G. S. Cox and G. de Jager *A Survey of Point Pattern Matching Techniques and a New Approach to Point Pattern Recognintion*

[13] *Prediction of Protein Sorting Signals and Localization Sites in Amino Acid Sequences*, http://psort.nibb.ac.jp

[14] *Predict the presence and location of signal peptide cleavage sites in amino acid sequences*, www.cbs.dtu.dk/services/SignalP/

[15] Jeffrey L. Bennetzen, Benjamin D. Hall *Codon Selection in Yeast*, The Journal of Biological Chemistry, 1982, 3026-3031.

[16] Paul M.Sharp, Wen-Hsiung Li *The codon adaption index - a measure of directional synonymous codon usage bias, and its potentional applications* Nucleic Acids Research, 1987, Volume 15, Number 3.

[17] *Electrophoretic Methods* http://www.np.edu.sg/~dept-bio/biochemistry/aab/topics/aab_electrophoresis.htm

[18] Frode S. Berven, *The establishment and use of 2-dimensional gel electrophoresis in proteome studies of Methylococcus capsulatus (Bath)*, Cand. Scient. thesis, October 2001.

[19] Petra Lutter et al. *Investigation of charge variants of rViscumin by two-dimensional gel electrophoresis and mass spectrometry*, Electrophoresis 2001, 22, 2888-2897

[20] Nakai, K. and Kanehisa, M., *Expert system for predicting protein localization sites in Gram-negative bacteria*, PROTEINS: Structure, Function, and Genetics 11, 95-110 (1991).

[21] Duret, Laurent *Detecting genomic features under weak selective pressure: The example of codon usage in animals and plants (invited talk)*, Bioinformatics 2002, Volume 18, ECCB2002 Proceedings.

[22] Andersson, Siv *Comparative genomics of microbial pathogens and symbionts (invited talk)*, Bioinformatics 2002, Volume 18, ECCB2002 Proceedings.

[23] Henrik Nielsen, et al. *Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites*, Protein Engineering vol. 10 no.1, pp. 1-6, 1997

[24] Bengt Bjellqvist, et al. *The focusing positions of polypeptides in immobilized pH gradients can be predicted from their amino acid sequences*, Electrophoresis 1993, 14, 1023-1031