

2D Protein Gel Analysis Tool for studying the protein phosphorylation
changes associated with contraction and relaxation of vascular
smooth muscle

An Internship Report Presented in Partial Fulfillment
of the Requirements for the Degree
of Master of Science

Submitted by
Pallavi Mudumby
Computational Biosciences,
Arizona State University,
Tempe, AZ 85287-1604 USA

Computational Biosciences
Advisor: Dr. Padmini Komalavilas

NOT CONFIDENTIAL

Report No: 05-20
ARIZONA STATE UNIVERSITY
October 2005

Graduate College
Arizona State University

This is to certify that this report submitted by Pallavi Mudumby has met the internship requirements for the Masters in Computational Biosciences Program at Arizona State University.

Approved By:
Supervisory Committee:

Dr.Padmini Komalavilas
Adjunct Professor
Harrington Department of Bioengineering
Arizona State University

Dr. Lokesh Joshi
Director, Center for Glycosciences and Technology
Associate Professor, Harrington Department of Bioengineering
Arizona State University

Dr. Kirkman Liff
Professor of Health Policy and Biotechnology
School of Health Management and Policy
W.P Carey School of Business
Arizona State University

Department Chair:

Dr. Rosemary Renaut,
Professor, Department of Mathematics and Statistics
Affiliated Professor, Department of Computer Science and Engineering
Director, Computational Biosciences Program
Arizona State University

Table of Contents	Page No
Acknowledgments	03
Chapter I:	
Abstract	04
Internship Details and Requirements	05
Goals	07
Significance of the project	07
Chapter II:	
Phase I: Two Dimensional Gel Electrophoresis	
Introduction	09
Goals	15
Methodology and Analysis	
Collecting and dissecting tissue to obtain coronary artery	15
Treatment of coronary artery with drugs	16
Tissue Pulverization	18
Protein Quantification	19
Protein Seperation	23
Staining	29
Scanning	30
Results	32
Chapter III:	
Phase II: 2D Protein Gel Analysis Tool	
Introduction	40
Goals	40
Methodology and Analysis	
Counting number of proteins on gel images	43
Displaying location and intensity of proteins	43
Comparing Control gel image with treated gel images	44
Results	45
Chapter IV:	
Conclusions	48
Future Work	51
Discussion	51
References	52
Review of Literature	54
Appendix	
Matlab Code	56

Acknowledgments

I would like to express my gratitude to my supervisor, Dr. Padmini Komalavilas, whose expertise, understanding, and patience, added considerably to my internship project. Her immense enthusiasm and guidance throughout the internship period has enabled me to finish the project on time.

I would like to thank Dr. Lokesh Joshi and Dr. Kirkman Liff for being on my committee and extending their support whenever required.

Very special thanks go to Mr. Vinod Swarna, without whose motivation and encouragement I would not have considered a graduate career in Computational Biosciences at Arizona State University. I doubt that I will ever be able to convey my appreciation fully, but I owe him my eternal gratitude.

I must also acknowledge Mr. Ashutosh Kulkarni for providing me technical assistance with Matlab Programming.

I would also like to thank my family for the support they provided me through my entire life and in particular, I must acknowledge my Brother and best friend, Ravi Chandra, without whose encouragement and editing assistance, I would not have finished this Report.

Chapter I:

Abstract:

Protein phosphorylation is the most important reversible post-translational modification that regulates eukaryotic cells. Analysis of phosphorylated proteins and identification of the phosphorylation sites helps to understand their biological functions. Two-dimensional gel electrophoresis is a powerful method for the analysis of complex protein mixtures extracted from cells, tissues, or other biological samples. This technique separates proteins according to their isoelectric points (first dimension) and then based on their molecular weights (second dimension). A crucial step in the analysis of two-dimensional protein gels is to match protein spots in different gel images that correspond to the same location. It requires extensive and time-consuming manual interference.

The goal of the project is to develop 2D Protein gel analysis tool for studying protein phosphorylation changes associated with the contraction and relaxation of vascular smooth muscle using two dimensional gel electrophoresis.

Systematic study is carried out which involved treatment of tissue with drugs, protein quantification, protein separation and staining with spyro ruby stain and phospho protein stain. 2D protein gel analysis tool is used for counting the number of proteins on the gel, displaying the location and intensity of the proteins and matching the control image with treated image for similar proteins based on location.

Results of two-dimensional gel electrophoresis when analyzed with 2D Protein gel analysis tool showed significant decrease in the number of phosphorylated proteins when compared to total number of proteins during contraction and relaxation of vascular smooth muscle.

Key words: Proteins, Phosphorylation, vascular smooth muscle, Spyro ruby stain, Phospho protein stain, Euclidean distance, Two dimensional gel electrophoresis and 2D Protein Gel Analysis Tool.

Internship details and requirements:

During the summer 2005, I have spent ten weeks as an intern with the Department of Bioengineering at Arizona State University. The courses I have done in Computational Bioscience Master's program at Arizona State University have given me a strong foundation in Biology, Mathematics, Computers and Technology. There is strong correlation between the experience I gained from internship and the classroom knowledge I obtained from Computational Biosciences Program.

The aim of the the project is to perform comprehensive proteomics and bioinformatics studies.It helped me gain education on methods and techniques in proteomics and different algorithms to analyze and interpret proteomic data.Having an internship like this not only helped me academically, but also professionally. Not only did I get internship credit, which helped me graduate, but I am now able to perform proteomics wet lab experiments and develop tools and applications for analyzing biological data.

Below are few reasons why a semester long internship in department of bioengineering at Arizona State University has proved to be very exciting for me

- I learnt to implement and integrate Bioinformatics with Proteomics. Everyone at Dr.Padmini's lab was very welcoming and always willing to help. I felt that they really cared and worked hard to make the experience valuable for me. It is fun to work with people who are knowledgeable and approachable.
- They clearly valued my participation and input. They also expressed appreciation well and often
- They paid me. Although it was a small amount, it was a fine gesture that indicated they considered me valuable

Requirements:

Proteomics wet-lab hands-on experience
Bioinformatics programming skills in Matlab.

Project is divided into two phases:

Phase I: Two Dimensional Gel Electrophoresis

Phase II: 2D Protein Gel Analysis Tool

Phase 1 deals with the Proteomics wet lab work in performing Two Dimensional Gel Electrophoresis and phase II is about development of 2D Protein Gel Analysis Tool for analyzing the results obtained from phase I.

Goals:

- To study protein phosphorylation changes associated with the contraction and relaxation of vascular smooth muscle using two dimensional gel electrophoresis.
- To study total proteins expressed during contraction and relaxation of vascular smooth muscle using spiro ruby stain
- To study phosphorylated proteins expressed during contraction and relaxation of vascular smooth muscle using phospho protein stain.
- To develop a 2D Protein Gel Analysis Tool to obtain the total count of proteins along with the location and intensity of proteins.
- To compare two protein gels by location of proteins using 2D protein analysis tool.

Significance of project:

The project's significance comes from its willingness to understand benefits of bioinformatics on proteomics.

Protein phosphorylation is the most important reversible post-translational modification. Analysis of phosphorylated proteins and identification of the phosphorylation sites helps to understand their biological functions.

Comparing total number of proteins expressed to the number of phosphorylated proteins during contraction and relaxation of vascular smooth muscle will help us to understand proteins phosphorylation changes.

A crucial step in proteomics is image analysis where the protein spot patterns in different images are to be matched. Manual analysis of protein spots on gel is time consuming, and sophisticated image analysis applications are essential to extract meaningful data. Relative positions of proteins on two different gels are to be compared, while new spots which are not present in the reference gel are to be detected and if appropriate identified. The need for developing 2D protein gel analysis tool has assumed particular prominence in the analysis of images of electrophoretic gels.

Chapter II:

Phase I: Two Dimensional gel Electrophoresis

Introduction:

Proteome:

The term 'proteome' was first coined in late 1994 by Marc Wilkins at Siena two-dimensional gel electrophoresis meeting 'PROTEins expressed by the genOME' is called 'Proteome', which explains complete set of proteins that ultimately results from genome transcription in a given cell, tissue or organism. Proteome is the complement of proteins expressed by a genome at a particular point in time.

A cellular proteome is the collection of proteins found in a particular cell type under a particular set of environmental conditions such as exposure to hormone stimulation. The complete proteome for an organism can be conceptualized as the complete set of proteins from various cellular proteomes [2].

The concept of the proteome is fundamentally different to that of the genome: while the genome is static and well-defined for an organism, the proteome continually changes in response to external and internal events [4].

The human proteome is much more complicated than the bacterial proteome, because of many modifications to proteins in higher organisms. The proteome of a resting cell is substantially different from that of an activated cell. The proteome of a cell changes based on what the cells need. For example, resting cells may only produce the proteins necessary to keep them alive. But, if the body calls proteins to perform a function, such as producing antibodies, then the cell generates new proteins to accomplish this function.

Complexity of Proteome:

Complexity of the human proteome is far beyond 30,000 human genes. The Complexity of the proteome arises because most proteins seem to be processed and modified in complex ways.

The proteome is much more complicated than the genome because

- Genome contains roughly 40,000 genes. One gene can give rise to many different proteins. Until now, some 30 to 50% of human proteins are of unknown function.
- Polypeptide folding into shape of three-dimensional structure.
- Post-translational modification.
- Behavior in different environment.
- The thousands of component proteins of a cell and their post translational modifications may change with the cell cycle, environmental conditions, developmental stage, and metabolic state [6].

Proteomics

The importance of proteomics is rapidly growing across essentially all areas of biological research. Proteomics is the next step from genomics. Although a complete map of the genome information enclosed within DNA is available this does not explain the complexity involved in creating a person. This complexity arises not because of the nucleic acids, but because of the proteins, and the interactions between proteins [8].

Essentially, there is certain information that a genome sequence can provide. This includes protein amino acid sequences and likely initiation and termination codons. mRNA expression research can provide some information about the protein expression levels and tissue distributions.

However, to obtain further information about a protein including subcellular location, turnover rate, post-translational modification, covalent and noncovalent associations, and how all this is affected by different external and internal conditions it is necessary to study the proteins themselves [4].

Proteomics is a leading edge technology that enables mass screening of proteins and their post-translational modifications. Proteomics research aims at characterizing the hundreds or thousands of proteins expressed by organisms in the context of whole organisms, specific tissues or normal vs. diseased states.

Proteome analysis is the direct measurement of proteins in terms of their presence and relative abundance. The overall aim of proteomics is the characterization of the complete network of cell regulation. Analysis is required to determine which proteins have been conditionally expressed, by how much, and what posttranslational modifications have occurred [8]. Proteomics can also be defined as the systematic large-scale analysis of protein expression under normal and perturbed (stressed, diseased, and/or drugged) states, and generally involves the separation, identification, and characterization of all of the proteins in a cell or tissue sample

Protein Phosphorylation:

Regulation of protein activity is very important in cells. Regulation is carried out through post translational modifications. Post-translational phosphorylation is one of the most common protein modifications that occurs in animal cells. Proteins undergo a huge number of post translational modifications. Only subsets of post translational modifications are reversible such as acetylation, fatty acid acylation, glycosylation and phosphorylation [9]. These modifications affect the activity, life span, or cellular location of the modified proteins. Approximately 10% of the proteins in the cell cytosol are phosphorylated. Beyond doubt, protein phosphorylation is the most

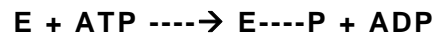
important regulatory event in eukaryotic cells. Many enzymes and receptors are turned on or off by phosphorylation and dephosphorylation.

In other words during phosphorylation phosphate is added and during dephosphorylation phosphate group is removed. The enzymes that phosphorylate proteins are termed kinases and those that remove phosphates are termed phosphatases.

Within a protein, phosphorylation can occur on several amino acids. Serine, threonine and tyrosine are the three amino acids subject to phosphorylation. Phosphorylation on serine is the most common, followed by threonine. Tyrosine phosphorylation is the rarest. The largest groups of kinases are those that phosphorylate either serines or threonines and as such are termed serine/threonine kinases [7]. The ratio of phosphorylation of the three different amino acids is approximately 1000/100/1 for serine/threonine/tyrosine. Although the level of tyrosine phosphorylation is minor, the importance of phosphorylation of this amino acid is profound.

The phosphorylation reaction involves ATP as the phosphoryl donor in the phosphorylation reaction and hydrolysis of the phosphoryl group in the dephosphorylation reaction. The net result of these reactions can be viewed as the hydrolysis of ATP, which has a ΔG of -12 kcal/mol under cellular conditions and is therefore, energetically favorable [10].

Phosphorylation



Dephosphorylation



Protein becomes phosphorylated and performs its action when an activating signal arrives. Upon the deactivating signal, the protein becomes dephosphorylated and stops working [5]. The network of phosphorylation can be very complex. Often, protein A phosphorylates B, and B phosphorylates C, but A also phosphorylates C directly, and B can phosphorylate D, which may in turn phosphorylate A [10].

Below are few advantages of phosphorylation:

Many carbohydrate, lipid, and amino acid metabolic pathways are synchronized by phosphorylation

Phosphorylation can modify the conformational equilibrium between different functional states.

Phosphorylation and dephosphorylation reactions under the control of kinases and phosphatases, can occur in less than a second or over a span of hours which makes this system perfect as a regulatory process.

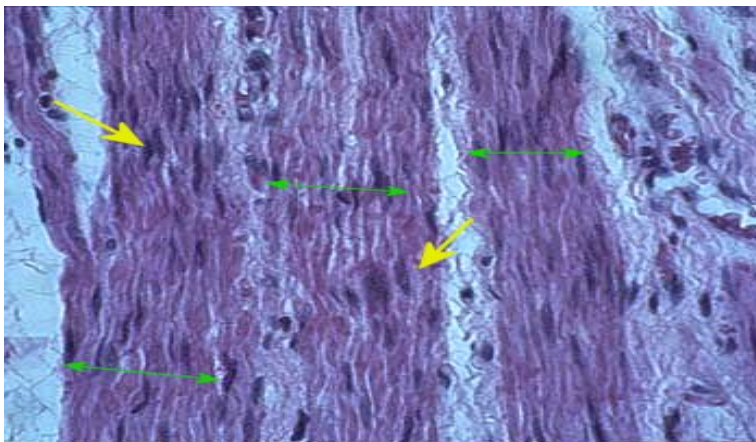
Phosphorylation and dephosphorylation reactions can be part of a flow of reactions which can amplify a signal which has an extracellular origin such as hormones and growth factors.

Phosphoryl group adds two negative charges to the protein modifying and disrupting the initial electrostatic interactions. This structural change can alter substrate binding and catalytic activity of a phosphorylated enzyme.

Smooth Muscle:

There are three types of muscle tissue: (1) skeletal muscle, (2) smooth muscle, and (3) cardiac muscle.

Smooth muscle cells are long (15 mm to 500 mm) and spindle-shaped with elongated nuclei. Smooth muscle is found in the walls of many hollow organs. In a smooth muscle cell the nucleus is centrally located and that smooth muscle is more eosinophilic and generally less refractive than connective tissue [12]. Smooth muscle can be found in the urinary bladder, uterus, arteries and veins. Each smooth muscle cell contains thick (myosin) and thin (actin) filaments that slide against each other to produce contraction of the cell. In general, Control of intracellular Ca^{2+} - principal mechanism initiates contraction and relaxation in smooth muscle. Contraction is associated with increased intracellular Ca^{++} concentrations and relaxation is associated with decreased intracellular Ca^{++} concentrations [11]. The mechanism of contraction and relaxation involves different signal transduction pathways, all of which converge to change intracellular calcium.



Yellow Arrow - Nuclei of Smooth Muscle Cell

Green Arrow - Width of Smooth Muscle Tissue

Figure 1: Smooth Muscle cell

Source

(<http://www3.umdnj.edu/histsweb/lab6/smoothmuscle/smcarrangement.html>)

Goals:

To study protein phosphorylation changes associated with the contraction and relaxation of vascular smooth muscle using two dimensional gel electrophoresis.

To study total proteins expressed during contraction and relaxation of vascular smooth muscle using spyro ruby stain

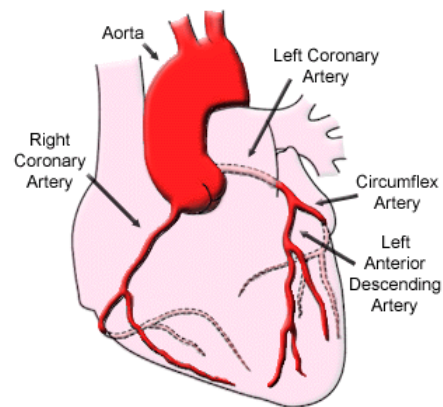
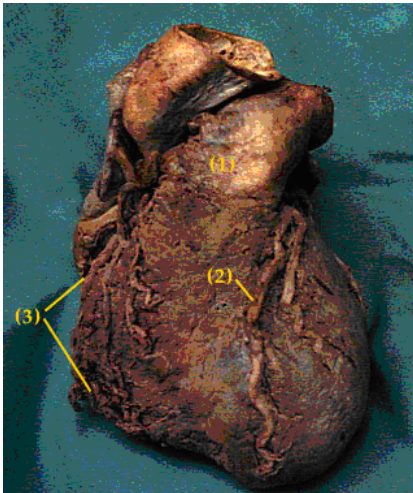
To study phosphorylated proteins expressed during contraction and relaxation of vascular smooth muscle using phospho protein stain.

Methods

Tissue acquisition

Fresh bovine heart tissue was obtained from Farabee Processing in organ preserving solution and immediately frozen with liquid N2.

Dissecting the tissue to obtain coronary artery



1 = pulmonary trunk; 2 = anterior descending (interventricular) coronary artery;
3 = right coronary artery with its right marginal branch.

Figure 2 and Figure 3: Bovine heart with left and right coronary arteries.

Source

(http://www.meddean.luc.edu/lumen/meded/grossanatomy/dissector/labs/thorax/heart/he2_1b.htm)

Procedure for dissecting:

1. After removing the heart, next step was to identify the right coronary and left coronary arteries. They arise from the aorta distal to two of the cusps of the aortic valve, the coronary cusps.
2. Fat and adventitial tissues are removed and artery rings containing vascular smooth muscle is equilibrated in a physiological buffer solution.

Treatment with drugs

Four treatment groups are set up. Tissue is either untreated or treated with drugs that induce contraction and relaxation.

Tissue that is not treated with drugs is called as control.

Treatment 1: Contraction is induced by treating the tissue with 1uM serotonin for 10 min.

Treatment 2: Smooth muscle tissue is treated with 1uM serotonin for 5 min followed by 10uM sodium nitro prusside for 5 min to induce relaxation.

Treatment 3: Tissue is treated with 1uM serotonin followed by 100uM papaverine for 5 min to induce relaxation.

Drug Serotonin

Serotonin was first recognised as a powerful vasoconstrictor in blood serum. It was isolated in 1948 by Page and was later found to be associated with the central nervous system.

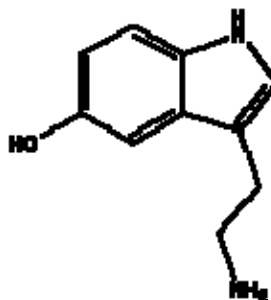


Figure 4: Drug Serotonin

Source

(http://www.chemsoc.org/exemplarchem/entries/2004/bristol_rosling/My%20Webs/introduction.htm)

The chemical name for serotonin is 5-hydroxytryptamine which is often abbreviated to 5-HT [14]. 5-HT acts as a neurotransmitter, allowing numerous functions in the human body including the muscle contraction, control of appetite, sleep, memory and learning, temperature regulation, mood, behaviour, cardiovascular function, endocrine regulation and depression.

Drug Papaverine

Papaverine is used to improve blood flow. It works by relaxing the blood vessels so that blood can flow more easily to the heart and through the body. Papaverine belongs to the group of medicines called vasodilators [15]. Vasodilators cause blood vessels to expand, thereby increasing blood flow. This medicine is used to treat problems resulting from poor blood circulation.

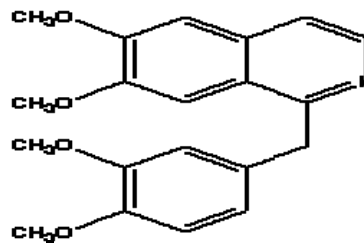


Figure 5: Drug Papaverine

Source (www.steve.gb.com/alkaloids/papaverine.png)

Drug Sodium Nitroprusside

Sodium nitroprusside acts by relaxation of vascular smooth muscle; it dilates peripheral arteries and veins. It is more active on veins than on arteries. Sodium nitroprusside breaks down in the blood and releases a chemical called nitric oxide (NO) [16]. Nitric oxide enters the muscle cells in the walls of the blood vessels and causes them to relax. When the muscles relax, the blood vessels become wider and the blood pressure decreases

Tissue Pulverization:

After treatment with drugs tissue is frozen in liquid nitrogen and pulverized using Bessman tissue Pulverizer. It is a two compartment stainless steel mortar with handle and pestle designed for pulverization of 10 to 100mg of tissue.

Below is the procedure for pulverizing tissue:

1. Liquid nitrogen is collected in Styrofoam container and mortar and pestle are left submerged in the liquid nitrogen.
2. Mortar is removed from Styrofoam container with some liquid nitrogen inside mortar.
3. Tissue is added into the liquid nitrogen after removing excess buffer on the tissue.
4. Tissue in the mortar is grinded well with pestle
5. Liquid nitrogen is allowed to evaporate and while it is slightly wet tissue is removed with the spatula and transferred to the labeled tube.
6. Tube is left open on dry ice for liquid nitrogen to evaporate and later closed and labeled.

The tissue is solubilized in urea, dithiothreitol, CHAPS (UDC) buffer by vortexing for 30 min at room temperature. Extracted proteins are separated by centrifuging the extract at 14,000 rpm for 10 min at 4°C using a centrifuge. Supernatant are transferred to a fresh tube.



Figure 6: White mortar and pestle



Figure 7: Centrifuge

Source (<http://www.fortune3.com/~comp72161/53438428.jpg> and www.roche-applied-science.com/centrifuge.jpg)

Protein Quantification: Protein assay

Proteins are quantitated using Bradford protein assay. The Bradford protein assay is one of several simple methods commonly used to determine the total protein concentration of a sample. The Bradford assay works by the action of Coomassie brilliant blue dye. The method is based on the proportional binding of the dye Coomassie to proteins and measuring the absorbance of the complex in a spectrophotometer and compared to known amount of standard proteins.

Coomassie Brilliant Blue dye :

A dye known as Coomassie Brilliant Blue was developed by the textile industry. This dye was known to bind to proteins and forms a wide variety of strong, but non-covalent, interactions including hydrogen bonding donor and acceptor interactions as well as hydrophobic (non-polar) interactions.

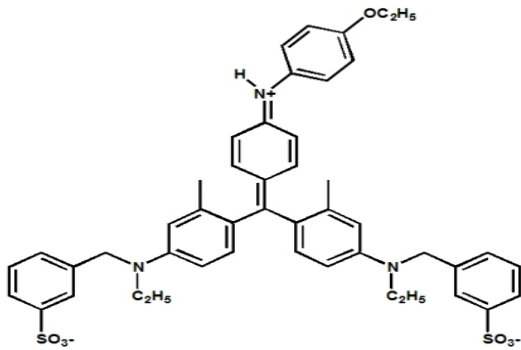


Figure 8: Coomassie Brilliant Blue dye

Source (<http://wine1.sb.fsu.edu/BCH40531/Lecture03/Lecture03.htm>)

The method is simple single step in which the dye is added to the protein solution under acidic conditions, and then the absorbance is read at 595nm [16]. Following is the procedure:

1. A standard curve using the standard BSA solution (1 mg/ml) for the range of 0-50 μg is constructed.
2. Protein samples are made up to 250 μl with dd.H₂O in 1.5 ml eppendorf microcentrifuge tubes
3. 750 μl of Bradford reagent is then added. The tubes are sealed and gently vortexed to ensure even mixing of the dye reagent
4. The reaction is allowed to continue for 10 minutes before absorbances are read at 595 nm wavelength using spectrophotometer.



Figure 9: Spectrophotometer

Source(www.beckman.com/duseries500_inst_dcr.asp)

Standard BSA solution

	Vol/ Standard	Vol H2O	Vol Assay mix
0	0ul	250 ul	750
0'	0ul	250 ul	750
2	10ul	240 ul	750
2'	10ul	240 ul	750
4	20ul	230 ul	750
4'	20ul	230 ul	750
6	30ul	220 ul	750
6'	30ul	220 ul	750
8	40ul	210 ul	750
8'	40ul	210 ul	750
10	50ul	200 ul	750
10'	50ul	200 ul	750

Protein samples

Protein	Vol of Protein	Vol H2O	Vol Assay Mix
Control 1	1ul	249	750
Control 2	1ul	249	750
5HT 1	1ul	249	750
5HT 2	1ul	249	750
5HT + SNP 1	1ul	249	750
5HT + SNP 2	1ul	249	750
5HT + Papa 1	1ul	249	750
5HT + Papa 2	1ul	249	750

Results of protein quantification:

ID	Ug	Net A
Blank		
1	0	0.450
2	0'	0.436
3	2	0.573
4	2'	0.572
5	4	0.673
6	4'	0.675
7	6	0.745
8	6'	0.760
9	8	0.828
10	8'	0.828
11	10	0.900
12	10'	0.898

Below are the proteins concentrations of control, 5HT, 5HT+ SNP and 5HT+ Papaverine

Control	4.1 ug/ul
5HT	7.63 ug/ul
5Ht + SNP	5.9 ug/ul
5HT + Papaverine	5.3 ug/ul

Protein samples (100 ug) are purified and are made upto 250 ul by adding UDC buffer

		100ug	UDC buffer
Control	4.1 ug/ul	24.4	225.6
5HT	7.63 ug/ul	13.1	236.9
5HT + SNP	5.9 ug/ul	16.9	233.1
5HT + Papa	5.3 ug/ul	18.8	231.2

Protein Separation

Two-dimensional polyacrylamide gel electrophoresis

A particular challenge of proteomics is the reproducible analysis of complex protein mixtures [21]. In 1975, two-dimensional polyacrylamide gel electrophoresis was simultaneously described by O'Farrell and by Klose. Their techniques allowed, for the first time, the separation of complex mixtures of proteins into individual components. Two-dimensional polyacrylamide gel electrophoresis (2D-PAGE) is currently the method of choice for separating complex mixtures of proteins. This method allows the simultaneous display of a large part of the entire set of proteins expressed by a given organism, organ or tissue. However, the technique is particularly powerful when comparing related samples, such as control tissue vs treated tissue. 2-D gels are not biased against any type of protein, and they have high enough load capacities to provide sufficient purified protein for characterization [22].

Key principles

- Proteins differ from each other in terms of their mass and charge.
- Both these properties can be used to separate proteins by gel electrophoresis.
- The successive application of both techniques in perpendicular directions (two dimensions) provides maximum separation and allows hundred of proteins to be resolved.
- Staining the gel reveals the positions of individual proteins as spots or smudges.

Steps:

2-D gel electrophoresis process consists of these steps:

- First dimension isoelectric focusing
- Second dimension gel electrophoresis
- Staining
- Imaging analysis

First Dimension

2-D gel electrophoresis uses a procedure called isoelectric focusing, which separates polypeptide chains depending on the surrounding pH and the charge of the protein (negative or positive). In the first step of 2D gel electrophoresis, a cell extract is fully denatured by high concentrations of detergent and layered on a glass tube filled with polyacrylamide that is saturated with a solution of ampholytes, a mixture of polyanionic[(-) charged] and polycationic [(+) charged] molecules [19]. When placed in an electric field, the ampholytes will separate and form a continuous gradient based on their net charge. The most highly polyanionic ampholytes will collect at one end of the tube, and the most polycationic ampholytes will collect at the other end. Charged proteins will migrate through the gradient until they reach their pI, or isoelectric point, the pH at which the net charge of the protein is zero [23]. This resolves proteins that differ by only one charge.

Day-1

Iso-Electric Focussing (IEF)

1. Protein samples (100 ug) are purified and are made upto 250 ul by adding UDC buffer.
2. 125 ul of sample in duplicates is loaded in rehydration tray.
3. 7 cm long IPG strip (pH 3-10) is rehydrated with protein sample on a disposable re-hydration tray for overnight at the room temperature.
4. Each strip is overlayed with 2-3 ml of mineral oil to prevent evaporation during the rehydration process.

Day-2

Wet electrode Wicks are placed on anode and cathode ends of electrode wires on an IEF tray. IPG strip is removed from rehydration tray, oil is drained and it is layed over on wet electrode wicks. Each strip is overlayed with 2-3 ml of mineral oil to prevent evaporation during the IEF process.

IEF can be started on Protean IEF cell (Bio-Rad # 165-4000) at 20°C using the following step cycles:

- (a) Step 1 250 volts, 20 min, linear ramp
 - (b) Step 2 8000 volts, 2.5 hr, linear ramp
 - (c) Step 3 8000 volts, 20,000 V-hr, rapid ramp
- Total: ~30,000 V-hr for a period of 5.3 hr

After IEF, IPG strips are removed from IEF tray, oil is drained and are either proceeded with 2nd dimension by equilibrating the strip or freezed at -70°C.



Figure 10: Bio-Rad Protean IEF cell



Figure 11: IEF Tray

Source (<http://proteomics.embl.de/images/IPStrips.jpg>
www.proteomesystems.com/Catalogue/Profile.asp)

Strip Equilibration.

Equilibration of the IPG strip is an important step in 2D gel electrophoresis as it enables the transfer of higher amounts of protein from the IPG strip into the SDS gel.

The strips are equilibrated before they can be used for SDS-PAGE.

Equilibration Solutions

Solution 1	10mL Stock Re-equilibration Buffer + 100mg DTT
Solution 2	10mL Stock Re-equilibration Buffer + 250mg iodacetamide

Steps in Strip Equilibration

- Strips are covered with solution 1, gel side up, and are placed on shaker for 20 minutes.
- Solution 1 is discarded.
- Strips are covered with solution 2, gel side up, and are placed on shaker for 20 minutes.
- Solution 2 is discarded.

Importance of Equilibration components:

DTT: DTT preserves the fully reduced state of denatured, unalkylated proteins.

Sodium dodecyl sulphate (SDS): SDS denatures proteins and forms negatively charged protein-SDS complexes. The amount of SDS bound to a protein, is directly proportional to the mass of the protein [17]. Electrophoresis of proteins in the presence of SDS separates proteins on the basis of molecular mass.

Iodoacetamide: Iodoacetamide alkylates thiol groups on proteins, preventing their reoxidation during electrophoresis. Protein reoxidation during electrophoresis can result in streaking and other artifacts. Iodoacetamide also alkylates residual DTT to prevent point streaking and other silver-staining artifacts [17]. Equilibration with iodoacetamide is used to minimize unwanted reactions of cysteine residues.

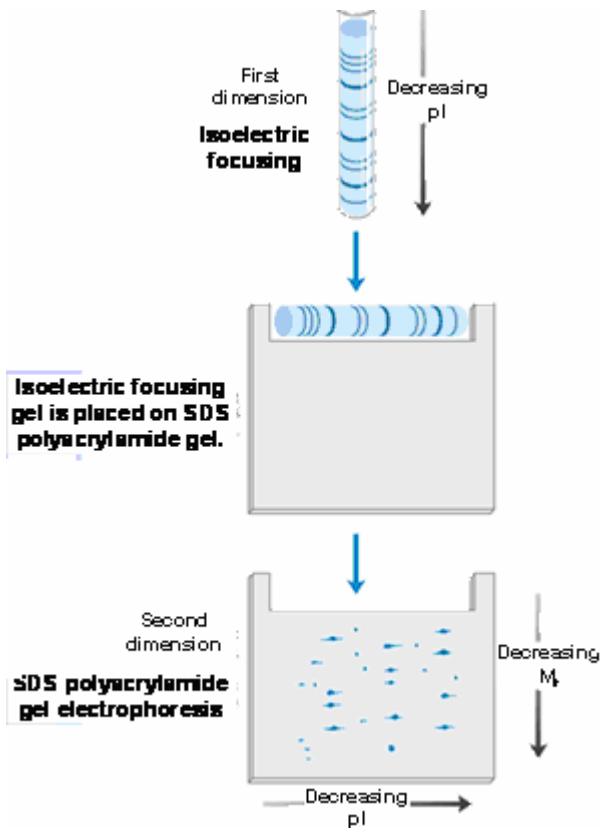


Figure 12: 2-D gel electrophoresis

Source (www.aber.ac.uk/parasitology/Proteome/Tut_2D.html)

Second Dimension:

Proteins that have been separated on an IEF gel are next separated in the second dimension based on their molecular weights. The second dimension is usually an SDS-polyacrylamide gel electrophoresis, which allows the separation of the proteins based on their molecular size [18]. The percentage of acryamide in the separation gel is chosen based on the size of the proteins to be separated (4-20%). The comb is removed from the gel after it has polymerized and the strip from above is applied horizontally on top of the stacking gel. The proteins in the strip are allowed to separate in the second dimension based of their apparent molecular weight in SDS.

When an electric field is imposed, the proteins migrate from the IEF gel into the SDS slab gel and then separate according to their mass. Sequential

resolution of proteins by their charge and mass can give excellent separation of cellular proteins [19].

Staining

Phosphoprotein gel stain

Phosphoprotein gel stain from molecular probes is a breakthrough technology that provides a method for selectively staining phosphoproteins in polyacrylamide gels

Stock Solution

1 Fix solution.

Fix solution for Phosphoprotein gel stain is a solution of 50% methanol and 10% acetic acid [20].

2 Destain solution:

Phosphoprotein gel destaining solution with catalog number P33310 is purchased from molecular probes. 1L Destain solution can also be prepared by combining and mixing throughly

- 50 mL of 1 M sodium acetate, pH 4.0
- 750 mL of ultrapure water
- 200 mL of acetonitrile

Staining procedure

1 Fix the gel. Gel is immersed in ~500 mL of fix solution and incubated at room temperature with gentle agitation for 30–60 minutes. This step is repeated overnight to ensure that all of the SDS is washed out of the gel.

2 Wash the gel. Gel is incubated in ~500 mL of ultrapure water with gentle agitation for 15 minutes. This step is repeated two times for a total of three washes. Gel is completely immersed in water in order to remove all of the methanol and acetic acid from the gel. Residual methanol or acetic acid will interfere with phosphoprotein staining.

3 Stain the gel. In the next step, Gel is incubated in the dark in 500 mL of phosphoprotein gel stain with gentle agitation for 1.5–2 hours.

4 Destain the gel. Destaining is important to reduce the gel background signal and to reduce the signal from nonspecific staining. Gel is incubated in 500 mL of destain solution with gentle agitation for 30 minutes at room temperature, protected from light. This step is repeated two more times. The optimal total destaining time is about 1.5 hours.

5 Wash the gel. Finally, gel is washed twice with ultrapure water at room temperature for 5 minutes per wash.

Imaging and Documenting the Gel

Phosphoprotein stain has an excitation maximum at ~555 nm and an emission maximum at ~580 nm.

Transillumination Stained gels can be visualized on a blue-light transilluminator or on a 300 nm UV transilluminator with an instrument capable of 532–560 nm excitations.

The scanner used to image the gels is Alpha Fluor Chem Scanner at Dr.Joshi's lab.

Spyro Ruby Stain:

Sypro Ruby protein gel stain is ideal for proteomics. It is used for the analysis of total proteins in 2-D polyacrylamide gels. This stain provides very low nanogram sensitivity, but stains more proteins and has a much broader linear quantitation range[20] .It is possible to obtain accurate protein quantitation for both highly expressed and minimally expressed proteins in the gel using spyro ruby protein gel stain.

Stock Solution:

Fix and destain Solution:

Fix and destain solution for spyro ruby stain is a solution of 50% methanol and 10% acetic acid.

Procedure:

1 Fix: After 2D electrophoresis, gel is placed into a microwavable container with 100 mL of fix solution and agitated on an orbital shaker for 15 minutes

2. Wash: Gel is transferred to a clean container and incubated in 100 mL of ultrapure water with gentle agitation for 15 minutes. Gel is completely immersed in water in order to remove all of the methanol and acetic acid from the gel.

3. Stain: After washing thoroughly, gel is stained with 100ml of spyro ruby protein stain for 3 hours on shaker. Staining is done overnight for better results.

4. Wash. After staining, gel is transferred to a clean container and incubated in 100 mL of wash solution for 30 minutes.

5. Destain: Destaining is important to reduce the gel background signal and to reduce the signal from nonspecific staining. Gel is incubated in 100 mL of destain solution with gentle agitation for 30 minutes at room temperature, protected from light.

6. Wash the gel. Finally, gel is washed twice with ultrapure water at room temperature for 5 minutes per wash.

Viewing and Photographing the Gel

Proteins stained with the spyro rubu protein stain are visualized using a 300 nm UV transilluminator, a blue-light transilluminator, or a laser scanner.

The scanner used to scan the gels is Alpha Fluor Chem Scanner from Dr.Joshi's lab.

Results of Two Dimensional Gel Electrophoresis:

Control with Spyro Ruby Stain

Gel image obtained after Two Dimensional Gel Electrophoresis when stained with spyro ruby stain. This gel is termed as control gel since it is not treated with any drugs.

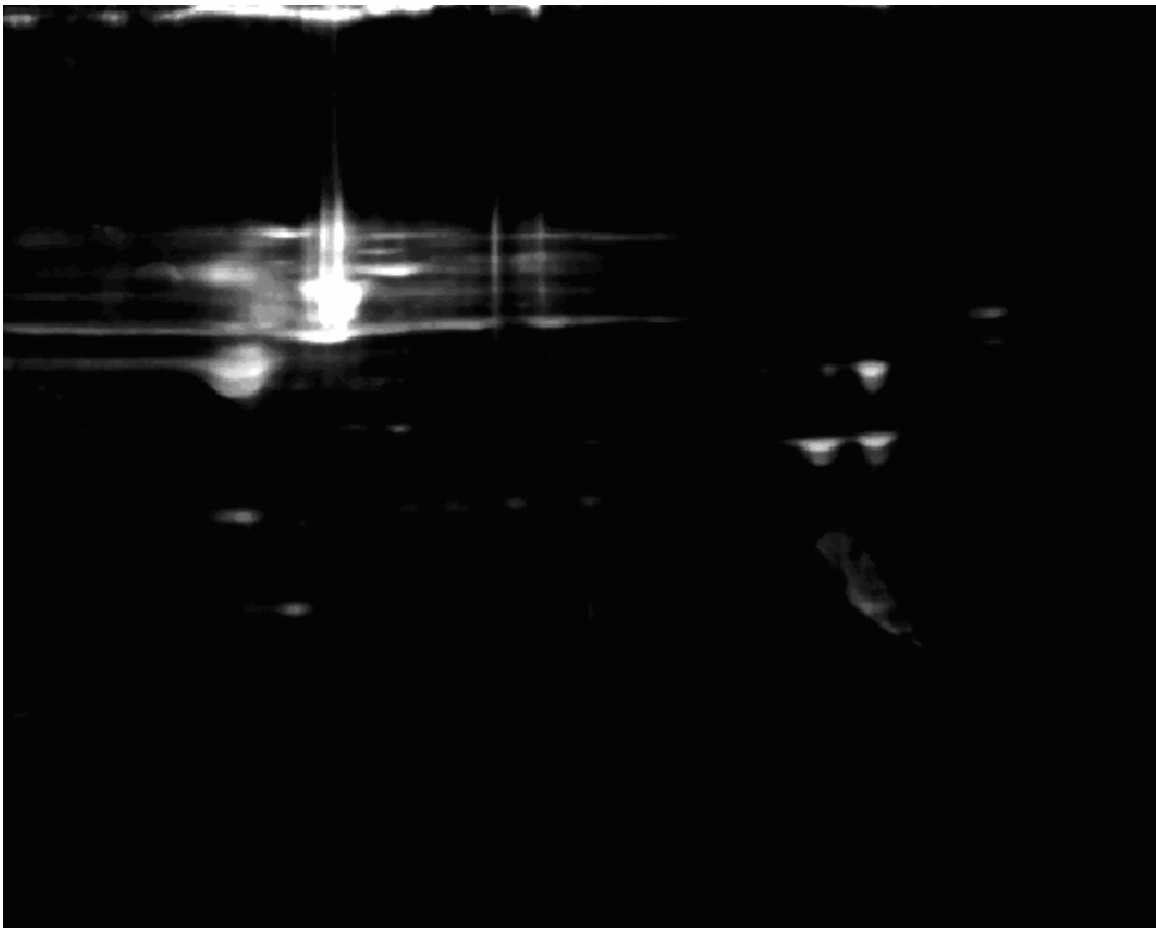


Figure 13: Control gel image obtained after staining with Spyro ruby stain

5HT with Spyro Ruby Stain

Gel image obtained after Two Dimensional Gel Electrophoresis when stained with spyro ruby stain. This gel is termed as 5HT gel since it is treated with serotonin (5HT) to induce contraction.

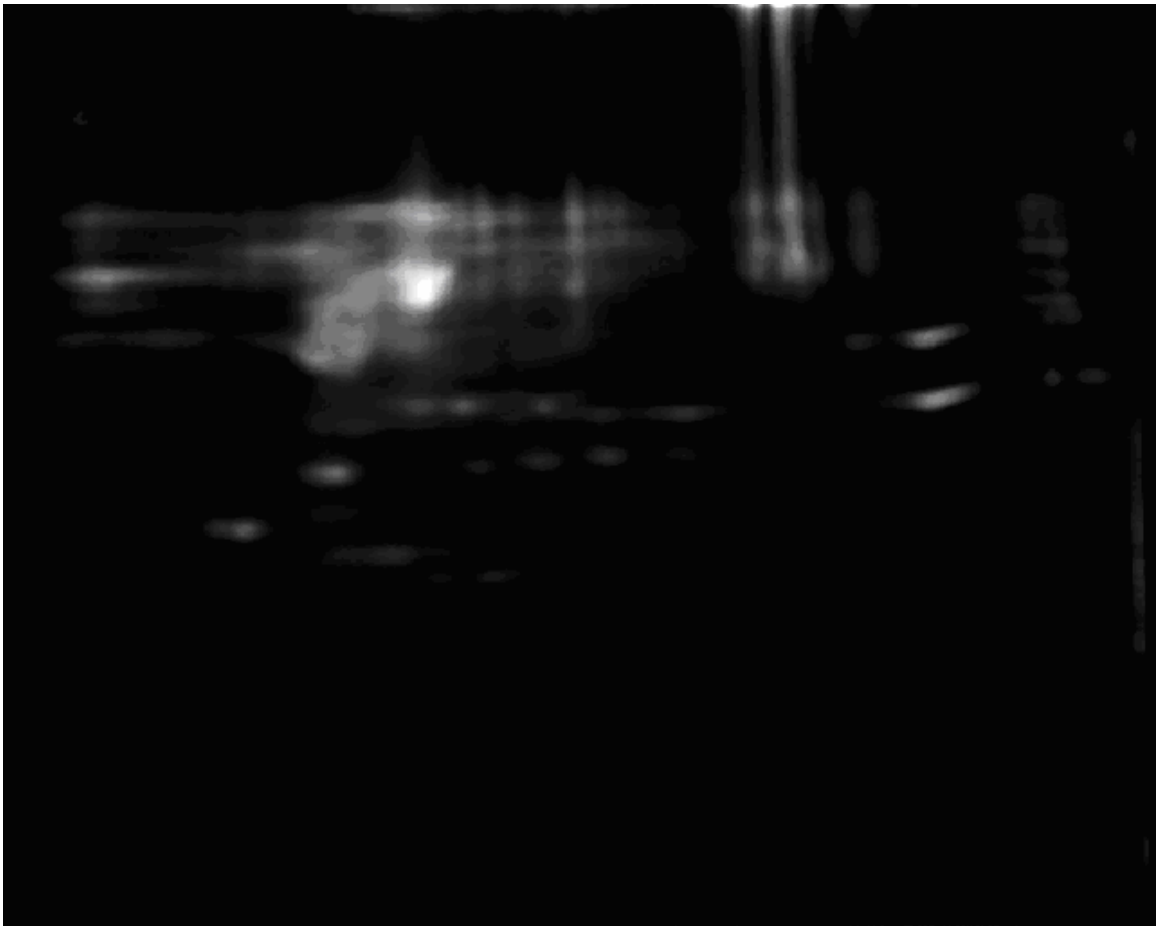


Figure 14: 5HT image obtained after staining with Spyro ruby stain

5HT + SNP with Spyro ruby stain

Gel image obtained after Two Dimensional Gel Electrophoresis when stained with spyro ruby stain. This gel is termed as 5HT + SNP gel since it is treated with serotonin followed by sodium nitro prusside to induce relaxation.

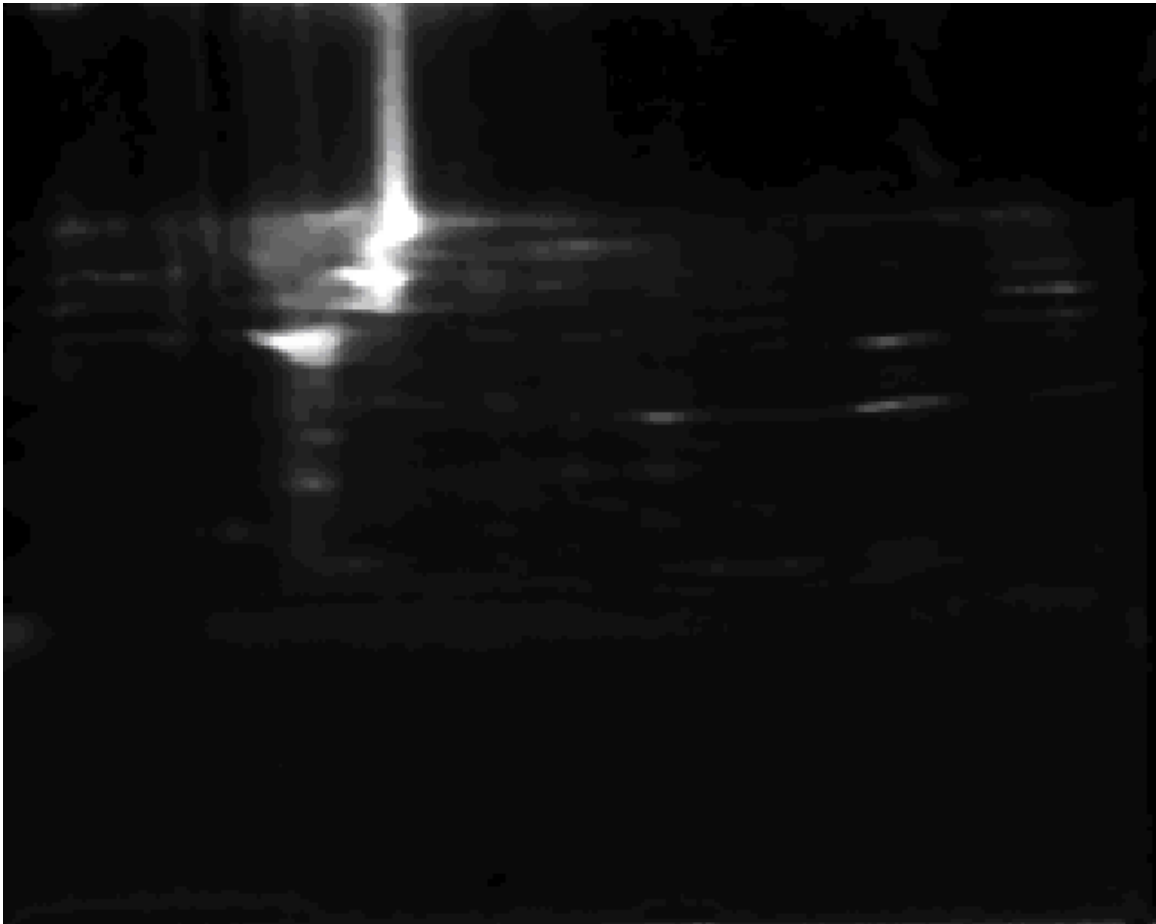


Figure 15: 5HT + SNP image obtained after staining with Spyro ruby stain

5HT + Papaverine with Spyro ruby stain

Gel image obtained after Two Dimensional Gel Electrophoresis when stained with spyro ruby stain. This gel is termed as 5HT + Papaverine gel since it is treated with serotonin followed by papaverine to induce relaxation.

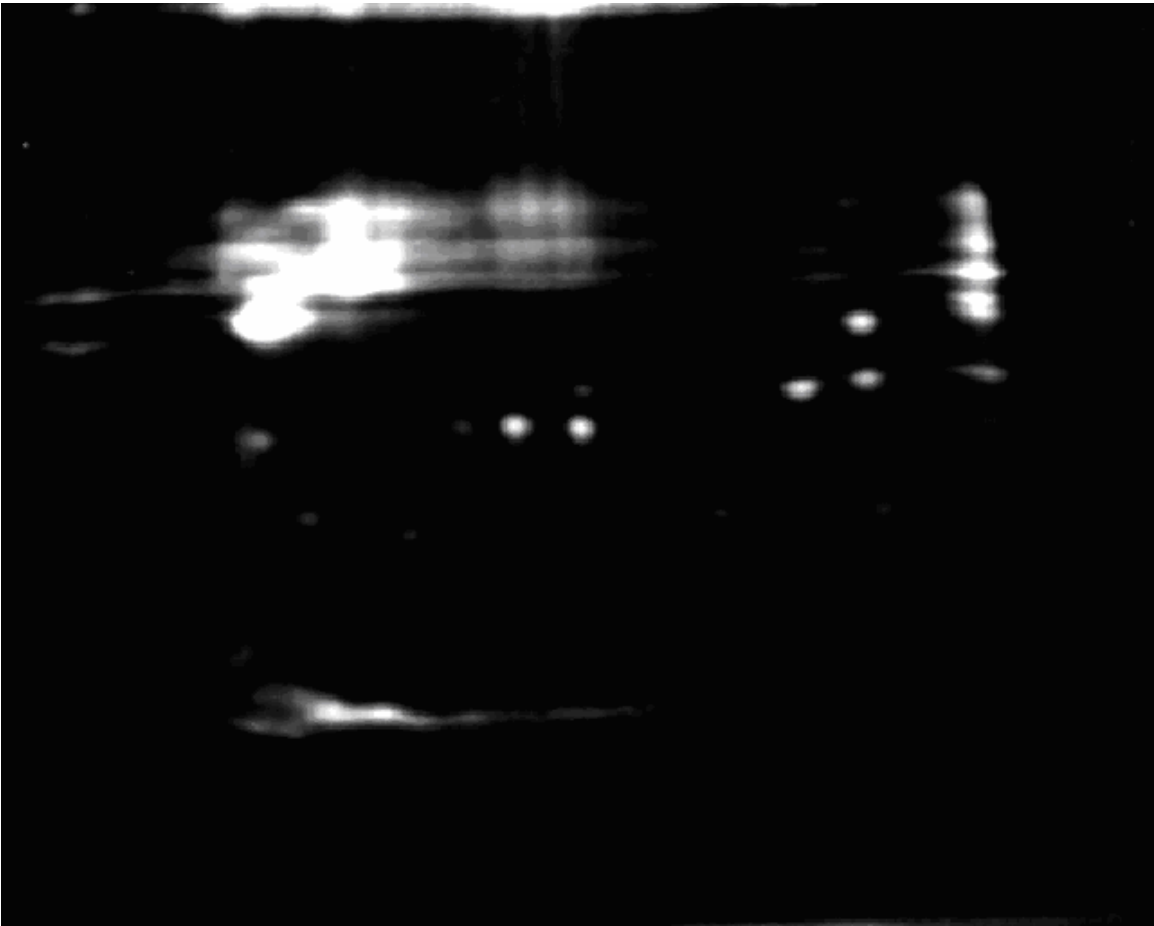


Figure 16: 5HT + Papaverine image obtained after staining with Spyro ruby stain

Control with Phosphoprotein stain:

Gel image obtained after Two Dimensional Gel Electrophoresis when stained with Phospho protein stain. This gel is termed as control gel since it is not treated with any drugs to induce contraction or relaxation.

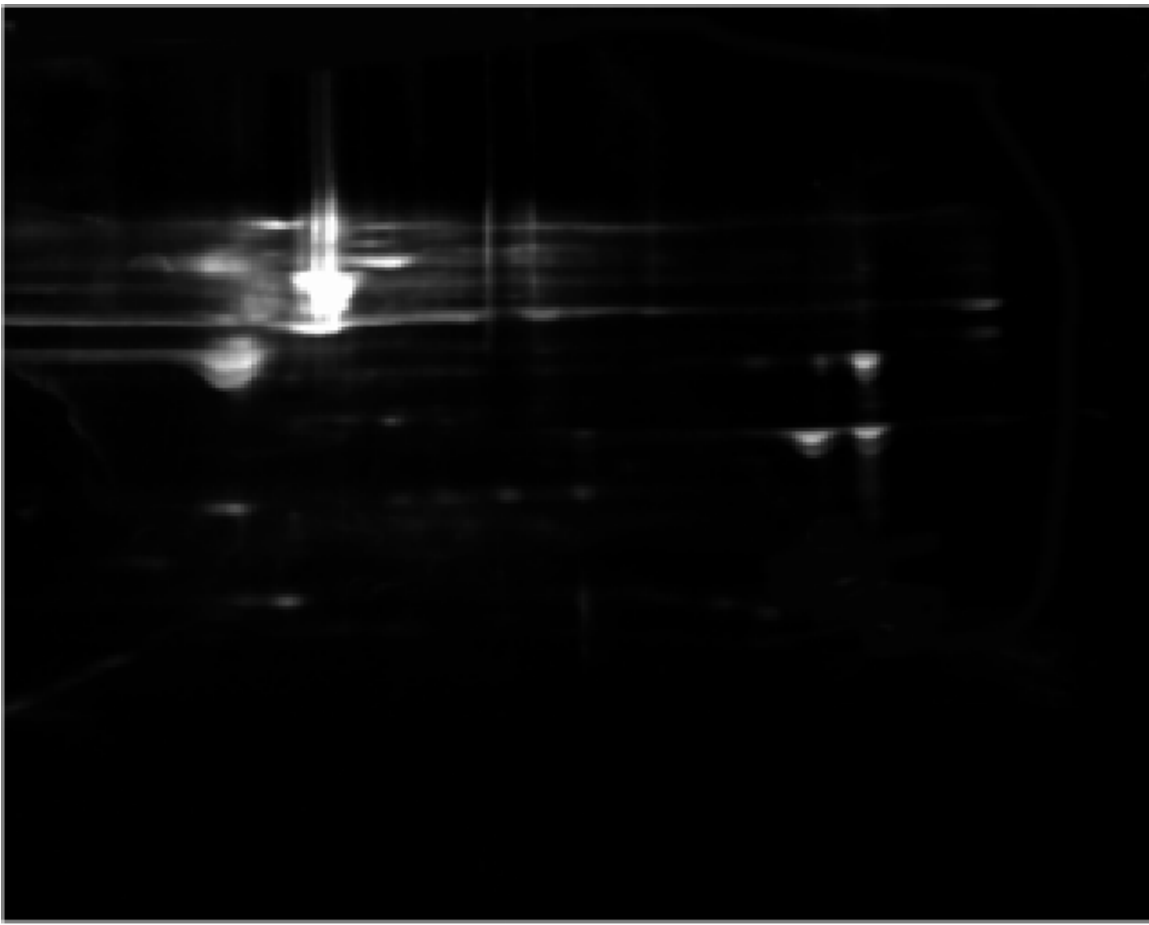


Figure 17: Control image obtained after staining with Phosphoprotein stain

5HT with Phosphoprotein stain

Gel image obtained after Two Dimensional Gel Electrophoresis when stained with Phospho protein stain. This gel is termed as 5HT gel since it is treated with serotonin (5HT) to induce contraction.



Figure 18: 5HT image obtained after staining with Phosphoprotein stain

5HT + SNP with Phosphoprotein stain

Below is the gel obtained after Two Dimensional Gel Electrophoresis when stained with Phospho protein stain. This gel is termed as 5HT + SNP gel since it is treated with serotonin followed by sodium nitro prusside to induce relaxation.

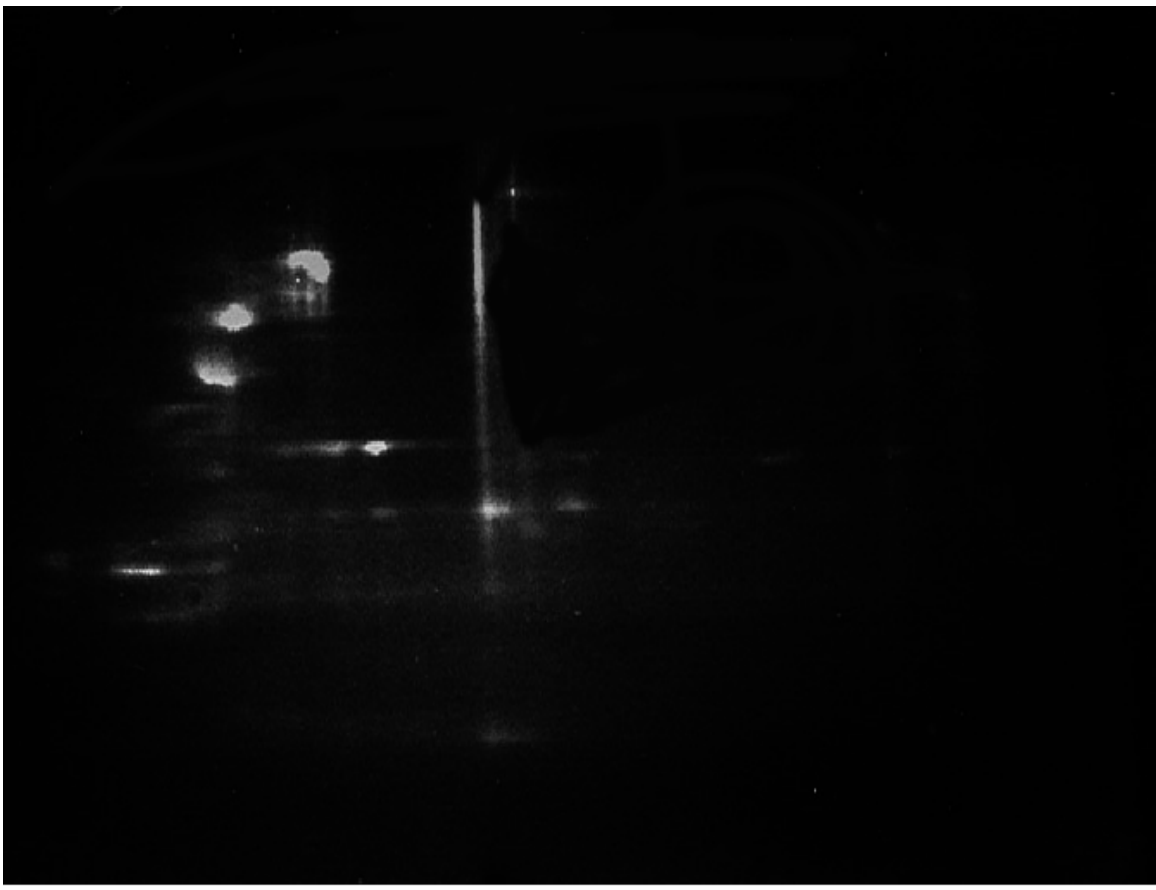


Figure 19: 5HT + SNP image obtained after staining with Phosphoprotein stain

5HT + Papaverine with Phosphoprotein stain

Below is the gel obtained after Two Dimensional Gel Electrophoresis when stained with Phospho protein stain. This gel is termed as 5HT + Papaverine gel since it is treated with serotonin followed by papaverine to induce relaxation.

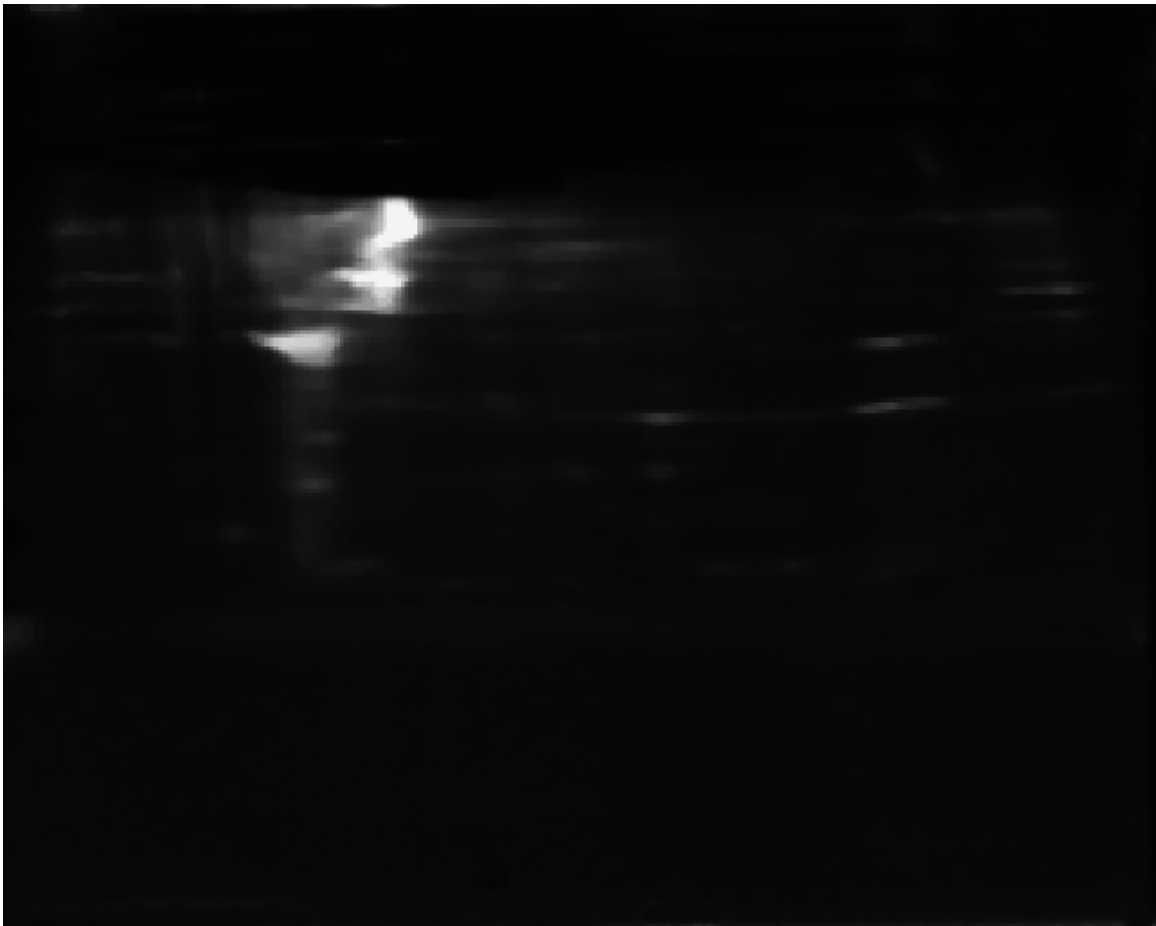


Figure 20: 5HT + Papaverine image obtained after staining with Phosphoprotein stain

Chapter III

Phase II: 2D Protein Gel Analysis tool

Introduction:

Despite its scientific potential, Two Dimensional Gel Electrophoresis technology is only partially exploited today. This is partially due to difficulties related to the experimental procedure, and even more due to the complexity of analyzing the results with the aid of existing tools. To extract biologically meaningful information, the output images of different Two Dimensional Gel Electrophoresis experiments have to be compared.

A crucial step in two-dimensional gel based protein expression analysis is to match protein spots in different gel images that correspond to the same location. It requires extensive and time-consuming manual interference. Gel analysts commonly do this manually by visually or semi-automatically identifying significant equivalent spots in both gels, selecting them manually, and then comparing the reference with the test gel so that these spots are matched against the chosen equivalents in the reference gel [24].

Goal:

The goal of the second phase project is

- To develop 2D Protein Gel Analysis Tool to obtain total count of proteins along with the location and intensity of proteins.
- To compare two two-dimensional protein gels by location of proteins using 2D Protein Gel Analysis Tool.

Methodolgy:

Segmentation is one of the most important steps in the analysis of image data. Segmentation includes number of different techniques that divide the image into Segments. Thresholding is the simplest segmentation process which is computationally inexpensive and fast. Thresholding is the operation of converting a multi-level image into a binary image. In a binary image, each pixel value is represented by a single binary digit. In its simplest form, thresholding is a point-based operation that assigns the values of 0 or 1 to each pixel of an image based on a comparison with some global threshold value T. Thresholding is an early processing step because it leads to significant reduction in data storage and results in binary images that are easy to analyze [31]. It is a way to get rid of noise by considering the significant information of the image and to improve the signal-noise ratio.

There are two basic methods in thresholding: "hard thresholding" and "soft thresholding":

In hard thresholding, any wavelet coefficient whose absolute value is less than the threshold value is set to zero. Coefficients whose absolute value is greater than or equal to the threshold remain unchanged [25].

Hard thresholding

$$\begin{aligned}\tilde{\theta}_j &= y_t^* |y_t^*| \geq \lambda, \\ &= 0 \quad |y_t^*| \leq \lambda\end{aligned}$$

In soft thresholding, any wavelet coefficient whose absolute value is less than the threshold value is set to zero. The threshold is subtracted from coefficients whose absolute value is greater than or equal to the threshold value. This moves these coefficients toward zero [25].

Soft thresholding

$$\tilde{\theta}_j = \text{sign}(y_j^*) (|y_j^*| - \lambda)_+.$$

Image is segmented using hard thresholding. The biggest challenge in thresholding an image is picking the right threshold value. For manual thresholding, a maximum (255) and minimum (0) intensity are specified to limit the range of valid intensities. This technique uses two values to define the threshold range. The thresholds are adjusted interactively by showing all pixels in the image whose value lies in this range are converted to black; pixels with values outside this range are converted to white.

User can slide a cursor along the slider to select a threshold value

Before threshold

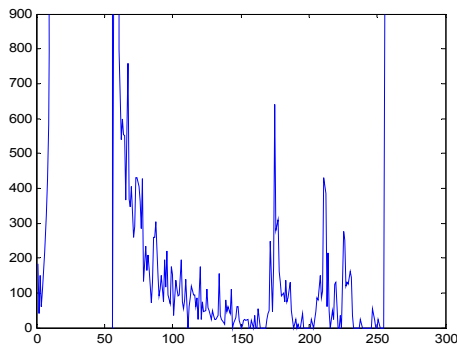


Figure 21

After thresholding at 128

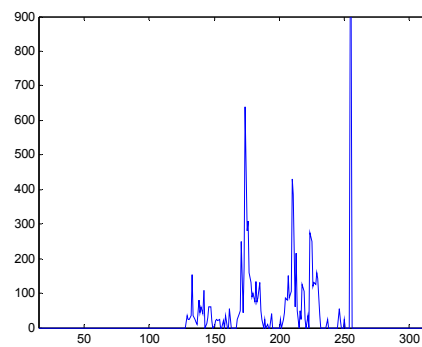


Figure 22

Figure 21: The intensity profile of the control image with spyro ruby stain (figure 13) before hard thresholding

Figure 22: The intensity profile of the control image with spyro ruby stain (figure 13) after hard thresholding at 128. Note that the coefficients whose absolute value is less than 128 are set to zero and coefficients whose absolute value is greater than or equal to 128 remain unchanged.

Counting the number of proteins

In the image analysis system, the centroid coordinates of each cell are important features of cell images.

Fastcentroid function returns the centroid coordinates for the connected components in BW (iind, jind) and the number of pixels of each component (numberofpixels) [26]. This software was developed by Marcelino Sanchez Gonzalez in May 2002.

The output obtained from FastCentroid function will be in the format of [iind, jind, numberofpixels]

The row coordinate of the center of the area for the ith protein is given by iind.

The column coordinate of the center of the area for the ith protein is given by jind.

The intensity of the ith protein is given by numberofpixels.

Fastcentroid is faster than bwmorph('shrink',Inf) and imfeature(L,'centroid') [26].

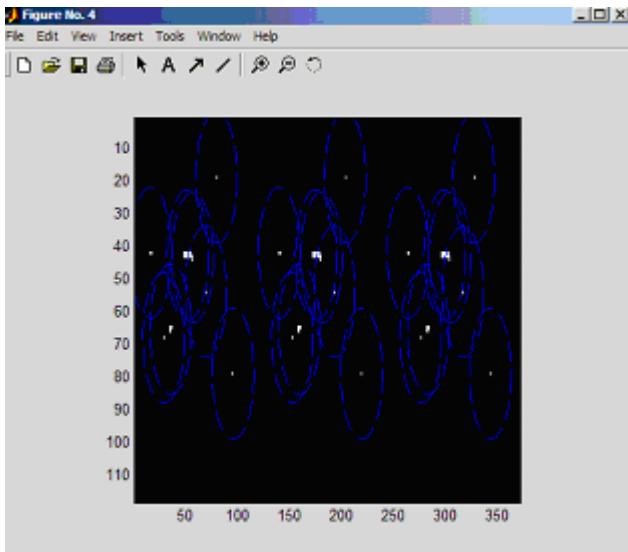


Figure 23: This figure is obtained as the result of Fastcentroid function: Each protein spot on the image is identified and Fastcentroid returns the position and intensity of each protein.

Finding similar proteins in control and treated gel images

Each protein to be matched is identified by describing its location and intensity so that it can be matched to proteins in another image. Every protein is assumed as simply a vector $(x(s); y(s); i(s))$ consisting of its nonnegative point coordinates $(x(s); y(s))$ in the Euclidean plane and a positive number $i(s)$ describing its intensity.

The protein spots are represented using a set of keypoints; and then a matching algorithm is applied to find the similar spots in the image 1 and image2. The matching criterion is based on the Euclidean distances between the keypoints on the control image and the keypoints on treated image. If a single keypoint on the control image is considered, its distance to all the keypoints of the entire treated image is computed. If the Euclidean distance between the control image and treated image for a particular pair of keypoints falls below the chosen threshold, this pair is termed a match. The threshold for matching images based on Euclidean distance is selected as '1'. This value is selected from trial and error method.

Euclidean distance is the most common distance measure. A given pair of cases is plotted on two variables, which form the x and y axes. The Euclidean distance is the square root of the sum of the square of the x difference plus the square of the y distance [43].

For two 2D points $P = [px, py]$ and $Q = [qx, qy]$, the distance is computed as

$$\sqrt{(px-qx)^2 + (py-qy)^2}$$

2D protein gel analysis tool is an open-source stand-alone computer program for finding position and intensity of proteins in a 2 Dimensional protein gel image. It can read black and white JPEG, GIF, and BMP images. Two images are required in order to do the comparison based on protein location.

Counting the number of proteins:

Load the image to count the number of proteins.

Adjust the threshold: user has the option of interactively selecting the threshold by moving the cursor over the slider.

Click the Apply button to get the total count of proteins

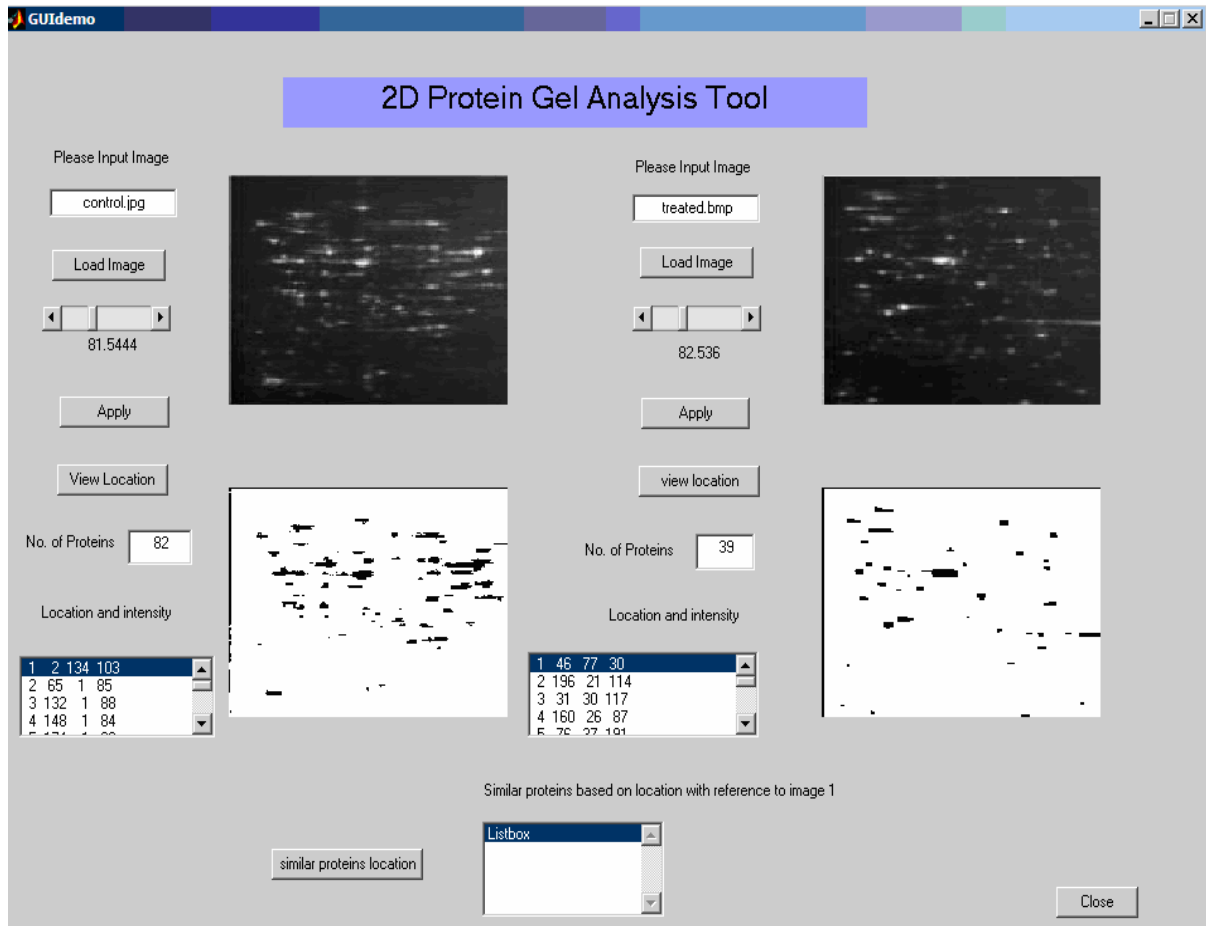


Figure 24: 2D Protein Gel Analysis Tool GUI

View Location and Intensity:

Load the image

Adjust the threshold

Click the apply button to view the protein location and intensity

Click the View location button to get the protein location and intensity interactively

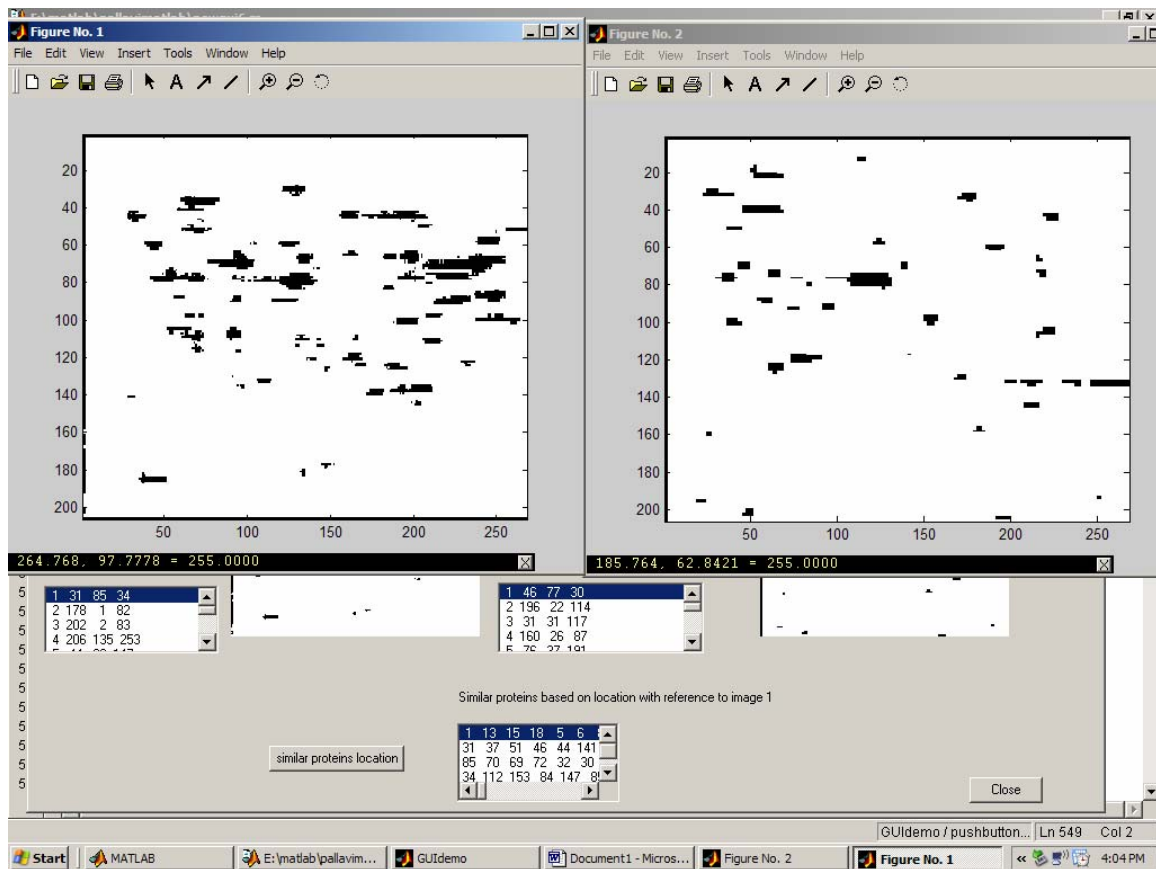


Figure 25: 2D Protein Gel Analysis Tool GUI

Get similar proteins in two images based on location

Load the image

Set the threshold

Click Similar protein location button to get the similar proteins in two images.

Similar protein spots can be identified by the red sign on the image.

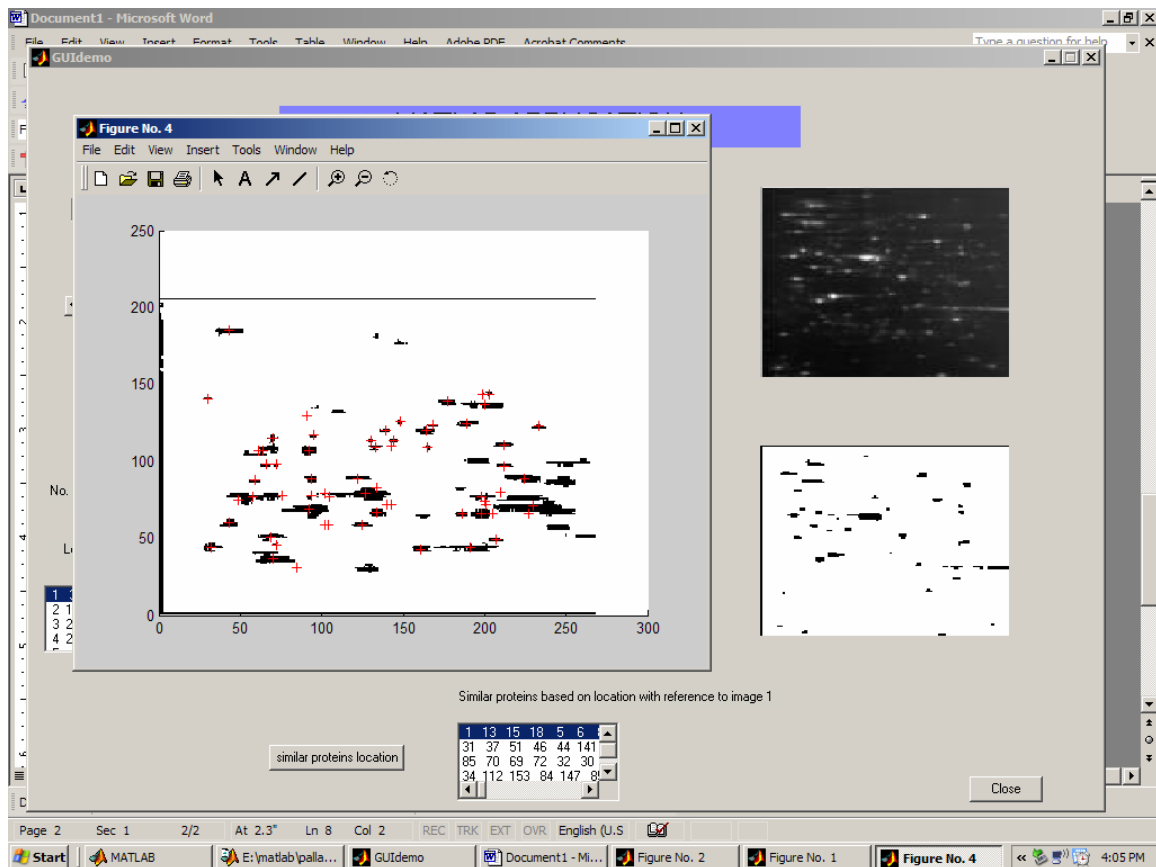


Figure 26: 2D Protein Gel Analysis Tool GUI

Chapter IV

Conclusions:

Below table explains the number of total proteins picked up by spyro ruby proteins stain and number of phosphorylated proteins picked up by phosphoprotein stain during four stages of control, contraction and relaxation at threshold 70

	Spyro ruby	Phosphoprotein	Common proteins
Control	27	18	7
5HT	21	13	5
5HT + SNP	23	19	7
5HT + Papaverine	19	11	4

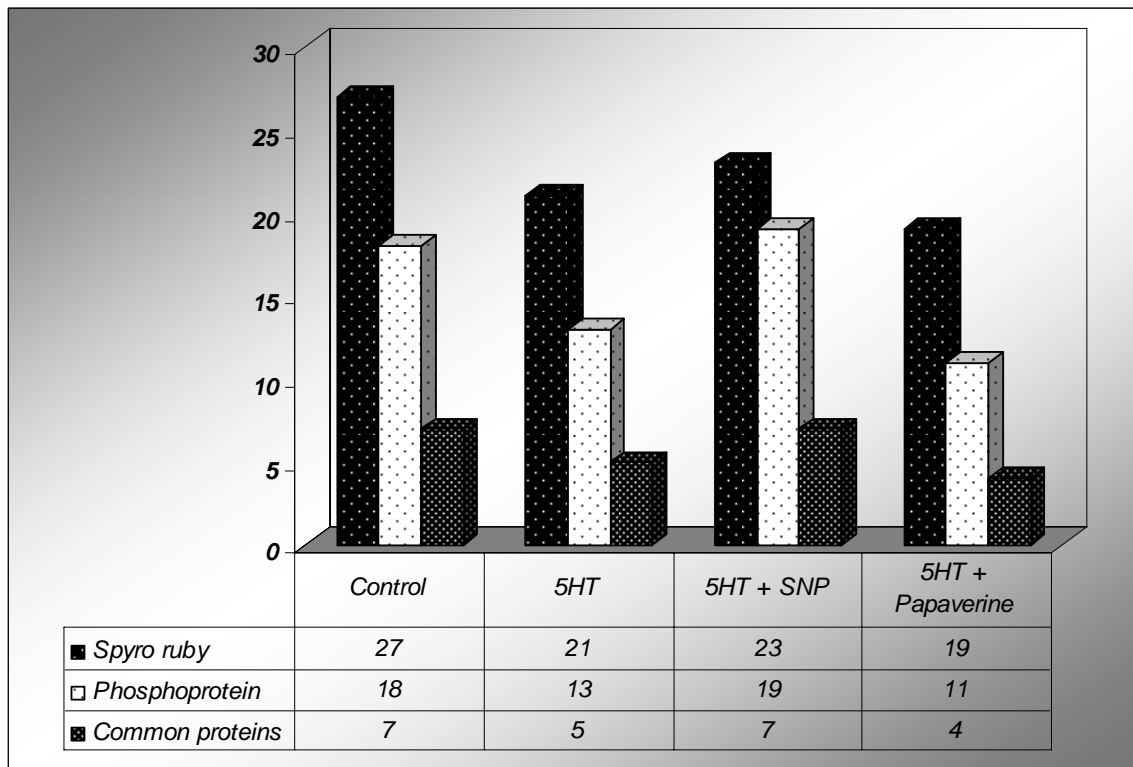


Figure 27: Total proteins picked up by spyro ruby stain, total phosphorylated proteins picked up by phosphoprotein stain and common proteins from spyrorubystain and phosphoprotein stain at threshold 70

Spyro ruby stain		Common proteins
Control	5HT	11
Control	5HT + SNP	12
Control	5HT + Papaverine	9
5HT	5HT + SNP	8
5HT	5HT + Papaverine	6
5HT + SNP	5HT + Papaverine	9

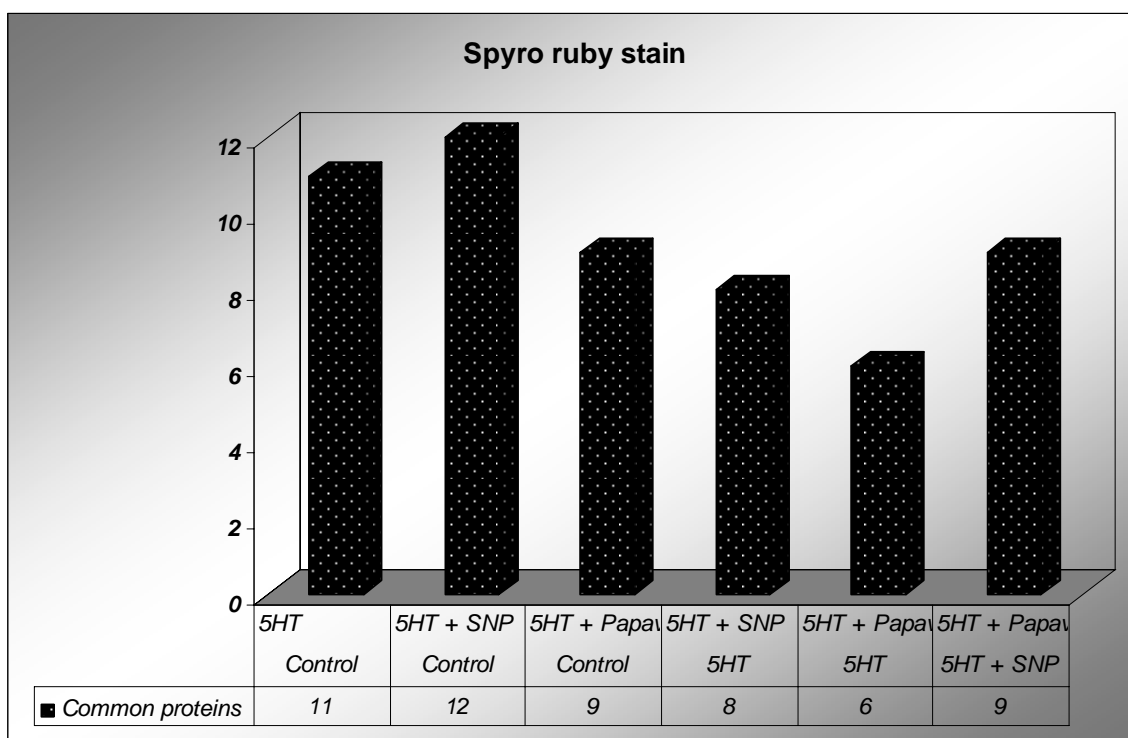


Figure 28: Total Number of proteins picked up by Spyro ruby stain

	Phosphoprotein stain	Common proteins
Control	5Ht	13
Control	5HT + SNP	10
Control	5HT + Papaverine	12
5HT	5HT + SNP	8
5HT	5HT + Papaverine	6
5HT + SNP	5HT + Papaverine	12

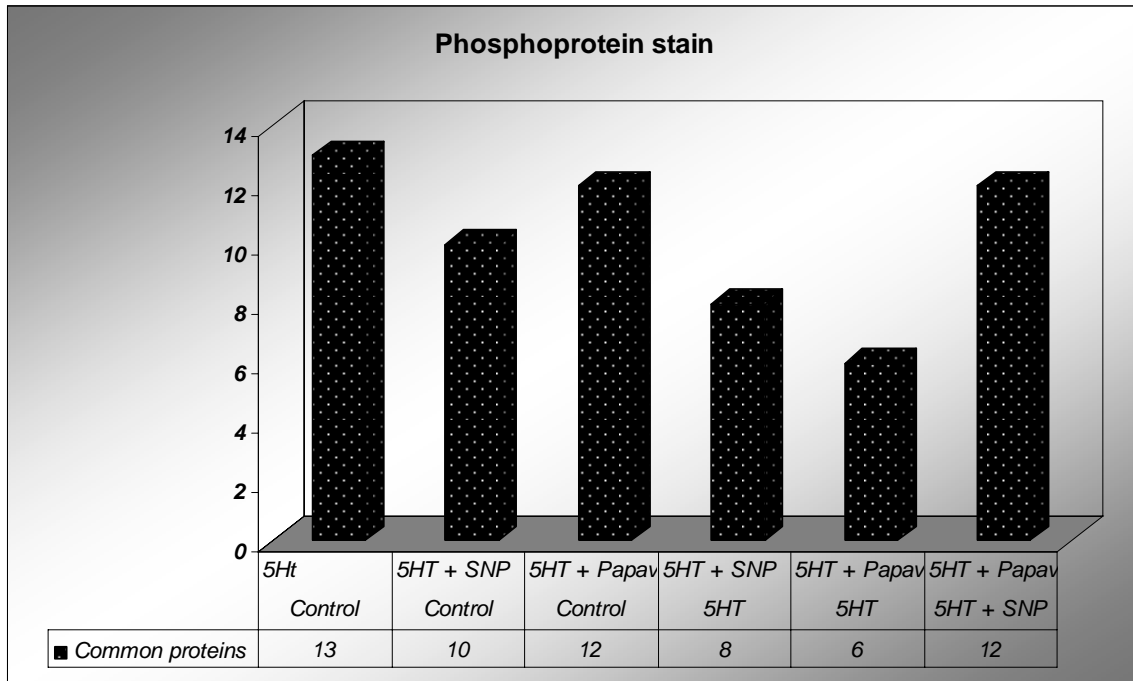


Figure 29: Total Number of Phosphorylated proteins picked up by Phospho Protein stain

There is minimal change in the total number of proteins picked up by spyro ruby stain during control, contraction and relaxation of vascular smooth muscle.

There is decrease in the total number of phosphorylated proteins picked up by phospho protein stain during control, contraction and relaxation of vascular smooth muscle when compared to the total number of proteins.

Future work:

To develop second version of 2D protein gel analysis tool which can match proteins based on the intensity.

To study phosphorylated proteins by using anti-phosphoamino acid antibodies.

Discussions:

2-D gel electrophoresis achieves very high separation efficacy. However, the technique does however have many serious limitations.

For instance it is not particularly good at resolving proteins or peptides with a low molecular mass as these migrate through the polyacrylamide gel too rapidly. It is also unsuitable for many proteins, particularly hydrophobic ones, as these interact unfavourably with the gel matrix.

Analyses performed by gels are not particularly quantitative due to the differing amounts proteins take up the staining reagents. Careful use of calibrated standards does allow some quantification, but even this is somewhat imprecise.

Proteins often react with the gel matrix. This leads to differing amounts of acrylamide-protein polymers which makes identification of proteins masses difficult.

2DE takes a long time to perform. Each of the individual stages requires quite a long time to run, and much user intervention is required. . This makes it a long and laborious process which requires some skill to master.

Performing 2DE is very much an art, requiring much experimentation to find the correct conditions for sample preparation, focusing times etc. If these are not optimised then artifacts can easily be introduced into the 2D pattern

References:

1. <http://www3.umdj.edu/histsweb/lab6/smoothmuscle/smcarrangement.html>
2. <http://www.answers.com/topic/proteome-2>
3. http://www.meddean.luc.edu/lumen/meded/grossanatomy/dissector/labs/thorax/heart/he2_1b.htm
4. <http://www.abrf.org/ABRFNews/1996/December1996/Proteome.html>
5. <http://academickids.com/encyclopedia/p/ph/phosphorylation.html>
6. <http://www.beyondgenome.com/pro.asp>
7. <http://www.answers.com/topic/phosphorylation>
8. <http://www.chemsoc.org/exemplarchem/entries/2002/proteomics/home.htm>
9. <http://www.pil.sdu.dk/Mann%20Jensen%20PTM%20Nature%20Biotech%20March2003.pdf>
10. <http://en.wikipedia.org/wiki/Phosphorylation>
11. <http://users.rcn.com/jkimball.ma.ultranet/BiologyPages/M/Muscles.html>
12. <http://ect.downstate.edu/courseware/histomanual/muscle.html>
14. http://www.chemsoc.org/exemplarchem/entries/2004/bristol_rosling/My%20Webs/introduction.htm
15. <http://www.nlm.nih.gov/medlineplus/druginfo/uspdi/202438.html>
16. <http://wine1.sb.fsu.edu/BCH4053I/Lecture03/Lecture03.htm>
17. [http://www4.amershambiosciences.com/applic/upp00738.nsf/vLookupDoc/172581038-R140/\\$file/80-6429-60AC.pdf](http://www4.amershambiosciences.com/applic/upp00738.nsf/vLookupDoc/172581038-R140/$file/80-6429-60AC.pdf)
18. http://www.biochem.mcw.edu/protein_facility/2D.html
19. http://cats.med.uvm.edu/cats_teachingmod/microbiology/courses/genomics/prokaryotic_proteomics/6.3_prok_prot_2dgelmeth.html

20. www.invitrogen.com/Content/SFS/ProductNotes/F_050810_ProQDiamond-TS-TL-MKT-HL.pdf
21. http://www.chemsoc.org/exemplarchem/entries/2002/proteomics/basic_proteins.htm
22. www.proteomesystems.com/uploads/PG6311e.pdf
23. http://nationaldiagnostics.com/article_info.php/articles_id/65
24. [dbk.ch.umist.ac.uk/Papers/analyst_129_\(542\).pdf](http://dbk.ch.umist.ac.uk/Papers/analyst_129_(542).pdf)
25. http://www.bearcave.com/misl/misl_tech/wavelets/histo/
26. <http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=1763>
27. www2.chass.ncsu.edu/garson/pa765/cluster.htm
28. <http://www.tau.ac.il/lifesci/units/proteomics/2dimgel.html>
29. www-lmmb.ncifcrf.gov/EP/EPemail.html
30. www.abrf.org/ABRFNews/1995/June1995/jun952dgelwkshop.html
31. <http://documents.wolfram.com/applications/digitalimage/UsersGuide/ImageSegmentation/ImageProcessing7.2.html>

Review of literature:

Proteome research: two-dimensional gel electrophoresis and identification methods / Th. Rabilloud (ed.). Berlin; New York : Springer, c2000

Two-dimensional gel electrophoresis of proteins : methods and applications / edited by Julio E. Celis and Rodrigo Bravo. New York : Academic Press, 1984

Image segmentation by histogram thresholding using fuzzy sets Tobias, O.J.; Seara, R.; Image Processing, IEEE Transactions on Volume 11, Issue 12, Dec. 2002 Page(s):1457 - 1465

P_aneek, J. & Vohradsk_y, J. (1999). Point pattern matching in the analysis of twodimensional gel electropherograms. In Electrophoresis, 20, 3483-3491

Takahashi K. Watanabe, Y. (1998). A Fast Structural Matching and its Application to Pattern Analysis of @-D Electroporesis Images. In Proc. 1998 Int. Conf. on Image Processing, 804-808

Lemkin, P.F. Lipkin, L.E. (1981). GELLAB: A computer system for 2D gel electrophoresis analysis I: segmentation of spots and preliminaries. In Computational Biomedical Resources, 14, 272-297

Ho_mann, F. Kriegel, & Wenk, C. (1998). Matching 2D patterns of protein spots. In Proceedings SoCG' 98, 231-239

Conradsen, K. & Pedersen, J. (1992). Analysis of Two-Dimensional Electrophoretic Gels. In Biometrics, 48 (4), 1273-1287

Bettens E., Scheunders P., Sijbers J., Van Dyck D., Moens L. (1996). Automatic segmentation and modelling of two-dimensional electrophoresis-gels. In Proceedings ICIP'96: vol.2 / IEEE International Conference on Image Processing, 665-668

Pro-Q Diamond phosphoprotein stain: a new reagent for detection of phosphoproteins and phosphopeptides in polyacrylamide gels and in microarrays (Presented at the Association of BioMolecular Resource Facilities conference, Denver, Colorado; February 10-13, 2003)

H. Ah, F. Hoffmann, K. Kriegel, C. Wenk, K.-R PleiBner, CAROL - New Algorithmic Tools for Comparing Two-Dimensional Electrophoretic Gel Images, Proceedings Electrophoresis Forum '97, Strasbourg, P 21

P. F. Lembkin, Comparing two-dimensional electrophoretic gel images across the Internet, Electrophoresis 18(3-4) (1997), 461-470

Roberto Manduchi, Carlo Tomasi: Distinctiveness Maps for Image Matching. ICIAP 1999: 26-31

F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets. In Proceedings of European Conference on Computer Vision, volume 1, pages 414–431. Springer-Verlag, 2002.

F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets. In Proceedings of European Conference on Computer Vision, volume 1, pages 414–431. Springer-Verlag, 2002.

David G. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 2004.

T. Tuytelaars V. Ferrari and L. Van Gool. Simultaneous object recognition and segmentation by image exploration. In Proc. 8th European Conference on Computer Vision, 2004.

Appendix

Matlab Code

```
function varargout = GUIDemo(varargin)
% GUIDEMO M-file for GUIDemo.fig
%   GUIDEMO, by itself, creates a new GUIDEMO or raises the existing
%   singleton*.
%
%   H = GUIDEMO returns the handle to a new GUIDEMO or the handle to
%   the existing singleton*.
%
%   GUIDEMO('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in GUIDEMO.M with the given input
%   arguments.
%
%   GUIDEMO('Property','Value',...) creates a new GUIDEMO or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before GUIDemo_OpeningFunction gets called. An
%   unrecognized property name or invalid value makes property
%   application
%   stop. All inputs are passed to GUIDemo_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only
%   one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUIDemo

% Last Modified by GUIDE v2.5 09-Aug-2005 20:39:51

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @GUIDemo_OpeningFcn, ...
                  'gui_OutputFcn',  @GUIDemo_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```

```

% --- Executes just before GUIDemo is made visible.
function GUIDemo_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUIDemo (see VARARGIN)

% Choose default command line output for GUIDemo
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUIDemo wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = GUIDemo_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

% --- Executes during object creation, after setting all properties.
function inEdit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to inEdit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc

```

```

    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function inEdit_Callback(hObject, eventdata, handles)
% hObject    handle to inEdit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of inEdit as text
%        str2double(get(hObject,'String')) returns contents of inEdit as a
double

```

```

% LOAD FIRST IMAGE--- Executes on button press in loadPush.
function loadPush_Callback(hObject, eventdata, handles)
% hObject    handle to loadPush (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

image_file = get(handles.inEdit,'String');
if ~isempty(image_file)
im_original=imread(char(image_file));
set(handles.original,'HandleVisibility','OFF');
set(handles.duplicate,'HandleVisibility','OFF');
set(handles.newIm,'HandleVisibility','OFF');
set(handles.orgIm,'HandleVisibility','ON');
axes(handles.orgIm);
image(im_original);
axis equal;
axis tight;
axis off;
set(handles.orgIm,'HandleVisibility','OFF');
end;

```

```

% SLIDER--- Executes during object creation, after setting all properties.
function intSlider_CreateFcn(hObject, eventdata, handles)
% hObject    handle to intSlider (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: slider controls usually have a light gray background, change
%       'usewhitebg' to 0 to use default. See ISPC and COMPUTER.
usewhitebg = 1;
if usewhitebg
    set(hObject,'BackgroundColor',[.9 .9 .9]);
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

akulk1 =0;

% --- Executes on slider movement.
function intSlider_Callback(hObject, eventdata, handles)
% hObject    handle to intSlider (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider
t = get(handles.intSlider,'value');
set(handles.valText,'String',num2str(t));

% -APPLY FIRST IMAGE-- Executes on button press in pushbutton17.
function pushbutton17_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton17 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
scale = get(handles.intSlider,'value');
image_file = get(handles.inEdit,'String');
orim=imread(char(image_file));

orim = double(orim);

[mic, nic , oic] =size(orim);

umc(mic,nic) = 0;
jmc(mic,nic) =0;
kmc(mic,nic) =0;
pc=orim;
for ic= 1:mic
    for jc = 1:nic

        jmc(ic,jc)=pc(ic,jc);

        if(jmc(ic,jc)>scale)
            umc(ic,jc) = 255;
            kmc(ic,jc) =0;
        else
            umc(ic,jc) = 0;
            kmc(ic,jc) =255;
        end
    end
end
nimav =kmc;

axis equal;
axis tight;
axis off;
set(handles.newIm,'HandleVisibility','ON');
set(handles.original,'HandleVisibility','OFF');
set(handles.orgIm,'HandleVisibility','OFF');
set(handles.duplicate,'HandleVisibility','OFF');

```

```

axes(handles.newIm);

imagesc(nimav);
colormap(gray);
%pixval on

axis equal;
axis tight;
axis off;
[iji,jij,n]=fastcentroid(umc);
len = size(iji)
nnn = size(iji)
for iu = 1:nnn(2)

% first figure

number = iu
x_cord = round(iji(iu))
y_cord = round(jij(iu))
intens = orim(x_cord,y_cord)

ashu(iu,1) = number;
ashu(iu,2) = x_cord ;
ashu(iu,3) = y_cord;
ashu(iu,4) = intens;
sprintf('numbers=%d x_cord=%d y_cord=%d
intensity=%d',ashu(1),ashu(2),ashu(3),ashu(4))
fid = fopen('ashu1.mat','w');
fprintf(fid,'%d %d %d %d \n',ashu');
fclose(fid);
set(handles.listbox2,'String',num2str(ashu));

set(handles.edit10,'String',num2str(number));
end
set(handles.newIm,'HandleVisibility','OFF');

% VIEW LOCATION FIRST IMAGE--- Executes on button press in
pushbutton18.
function pushbutton18_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton18 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
scale = get(handles.intSlider,'value');
image_file = get(handles.inEdit,'String');
orim=imread(char(image_file));
orim = double(orim);
gen =rgb2gray(orim);
[mic, nic , oic] =size(orim);
count1 = 0;

```

```

umc(mic,nic) = 0;
jmc(mic,nic) =0;
kmc(mic,nic) =0;
umn(512)=0;
umx(512)=0;
pc=orim;
for ic= 1:mic
    for jc = 1:nic

        jmc(ic,jc) = pc(ic,jc);

        if(jmc(ic,jc)>scale)
            count1 = count1+1;
            umc(ic,jc) = 255;
            kmc(ic,jc) =0;
            umn(count1)=jmc(ic,jc);
            umx(umn(count1) )= umx(umn(count1) )+1;
        else
            umc(ic,jc) = 0;
            kmc(ic,jc) =255;
        end
    end
end

end
% umx
nimav =kmc;
%figure(15);
% ux= imhist(gen);
%plot(umx);

figure(1)
imagesc(nimav);

% fclose(fid);

colormap(gray);
pixval on

% NUMBER OF PROTEINS- FIRST IMAGE-- Executes during object
creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
end

```

```

function edit10_Callback(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit10 as text
%        str2double(get(hObject,'String')) returns contents of edit10 as a
double

% --- Executes during object creation, after setting all properties.
function listbox2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listbox2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

% --- Executes on selection change in listbox2.
function listbox2_Callback(hObject, eventdata, handles)
% hObject    handle to listbox2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns listbox2 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from listbox2

% --- Executes on button press in closePush.
function closePush_Callback(hObject, eventdata, handles)
% hObject    handle to closePush (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close all;

%%
%%SECOND IMAGE
%
% --- LOAD SECOND IMAGE Executes on button press in pushbutton13.
function pushbutton13_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles    structure with handles and user data (see GUIDATA)
image_filea = get(handles.edit6,'String');
if ~isempty(image_filea)
im_originala=imread(char(image_filea));
set(handles.original,'HandleVisibility','ON');
set(handles.duplicate,'HandleVisibility','OFF');
set(handles.newIm,'HandleVisibility','OFF');
set(handles.orgIm,'HandleVisibility','OFF');
axes(handles.original);
image(im_originala);
axis equal;
axis tight;
axis off;
set(handles.original,'HandleVisibility','OFF');
end;

% Hint: get(hObject,'Value') returns toggle state of checkbox5

% --- Executes during object creation, after setting all properties.
function slider5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background, change
%       'usewhitebg' to 0 to use default.  See ISPC and COMPUTER.
usewhitebg = 1;
if usewhitebg
    set(hObject,'BackgroundColor',[.9 .9 .9]);
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

% --- Executes on slider movement.
function slider5_Callback(hObject, eventdata, handles)
% hObject    handle to slider5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider

ta = get(handles.slider5,'value');
set(handles.text13,'String',num2str(ta));

akulk = 0;
% --- Executes on button press in pushbutton14.
%APPLY SECOND IMAGE BUTTON
function pushbutton14_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton14 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
%Flaga = get(handles.checkbox6,'value');%This flag is to indicate if the
check box is selected.
scalea = get(handles.slider5,'value');
image_filea = get(handles.edit6,'String');
orima=imread(char(image_filea));

orima = double(orima);

[mi, ni , oi] =size(orima);

um(mi,ni) = 0;
jm(mi,ni) =0;
km(mi,ni) =0;
p=orima;
for i= 1:mi
    for j = 1:ni

        jm(i,j) = p(i,j);

        if(jm(i,j)>scalea)
            um(i,j) = 255;
            km(i,j) =0;
        else
            um(i,j) = 0;
            km(i,j) =255;
        end
    end
end
nimac =km;

axis equal;
axis tight;
axis off;
set(handles.newIm,'HandleVisibility','OFF');
set(handles.original,'HandleVisibility','OFF');
set(handles.orgIm,'HandleVisibility','OFF');
set(handles.duplicate,'HandleVisibility','ON');

axes(handles.duplicate);

imagesc(nimac);
colormap(gray);

axis equal;
axis tight;
axis off;

[iji,jij,n] =fastcentroid(um);
len = size(iji);

```

```

nnn = size(iji);
for iu = 1:nnn(2)

% second figure

    numbervin = iu ;
    x_cordvin = round(iji(iu)) ;
    y_cordvin = round(jij(iu));
    intensvin = orima(x_cordvin,y_cordvin);
    ashuvin(iu,1) = numbervin;
    ashuvin(iu,2) = x_cordvin;
    ashuvin(iu,3) = y_cordvin;
    ashuvin(iu,4) = intensvin;
    fid = fopen('ashu.mat','w');
    fprintf(fid,'%d %d %d %d \n',ashuvin');
    fclose(fid);
    sprintf('numbers=%d x_cordinate=%d y_cordinate=%d
intensity=%d',ashuvin(1),ashuvin(2),ashuvin(3),ashuvin(4))
    set(handles.listbox1,'String',num2str(ashuvin));
set(handles.edit9,'String',num2str(numbervin));
end

set(handles.duplicate,'HandleVisibility','OFF');

% --- Executes on button press in pushbutton15.
function pushbutton15_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton15 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close all;
% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
function edit9_Callback(hObject, eventdata, handles)

% ccc=pallavi;
% set(handles.ccc,'String', ccc);

```

```

% --- VIEW LOCATION SECOND IMAGE Executes on button press in
pushbutton16.
function pushbutton16_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton16 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%Flaga = get(handles.checkbox6,'value');%This flag is to indicate if the
check box is selected.
scalea = get(handles.slider5,'value');
image_filea = get(handles.edit6,'String');
orima=imread(char(image_filea));

orima = double(orima);

    [mi, ni , oi] =size(orima);

um(mi,ni) = 0;
jm(mi,ni) =0;
km(mi,ni) =0;
p=orima;
    for i= 1:mi
        for j = 1:ni

            jm(i,j) = p(i,j);

            if(jm(i,j)>scalea)
                um(i,j) = 255;
                km(i,j) =0;
            else
                um(i,j) = 0;
                km(i,j) =255;
            end
        end
    end

    end
    nimac =km;
figure(2)
imagesc(nimac);

    colormap(gray);
pixval on
%end

% --- Executes during object creation, after setting all properties.
function listBox1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listBox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: listbox controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

% --- Executes on selection change in listbox1.
function listbox1_Callback(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns listbox1 contents as cell
array
%     contents{get(hObject,'Value')} returns selected item from listbox1

```

```

% --- Executes during object creation, after setting all properties.
function listbox3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listbox3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: listbox controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

% --- Executes on selection change in listbox3.
function listbox3_Callback(hObject, eventdata, handles)
% hObject    handle to listbox3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns listbox3 contents as cell
array
%     contents{get(hObject,'Value')} returns selected item from listbox3

```

```

% --- Executes during object creation, after setting all properties.
function listbox4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listbox4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

% --- Executes on selection change in listbox4.
function listbox4_Callback(hObject, eventdata, handles)
% hObject    handle to listbox4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns listbox4 contents as cell
array
%       contents{get(hObject,'Value')} returns selected item from listbox4

% --- SIMILAR PROTEINS IN FIRST AND SECOND IMAGE Executes on
button press in pushbutton19.
function pushbutton19_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton19 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
uintx = [];
set(handles.listbox3,'String',num2str(uintx));

fid = fopen('ashu.mat','r')
ashu = fscanf(fid,' %d %d %d %d ' )
n = size(ashu);
n1 = n(1)/4;
ashu = reshape(ashu,4,n1);
fclose(fid);

clear midni;
fid = fopen('ashu1.mat','r')
ashu1 = fscanf(fid,' %d %d %d %d ' )
na = size(ashu1);
na1 = na(1)/4;
ashu1 = reshape(ashu1,4,na1);
fclose(fid);
ak = ashu;

```

```

ak1 =ashu1;

% [axc ]= corre(ak,ak1);
arry =ak;
arry1 =ak1;
m = size(arry);
n = size(arry1);
u =1;
clear midni;
midni(1:300) = 0;
mix =midni;
u=0;
ux=0;
m
n
uintx(1:4,1:50) =0;
if(m(2)>=n(2))
for i = 1:m(2)
for j = 1:n(2)
u= u+1;
midni(u) = abs((arry(2,i) - arry1(2,j))) + abs((arry(3,i) - arry1(3,j)));
if( midni(u) <=5)
ux= ux+1;
mix(ux) = i;
uintx(1:4,ux) = arry(1:4,i);
end
end
end

u =0;
ux =0;
if(n(2)>=m(2))
clear uintx;
for i = 1:m(2)
for j = 1:n(2)
u= u+1;
midni(u) = abs((arry(2,i) - arry1(2,j))) +abs( (arry(3,i) - arry1(3,j)));
if( midni(u) <=5)
ux= ux+1;
mix(ux) = j;
uintx(1:4,ux) = arry1(1:4,j);
end
end
end

scalea = get(handles.slider5,'value');
image_filea = get(handles.edit6,'String');
orima=imread(char(image_filea));

```

```

orima = double(orima);

[mi, ni , oi] =size(orima);

um(mi,ni) = 0;
jm(mi,ni) =0;
km(mi,ni) =0;
p=orima;
for i= 1:mi
    for j = 1:ni

        jm(i,j) = p(i,j);

        if(jm(i,j)>scalea)
            um(i,j) = 255;
            km(i,j) =0;
        else
            um(i,j) = 0;
            km(i,j) =255;
        end
    end
end
[nx,mx,wx] = size(uintx);
nimac =km;
figure(3)
hold on;
imagesc(km);
colormap(gray)

for j = 1:mx
    plot(uintx(3,j),uintx(2,j),'r+')% the c olor of the
end

scale = get(handles.intSlider,'value');
image_file = get(handles.inEdit,'String');
orim=imread(char(image_file));
orim = double(orim);

[mic, nic , oic] =size(orim);

umc(mic,nic) = 0;
jmc(mic,nic) =0;
kmc(mic,nic) =0;
pc=orim;
for ic= 1:mic
    for jc = 1:nic

        jmc(ic,jc) = pc(ic,jc);

        if(jmc(ic,jc)>scale)
            umc(ic,jc) = 255;
            kmc(ic,jc) =0;
        end
    end
end

```

```

        else
            umc(ic,jc) = 0;
            kmc(ic,jc) =255;
        end
    end
end

end
nimav =kmc;
figure(4)
hold on;
imagesc(kmc);
colormap(gray)

for j = 1:mx
    plot(uintx(3,j),uintx(2,j),'r+')% the color of the
end

uintx;
mix;
set(handles.listbox3,'String',num2str(uintx));

```