A Data Mining Approach for High-Content Fluorescence Microscopy Images of Tissue Samples

Der Technischen Fakultät der Universität Bielefeld
vorgelegt zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften

B.Sc. Julia Herold
Bielefeld, Germany

Biodata Mining & Applied Neuroinformatics Group
Faculty of Technology
Bielefeld University

Gedruckt auf alterungsbeständigem Papier gemäß ISO 9706

Acknowledgments

This work was carried out in the Biodata Mining & Applied Neuroinformatics Group, headed by Jun. Prof. Tim W. Nattkemper, at the Faculty of Technology, University of Bielefeld. It was performed in collaboration with the Molecular Pattern Recognition Research Group, at the Institute of Medical Neurobiology, University of Magdeburg, headed by Dr. Walter Schubert, and the Department of Biological Sciences, University of Warwick. The work was supported by a grant of the ”International NRW Graduate School in Bioinformatics and Genome Research”.

First of all I thank my supervisor Tim W. Nattkemper for giving me the opportunity to work in this exciting field of research. I thank him for his constant support and encouragement and the inspiring ideas he gave throughout all stages of my work. His optimism and enthusiasm had an invaluable impact.

I thank Walter Schubert for providing the interesting biological topic and for the exciting discussions. Thanks also go to David Epstein for initializing and supporting the cooperation with the Biological Department at the University of Warwick. I thank Karsten Niehaus for always being available to answer biological questions and Jens Stoye for encouraging me to start the PhD studies and for his continuous support.

The Biodata Mining & Applied Neuroinformatics Group, as well as the Neuroinformatics Group, were excellent and enjoyable workplaces. Not just because of the well maintained infrastructure, where special thanks go to Olli, but also because of the friendly atmosphere among all colleagues. I especially want to thank Christian, who shared the office with me, for the numerous discussions and the cheerful moments, and Jörg for continuous support in scientific topics. I also thank Yvonne, Timm, Andrea and Yevgen who made their thesis or a student project under my supervision and gave important contributions to this work.

I thank my collaborators at the University of Magdeburg and at the University of Warwick, for the fruitful cooperation and for answering all my biological questions.

Special thanks go to Kai, Michael, Jörg, Christian, Sebastian, Thomas, and Wiebke for proofreading the whole or parts of this manuscript.

Finally, I would like to thank my family and Thomas for their belief in me, their patience, and all their emotional support. Without them, this work would not have been possible.

Originality of Work

The work presented in this thesis was conceived and carried out by myself under supervision of Tim W. Nattkemper. Clustering with the H^{2}SOM was carried out in collaborative work by Jörg Ontrup who also supervised the construction of the web-based H^{2}SOM visualization.

To my parents.
For all their support throughout my life.

Contents

1 Introduction **1**

1.1 Organization of the Thesis . . . 2

2 Imaging for Protein Colocation Studies **5**

2.1 Multi Protein Fluorescence Microscopy . . . 7

2.2 Image Data . . . 8

2.2.1 Pancreas Tissue Samples . . . 8

2.2.2 Brain Tissue Samples . . . 10

2.3 Summary . . . 15

3 An Exploration System for Multivariate Fluorescence Tissue Images **17**

3.1 Existing Methods for the Exploration of Multivariate TIS Data . . . 18

3.2 Demands on an Object-Based Analysis Strategy for Non-Binary Multivariate Images . . . 20

3.2.1 Semantic Image Annotation . . . 22

3.2.2 Extraction of Object-Specific Features . . . 25

3.2.3 Data Interpretation . . . 25

3.3 Summary . . . 28

4 Supervised Learning-Based Object Detection in Tissue Micrographs **29**

4.1 Accuracy Assessment . . . 29

4.2 Human Synapse Detection Performance . . . 31

4.2.1 Discussion . . . 34

4.3 Requirements Posed on a Supervised Learning-Based Object Detection Strategy 35

4.4 SVM Theory . . . 35

4.4.1 Linear Support Vector Machines . . . 36

4.4.2 Non Linear Support Vector Machines . . . 39

4.4.3 Parameter Selection . . . 40

4.4.4 Probabilistic Outputs for SVMs . . . 41

4.5 The i3S for Object Detection . . . 42

4.5.1 Image Preprocessing . . . 42

4.5.2 Training Set Generation . . . 47

4.5.3 i3S Training . . . 48

4.5.4 Computation of Object Positions . . . 48

4.5.5 Estimating a Constant Confidence Threshold . . . 48

4.6 Case study I: Brain Tissue . . . 49

4.6.1 The Influence of Training Set and Threshold Choice 49
4.6.2 The Influence of the Labeling Strategy and Constant Thresholds in an Inter-Image Detection Setup 54
4.6.3 i3S Performance in a Multi Protein Detection Setup 61
4.6.4 Discussion 63
4.7 Case study II: Pancreas Tissue 65
4.7.1 Results 66
4.7.2 Discussion 66
4.8 Summary 67

5 A Direct Visual Data Mining Tool for Three-Channel High-Content Micrographs 69
5.1 Image Preprocessing 69
5.2 Feature Extraction 70
5.3 Interactive Visual Data Exploration and Cell Type Classification 72
5.4 Results 74
5.5 Discussion 76

6 An Unsupervised Learning Approach for Data Mining d-dimensional Fluorescence Images 81
6.1 Feature Calculation 82
6.1.1 Image Enhancement and Normalization 82
6.1.2 Synapse Specific Feature Calculation 83
6.2 Cluster Analysis 85
6.2.1 Online K-means 86
6.2.2 Neural Gas 87
6.2.3 Hierarchically Growing Hyperbolic Self Organizing Maps (H^{2}SOM) . . 88
6.2.4 Cluster Validation 92
6.2.5 Selecting a Distance Function 94
6.3 Visualizations of the Feature Domain for TIS Data Exploration 94
6.3.1 Prototype Visualization 96
6.3.2 Prototype Visualization via Combinatorial Intensity Profile Archetypes 97
6.3.3 Cluster Visualization 99
6.4 Visualizations in the Image Domain 101
6.4.1 Dimensionality Reduction Methods 103
6.4.2 Color Assignment 106
6.5 Statistical Comparison of Image Stacks 108
6.5.1 Pearson's Correlation Coefficient 109
6.5.2 Spearman's Rank Correlation Coefficient 110
6.6 Results 110
6.6.1 Estimating the Appropriate Number of Clusters 111
6.6.2 Stability of Cluster Results 113
6.6.3 Analysis of Protein Colocation and Inter Image Correlation 114
6.6.4 Comparison of Cluster Results of K-means, NG and H^{2}SOM 129

6.6.5 Analysis of the Whole Protein Set 135
6.6.6 The Impact of Feature Calculation 137
6.6.7 The Impact of Semantic Image Annotation 140
6.7 Discussion 144
6.8 Summary 145

7 Statistical Synapse Distribution Analysis 147
7.1 Ripley's K and O-Ring Statistic 147
7.1.1 Ripley's K Statistic 148
7.1.2 O-ring Statistic 148
7.1.3 Numerical Estimation 149
7.2 Null Models for Univariate and Bivariate Point Patterns 150
7.2.1 Univariate Null Models 150
7.2.2 Bivariate Null Models 152
7.3 Sample Application 152
7.3.1 Analysis of Synapse Distribution 153
7.3.2 Class Specific Synapse Distribution 155
7.4 Discussion 157

8 Conclusion and Outlook 159

Appendix 165
A.1 Acronyms 165
A.2 Technical Abbreviations 166
A.3 Supplementary Figures 169
A.4 Supplementary Tables 175

Bibliography 177

Publications

Parts of this thesis have been published in:

- J. Herold, W. Schubert, T.W. Nattkemper. *Automated detection and quantification of fluorescently labeled synapses in murine brain tissue sections for high throughput applications*. Journal of Biotechnology, 2010.

- J. Herold, L. Zhou, S. Abouna, S. Pelengaris, D. Epstein, M. Khan, T.W. Nattkemper. *Integrating Semantic annotation and information visualization for the analysis of multichannel fluorescence micrographs from pancreatic tissue*. Computerized Medical Imaging and Graphics, 2009.

- J. Herold, S. Abouna, L. Zhou, S. Pelengaris, D. Epstein, M. Khan, T.W. Nattkemper. *A way towards analyzing high-content bioimage data by means of semantic annotation and visual data mining*. SPIE Medical Imaging, Orlando, 2009.

- J. Herold, M. Friedenberger, M. Bode, N. Rajpoot, W. Schubert, T.W. Nattkemper. *Flexible synapse detection in fluorescence micrographs by modeling human expert grading*. Proc. of 2008 IEEE International Symposium on Biomedical Imaging (ISBI), Paris, 2008.

- J. Herold, S. Abouna, L. Zhou, S. Pelengaris, D. Epstein, M. Khan, T.W. Nattkemper. *A machine learning based system for multichannel fluorescence analysis in pancreatic tissue bioimages*. 8th IEEE International Conference on Bioinformatics and Bioengineering (BIBE), Athen, 2008.

Chapter 1

Introduction

The understanding of cellular function has intrigued researchers since the discovery of the cell as a basic module of all organisms, and the current knowledge about the cell is the result of many years of cell biological research. In the last few decades, an enormous gain in knowledge about cellular function has been achieved through genomic, transcriptomic and proteomic research. Several whole genome sequencing projects have successfully been completed, among them the sequencing of the human genome (Venter et al. (2001)). Transcriptomic and proteomic methods have helped to shed light into protein function, regulation, and interaction (Starkuviene and Pepperkok (2007)). With these methods, a wealth of information has been obtained which allows researchers to gain insight and understanding into how each single part (e.g. gene sequence, RNA molecule, protein) contributes to the whole system and thus to the function of a complete organism. This growing field of study is referred to as *systems biology* (Megason and Fraser (2007)).

Despite these advances, we are still very far from entirely understanding how a cell spatially and temporally organizes and coordinates its individual parts to establish specific cellular functions. Knowing the sequence of a gene does not directly enable to predict the function of the encoded protein, when and where it will be expressed, and how it will contribute to the whole protein network (Megason and Fraser (2007)). Furthermore, most proteomic methods do not provide spatial information of the proteins due to the need for cell homogenization (Starkuviene and Pepperkok (2007)). The spatiotemporal organization of the individual molecules, however, is existential for cellular function. Each protein has to be at the right place, at the right time and with the right concentration (Schubert et al. (2006)). Diseased and healthy samples may show similar protein concentration when analyzed by traditional -omics approaches but can reveal significant differences in their sub-cellular protein location. In order to further understand cellular function, these complex location processes and molecular networks, referred to as the *toponome* (Schubert (2003)), have to be regarded complementary to traditional -omics approaches.

This is where imaging can play an important role in systems biology (Megason and Fraser (2007)). Fluorescence microscopy has increasingly been used in recent decades to study protein localization and protein-protein interaction, not least due to the discovery of the green fluorescent protein (GFP). Fluorescence microscopy allows to obtain spatiotemporal information with single-cell or even sub-cellular resolution. For example, the genetic fusion of a protein of interest with a GFP protein, or one of its spectral variants, permits the study of the distribution of a specific protein in vivo. In combination with fluorescence resonance energy transfer (FRET), fluorescence imaging enables researchers to analyze protein-protein interaction in vivo for up to three proteins (Galperin et al. (2004)).

Furthermore, a common application of fluorescence microscopy in systems biology is the study of sub-cellular protein colocation to identify protein networks. Multiple proteins are fluorescently labeled in one sample by different fluorophores and the location of each protein is recorded in a separate image channel. This is referred to as high-content imaging (Starkuviene and Pepperkok (2007)) or multivariate imaging. Usually, these colocation studies are carried out only for a very limited number of proteins (two to three). This is mainly because it allows for a direct visual exploration of protein colocation by generating RGB fusion images and because current statistical measures for colocation analysis are limited to two proteins (cf. Bolte and Cordelières (2006)). Additionally, the number of distinct fluorophores which can be simultaneously imaged is limited to around ten due to their spectral characteristics (Murphy (2006)).

Recently, a new imaging technology has been introduced, the *multi-epitope ligand cartography* (MELC), also referred to as *toponome imaging system* (TIS) (Schubert et al. (2006); Bode et al. (2008)). This technique allows the spatial location of at least a hundred proteins on the *same* tissue or cell sample to be imaged in situ, overcoming the spectral limitation of fluorescence microscopy (Schubert et al. (2006)). Hence, colocation information of many proteins is obtained and three-channel high-content imaging is extended to d-channel high-content imaging (with d being the number of imaged proteins). While this opens up new vistas on the high complexity of the toponome, it represents a new and challenging problem for biodata analysis. Little a priori knowledge is available of what can be expected to be seen in these multivariate data sets. It is not known which proteins will be colocalized, how diverse these protein colocations can be, and what differences will be seen in, for example, diseased and healthy samples. Due to the sheer amount of data and its complex nature, manual visual analysis of the raw data is not feasible. Thus, computational approaches are required which aid the biologist in obtaining an insight and understanding of this new data domain. However, up to now there only exist very few methods for TIS image analysis.

In this thesis, I will address the question how such highly complex multivariate image data can be analyzed by means of computational methods in order to gain insight into the colocation of proteins and thus into molecular networks. Special focus is put on the exploration of protein colocation in synapses in brain tissue samples, and analysis strategies are tailored to its needs. Furthermore, a system for protein-based cell type classification in images of mouse pancreas tissue samples will be introduced. Strategies will be presented which combine concepts from the field of image processing, data mining, and information visualization to explore three-channel and especially d-channel high-content image data.

1.1 Organization of the Thesis

Chapter 2 of this thesis presents imaging methods and analysis strategies which allow for image based protein colocation studies, and provides an introduction to the toponome imaging system. Furthermore, an overview of the data sets used throughout this work, as well as the associated biological questions, are given. Chapter 3 focuses on the requirements posed on a system for TIS-based protein colocation analysis. Available methods for TIS data evaluation, as well as methods related to this field will be discussed and a modular, object-based analysis strategy is proposed. In chapter 4, a system for object detection in tissue micrographs, based on supervised machine learning, is introduced and evaluated. It serves as a basis for the object-based protein colocation studies in pancreas and brain tissue sections, introduced in chapter 5 and 6, respectively. Here, concepts from the field of data mining and information visualization are combined and extended to allow for an interactive, visual exploration of TIS data. An introduction to statistical object distribution analysis is given in chapter 7. Its applicability in object-based colocation studies will be briefly shown on the basis of the synapse data set. The thesis ends with a short conclusion and outlook in chapter 8.

Chapter 2

Imaging for Protein Colocation Studies

Analyzing the spatial organization of a protein is one crucial aspect to understand its role in biological processes. One specific cellular function is the result of multiple proteins which are at the right time, at the right place with the correct concentration and therewith form a molecular network. Traditional -omics approaches can be applied to study protein concentration, however they mostly do not provide the information of protein location. Thus, imaging and analysis strategies which allow the study of colocation for multiple proteins have to be provided as a complementary method to the traditional approaches.

Fluorescence microscopy has proven to be a powerful tool to obtain protein location information at the single-cell or sub-cellular level for both in situ and in vivo studies (Starkuviene and Pepperkok (2007); Megason and Fraser (2007); Glory and Murphy (2007)). In general, two strategies can be applied to achieve a selective labeling of one specific protein. First, a protein can be genetically fused with a fluorescent protein as the GFP or one of its spectral variants. It allows for in vivo analyses of the protein distribution or protein transport (Huh et al. (2003); Simpson et al. (2000)). Another widely used strategy is the immunolabeling of proteins. In immunolabeling, a protein is labeled by an antibody conjugated to a fluorescent dye, which is specific to the protein of interest. The drawback of this strategy is that it requires fixation of the sample and thus limits the experiment to one time point. However, no laborious genetic construction of fusion proteins is needed and potential location artifacts introduced through the fusion with GFP are avoided (Barbe et al. (2008)).

The number of available protein specific antibodies has continuously been increasing in the recent years. Hence, a great amount of molecules can selectively be labeled as has been done, for example, to construct a human protein atlas by immunocytochemistry (Uhlén et al. (2005)). This suggests that fluorescently imaging the location of a great number of proteins in one sample, and hence studying colocation for many proteins, is of no difficulty. Yet, the maximum number of fluorophores, and thus proteins, which can be simultaneously imaged in one sample is currently around ten (Murphy (2006)), a number not sufficient to analyze complex molecular networks (Schubert (2007)). Still, this field of imaging is referred to as high-content imaging. The spectral limitation of fluorescence imaging is due to the nature of the fluorophores themselves, which are characterized by distinct excitation and emission spectra. When choosing fluorophore combinations for colocation studies, their spectra have to be unambiguously distinctive. Otherwise, two fluorophores can be excited with the same wavelength because their excitation spectra overlap, known as cross-talk. Furthermore, the emission spectrum of one fluorophore can overlap with that of another fluorophore, referred to as bleed-through. Multispectral imaging followed by linear unmixing can eliminate these effects to some extent, but still only a very limited number of molecules can be imaged simultaneously (Starkuviene and Pepperkok (2007)). Due to these spectral limitations of fluorescence microscopy, alternative strategies are required which allow for large scale colocation studies despite these restrictions.

One strategy, which is completely unaffected by spectral limitations, arises from the field of pattern recognition and covers one aspect of *spatial proteomics* (Murphy (2009)). Solely one protein is labeled in one sample, i.e. only one protein image is generated. The localization pattern of this protein is then assigned to a sub-cellular location class, for example localization in the endoplasmatic reticulum, with a supervised or unsupervised machine learning strategy (see for example Huang and Murphy (2004); Conrad et al. (2004); Chen et al. (2007); Glory et al. (2008)). Supervised refers to the fact that the system was trained on samples for which the location classes were known and new images are assigned to these classes. Unsupervised strategies, on the other hand, deduce the grouping of images with similar location patterns solely from the data itself. The underlying assumption of this type of colocation study is that two or more proteins, which show the same localization pattern, are de facto colocated and form a molecular network, although they are not imaged in the same sample (Murphy (2006)). The disadvantage of this pattern recognition based approach is the fact that it is restricted to cell culture images and colocation of proteins is not tested in one single cell sample which might possibly affect the outcome. Labeling the sample for two proteins would allow to test for exact colocation of two proteins, but imaging each pairwise combination would square the number of recorded images, i.e. required samples.

Imaging mass spectrometry could be used as an alternative in order to image all proteins in one sample, as it allows protein location information to be obtained without the requirements of labeling the protein. However, its spatial resolution is still inferior to that of a fluorescence microscope (Schubert et al. (2006); McDonnell and Heeren (2007)).

An effective way to overcome the spectral limitation and image all proteins of interest in *one* cell or tissue sample is to bleach the fluorescent dye with its excitation wavelength after imaging, and re-stain another protein in the next cycle (Schubert (2007)). Hence, d-dimensional high-content imaging is possible. Since the introduction of this technique and first demonstration of its feasibility and specificity in 1990 (Schubert (1990)), an increasing amount of applications using re-staining techniques have been reported (see Schubert (2007) for an overview). A related methodology which applies elution of the antibody after imaging has recently proposed by Micheva and Smith (2007). When it comes to the imaging of a large number of proteins on one tissue or cell sample, the *multi-epitope ligand cartography* method (MELC) (Schubert et al. (2006)) is up to now unparalleled (Friedenberger et al. (2007)). In recent years, this technique has been further improved and developed as the *toponome imaging system* (TIS) (Bode and Krusche (2007)). It allows the imaging of at least 100 proteins in the same sample, providing an unprecedented insight in the complex topological ordering and colocation of proteins in intact tissue samples. However, it also raises many questions on how to interpret and analyze this high-dimensional data, i.e. extract knowledge from the many images recorded for one sample.

As this work focuses on developing strategies for the analysis of protein colocation in images of tissue samples obtained via the MELC/TIS technology, a detailed introduction to

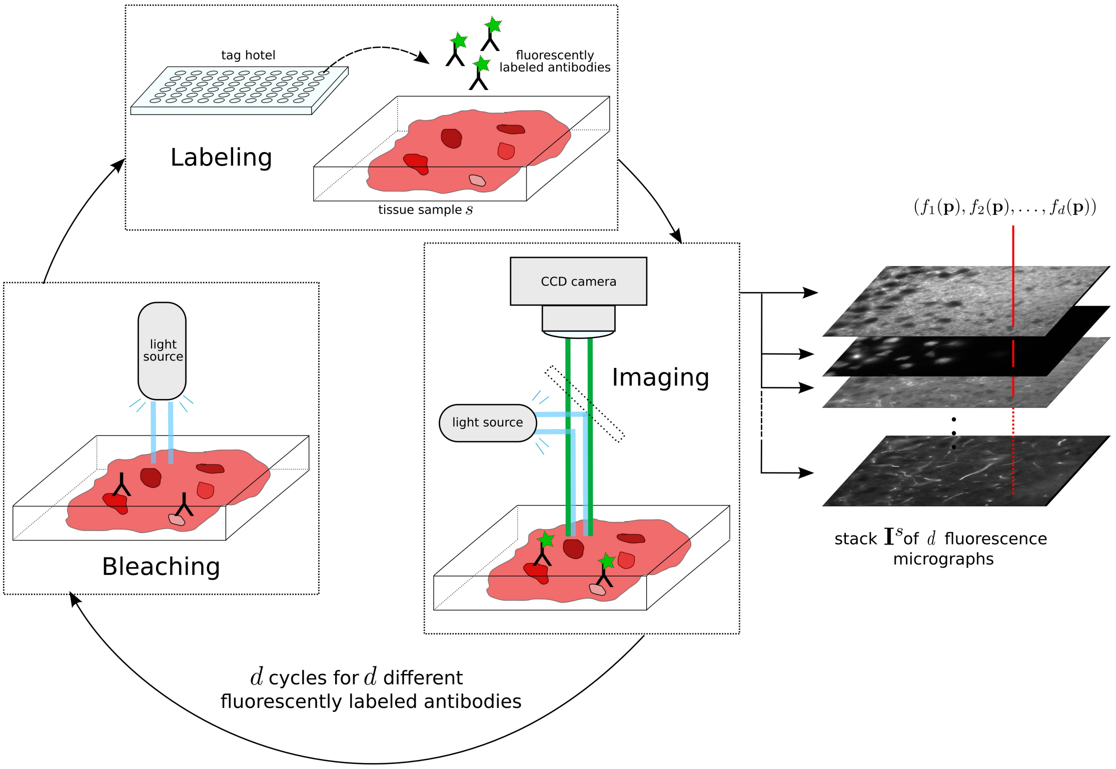

tag hotel
fluorescently labeled antibodies
Labeling
tissue sample s
$(f_1(\mathbf{p}), f_2(\mathbf{p}), \dots, f_d(\mathbf{p}))$
CCD camera
Imaging
light source
light source
Bleaching
stack $\mathbf{I}^s$of d fluorescence micrographs
d cycles for d different fluorescently labeled antibodies

Figure 2.1: Schematic illustration of the TIS imaging method. In the labeling step, a fixed tissue or cell sample s is incubated with one of d tags, e.g. antibody, conjugated to a fluorescent dye. The fluorophore is excited and the emitted light is recorded with a cooled CCD camera for image acquisition. In a bleaching phase, the fluorophore is bleached with the excitation wavelength. The rounds of labeling, imaging and bleaching are repeated for d different tags resulting in a stack of d fluorescence micrographs (images), each representing the location of a distinct protein. For each pixel location $\mathbf{p}^{(x,y)}$ (for the sake of compactness the superscript (x, y) is further omitted), a protein pattern can be extracted which represents the fluorescence intensities $f_n(\mathbf{p})$ of each protein n.

this imaging modality will be given in the following.

2.1 Multi Protein Fluorescence Microscopy

In MELC/TIS imaging, hereafter referred to as TIS only, repeated rounds of labeling, imaging, and bleaching are carried out to obtain multiple images for one sample each showing the location of a distinct protein. Figure 2.1 illustrates the imaging process via the TIS method for d different proteins. In the labeling step, a fixed tissue or cell sample is incubated with a tag conjugated to a fluorescent dye, which only binds to those regions in the sample which feature the tag specific protein. Most often, the tag is a protein specific monoclonal antibody but other affinity reagents may also be applied (Friedenberger et al. (2007)). After labeling, the fluorophores are excited and the emitted light is digitally recorded by an integrated, cooled charge-coupled device (CCD) camera, providing an intensity image showing the localization of the labeled protein. Subsequently, the fluorophores are bleached by using the excitation wavelength, resulting in the destruction of the fluorescent dye. By repeating the rounds of labeling, imaging, and bleaching for d different antibodies, a set of d fluorescent micrographs (images) is obtained, each showing the location of a different protein in the same sample. The images are automatically registered to each other through affine transformation with an accuracy of ± 1 pixel and a correction for background signals is carried out by flat-field correction (Schubert et al. (2006)). In the following, the stack of d images for a sample s is referred to as multivariate or multichannel image stack $\mathbf{I}^s$. Individual images in one stack are denoted by I_n^s, with $n = 1, \ldots, d$. For each pixel $\mathbf{p}^{(x,y)}$in the stack, a feature vector, also referred to as protein pattern, can be extracted (cf. figure 2.1). It represents the fluorescent intensity of each protein channel, i.e. image, at that specific position and thus holds colocation and anti-colocation information. For the sake of compactness, I further omit the (x, y) superscript and refer to a specific pixel as $\mathbf{p}$.

It has been shown that at least 100 proteins can be reliably localized. Furthermore, the order of the antibodies does not affect the imaging outcome (Schubert et al. (2006)). More details about the imaging technology, sample preparation etc. can be found for example in Schubert et al. (2006) and Friedenberger et al. (2007).

2.2 Image Data

In the following sections, the image data used throughout this work is presented. Two different tissue types, raising different biological questions, were used for the design of a data mining approach for high-content fluorescence microscopy images of tissue samples.

The first data set comprises only three channels per sample and thus does not reach the high-dimensionality of TIS images. However, it can be used very well to study concepts for multivariate image analysis as it allows for a direct visual interpretation of the images, unlike higher dimensional data sets. It thus provides the possibility to easily verify results generated by the proposed strategy. The second image set was obtained by the TIS technology, representing a very challenging high-dimensional data set. Highly optimized analysis strategies are required for this image set. In the following, the imaging modalities, the biological questions at hand, as well as relevant image characteristics will be introduced.

2.2.1 Pancreas Tissue Samples

The pancreas plays a fundamental role in the production of enzymes and hormones for digestion and regulation of the level of blood glucose. The regulating hormones are produced by cells forming cell agglomerations within the pancreas, known as the islets of Langerhans (Pschyrembel (2002)). The islets are mainly formed by beta (β) cells, producing the sugar lowering hormone insulin, and by alpha (α) cells which produce glucagon leading to an increased glucose level in the blood.

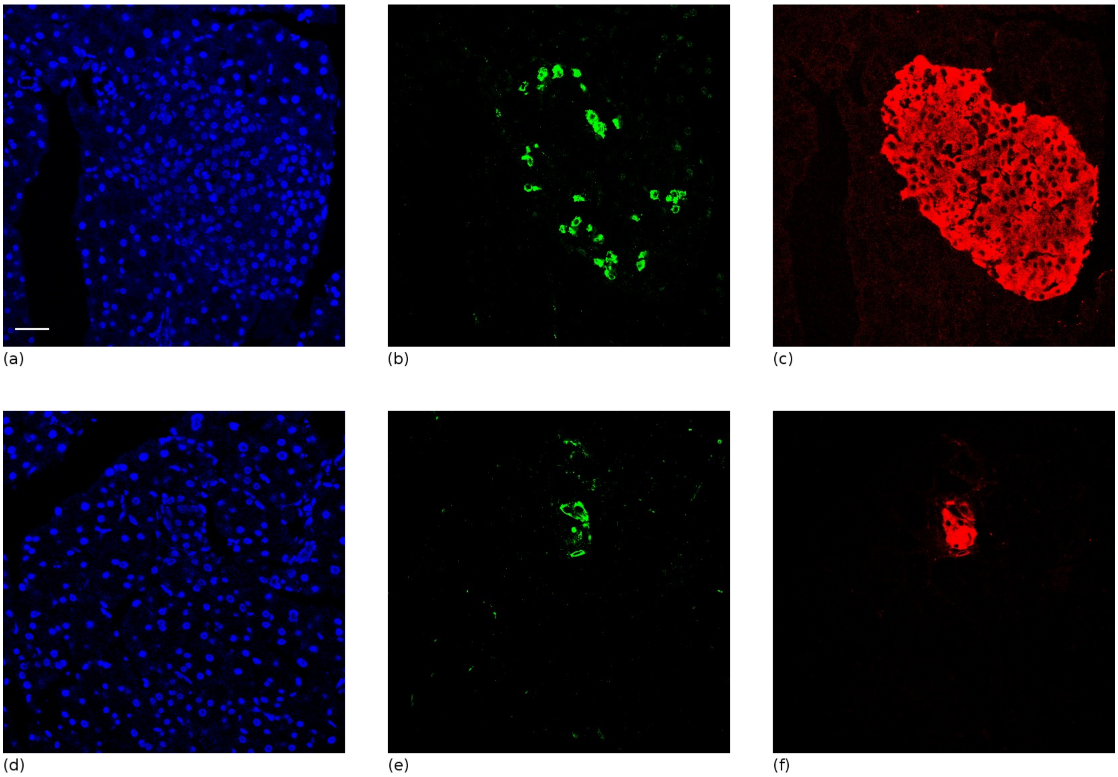


Figure 2.2: Images obtained for pancreatic tissue sections. The first row (a - c) shows images obtained from a transgenic mouse with c-Myc switched off. These images show features similar to that of healthy mice except for a slightly higher β-cell mass. Images (d-f) are taken from a transgenic mouse with c-Myc switched on to induce β-cell death, mimicking diabetes. (a) and (d) show nuclei labeled with DAPI, (b) and (e) display the α-cells and (c) and (f) β-cells. Scale bar of 37μm refers to all images.

Type 2 diabetes is a disease which affects around 200 million people worldwide and is primarily caused by a defective number of β-cells (Kulasa and Henry (2009)). Excess of α-cells and thus glucagon production, however, may contribute to hyperglycaemia. Furthermore, current drugs partly work by reducing the level of glucose. Therefore, researchers are increasingly interested in the ratio or balance of α- and β-cells for diabetes study and monitoring of new anti-diabetes treatments.

In this work, images were obtained for pancreatic tissue sections from transgenic mice. In these mice, a c-Myc protein can be selectively switched on and off to induce β-cell apoptosis in order to study β-cell regeneration (Cano et al. (2008)). Imaging was carried out with a Leica (SP2) confocal fluorescence microscope with a ×40 objective at the Biomedical Research Institute, Department of Biological Sciences, University of Warwick. For each section, three images showing cell nuclei, α-cells and β-cells were recorded. For cell nucleus identification, the sections were stained with 4',6-Diamidino-2-phenylindol (DAPI), labeling

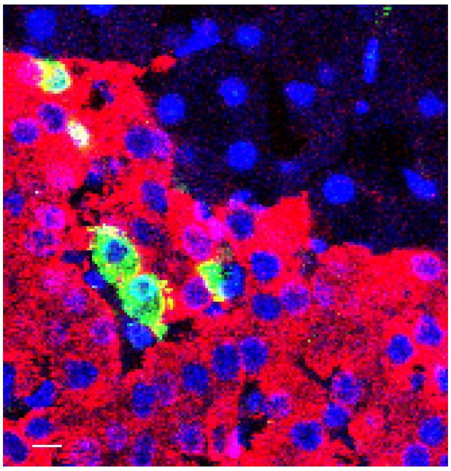


Figure 2.3: Sub-image of an overlay of all three channels of figure 2.2 (a)-(c). These RGB images are used for manual counting of alpha and beta cells. (Red: Insulin staining, Green: Glucagon staining, Blue: Nucleus staining). Scale bar is 7.5μm.

the DNA blue. For α- and β cell localization, each section was additionally immunostained against glucagon and insulin, respectively. Antibodies against glucagon were labeled with Fluoresceinisothiocyanat (FITC, Excitation wavelength 494nm) and antibodies against insulin with Alexa633 (Excitation wavelength approx. 633 nm). The pixel size is approximately 0.75μm/pixel. Image stacks were recorded for two samples with c-Myc switched off, referred to as $\mathbf{I}^{A1}$ and $\mathbf{I}^{A2}$, and for two samples with c-Myc switched on, referred to as $\mathbf{I}^{B1}$ and $\mathbf{I}^{B2}$. Figure 2.2 shows the images obtained for one section with (a) and (d) displaying cell nuclei of the islet of Langerhans and surrounding pancreatic tissue, (b) and (e) showing the glucagon cells in the islet and (c) and (f) the insulin cells. Here, figure 2.2 (a-c) correspond to a transgenic mouse with c-Myc switched off. These images show features similar to that of healthy mice except of a slightly higher β-cell mass. In figure 2.2 (d-f) c-Myc was switched on to induce β-cell death.

Traditionally, the images recorded for one sample are manually evaluated to obtain α- and β cell counts in order to study, for example, β-cell regeneration. Therefore, overlays of all three images, as shown in figure 2.3 for a sub-image of 2.2 (a)-(c), are manually analyzed and the local characteristic of insulin and glucagon staining around each nucleus is evaluated. Usually, the cell counting is performed multiple times in order to account for human experts' variability. However, for a high throughput study, manual evaluation is not feasible and a fully- or semi-automated computational approach is required.

2.2.2 Brain Tissue Samples

The brain is one of the most complex structures of an organism and up to now very little is known how it constitutes its function (Anderson and Grant (2006)). Throughout life, it continuously changes by modifying the contacts between neurons, which are referred to as synapses. Synapses are responsible for transmitting information between neurons and for processing patterns of neural activity to change the neurons' properties (Squire et al. (2008)). They have the ability to vary their function, can be replaced and can increase or decrease in number. It is, however, not well understood how these processes are organized.

The number of synapses in a brain approaches 10^{15}. It is assumed that changes in number of synapses or their effectiveness underlie neural plasticity, which is associated both with behavioral changes, such as skills and emotions, as well as mental or neurodegenerative diseases and addiction. Hence, the quantitation of synapses in defined regions of the brain or even throughout the neocortical or archeocortical brain regions would be important information for investigations on behavioral or structural brain diseases as well as for exact measurements of neuron-targeting drugs. Usually, this quantification is carried out on images obtained by electron microscopy, manual counting of synapses and subsequent extrapolation of the total number of the synapses in a defined region of the brain (Geinsman et al. (1996); Marrone et al. (2003)). With this strategy, synapses can be imaged at high resolution, which makes synapse detection very accurate.

However, not only the number of synapses but especially the proteins they express are of great interest in order to understand brain disease and brain function. Around 1000 proteins have been identified as the "average synaptic proteome" contributing to specific synaptic functions and states (Schubert et al. (2008); Anderson and Grant (2006)). Because of physiological constraints, however, it is not possible that all proteins are expressed in each single synapse. Hence, the combination of synaptic proteins is likely to be the key feature of synaptic function (Anderson and Grant (2006)). It is therefore of great importance to identify those protein colocation patterns of individual synapses which are, for example, related to neurodegenerative diseases.

By imaging with the TIS technology, both aspects, i.e. synapse number and synapse protein colocation, can be analyzed simultaneously. Synapses can be made visible by labeling with a robust synapse marker. With respect to electron microscopy, it features the benefit that synapses can be mapped on a morphological level of brain tissue and larger regions of the brain can be imaged with faster recording times. Mosaics of light microscopic images can even give an overview of entire brain regions. Furthermore, the labeling and bleaching strategy of TIS allows multiple protein channels to be imaged in one sample, and thus provides exact protein colocation information at synapses.

The most common strategy to label synapses is to use synaptophysin as a robust synapse marker, because a wealth of literature indicates that the corresponding punctuate fluorescence signal is indeed confined to the area of the synapse (for example Calhoun et al. (1996); Silver and Stryker (2000); Mouton et al. (1997)). Hence, presence of synaptophysin, which is associated with synaptic vesicles (Schubert et al. (1991)), is an indicator of the presence of chemical synapses in brain tissue sections, while absence of this signal does not exclude presence of synapses. For example, immature synapses occurring during early stages of synaptic turnover may not (yet) contain synaptophysin. Thus, certain stages of synaptic plasticity may not be detected by this marker and therefore not all present synapses might be visible. Hence, if the intention is to detect almost all synapses in one sample, additional markers need to be imaged. As the TIS allows multiple proteins to be imaged, it is obvious that any other marker protein, which shows a good confinement to the synaptic area, could be used to detect all stages of synaptic plasticity. Such markers may include, for example, proteins of postsynaptic structures, such as postsynaptic NMDA or AMPA families of receptors, or presynaptic cytoskeletal proteins expressed earlier than synaptophysin in the maturing synapse. Detecting synapses in images labeled for synaptophysin, synapsin and `nmdr1` (cf. table 2.1), will allow the imaging of almost all present synapses (Schubert (2006)).

It is evident that neither manual counting of synapses nor the manual analysis of protein colocation at synapses is feasible when a larger number of samples has to be analyzed and/or several protein channels are imaged. Thus, computational approaches are required which aid the investigator in their process of image analysis and knowledge discovery. In order to set up a computational system which allows for synapse counting and protein colocation analysis in TIS image data, two sets of image stacks were used throughout this work.

For the analysis of synaptic protein colocation and implementation of a computer aided analysis strategy for protein colocation study, five tissue samples of mouse brain hippocampal CA3 regions were provided by the Molecular Pattern Recognition Research Group, Magdeburg University. Each sample comprises the stratum radiatum (SR) and the stratum pyramidale (SP) region of the CA3 area with synapses densely packed in the SR area and arranged around neuronal parikarya in the SP. Each sample was imaged with TIS for 22 different proteins (cf. table 2.1). Out of these 22 proteins, 13 are clear-cut synaptic markers belonging to AMPA and NMDA protein families, and 9 are intermediate neuronal filaments and neuronal cell bodies, mainly for visual orientation. Sample preparation and imaging followed a strict protocol with 5μm thick sections, a ×63 oil immersion objective (1.4 aperture), sampling frequency of two pixels and specific incubation times and concentrations of antibodies. Images of size 658×517 with a pixel size of 216nm/pixel and 16bit/pixel were obtained. This set of image stacks is referred to as $\mathcal{I} = \{\mathbf{I}^1, \mathbf{I}^2, \ldots, \mathbf{I}^5\}$. Figure 2.4 displays two synaptic located proteins, `nmdr1` and `syphys`, in more detail. Synapses appear as small 3×3 to 5×5 pixel sized, glowing dots in both the SR and SP region of the images.

A set of eight images was employed for a detailed analysis of automated synapse detection and the evaluation of labeling strategies required for reliable detection. Therefore, eight sections of the CA3 region of a male C57BL/6 mouse (15 months) were imaged with TIS at the Molecular Pattern Recognition Research Group, Magdeburg University. Each sample comprised SP and SR regions of the CA3 area. 5μm horizontal sections were immunostained against synaptophysin in two different ways. Four serial sections were incubated with monoclonal anti-synaptophysin antibody directly conjugated to the fluorophore FITC, referred to as *directly* labeled sections. The remaining four sections were first incubated with mouse anti-human synaptophysin monoclonal antibody and then with a secondary FITC-conjugated polyclonal goat anti-mouse antibody, referred to as *indirectly* labeled sections. A detailed description of the labeling and imaging setup can be found in Herold et al. (2010). Each section was imaged with a ×40 water immersion objective obtaining a pixel size of 180×180nm/pixel (0.0324 μm^2) for each image and a size of 768×512 pixels with an intensity range of 8bit/pixel. Figure 2.5 shows one sample directly (a) and indirectly (b) labeled for synaptophysin with an enlarged view in (c) and (d), highlighted in (a) and (b) with a red

molecules/moiety recognized	abbreviation	name
Apolipoprotein E	`apoli`	Apolipoprotein E [*Homo sapiens*]
Carnitine acetylase	`cat`	Carnitine acetyltransferase [*Homo sapiens*]
ConA ligand	`cona`	Concanavalin A, α-man, α-glc
* GRIP-1,CT	`grip1ct`	Glutamate receptor interacting protein-1 [*Homo sapiens*]
* GluR1	`glur1`	Glutamate receptor, ionotropic, AMPA-1 [*Homo sapiens*]
* GluR2	`glur2`	Glutamate receptor, ionotropic, AMPA-2 [*Homo sapiens*]
* GluR2/3	`glur23`	Glutamate receptor, ionotropic, AMPA-2 (α2) [*Mus musculus*]
* GluR5	`glur5`	Glutamate receptor, ionotropic, kainate 1 [*Homo sapiens*]
* GAP43	`gap43`	Growth associated protein-43 [*Homo sapiens*]
IgG1-binding moiety	`igg1`	IgG isotype control / microglia associated Fcγ-receptor
Internexin-α	`inter`	Internexin-α, neuronal intermediate filament protein
* mGluR5	`mglur5`	Glutamate receptor, metabotropic-5
NEFHp	`nefhp`	Neurofilament triplet H protein
Neurofilament	`neurof`	Neurofilament light polypeptide
NeuN	`neun`	Neuronal nuclear antigen A60
* NMDA1	`nmdr1`	Glutamate receptor, ionotropic, *N*-methyl *D*-aspartate-1 [*Homo sapiens*]
* nNos	`nnos`	Nitric oxide synthase-1
* NR2A	`nr2a`	Glutamate receptor, ionotropic, *N*-methyl *D*-asparteate 2A [*Homo sapiens*]
* NR2B	`nr2b`	Glutamate receptor, ionotropic, *N*-methyl *D*-asparteate 2B [*Homo sapiens*]
* Synapsin I	`synap`	Brain protein 4.1
* Synaptophysin	`syphys`	Synaptophysin [*Homo sapiens*]
Ubiquitin activating enzyme-1	`ubiqui`	Ubiquitin activated enzyme E1
Propidium ligand	`prop`	Nucleic acids

Table 2.1: Summary of molecules labeled by TIS imaging for brain tissue analysis. For each molecule, an abbreviation is given which is used in the text. Proteins with synaptic localization are marked by an * (asterisk).

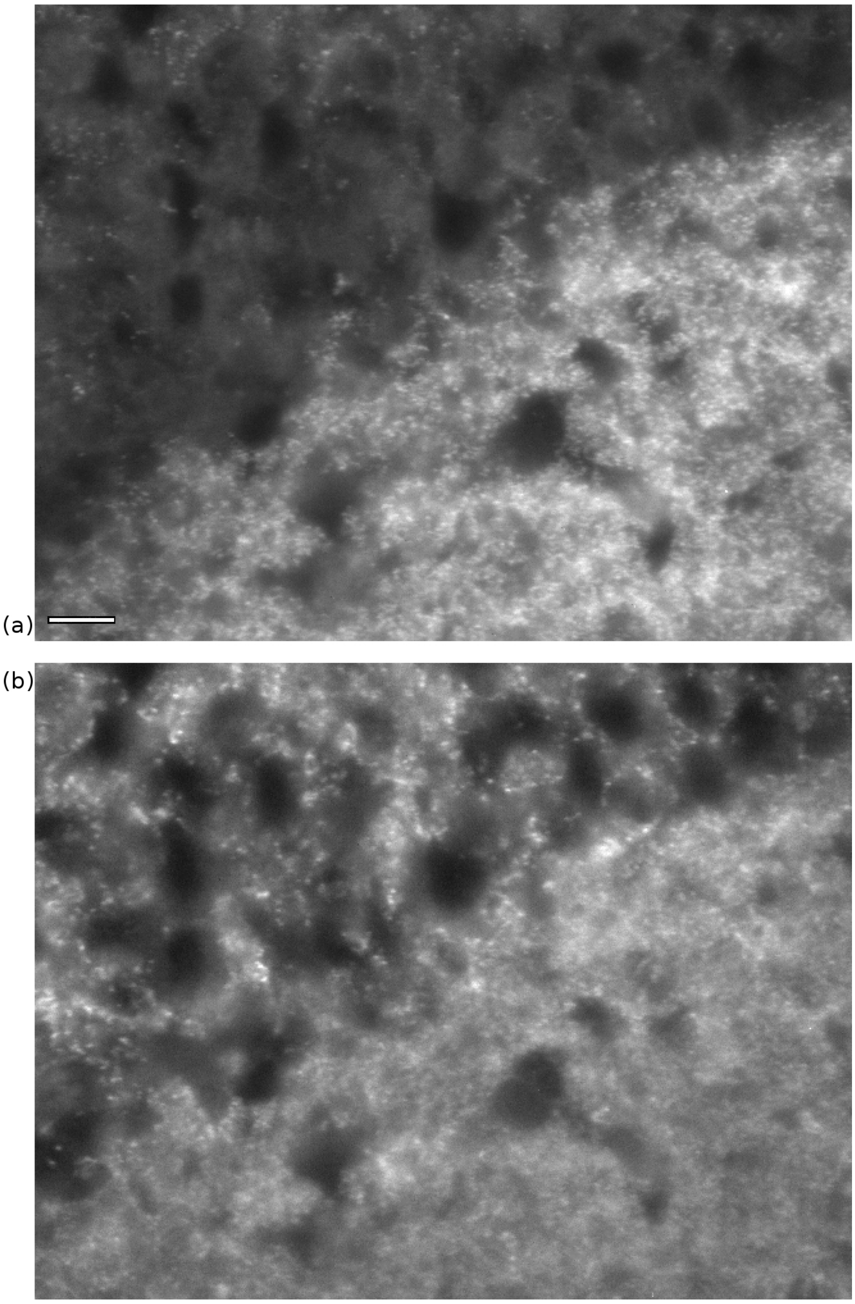


Figure 2.4: Images of a brain tissue sample labeled for (a) `nmdr1` and (b) `syphys`. Both are stable markers for synaptic regions. Images were rescaled to 8bit intensity values and manually enhanced slightly for display purposes. Scale bar of 10.8μm corresponds to both images.

rectangle, respectively. This set of eight images is referred to as $\mathcal{I}' = \{\mathbf{I}^{11}, \mathbf{I}^{12}, \ldots, \mathbf{I}^{18}\}$[1]. Each stack $\mathbf{I}^s \in \mathcal{I}'$ only consists of one image labeled for synaptophysin. Hence, for the sake of compactness, individual images are referred to as I^s, rather than I^s_n, omitting the extra identifier n to refer to one specific protein image in the stack.

For both image sets introduced, it has to be considered that one has to operate at the diffraction limited resolution of $\sim$200nm to image synapses. Synapses itself have a diameter of around 200nm to 300nm thus they can even be smaller than the optical resolution of the microscope. However, due to labeling with fluorescent dyes and the applied sampling frequency of the detector, synapses are visible as 3×3 to 5×5 pixel sized, glowing dots (Bolte and Cordelières (2006)). Furthermore, operating at the diffraction limited resolution with a conventional wide field microscope often results in low signal-to-noise ratios and low contrast but at the same time a great amount of small, diffuse objects of interest are to be identified. These image characteristics have to be considered when dealing with the synapse image data sets. Clearer images could be obtained by using confocal microscopy, however, here valuable information may get lost due to the exclusion of out-of-focus light (Bolte and Cordelières (2006)). Another strategy would be to apply a deconvolution to the images which is, however, quite complex and time consuming. Thus, no deconvolution was applied to the images and synapse detection and colocation analysis was carried out on the standard wide field microscopy images.

2.3 Summary

In this chapter, imaging approaches for protein colocation analysis were presented and discussed and the principles of the TIS imaging system have been presented. Two data sets, one 3-channel and one 22-channel data set, were introduced which will be used in the following chapters as a data basis to set up a strategy for high-content fluorescence image analysis. Additionally, a synapse data set with only one channel was introduced which will be applied to study synapse detection strategies in detail.

[1] Counting is started from 11 instead of 6 for easier interpretation.

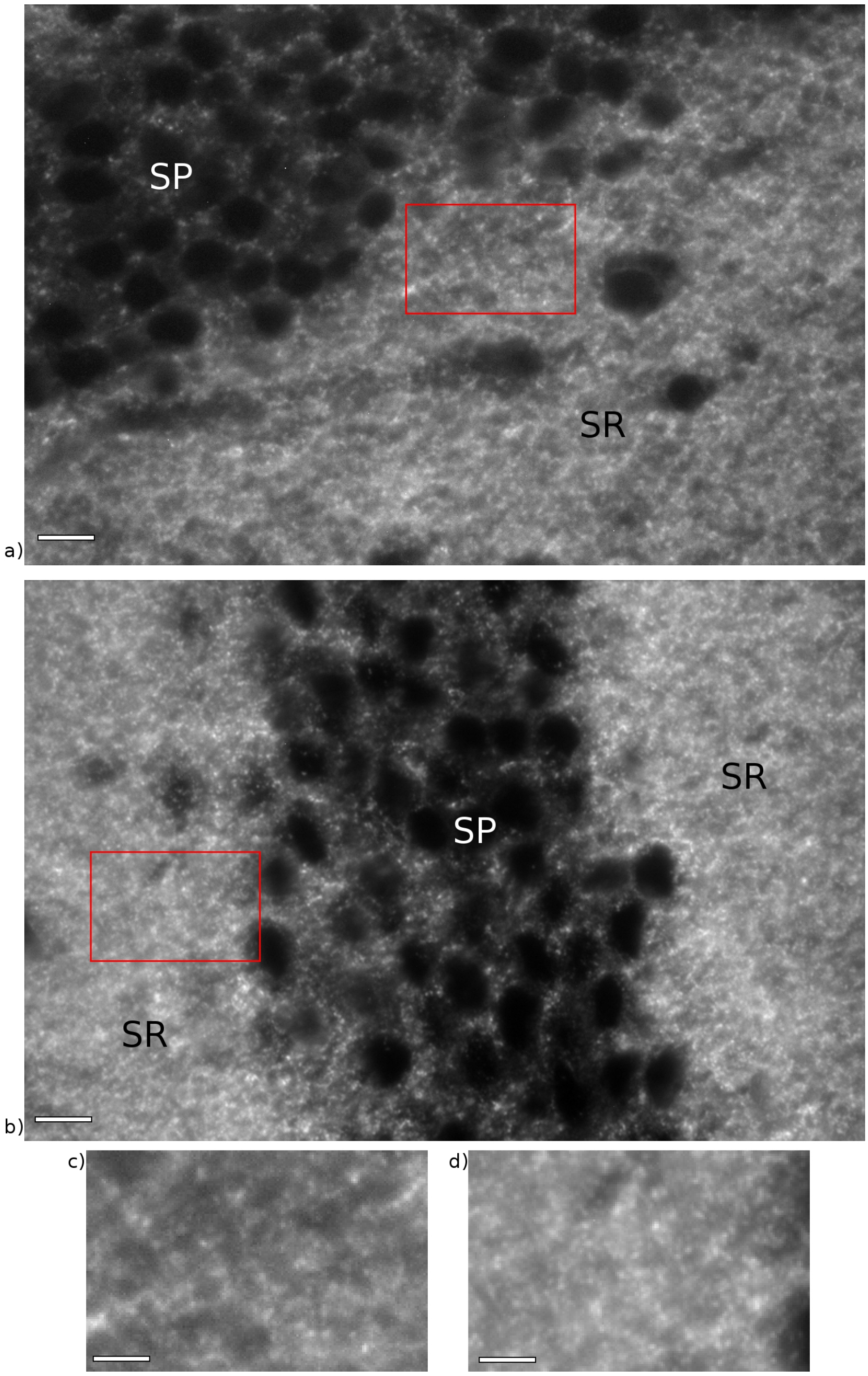


Figure 2.5: Images of two brain tissue sections labeled for synaptophysin. (a) and (c) display images directly labeled for synaptophysin, with (c) being an enlarged sub-image marked in (a) with a red box. (b) and (d) are obtained by indirect labeling of synaptophysin, with an enlarged sub-image in (d). The stratum radiatum and stratum pyramidale regions are marked by SP and SR. It is clear to see that synapses are densely packed in SR while arranged around neuronal parikarya in SP. Scale bars are (a),(b) 9μm, and (c)(d) 4.5μm.

Chapter 3

An Exploration System for Multivariate Fluorescence Tissue Images

While the gain in molecular information through TIS imaging can lead to a new understanding of functional molecular networks, the exploration of TIS data is a new challenge for computational biodata analysis.

For colocation studies of two labeled proteins, several methods have been proposed. The most basic approach consists of an overlay of both protein channels to obtain a colored fusion image which is visually inspected. In addition, statistical measures can be calculated to assess colocation such as the Pearson correlation (Pearson (1901)) or the Manders' coefficients (Manders et al. (1992)). Furthermore, object-based colocation analysis for two channels have been proposed. They rely on manual object identification and analysis of intensity values or apply basic image segmentation strategies as thresholding or edge detection to obtain object information. An introductory review has recently been given by Bolte and Cordelières (2006).

However, all these methods can only be applied for the analysis of two proteins, which is far below the complexity of TIS. It is evident that a pure visual exploration is not feasible for colocation analysis in d-dimensional high-content images. Through visual inspection of each single gray value image, colocation can hardly be identified. Iteratively superimposing k out of the d images of the stack or even all images to obtain RGB fusion images is also not feasible for protein network identification since an observer would need to analyze a number of $d!/(k!(d-k)!)$ visualizations[1]. Thus, methods are needed which process the image content in such a way that it is comprehensible by the human expert and hence facilitates sophisticated TIS data analysis.

In the following, an introduction to existing methods for TIS data exploration will be given and advantages and disadvantages will be discussed. Subsequently, demands on a strategy for high-content TIS data analysis are specified and the concept for an object-based multivariate tissue analysis system developed in this work is presented.

[1] For $d = 22$ and $k = 3$ already 1540 image combinations have to be regarded and evaluated.

3.1 Existing Methods for the Exploration of Multivariate TIS Data

Binary protein colocation analysis

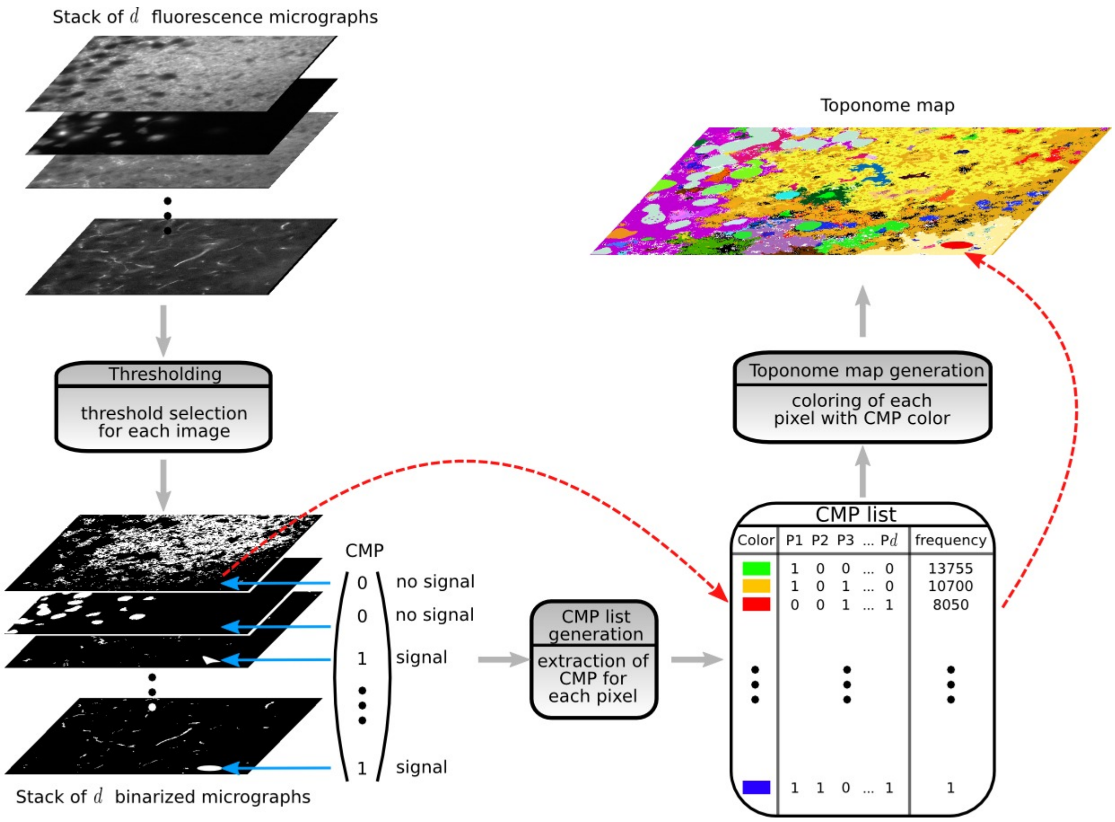
Schubert et al. (2006) has proposed a method for the evaluation of the obtained multivariate image data by means of analyzing binary protein colocation patterns, termed *combinatorial molecular phenotypes* (CMPs). To this end, thresholds $\theta_1^s, \theta_2^s, \ldots, \theta_d^s$ are manually selected for each image I_n^s of a multivariate image stack $\mathbf{I}^s = \{I_1^s, I_2^s, \ldots, I_d^s\}$ of a sample s. For each image I_n^s and each pixel location $\mathbf{p}$ the intensity $f(\mathbf{p})$ of pixel $\mathbf{p}$ is set to 0 if $f(\mathbf{p}) < \theta_n^s$, thus if protein n is absent, and otherwise to 1. Hence, each image is converted into a binary black and white image. For each pixel location a binary CMP can then be extracted as depicted in figure 3.1 (left side). Thus, each pixel can only be associated with one individual CMP but different pixels can feature the same CMP. The whole list of CMPs, usually sorted according to their frequencies, i.e. the number of pixels, can then be visually analyzed to find interesting protein combinations.

CMPs can be further grouped into a set of CMP motifs by combining CMPs which (i) contain one or more protein(s), called lead proteins, (ii) have one or more absent protein(s) and (iii) have variable occurrence of additional proteins (wild card proteins). These CMP motifs denote a functional region of a tissue or cell sample (Friedenberger et al. (2007)).

Thresholding the data and analyzing the obtained CMP with the proposed approach has many benefits: (i) The obtained binary protein patterns can easily be interpreted since proteins are only either absent or present. (ii) Patterns extracted from different images stacks can instantaneously be compared to find common or separating CMPs. (iii) The binarized protein patterns reflect the most prominent features of protein localization (Schubert et al. (2006)). Thus, analyzing binary images instead of gray value images is a reasonable strategy to gain insight in high-content images as it reduces the complexity of the data. However, binarization of images requires a high level of expertise and manual interaction for each image, which is quite time consuming. Slight modifications of the threshold can lead to different CMP lists, potentially affecting the interpretation of the data, and for some images, it might not even be possible to set one specific threshold.

To support the spatial analysis of CMPs and CMP motifs, Schubert has proposed a so called *toponome map*. To construct this map, a random color is assigned to each CMP (motif) of the list and each pixel is visualized in the color assigned to its CMP. Figure 3.1 (top right) exemplarily shows a toponome map. The red dotted arrow indicates the process of CMP extraction at one pixel location, color assignment and image coloring. Through the analysis of the toponome maps, spatial ordering of CMP (motifs) can be evaluated.

By applying this strategy of analyzing binary protein colocation information, Schubert's group was able to identify, for example, CMPs in images of skin allowing them to distinguish between healthy patients, patients with psoriasis and patients with atopic dermatitis (Schubert et al. (2006)). Furthermore, they discovered the lead proteins controlling the molecular networks in rhabdomyosarcoma tumor cell lines (Schubert et al. (2006)), and identified synaptic classes (CMPs) which were restricted to defined subregions of the CA3 hippocampus (Bode et al. (2008)). Other examples of the application of MELC/TIS for the


Stack of d fluorescence micrographs
Thresholding
threshold selection for each image
Stack of d binarized micrographs
CMP
0 no signal
0 no signal
1 signal
1 signal
CMP list generation
extraction of CMP for each pixel
Toponome map
Toponome map generation
coloring of each pixel with CMP color
CMP list

Color	P1	P2	P3	...	Pd	frequency
🟩	1	0	0	...	0	13755
🟧	1	0	1	...	0	10700
🟥	0	0	1	...	1	8050
🟦	1	1	0	...	1	1

Figure 3.1: Analysis strategy for TIS data based on binary protein patterns. Each image is manually thresholded to obtain binary images. For each pixel in the image, a binary protein pattern, termed combinatorial molecular phenotype (CMP), can be extracted. A list of CMPs is generated, holding the frequency and characteristic of each CMP. For spatial analysis of CMPs, a toponome map is constructed by randomly assigning a color to each CMP and coloring each pixel location according to its CMP color.

analysis of protein colocation can be found, for example, in Schubert et al. (2009), Schubert et al. (2008) or Somani A. K. (2008).

Interactive image coloring

Choosing random colors to generate a toponome map, as proposed by Schubert, follows the idea to treat the CMP as a nominal variable, thereby presuming that each CMP represents its own category and there exists no relation between CMPs. The main advantage is that borderlines between neighboring CMPs in the toponome map are clear to read, even if they share common features. The drawback, however, is that morphological structures can vanish behind a complex, colorful map overburdening the cognitive skills of a human observer. Furthermore, random coloring is not a wise choice if CMPs are not interpreted as nominal variables but it is assumed that a relationship between CMPs exists. Similar CMPs may be mapped to complementary colors and vice versa, making a pseudo color visualization difficult to interpret.

Serocka (2007) has therefore proposed a strategy where similar CMPs are mapped to similar colors, following the assumption that CMPs which differ only in few proteins are similar to each other. Similar to the work of Schubert, thresholds are set for each image channel. At the same time, a toponome map is generated by assigning a color to each binarized image which are then superimposed to generate a false color fusion image. However, for a truthful interpretation of mixtures of colors, the set of explored proteins needs to be restricted to a small number. It is evident that for a set of 100 proteins, it is not possible to identify those protein channels which contribute to the resulting mixed color.

Object based binary colocation analysis

An analysis strategy of multivariate TIS data based on object identification has been introduced by Nattkemper et al. (1999). In his work, protein combination patterns present at lymphocytes in human muscle tissue samples were analyzed. Each sample was immunostained against seven cell surface proteins. In each of the seven images in the stack, a cell detection was performed and the resulting list of cell positions were combined to a master cell position list. For each position in this master list a binary protein combination vector $\mathbf{x} = (x_1, x_2, \cdots, x_7)$ was extracted by setting $x_n = 1$ if a cell was detected in image I_n^s at the given position, i.e. protein n was present, otherwise $x_n = 0$. Thus, for each cell in the sample, a binary vector similar to the CMP vector was obtained, however corresponding to a whole cell rather then a single pixel location. Hence, an extensive reduction of data complexity can be obtained. However, it has to be considered that object detection is required for each image of the stack. This can be a difficult task if the proteins of interest label different cellular structures. In this case, for each cellular structure of interest an individual detection approach would need to be designed.

Nattkemper has furthermore suggested different visualization strategies for the analysis of the obtained colocation information such as a list view similar to the CMP list, a histogram plot of the pattern frequencies or star glyph visualizations combined with a sonification approach for each pattern. More details can be found for example in Nattkemper et al. (1999), Hermann et al. (2000), Nattkemper (2001), or Nattkemper et al. (2003b).

3.2 Demands on an Object-Based Analysis Strategy for Non-Binary Multivariate Images

The great challenge when analyzing TIS data is the high complexity of the image data. A large number of high-dimensional protein patterns which is not perceivable by visual inspection need to be analyzed. Hence, the existing TIS analysis methods are all aiming at reducing this data complexity by converting the non-binary data into binary protein colocation information. The approach of Nattkemper furthermore reduces the number of protein patterns which have to be regarded by applying an object detection approach. This reduction of complexity eases the interpretation of the data and the knowledge discovery process. However, besides being laborious, the process of thresholding features the great disadvantage that information

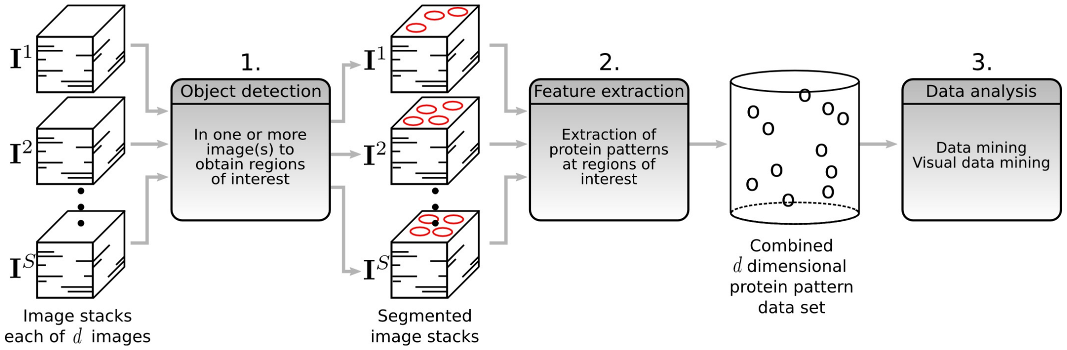


Figure 3.2: Analysis pipeline for multivariate fluorescence tissue images for protein colocation pattern studies. 1. Object detection is carried out in one or more images of the stacks $(\mathbf{I}^1, \ldots, \mathbf{I}^S)$ to obtain regions of interest. 2. Object specific features are extracted from each image stack. 3. The extracted image features are processed and analyzed by data mining and visual data mining approaches.

inherent in the data is discarded. Analyzing non-binarized images is much closer to the "reality" of protein abundances in the cell (Friedenberger et al. (2007)). Yet, the protein patterns are much more difficult to interpret and to compare than binary patterns. While in the binary case a pattern for d images can feature 2^d different protein combinations, for images recorded with 8 bit/pixel and integer precision already 255^d different combinations are possible. Thus, methods are required which allow for an efficient interpretation of non-binary data. Sophisticated visualization strategies have to be combined with concepts which reduce the data complexity without the requirement to set a threshold. This could be achieved, for example, by analyzing only selected protein channels. Yet, this would discard the special benefit of TIS imaging that colocation information for many proteins is obtained. In this work it is proposed to filter the amount of data on a morphological level by restricting the analysis to biological relevant structures, similar to the approach of Nattkemper. Furthermore, a data reduction is obtained by filtering at the combinatorial level, i.e. grouping protein patterns which are similar to each other.

In the following, I present the concept for a system for object-based automated analysis of multivariate tissue data obtained with the TIS technology which takes into account the whole intensity information. Its main focus lies on the analysis of synapse tissue micrographs. Furthermore, related methods which were considered for the design of the proposed strategy are presented. The system introduced consists of three major steps as depicted in figure 3.2. In the first step, object detection is carried out on one or few images of the image stacks to restrict the analysis to regions of interest. Hence, the amount of data which needs to be analyzed is largely reduced in this first step through filtering at the morphological level. Second, object-specific feature vectors holding protein colocation information are extracted. Third, methods from the field of data mining are applied to further reduce the data complexity by filtering on the combinatorial level. Furthermore, visualizations tailored to the needs of non-binary protein colocation analysis are provided to aid the user in getting an insight into this complex data domain.

3.2.1 Semantic Image Annotation

I have pointed out that reduction of the data complexity is one very reasonable strategy when analyzing TIS data. In this work a filtering of TIS data on a morphological level is proposed to obtain such a data reduction, i.e. restrict the analysis to regions of interest (ROIs). This strategy is very appropriate for tissue samples, as here ROIs often only constitute a small fraction of the whole image unlike in cell culture samples where all cells are of interest. In the case of tissue samples, relevant information would supposably be lost in the wealth of data when the whole image is analyzed. Furthermore, by restricting the analysis to biological relevant regions, extracted protein patterns can be directly linked to the corresponding biological object.

To obtain such a filtering on the morphological level, an object detection has to be carried out prior to the extraction of the protein patterns. I refer to this step as *semantic annotation* of the image, because it assigns meaning to the distinct object regions and thus also to the subsequently extracted features.

In the majority of cases, it is sufficient to perform such a semantic annotation only on one or few channels of the multivariate image stack. For example, in the case of the pancreas images, it is sufficient to detect cell nuclei in order to obtain object information. For synapse identification in the brain images, object detection in one or few images obtained by labeling proteins considered as stable synaptic markers, is enough (cf. section 2.2.2).

In the case of synapse data, the need for semantic annotation is evident. On average, around 2500 synapses are present in one image of a CA3 brain tissue sample (658×517 pixels). Assuming that all synapses have the maximum size of 5×5 pixels, only around 18% of all protein patterns of the whole image are of interest when analyzing synaptic protein colocation[2]. Restricting the analysis to synaptic pixels therefore results in a large amount of data reduction while no relevant information is discarded. It allows the biologist to concentrate on relevant data rather then interpreting irrelevant image regions. In the case of the pancreas data, a semantic image annotation is a prerequisite to analyze the multivariate image characteristics. To differentiate between different cell types, object specific features have to be extracted holding the information of which protein surrounds the nucleus. Extracting this information is only possible with a preceding nucleus detection step.

There are many examples of automated object detection in the field of microscopy (Nattkemper et al. (2001); Barber et al. (2001); Carpenter (2007); Gordon et al. (2007)). A review of automated object detection focusing on the analysis of microscopy images has been given by Nattkemper (2004). Traditional object detection approaches usually consist of highly tuned pipelines of simple algorithmic steps. Often, the core of such an approach is built on gray value thresholding followed by a tuned protocol of morphological operators (Sonka and Fitzpatrick (2000); Nedzved et al. (2000)). These operators are strongly related to morphological parameters and must be carefully adapted by experts in case of changes in the biological sample. Another common segmentation strategy is the use of watershed algorithms, which interpret the intensity of the image as a pattern of "mountains" and "val-

[2] In an image of size 658×517 pixels, 340,186 protein patterns would need to be regarded. This can be reduced to 62,5000 protein patterns if only patterns at ROIs are regarded (2500 ROIs of size 5×5).

leys" (Malpica et al. (1997); Wählby et al. (2004)). Starting at the lowest valley, the image is successively flooded. If two flooded valleys merge by reaching a common watershed, this watershed is interpreted as the object border. The crucial parameter is the minimum height of a watershed which is accepted as an object border. Setting a meaningful height can be very difficult in microscopy images, and often this strategy leads to over-segmentation of the image and a subsequent tuned post processing has to be performed (Wählby et al. (2002)). Other approaches rely on textural features to segment the image or are model-based, as for example the Hough transformation (Gonzalez and Woods (2002), p. 587), which require a precise description of the objects shape. In recent years, approaches are increasingly used which expanding a curve, referred to as *snake*, from a given starting point until they reach the boundary of an object (Hu et al. (2004); Solorzano et al. (2000)).

In various applications, these traditional object detection strategies were successfully applied for nucleus identification and segmentation (Wählby et al. (2004); Malpica et al. (1997); Beliën et al. (2002); Hu et al. (2004)). Synapse detection, however, is only rarely carried out by automated approaches. Two prominent examples are the works proposed by Micheva and Smith (2007) and Silver and Stryker (2000). Micheva and Smith (2007) have proposed an image processing pipeline for the detection of punctuate synaptic structures in ultrathin (200nm) sections, indirectly immunolabeled for synapsin. Here, the obtained fluorescence images were manually thresholded to remove background noise and obtain punctual segments matching those apparently visible. In a subsequent step, a watershed algorithm was applied in which the watershed level was adjusted to provide for separation of nearby objects, which were visually identified as separate. Objects of sizes below a user defined threshold were then discarded. With this strategy, synapse counts similar to stereological estimates could be obtained (Micheva and Smith (2007)) suggesting that synapses were correctly detected. Another synapse detection pipeline based on traditional image processing strategies has been introduced by Silver and Stryker (2000). Confocal images of serial sections, indirectly immunolabeled for synaptophysin, were processed in the following way. First, cell bodies and blood vessels were manually masked and an image threshold was computed based on the total number of pixels in the image and the masked image pixels. By manually placing seed points, which were used as starting points for an iterative dilation, subsequently under-segmented synaptic regions were separated. Out of focus synapses were discarded by analyzing the intensity values in adjacent serial sections, where optimal section spacing was estimated by calculating the point spread function.

The strategies proposed by Micheva and Smith (2007) as well as Silver and Stryker (2000) clearly show great potential of computerized synapse quantification in fluorescence micrographs. However, a high amount of human interaction is required in, for example, setting appropriate thresholds, watershed levels or seed points. Despite being very laborious and time consuming, this manual interaction have other disadvantages. Each time a new image is analyzed, new thresholds or seeds need to be set, thus the processing is influenced by the varying performance of the human observer(s), known as *intra-* and *inter-observer variability*. It is thus not guaranteed that the same result is obtained when one image is analyzed twice by one observer or by different observers. Furthermore, it might not even be possible to find, for example, one watershed level suitable for the whole image as the valleys' depth and the mountains' steepness can highly vary if the image shows low contrast, high and uneven background, or out of focus signals. In standard fluorescence microscopy, however, these phenomena are quite common, especially when synapses are directly labeled and imaging is carried out at the diffraction limited resolution, as it is required for synapse detection. Thus, for standard wide field fluorescence microscopy, the setting of appropriate pipeline parameters is even more challenging. Approaches relying on textural or shape features would presumably fail as well in synapse detection because synapses are mostly densely packed with no particular structure or substructure at light microscopic resolution, often lacking sharp edges that would separate them from the environment. Hence, traditional approaches are not well applicable for studies with considerable throughput.

To enable a higher throughput in image analysis, I propose a novel synapse detection approach which greatly reduces the amount of manual interaction and applied heuristics. The basic idea is a point-wise application of a supervised learning algorithm which has already been proposed and successfully applied for the detection of cells in micrographs (Wei et al. (2007); Sjöström et al. (1999); Nattkemper et al. (2002))) or similar bioimage informatics problems (El-Naqa et al. (2002); Theis et al. (2004); Chen et al. (2007); Long et al. (2008)). One advantage of this learning based method is it that it allows direct incorporation of human expert knowledge in the object detection task by providing hand selected training data. In general, this is the only human interaction required to set up the proposed system so that it is capable of detecting synapses in new images.

To prove the real world applicability of the proposed computational object detection system, a thorough evaluation that measures accuracy and stability needs to be performed. It has to be considered that (i) changes in the image quality is a common source of variation and (ii) the system is applied by different users. To simulate these facts, two studies are carried out. To assess the influence of (i), the average accuracy of the system is computed. This is done by using different example images (i.e. data subsets) for tuning of the detection system. The different parameterized versions are applied to the data and the average accuracy and standard deviation for each system is computed. This kind of an analysis is known as k-fold cross validation with k as the number of data subsets, i.e. images. To study the influence of (ii), each image is processed by all different parameterized versions (except the one derived from this particular image) and the average and standard deviation of detected objects are computed. Many works in bioimage segmentation do not carry out such a thorough evaluation, although it is especially important for those approaches which need a critical amount of human interaction. One possible reason for this shortcoming is the requirement of a *gold standard*, also referred to as *ground truth*, to access the accuracy of a system. The gold standard holds a template of the correct detection result. Especially in microscopy image analysis, obtaining such a gold standard is often non-trivial. In the case of synapse detection, it is even not possible. Nevertheless, in this work synapse detection performance is carefully evaluated regarding both aspects, accuracy and stability, by comparing against expert reference lists, coming close to a gold standard.

The object detection setup and the application to the synapse and pancreas data set will be described in detail in chapter 4. Although nucleus detection can be performed without this learning-based strategy (see above), it was applied to show the applicability of this method to different bioimage problems. In general, learning based strategies are applicable to a wide field ob object detection tasks, as has been shown by the citations above, making them well suited when the biological data and question at hand changes.

3.2.2 Extraction of Object-Specific Features

Once semantic image annotation has been carried out so that ROIs are identified, it is then possible to extract object-specific multivariate image feature vectors, i.e. protein patterns. In general, feature vectors can be extracted in multiple ways, always dependent on the image domain and biological question at hand. It might be sufficient to extract the intensity value at each pixel of the ROIs. Alternatively, the average intensity values of each ROI in each channel could be used as a feature, or even more complex feature vectors could be calculated. Extracting feature vectors tailored to the needs of the particular biological question is the prerequisite for a meaningful data interpretation. In section 5.2 and 6.1.2, I will present two strategies for feature extraction for the pancreas and brain data set, showing the diversity of how features can be obtained.

Although no feature extraction procedure can be proposed which will be applicable to all image domains and biological problems, there is one aspect which has to be kept in mind for all methods. The biologists are interested, for example, in the variance between diseased and control samples with respect to the observed protein patterns. Thus, protein patterns have to be comparable within and between images. However, often intensities of different images, between stacks or within one stack, are not directly comparable. Therefore, image normalization should be carried out prior to feature extraction.

It is always dependent on the image domain which normalization strategy is suitable. For some cases, it might even be necessary to apply different methods to the individual images of one image stack or perform a manual normalization to obtain reasonable normalization results. Common image normalization strategies are histogram equalization or linear rescaling. Histogram equalization maps the histogram to a uniform distribution, while linear rescaling transforms the histogram to cover the whole intensity range. Alternatively, a log transformation is often used to enhance low intensity values. Another strategy relying on statistical measures of the image histogram is the z-normalization which rescales the image histogram to zero mean and unit variance.

3.2.3 Data Interpretation

Probably the most difficult and diverse part of TIS data analysis is the data interpretation itself. Based on the high dimensional feature vectors, information has to be extracted which helps, for example, in understanding which protein patterns make the difference between healthy and diseased samples. Other possible exploration goals are to link a protein network to a cellular function, or analyze the spatial distribution of individual protein patterns. As in this work non-binary patterns are considered, data analysis, especially the comparison of patterns extracted from different image stacks, becomes much more difficult. Finding the valuable information inherent in the data is often like searching for a needle in a haystack.

From an image analysis point of view, the non-binary TIS data analysis problem is related to the field of multispectral imaging in geology and astronomy, where images originate from satellite- or air-borne remote sensing. For each spatial location (pixel), the spectral signature of a material is recorded over multiple wavelength bands, resulting in high dimensional feature vectors. Several data mining techniques have successfully been applied to link spectral feature vectors to terrain information, as soil, vegetation etc., for example in Villmann et al. (2003), Bandyopadhyay et al. (2007) or Melgani and Bruzzone (2004). Similar strategies have been applied to the relatively new field of multispectral imaging in biology, were biological samples are imaged for multiple spectral bands. In Harris (2006), an analysis strategy developed for the earth sciences was applied to microscopy image data. Boucheron et al. apply support vector machines and other learning-based approaches to classify hematoxylin and eosin stained images (Boucheron et al. (2007)). Furthermore, a genetic algorithm was used for pixel-wise classification of benign and malignant cells in Angeletti et al. (2005).

Although data mining strategies, especially from the field of supervised learning, have successfully been introduced in the field of multispectral image analysis, applying solely these concepts to TIS data would not lead to valuable information. It can be compared very well with the genomic sequence analysis problem. Many approaches could be adapted from the field of text mining. However, applying only the available approaches was not sufficient but problem specific methods needed to be provided to get an in detail understanding of the data. The same applies to TIS data analysis with respect to multispectral imaging. The main difference between multispectral and TIS imaging is the meaning of each recorded image. In multispectral imaging, the recorded bands have no specific biological meaning besides its wavelength. Not the characteristic of the spectral feature vector itself, but the class it encodes is of interest. The spectral features only aid in better discriminating between individual classes which are often known beforehand. In addition, adjacent channels in multispectral imaging are usually highly correlated and the intrinsic dimension of the data is comparably low. In TIS imaging, however, each image is associated with a specific protein and the characteristics of the protein vectors are the essential information which, in later steps, will aid in understanding what biological class, i.e. network, they encode. In most cases, no prior knowledge is available about the intrinsic dimensionality of the manifold in d-dimensional space which is described by the TIS data. This is, last but not least, due to the limited knowledge of protein networks and the fact that TIS is a new technology. A detailed spatial analysis of protein colocation has not been possible before, so that only very few assumptions can be made of protein patterns and topological ordering that can be expected.

Interactive visual data mining is considered as a powerful tool when analyzing data where the goal of the exploration is vague and little a priori knowledge is available (Keim (2002)). Visual data mining combines concepts from traditional data mining with information visualization techniques and includes the human expert directly in the data exploration step. Thus, the human expert's general knowledge of the data domain is incorporated in the knowledge discovery process.

In this work, the application of unsupervised learning, more precisely clustering, is proposed for the data mining part since it is well applicable when little a priori knowledge is available

(Nattkemper (2004)). In these cases, supervised learning strategies as proposed in most multispectral image analysis concepts, are not feasible. Clustering groups L data items into K clusters ($K \ll L$) so that the similarity of protein patterns within one cluster is high and the similarity between patterns belonging to different clusters is low. For TIS data analysis, clustering has many benefits. First, it allows for a purely data driven analysis so that the obtained results are based solely on the underlying image information. Second, protein patterns are grouped and frequency information of patterns with similar protein patterns can be obtained, similar to the frequency of CMPs. This information is relevant for the comparison of different samples, which might show similar protein patterns but with different frequencies. Third, vector quantization can be applied, a strategy directly linked to clustering. In vector quantization, each of the K clusters is approximated by one of K *prototypes*, also referred to as *reference* or *codebook vectors*. Representing the data set by K prototypes allows for a further reduction of data complexity on the combinatorial level. It is reasonable to analyze the protein patterns of the prototype vectors in a first overview step instead of the patterns of each individual data item.

For the information visualization part, it has to be considered that two data domains have to be visualized and explored simultaneously. Unlike in other visual data mining problems where solely the content of a data base is analyzed, the feature vectors have an origin in the spatial domain. This, as mentioned before, is the special benefit of imaging in systems biology and has to be considered when designing a strategy for TIS data analysis. Furthermore, the biological experts are very familiar in analyzing image data. Hence, providing sophisticated visualizations in this domain is very important. Besides providing suitable visualizations for the image domain, also visualizations in the feature domain are required which allow for a rapid identification of interesting protein patterns. Due to the complexity of the data, it is reasonable to follow the information mantra of Ben Shneiderman "Overview first, zoom in and filter, details on demand" (Shneiderman (1996)) when designing visualizations for both domains. Hence, in this work several different visualizations are generated which allow first to roughly explore the data domain and then focus on interesting findings in more detail. Visualizations of the image domain, for example, can well be used to obtain an overview of the data and to allow for a navigation in the spatial domain (Nattkemper (2004)). Interesting findings, for example spatial clusters of objects with similar protein patterns, can then be analyzed in a more detailed view of the feature domain. One such possible visualization of the image domain has already been proposed by Schubert with the toponome map (see section 3.1). The concept of assigning a color to each CMP is well suited to link the feature as well as the image domain. However, I have already discussed that assigning random colors to each CMP has advantages and disadvantages. Random colors are perfectly suited if it is assumed that two CMPs which only differ in one protein contribute to two totally different functions and are not related in any way. However, if a relation between protein networks is assumed, a color encoding reflecting this relationship would be more appropriate. Hence, several different color encodings should be provided. To allow for a detailed analysis of protein patterns, several strategies proposed by the field of information visualization for the display of multivariate data are applied in this work (Ware (2004); Spence (2007); Wong and Bergeron (1997)). Additionally, a novel visualization strategy tailored to the needs of protein colocation study is proposed which allows for a rapid identification of valuable information. To explore multiple variables in different domains, the field of information visualization has proposed several techniques. One of the most powerful approaches is the link and brush principle. The user can interactively select a sub set of the data in one display which automatically highlights the selected items in the other visualizations. By providing linked visualizations in the image and in the feature domain, the user can be aided in forming a mental model of the data and in identifying hidden regularities.

One example that shows the potential of visual data mining in microscopy analysis has been recently proposed by Zamir et al. (Zamir et al. (2008)). Here, cell-matrix adhesion in cell cultures of rat embryo fibroblasts is studied by analyzing the location of eight proteins involved in adhesion. Because of the spectral limitation at hand, iteratively combinations of five out of the four proteins are labeled and imaged. A pixel-wise clustering is performed for each protein combination set and clustered of different combinations are manually compared and matched. A random color code is assigned to each cluster for the purpose of visualization and rendering of images similar to the toponome map.

In chapter 5 and 6 I will give an overview of how the pancreas as well as synapse data can be interactively and efficiently analyzed by applying concepts from the field of visual data mining. Visualization strategies, tailored to the needs of colocation analysis will be presented. In chapter 7, I will introduce a strategy to statistically evaluate the topological distribution of protein patterns.

3.3 Summary

In the last sections, several requirements posed on a system for the analysis of non binarized multivariate fluorescence tissue images were put forth. It is evident that the analysis of such data is not restricted to one scientific domain. Aspects of image analysis are covered, such as object detection and image enhancement. Additionally, concepts of the field of data mining need to be applied, as clustering and statistical analysis of the data. Furthermore, visualization approaches are required for the interactive, visual interpretation of the data. In each of the fields, one can choose from a wealth of available methods which are more or less suited for the evaluation of TIS data, or novel approaches have to be proposed. However, TIS data analysis is a very new field and little is known about the pitfalls and challenges which will be encountered.

In this chapter, I have proposed a first approach for the analysis of non-binary protein colocation information extracted form multivariate TIS fluorescence tissue micrographs. It combines supervised-learning based object detection with concepts from the field of data mining and information visualization to allow for a reduction of data complexity and for an efficient visual exploration of TIS image data.

Chapter 4

Supervised Learning-Based Object Detection in Tissue Micrographs

Section 3.2.1 has pointed out the suitability of object detection when analyzing multivariate fluorescence tissue micrographs. It offers the benefit to greatly reduce the amount of data which has to be analyzed and allows to link extracted multivariate information directly to biological objects.

In this chapter, an automated object detection system based on supervised learning and tailored to the need of synapse detection will be presented. To fathom the requirements such an automated detection system has to meet, the human performance in detecting synapses is considered first.

4.1 Accuracy Assessment

To analyze the detection performance of human experts, as well as of an automated system, the accuracy of a detection result has to be measured. Several accuracy indices are mentioned in the literature as for example *sensitivity, specificity, positive/negative predictive value, precision* or *accuracy*, which cover different aspects of accuracy (Fawcett (2004); Bankman (2000)). Each of these indices requires the presence of a correct detection outcome, referred to as *ground truth* or *gold standard*. Given a two class problem with a positive and a negative class and a test set which is classified either as a positive or negative item by a human expert or an automated system, the evaluation of the obtained classification result against a gold standard can produce four possible outcomes (see also table 4.1): (i) *true positives* (TP), defined as a positive item classified as positive; (ii) *true negatives* (TN), defined as a negative item classified as negative; (iii) *false positive* (FP), a negative item classified as positive; and (iv) *false negative* (FN), a positive item classified as negative. Based on these four classification outcomes, the different accuracy indices can be calculated, for example the *sensitivity* (SE) is defined as $\mathrm{SE} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$. It is a common way to plot complementary indices as so called *Receiver-operating characteristic* (ROC) curves. ROC analysis has first been introduced for signal processing (Egan (1975)) and now features a wide application especially in the field of medical image analysis and machine learning (Metz (2008); Bankman (2000); Fawcett (2004)).

In the case of object detection in fluorescence images of tissue samples, obtaining a gold standard would require the generation of a complete set of object position coordinates in an image, representing a template of the correct detection result. In most cases in bioimaging,

		True Class	
		Positive	Negative
Detected	Positive	True Positives	False Positives
Class	Negative	False Negatives	True Negatives

Table 4.1: Possible evaluation outcomes for a two class problem when evaluated against a gold standard.

there is no alternative to manual labeling of the raw fluorescence micrographs by a human expert. Theoretically, the variability within and between experts should be almost zero. However, it is a well known fact that such a human evaluation is influenced by many factors resulting in variable performances between and within experts (Nattkemper et al. (2003a); Jagoe et al. (1991); Bankman (2000); Sjöström et al. (1999)). It is known, for example, that the degree of spatial attention needed for such a labeling task declines rapidly, which decreases the detection performance (Jagoe et al. (1991)). Furthermore, the level of expertise influences the labeling outcome, as has been analyzed in several studies of medical diagnostics (cf. Krupinski (2000)). In general, observer variation has been studied more frequently in medical imaging than in bioimaging. Still, many aspects also hold for bioimage data. A recent review of reasons for observer performance variation in radiology is given in Krupinski (2000).

Due to the within and between observer variability, it is reasonable to obtain labeling results of more than one human expert. This allows (i) to combine labeling results of different experts to one master label list, more trustworthy than the individual experts and (ii) to estimate the variability of the labeling results of different human experts.

As a consequence of variable human expert performance, the term gold standard or ground truth is not used throughout this work but the set of object positions obtained for one image I_n^s by manual human expert labeling is referred to as *expert reference* (ER_n^s). This change of terminology will reflect the fact that the expert labeling can not be interpreted as a ground truth but only as a reference set which still can contain falsely labeled objects.

The detection accuracy of an automated system, or of a human expert, can thus be assessed by using two complementary measures analogous to the sensitivity (SE) and positive predictive values (PPV) applied in ROC analysis. Let $\mathcal{S} = \{\mathcal{S}_{+,+}, \mathcal{S}_{-,+}, \mathcal{S}_{+,-}\}$ be the set of synapse positions either contained in the expert reference list and/or labeled by the system. If Λ_n^s is defined as the set of object positions detected by the system in image I_n^s, the sub sets $\mathcal{S}_{+,+}, \mathcal{S}_{-,+}$, and $\mathcal{S}_{+,-}$ can be defined as follows: $\mathcal{S}_{+,+}$ contains all object positions listed in ER_n^s as well as Λ_n^s, i.e. the expert and the system agree. The sub set $\mathcal{S}_{-,+}$ subsumes those synapses which are listed in ER_n^s but are missed by the system. In contrast, $\mathcal{S}_{+,-}$ comprises all synapses listed in Λ_n^s but not in ER_n^s. Table 4.2 shows a data confusion matrix for the adapted nomenclature similar to the one displayed in table 4.1. A corresponding definition to the TN class is not given as all pixels of the image not being a center of an object are true negatives. The number of negative items is thus considerably larger than the number of positive items, making all accuracy measures adapted from TN counts, such as for example the specificity, unfeasible.

		Expert Reference	
		Positive	Negative
Detected	Positive	$\mathcal{S}_{+,+}$	$\mathcal{S}_{+,-}$
Class	Negative	$\mathcal{S}_{-,+}$	na

Table 4.2: Nomenclature of possible outcomes for an object detection result evaluated against an expert reference list.

The SE and PPV measures corresponding to one ER_n^s and one Λ_n^s can then be computed as:

$$\mathsf{SE} = \frac{|\mathcal{S}_{+,+}|}{|\mathcal{S}_{+,+}| + |\mathcal{S}_{-,+}|}, \mathsf{PPV} = \frac{|\mathcal{S}_{+,+}|}{|\mathcal{S}_{+,+}| + |\mathcal{S}_{+,-}|} \,. \tag{4.1}$$

Thus, SE measures the percentage of those objects listed in ER_n^s which are also listed in Λ_n^s. PPV reflects the percentage of objects listed in Λ_n^s also listed in ER_n^s. In the case of synapses, a distance of less than two pixels is allowed for two locations in ER_n^s and Λ_n^s to be considered the same throughout this thesis. For cell nuclei in pancreas images, this distance is increased to a distance of five pixels, as the objects are much larger.

A measure combining SE and PPV into one accuracy index is the *f-measure* defined as

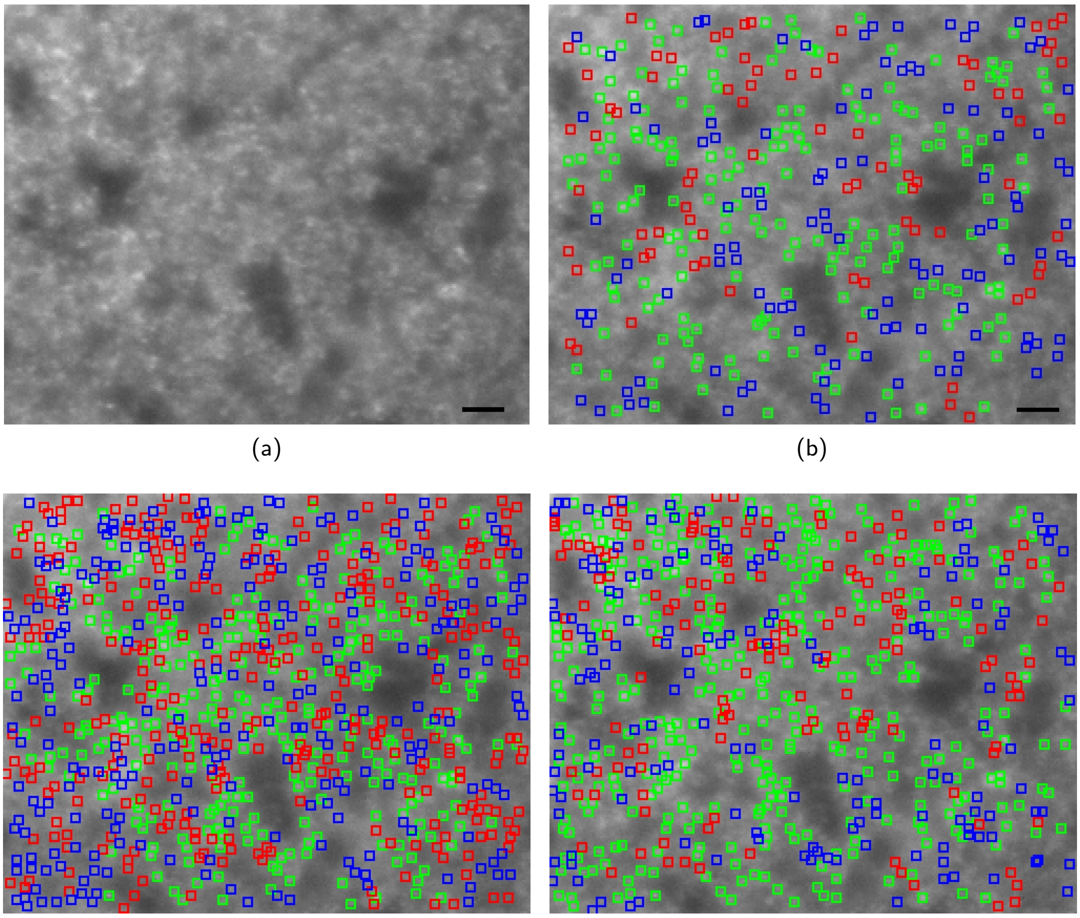
$$FM = \frac{2 * \mathsf{PPV} * \mathsf{SE}}{\mathsf{PPV} + \mathsf{SE}} \,, \tag{4.2}$$

which is the harmonic mean of SE and PPV.

4.2 Human Synapse Detection Performance

To specify the requirements posed on a system for automated synapse detection, the performance of human experts in labeling synapses is evaluated. To this end, the following evaluation strategy was performed.

Three experts (two on expert-, one on novice level) were asked to label a 250×200 pixel sized sub-image in an image stained for syphys ($\hat{I}_n^s$), and in an image stained for nmdr1 ($\hat{I}_m^s$), both belonging to the same stack $\mathbf{I}^s \in \mathcal{I}$. With $\hat{I}$ it is referred to a sub-image. To obtain an impression if it is possible to differentiate between different synapse qualities, the experts were additionally asked to assign one of three quality or certainty gradings to each synapse position based on their personal judgment. Quality A represents a position in the image which certainly represents a synapse, quality B stands for image locations where there are quite sure synapses, and quality C refers to an uncertain synapse position. Thus, for each sub-image ($\hat{I}_n^s$ and $\hat{I}_m^s$) and each expert, three expert references were obtained. For example, for one expert and sub-image $\hat{I}_n^s$, expert references $\mathsf{ER}_n^s(A)$ containing all quality A synapse positions, $\mathsf{ER}_n^s(B)$ subsuming all quality B synapses and $\mathsf{ER}_n^s(C)$ being quality C synapses, are extracted. By simple set theory, any combination of these three sets can be generated, for example the union of all three sets produces the reference set $\mathsf{ER}_n^s(ABC)$.



(a)

(b)

(c)

(d)

Figure 4.1: Human expert labels on sub-image $\hat{I}_n^s$. Three quality levels are displayed through a color code. Green: certainly a synapse, blue: quite sure a synapse, red: might be a synapse. (a) Original sub-image of size 250×200 pixels. Rescaled to 8bit intensity range for display purpose. (b) labeling result of expert 1 (novice level), (c) labeling result of expert 2 (expert level), (d) labeling result of expert 3 (expert level). Scale bar of 4.3μm refers to all images.

Figure 4.1 displays the original sub-image $\hat{I}_n^s$ in (a) as well as the labeling results of the three experts in (b-d) as an overlay to the image. Synapse positions of quality A are depicted in green, quality B in blue and quality C in red. Solely by visually evaluating the labeling results, it becomes evident that the human experts show a considerable amount of variation in the number of labeled synapses (expert 1: 376, expert 2: 841, expert 3: 652) as well as in their assignment of qualities. It is interesting to notice that the most cautious labeling result with 376 labeled synapses was obtained by expert 1 being a novice in synapse labeling (see figure 4.1 (b)). The two other experts, both more experienced in synapse labeling, obtained much higher counts (see figure 4.1 (c) and (d)). Similar results were obtained for the image

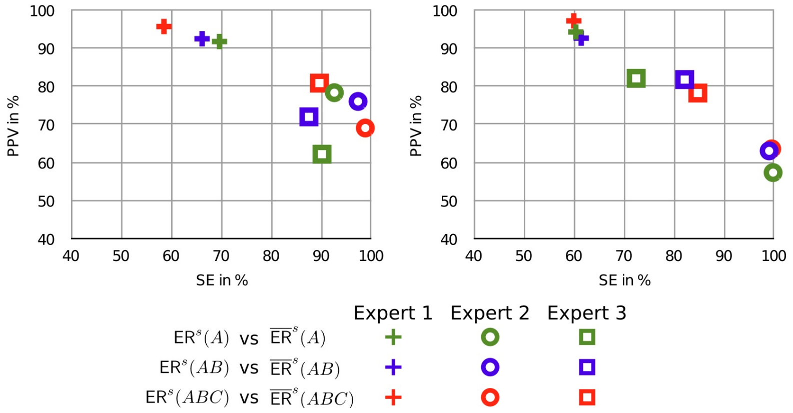


Figure 4.2: Individual human detection performance (SE and PPV) for image I^s_n (left) and I^s_m (right) evaluated against $\overline{\text{ER}}^s(A)$, $\overline{\text{ER}}^s(AB)$, and $\overline{\text{ER}}^s(ABC)$.

$\hat{I}^s_m$ stained for `nmdr1`. Here, expert 1 in total labeled 522, whereas expert 2 labeled 1323 and expert 3 labeled 916 synapses. Labeling of one of the 250×200 pixel sized micrographs took up to 3 hours for one human expert.

To obtain a statistical measure for the human observer performance, SE and PPV indices were calculated. Therefore, a reference had to be provided against which the expert labeling could be evaluated. As there exists no ground truth, as has be discussed in section 4.1, the three experts were combined to artificial master expert references $\overline{\text{ER}}^s_n(X)$ (with X being a placeholder for the quality encoder). A synapse position was accepted in the master reference $\overline{\text{ER}}^s_n(X)$ if it was labeled by at least two experts. Two label positions $\mathbf{p}$ and $\mathbf{p}'$ of the expert references of two different experts were considered the same if their Euclidean distance $d(\mathbf{p}, \mathbf{p}') < 2$. By combining the lists of all three experts of ER$^s_n(A)$, ER$^s_n(B)$, ER$^s_n(C)$, as well as ER$^s_n(AB)$ and ER$^s_n(ABC)$, five master reference list $\overline{\text{ER}}^s_n(A)$ (244 synapses), $\overline{\text{ER}}^s_n(B)$ (173 synapses), $\overline{\text{ER}}^s_n(C)$(172 synapses), $\overline{\text{ER}}^s_n(AB)$ (417 synapses), and $\overline{\text{ER}}^s_n(ABC)$ (589 synapses) were obtained, respectively. Similarly, the master reference lists for image I^s_m were computed with 530 synapses in $\overline{\text{ER}}^s_m(A)$, 181 synapses in $\overline{\text{ER}}^s_m(B)$, 136 synapses in $\overline{\text{ER}}^s_m(C)$, 711 synapses in $\overline{\text{ER}}^s_m(AB)$ and 847 synapses in $\overline{\text{ER}}^s_m(ABC)$. Subsequently, the individual expert references ER$^s(X)$ were evaluated against the corresponding master reference list $\overline{\text{ER}}^s(X)$ to obtain SE and PPV measures.

Figure 4.2 shows the individual performances of each expert evaluated against $\overline{\text{ER}}^s(A)$, $\overline{\text{ER}}^s(AB)$ and $\overline{\text{ER}}^s(ABC)$ for image I^s_n (left) and I^s_m (right). It is clear to see that expert 1, depicted with crosses, achieves a detection performance with high PPV but low SE for both images, whereas expert 2 and 3 achieve lower PPV but higher SE values in both images for all quality levels. Expert 1 thus shows a very cautious labeling with very few synapses. However, in most of the cases expert 1 labeled synapses which were also labeled by one of the other experts, and was thus contained in the master expert list.

Table 4.3 displays the mean and standard deviation for SE and PPV averaged over all three experts, evaluated against each of the master reference lists for each image. It can be inferred by examining table 4.3 that for each of the three quality levels (A, B, C), there is a core set

		I_n^s		I_m^s	
		SE$_\mu$	PPV$_\mu$	SE$_\mu$	PPV$_\mu$
master expert reference	$\overline{\text{ER}}^s(A)$	84±10.33	77±12.11	77±16.50	78±15.20
	$\overline{\text{ER}}^s(B)$	70±13.63	36± 6.16	67±24.68	15± 6.25
	$\overline{\text{ER}}^s(C)$	67±23.25	25± 5.19	68±17.30	17± 9.40
	$\overline{\text{ER}}^s(AB)$	83±12.95	80± 8.63	81±15.36	79±12.22
	$\overline{\text{ER}}^s(ABC)$	82±17.14	81±10.72	81±16.47	79±13.42

Table 4.3: Mean and standard deviation (in percent) for SE and PPV averaged over all three human experts. Evaluation was performed against four master reference lists, each subsuming synapse locations of different quality levels.

of positions all three experts agree upon. However, with decreasing quality, the agreement becomes much weaker. When combining different quality levels, performance increases. For example, for high and medium quality synapses ($\overline{\text{ER}}^s(AB)$), the average performance is SE$_\mu$ of 83% and PPV$_\mu$ of 80% for image I_n^s and SE$_\mu$ = 81% and PPV$_\mu$ = 79% for image I_m^s. Thus, there is higher agreement in what is a synapse than in the assignment of a quality label to a synapse. The overall detection performance of an average human expert, irrespective of the quality assignments, achieves a mean SE$_\mu$ of 82% and a mean PPV$_\mu$ of 81% for image I_n^s. Slightly lower performance measures are obtained for image I_m^s with an average SE$_\mu$ of 81% and a PPV$_\mu$ of 79%.

4.2.1 Discussion

In general, it can be stated that synapse detection performance depends upon the expertise of the observer and can vary considerably, as has been shown in figure 4.2. The untrained expert was very cautious but precise in the labeling, whereas more trained experts labeled a higher number of synapses. This might be due to the particular challenging task of synapse detection which requires some training. Densely distributed, small objects with no particular structure or substructure need to be identified in a gray value image with "lumpy" background. Hence, synapse detection is not comparable to other bioimage detection tasks where objects, for example cells, show a clear structure. A theoretical study about the efficiency of human observers in detecting small signals in lumpy backgrounds has been performed in Park et al. (2007, 2009), showing that this object detection task is very challenging. The performance of an expert is furthermore influenced by the amount of time the labeling takes. The degree of spacial attention decreases rapidly (Nattkemper et al. (2003a); Jagoe et al. (1991)) but is not perceived by the expert. This phenomenon might be a cause for the lower performances in image I_m^s, as around 30% more synapses had to be labeled. Thus, a complete labeling of an image containing more than 1000 densely packed synapses would not be possible as the number of mistakes would increase dramatically over time.

With regards to the human detection performance, it could be shown that there is consensus between the expert of what is a synapse and what is not a synapse. Overall performances of $\mathrm{SE}_\mu \geq 80\%$ and $\mathrm{PPV}_\mu \geq 79\%$ could be obtained if quality assignments were neglected. When it comes to the assignment of distinct quality labels, however, there is a much weaker agreement between the experts. While for synapses of high quality good performances could still be achieved, assignment of lower quality labels was not stable across the experts.

When using expert labels as a training basis for a supervised learning architecture as well as for the evaluation of the system's performance, the findings made in this section should be kept in mind. The obtained expert references will always feature a mixture of different synapse qualities where lower quality synapses are labeled while some synapses of higher quality are overlooked.

4.3 Requirements Posed on a Supervised Learning-Based Object Detection Strategy

It could be shown by evaluating the human synapse detection performance that there exists a reasonable agreement between the experts when labeling the position of a synapse. However, no stable quality assignment could be obtained. It is therefore adequate to formulate the problem of synapse detection as a classification problem rather than as a regression problem. There exist several methods for supervised learning for classification such as random forests (Breiman (2001)) or multi-layer perceptrons (Bishop (2004)). However, they mostly feature a difficult parameter optimization problem. To provide for a system which can easily be used also by non-computer experts, only a very limited number of parameters should need to be tuned by the user or an automated tuning of the systems parameter should be possible.

Therefore, in this work *support vector machines* (SVMs) (Vapnik (1996)), are applied for the task of automated object detection. SVMs are widely used because of their good generalization performances, i.e. their ability to correctly classify previously unseen data and the absence of local minima in their optimization problem (Cristianini and Shawe-Taylor (2003)). Furthermore, only a low number of parameters need to be tuned for which sophisticated automated approaches exist. They have found a wide variety of applications in the field of bio(medical) image analysis achieving good performances, as for example in cell detection (Twellmann et al. (2001); Wei et al. (2007); Long et al. (2008)), detection of microcalcifications (El-Naqa et al. (2002)) or the classification of localization patterns (Chen et al. (2007)).

In the following, the basic concepts of SVMs will be addressed. Afterwards, the *intelligent synapse screening system* (i3S) will be introduced which is a SVM-based object detection system optimized to the needs of automated synapse detection in fluorescently labeled brain tissue micrographs.

4.4 SVM Theory

In this section, the theory of Support Vector Machines will be introduced. The following explanations are not intended to be a full review on SVMs or machine learning but shall provide the basic concepts required to understand the subsequent sections of SVM-based

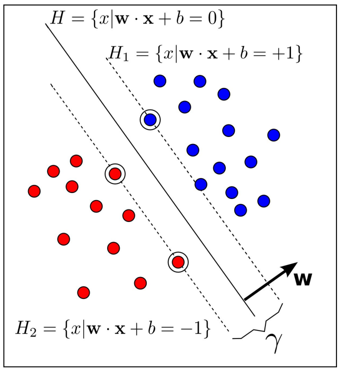

$H = \{x|\mathbf{w} \cdot \mathbf{x} + b = 0\}$
$H_1 = \{x|\mathbf{w} \cdot \mathbf{x} + b = +1\}$
$H_2 = \{x|\mathbf{w} \cdot \mathbf{x} + b = -1\}$
$\mathbf{w}$
γ

Figure 4.3: The optimal separating hyperplane H is the hyperplane with largest margin γ which linearly separates negative (red) and positive (blue) items. The hyperplane is defined by the weight vector $\mathbf{w}$ and the threshold b. Points lying on the parallel hyperplanes H_1 and H_2, which correspond to the largest margin γ, are termed support vectors and are highlighted by an extra circle. Adapted from Schölkopf (2002).

object detection. The interested reader can find more details in, for example, Burges (1998), Cristianini and Shawe-Taylor (2003) or Schölkopf (2002).

4.4.1 Linear Support Vector Machines

Given a set of training data items consisting of N input-output pairs $\Gamma = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1,\ldots,N}$, with $\mathbf{x}^{(i)} = (x_1^{(i)}, x_2^{(i)}, \ldots, x_q^{(i)})^T$ being a q-dimensional training data item and $y^{(i)} \in \{-1, +1\}$ being the class label, representing a two class problem. In a geometric interpretation of the SVM algorithm, a hyperplane H is determined, which separates the training data in order to perform binary classification as depicted in figure 4.3.

An arbitrary hyperplane in the q-dimensional space can be described through $H = \{\mathbf{x}|\mathbf{w} \cdot \mathbf{x} + b = 0\}$, where $\mathbf{w}$ is a vector normal to the hyperplane, $\parallel \mathbf{w} \parallel$ is the Euclidean norm of $\mathbf{w}$, and $\frac{|b|}{\|\mathbf{w}\|}$ defines the perpendicular distance from the origin to the hyperplane. The margin γ of a hyperplane can now be defined as the sum of (i) the shortest distance from the hyperplane to the closest positive item and (ii) the shortest distance from the hyperplane to the closest negative item. There are many possibilities of fitting a hyperplane to linearly separate positive and negative data items. The SVM searches for an optimal hyperplane H with largest margin γ.

Without loss of generality, $\mathbf{w}$ and b are normalized so that the hyperplane is given in canonical form, leading to the following constraints.

$$\mathbf{w} \cdot \mathbf{x}^{(i)} + b \geq +1 \quad \text{for} \quad y^{(i)} = +1 \tag{4.3}$$

$$\mathbf{w} \cdot \mathbf{x}^{(i)} + b \leq -1 \quad \text{for} \quad y^{(i)} = -1 \; , \tag{4.4}$$

which can be summarized to

$$y^{(i)}(\mathbf{w} \cdot \mathbf{x}^{(i)} + b) - 1 \geq 0 \quad \forall i \; . \tag{4.5}$$

Data items fulfilling the equality constraints of 4.3 and 4.4 lie on two hyperplanes H_1 and H_2, respectively, which are parallel to the optimal hyperplane H (see figure 4.3). The distance of a data point lying on H_1 to the origin is $\frac{|1-b|}{\|\mathbf{w}\|}$ and lying on H_2 is $\frac{|-1-b|}{\|\mathbf{w}\|}$. The margin of the hyperplane H can thus be defined as $\gamma = \frac{2}{\|\mathbf{w}\|}$. Thereby, maximizing the margin to find an optimal hyperplane is equivalent to minimizing $\| \mathbf{w}^2 \|$, subject to constraint 4.5. The data items fulfilling equality 4.5, thus lying on H_1 and H_2, are termed *support vectors* and are highlighted in figure 4.3 with an additional circle. Only these items affect the training of the SVM. If the SVM would be trained again excluding all other items than the support vectors, the optimal hyperplane would still be the same. Removal of a support vector, however, would change the solution as the maximum margin γ would change.

If an optimal hyperplane with largest margin, thus optimal parameters $\mathbf{w}$ and b, has been found, a new data point $\mathbf{x}$ can be classified by evaluating

$$y = \operatorname{sgn}(h(\mathbf{x})) \; , \tag{4.6}$$

with classification function

$$h(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b \; . \tag{4.7}$$

In a geometrical interpretation, the class label is determined by evaluating on which side of the optimal hyperplane H the data point $\mathbf{x}$ falls.

A feasible solution, however, only exists if the training items $\mathbf{x}^{(i)} \in \Gamma$ are linearly separable. In real world applications, this is rarely the case. The constraint 4.5 is therefore relaxed by introducing slack variables $\xi_i \geq 0$ allowing also data items lying within the margin or on the wrong side of H. This leads to the *soft margin* formulation of the optimization problem defined as

$$\min_{\mathbf{w},b} \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{N} \xi_i \tag{4.8}$$

$$\text{subject to} \quad y^{(i)}(\mathbf{w} \cdot \mathbf{x}^{(i)} + b) \geq 1 - \xi_i \quad \forall i \tag{4.9}$$

$$\xi_i \geq 0 \quad \forall i \tag{4.10}$$

$$C \geq 0 \; . \tag{4.11}$$

The sum $\sum_{i=1}^{N} \xi_i$ can be interpreted as an upper bound on the number of training errors. The parameter C, which needs to be chosen by the user, is termed *regularization parameter* and controls the penalization of errors made. A large C corresponds to a higher penalty to errors. It must be mentioned, however, that data items lying on the correct side of the hyperplane, but within the margin, are also penalized.

The optimization problem 4.8 with constraints 4.9 to 4.11 can now be solved by use of *Lagrange theory* (Cristianini and Shawe-Taylor (2003)). The *primal Lagrangian form* of the problem thus is

$$L_P = \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \alpha_i \{ (\mathbf{w} \cdot \mathbf{x}^{(i)} + b) - 1 + \xi_i \} - \sum_{i=1}^{N} r_i \xi_i \; . \tag{4.12}$$

Here, $\alpha_i \geq 0$ and $r_i \geq 0$ are referred to as *Lagrange multipliers*. Necessary and sufficient conditions for a solution to a convex optimization problem as stated in 4.8 with equality and inequality constraints are the *Karush-Kuhn-Tucker* (KKT) conditions:

$$\frac{\partial L_P}{\partial \mathbf{w}} = 0 \tag{4.13}$$

$$\frac{\partial L_P}{\partial \xi_i} = 0 \tag{4.14}$$

$$\frac{\partial L_P}{\partial b} = 0 \tag{4.15}$$

$$y^{(i)}(\mathbf{w} \cdot \mathbf{x}^{(i)} + b) - 1 + \xi_i \geq 0 \tag{4.16}$$

$$\xi_i \geq 0 \tag{4.17}$$

$$\alpha_i \geq 0 \tag{4.18}$$

$$r_i \geq 0 \tag{4.19}$$

$$\alpha_i[y^{(i)}(\mathbf{w} \cdot \mathbf{x}^{(i)} + b) - 1 + \xi_i] = 0 \tag{4.20}$$

$$r_i \xi_i = 0\ . \tag{4.21}$$

Equation 4.20 and 4.21 are known as the KKT complimentary conditions.

In practice, the primal Lagrangian is not solved but it is transformed into a dual form. This form is easier to solve and furthermore enables the latter use of kernels as the training items only appear in the form of dot products, as will be discussed in section 4.4.2. Differentiating the primal form with respect to $\mathbf{w}, \xi$ and b, imposing stationary results in

$$\frac{\partial L_P}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N} y^{(i)} \alpha_i \mathbf{x}^{(i)} = 0$$

$$\Leftrightarrow \mathbf{w} = \sum_{i=1}^{N} y^{(i)} \alpha_i \mathbf{x}^{(i)} \tag{4.22}$$

$$\frac{\partial L_P}{\partial \xi_i} = C - \alpha_i - r_i = 0 \tag{4.23}$$

$$\frac{\partial L_P}{\partial b} = \sum_{i=1}^{N} y^{(i)} \alpha_i = 0\ . \tag{4.24}$$

Substitution of the relations obtained gives the *dual Lagrangian form*

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} y^{(i)} y^{(j)} \alpha_i \alpha_j \mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)}\ . \tag{4.25}$$

The KKT condition 4.23 (same as 4.14) together with 4.19 enforces $\alpha_i \leq C$, and the complementary condition 4.21 implies that $\xi_i \neq 0$ only if $r_i = 0$ and thus $\alpha_i = C$. Together with complementary condition 4.20, this implies that non zero slack variables can only occur when $\alpha_i = C$, thus the distance of those training items is less than the desired margin $\frac{1}{\|w\|}$.

Points fulfilling the inequality constraint $0 < \alpha_i < C$ (see eq. 4.23), lie at the distance of $\frac{1}{\|w\|}$. All items with non-zero α_i are referred to as *support vectors*. This is similar to the linearly separable case, however now all items with $\xi_i > 0$ also belong to the set of support vectors.

With the dual form available, training of the SVM is equivalent to maximizing L_D subject to

$$\sum_{i=1}^{N} y^{(i)} \alpha_i = 0 \tag{4.26}$$

$$0 \leq \alpha_i \leq C \quad \forall i \; . \tag{4.27}$$

The classification function for a new item $\mathbf{x}$ of the optimal hyperplane is defined as

$$h(\mathbf{x}) = \sum_{i=1}^{N} y^{(i)} \alpha_i \mathbf{x} \cdot \mathbf{x}^{(i)} + b \; , \tag{4.28}$$

and the class label can again be derived by

$$y = \operatorname{sgn}(h(\mathbf{x})) \; . \tag{4.29}$$

4.4.2 Non Linear Support Vector Machines

The introduction of the slack variables ξ_i already allows to solve the optimization problem even for non perfectly linear separable training data, however, it might not always lead to the desired result. In order to allow for further discriminative power of the SVM approach, the training data is mapped by a non linear transformation

$$\Phi : \mathbb{R}^q \rightarrow \mathbb{F} \tag{4.30}$$

$$\mathbf{x} \mapsto \Phi(\mathbf{x}) \tag{4.31}$$

into a higher dimensional feature space $\mathbb{F}$. In the higher dimensional space $\mathbb{F}$ the linear SVM algorithm can then be executed as schematically depicted in figure 4.4.

The explicit calculation of the higher dimensional feature space and the optimal hyperplane within this space is computationally very expensive. However, in the dual formulation of the optimization problem, only the dot product of two training data items $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ need to be calculated (see eq. 4.25). Thus, in order to implicitly obtain the optimal hyperplane in $\mathbb{F}$, only the dot product $\Phi(\mathbf{x}^{(i)}) \cdot \Phi(\mathbf{x}^{(j)})$ has to be computed. The dot product can be replaced by a *kernel function* $K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \Phi(\mathbf{x}^{(i)}) \cdot \Phi(\mathbf{x}^{(j)})$, given that the kernel function fulfills *Mercer's condition* (Schölkopf (2002)). Thereby, the explicit values of $\Phi(\mathbf{x}^{(i)})$ and Φ itself never need to be known, while all other aspects of the previous chapter still hold.

The decision function of the SVM, can thus be written as

$$h(\mathbf{x}) = \sum_{i=1}^{N} y^{(i)} \alpha_i K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) + b \; . \tag{4.32}$$

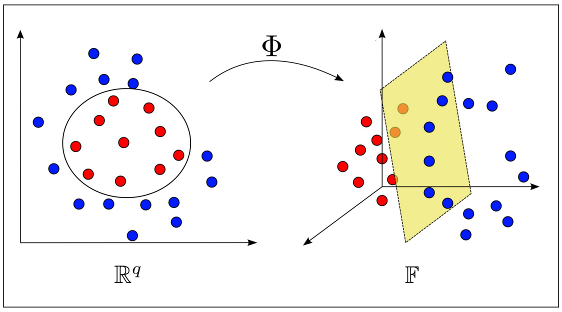


Figure 4.4: The function Φ maps the data items from $\mathbb{R}^q$ into a higher dimension feature space $\mathbb{F}$. In $\mathbb{F}$, the data items can be linearly separated.

A commonly used kernel function is the *Gaussian or radial-basis function* (RBF) kernel, defined as

$$K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \exp - \frac{\| \mathbf{x}^{(i)} - \mathbf{x}^{(j)} \|^2}{2\sigma^2} , \tag{4.33}$$

for which the dimensionality of $\mathbb{F}$ is infinite (Cristianini and Shawe-Taylor (2003)). Other frequently used kernel functions are linear or polynomial kernels.

$$K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)} \qquad \text{(linear kernel)} \tag{4.34}$$

$$K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = (\mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)})^p \qquad \text{(polynomial kernel)} \tag{4.35}$$

4.4.3 Parameter Selection

Crucial to the discriminative power of the SVM is the choice of the regularization parameter C and the kernel parameters, for example in the case of RBF kernels the choice of σ. Thus, a strategy is required which allows to obtain the best parameter setting for the given training data and a kernel selection. There are several works on automatically choosing appropriate parameters, for example the work of Chapelle et al. (2002), Duan et al. (2003), Gold et al. (2005) or Glasmachers and Igel (2008). However, still the most common way to determine a low number of regularization and kernel parameters is the application of a grid search method. To this end, a finite set of values for each parameter is chosen and for each parameter value combination the performance of the SVM trained with these parameters is evaluated. Here, the performance evaluation can either be performed on a separate test set or by performing cross validation on the training data. For example, the training set is split into four equally sized, mutually exclusive, sub sets in a four fold cross validation. In each round, one of the sub sets is held back as a validation set whereas the remaining three are used for training of the SVM. The average performance over all the four validation sets is used as the SVM performance. It is evident that for a higher number of parameters, a higher number of finite values for a fixed parameter set (i.e. a finer grid), and/or a larger number of cross validation steps, grid searching can become quite time consuming. It can therefore be reasonable to first use a larger grid and then refine it for areas which have already shown good performance.

4.4.4 Probabilistic Outputs for SVMs

One potential disadvantage of SVMs is that the output of its decision function (see eq. 4.29) assigns each input $\mathbf{x}$ either to the positive ($\mathrm{sgn}(h(\mathbf{x})) = +1$) or to the negative class ($\mathrm{sgn}(h(\mathbf{x})) = -1$). In some applications, however, it is more reasonable to obtain a measure for each item $\mathbf{x}$ on how likely it is assigned to the positive class. Regular SVMs outputs do not provide such a likelihood measure and solely provide a binary classification output.

One frequently used method to obtain probabilistic SVM outputs is the one proposed by Platt, which converts the unbounded SVM outputs to posterior probabilities on [0,1] (Platt (1999)). Therefore, the posterior probability $P(y = +1|h(\mathbf{x}))$ is expressed through a sigmoid function

$$P(y = +1|h(\mathbf{x})) = \frac{1}{1 + \exp(Ah(\mathbf{x}) + B)} \; . \tag{4.36}$$

In order to obtain a calibrated output, the parameters A and B have to be fitted. This is done on a training set $(h(\mathbf{x}^{(i)}), p_i)$ where

$$p_i = \frac{y^{(i)} + 1}{2} \tag{4.37}$$

defines the target probabilities derived from the class label $y^{(i)}$. Parameters A and B are then found by minimizing the negative log likelihood of the training data

$$\arg\min_{A,B} \left\{ -\sum_i p_i \log(c(\mathbf{x}^{(i)})) + (1 - p_i)\log(1 - c(\mathbf{x}^{(i)})) \right\} \tag{4.38}$$

$$\text{with}\, c(\mathbf{x}^{(i)}) = \frac{1}{1 + \exp(Ah(\mathbf{x}^{(i)}) + B)} \tag{4.39}$$

referred to as the *calibrated output*.

Now the question how to choose the training set and how to prevent overfitting to the training set needs to be considered. According to Platt (Platt (1999)), a good choice is the use of cross validation to obtain a training set for the calibration of the SVM outputs. In a three fold cross validation, for example, the training set used for SVM training is split into three parts. Three SVMs are trained on permutations on two out of the sub sets and the remaining third is used to obtain the $h(\mathbf{x}^{(i)})$. The union of all three $\{(\mathbf{x}^{(i)}, h(\mathbf{x}^{(i)}))\}$ sets can be used to form the actual training set for output calibration.

Performing cross validation to obtain a training set reduces overfitting of A and B to the training data, however, still the sigmoid can be overfit especially for a small number of items. Therefore, Platt introduces a regularization term, which requires a distribution of out-of-sample data. This data is modeled with the same empirical distribution as the training data, however with a finite probability of an opposite label. Thus, if a positive item is observed for some SVM output $h(\mathbf{x}^{(i)})$, the target value is not set to $p_i = +1$ but to $p_i = 1 - \epsilon_+$ with some ϵ_+. This reflects the fact that there is a finite chance of observing an opposite label for the same $h(\mathbf{x}^{(i)})$. A similar strategy applies for negative items. The MAP estimate for the target probabilities can be obtained by Baye's rule. For N_+ positive items and N_- negative items, the target values are given as:

$$p_+ = \frac{N_+ + 1}{N_+ + 2} \quad \text{and} \quad p_- = \frac{1}{N_- + 2} \, . \tag{4.40}$$

These target values are used throughout the fitting of the sigmoid function instead of the target values defined in equation 4.37. These non binary target values will converge to $\{0,1\}$ when the size of the training set reaches infinity. Further motivation and details for these target values can be found in Platt (1999) which also provides a pseudo code for sigmoid training. It has to be pointed out that, by applying output calibration, the time required for SVM training will increase as an additional cross validation is introduced.

An alternative to fitting a sigmoid to the SVM output is the application of SVM regression (Smola and Schölkopf (2003)). This strategy would require training labels with probability assignments. For synapses, however, it is not possible to obtain labels with reliable probability assignments as has been shown in section 4.2:

Another recently introduced alternative is the relevance vector machine (RVM) (Tipping (2000)). It has the same functional form as the SVM but obtains probabilistic outputs through Bayesian interference. However, due to its expectation-maximization learning, it can potentially get stuck in local minima, unlike the SVM algorithm. Furthermore, for larger data sets, training is more expensive than for SVMs.

4.5 The i3S for Object Detection

In the last sections, the basics for SVM-based object detection have been described. Now, a detailed description of the individual steps of the intelligent synapse screening system will be given. In general, it consists of three steps as can be seen in figure 4.5. First, image preprocessing is carried out, subsuming elimination of image artifacts, noise reduction and image normalization as depicted in step a). Since in most cases synapses are detected only on one or few images of the stack, the process of image selection is also listed. In the next step, a training set is constructed based on a human expert labeling and a SVM is trained (see (b) in figure 4.5). In step c), the trained SVM is then applied to detect synapses in previously unseen images. This process requires filtering the classification result, termed *confidence map*, with a threshold t. Since this threshold is not applied to the gray values of the original image but to the confidence map, it is referred to as *confidence threshold*. In this work, it will be described how this confidence threshold can be deduced based on a human expert reference list. Furthermore, a strategy is proposed to estimate a constant threshold which can be applied to many images. Figure 4.6 displays a screenshot of the graphical user interface of the i3S system which allows a user to setup expert reference lists, train the i3S as well as perform object detection in new images.

4.5.1 Image Preprocessing

Image preprocessing, especially image normalization, plays a significant role for the performance of the i3S system for object detection. For a real world application, it needs to be

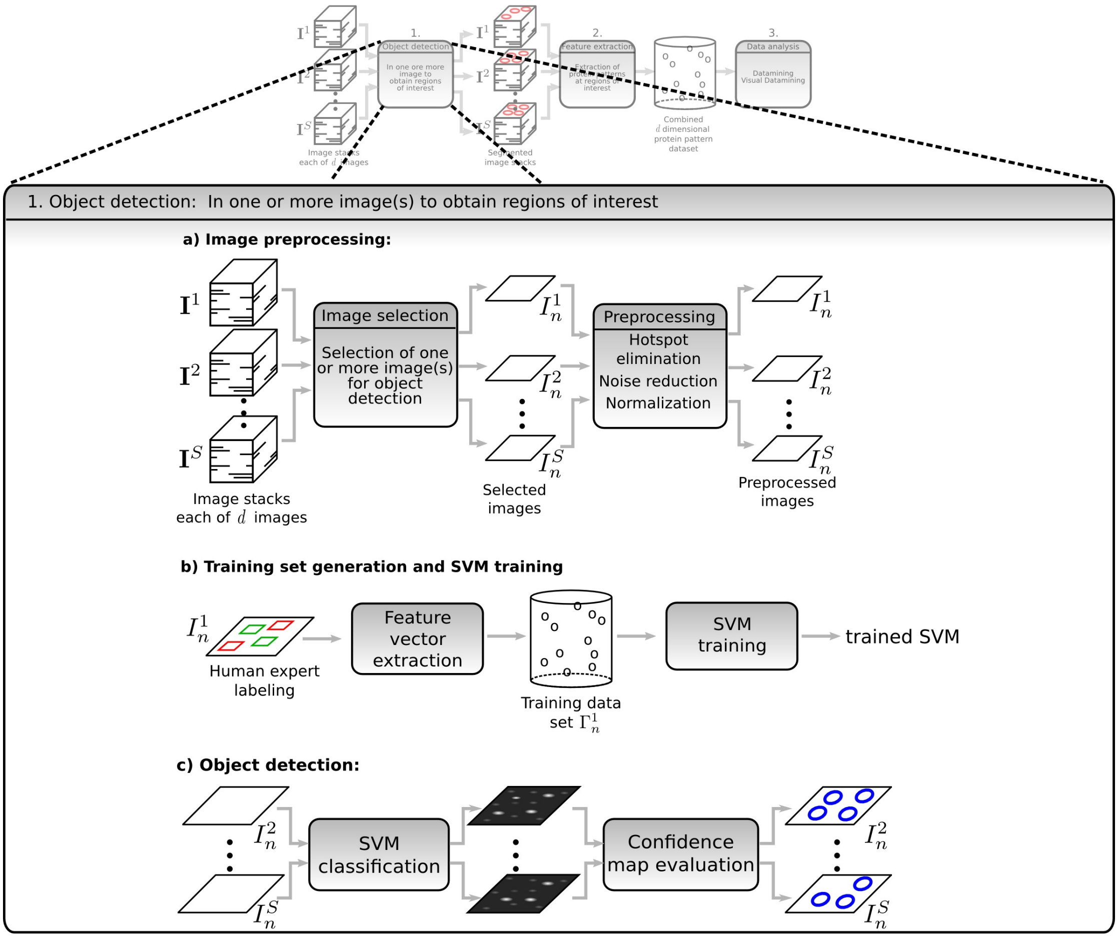
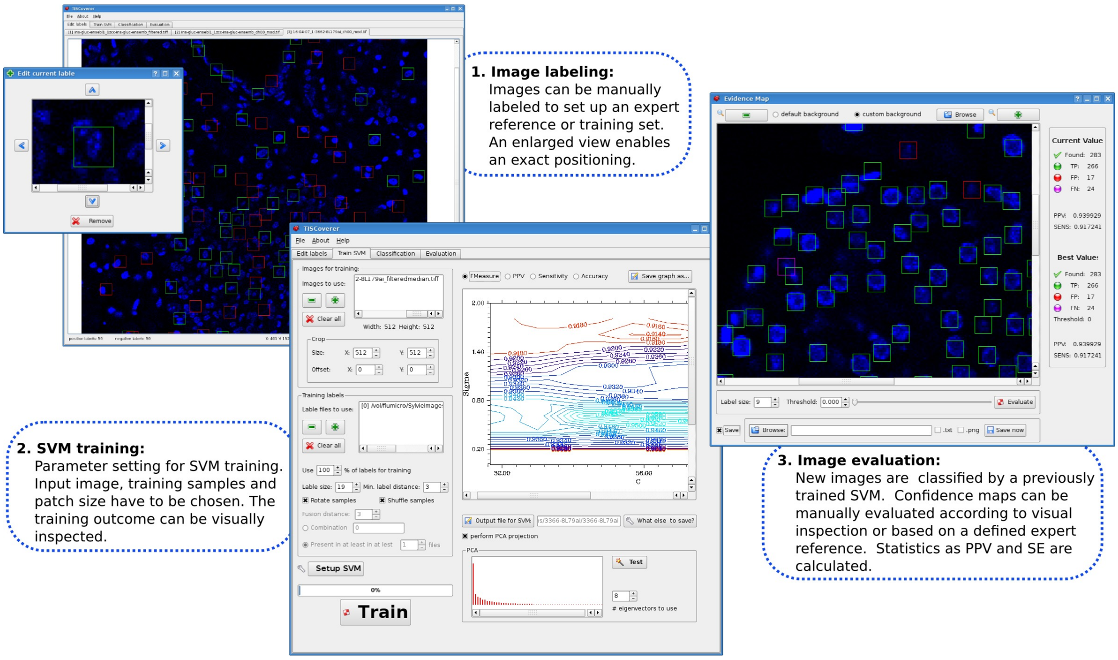

$\mathbf{I}^1$
$\mathbf{I}^2$
$\mathbf{I}^S$
Image stacks each of d images
1.
Object detection
In one ore more image to obtain regions of interest
Segmented image stacks
2.
Feature extraction
Extraction of protein patterns at regions of interest
Combined d dimensional protein pattern dataset
3.
Data analysis
Datamining Visual Datamining
1. Object detection: In one or more image(s) to obtain regions of interest
a) Image preprocessing:
Image selection
Selection of one or more image(s) for object detection
I_n^1
I_n^2
I_n^S
Selected images
Preprocessing
Hotspot elimination
Noise reduction
Normalization
Preprocessed images
b) Training set generation and SVM training
I_n^1
Human expert labeling
Feature vector extraction
Training data set Γ_n^1
SVM training
trained SVM
c) Object detection:
I_n^2
I_n^S
SVM classification
Confidence map evaluation

Figure 4.5: Schematic description of the object detection via the i3S, which is the first step in the analysis of multivariate fluorescence tissue micrographs. (a) Images are preprocessed, which includes hotspot elimination, noise reduction and normalization of the images. As usually object detection is only carried out on one or few images of the stacks, also the image extraction is included. (b) Based on a human expert labeling, a data set is generated for SVM training. Usually, this training is only carried out on one sample. (c) Test samples are classified by the trained SVM. The obtained classifier outputs, termed *confidence maps*, are evaluated to obtain the final object positions.

possible that a SVM trained on one image is capable of detecting synapses in previously unseen images. Otherwise, for each new image encountered a new SVM would need to be trained, which is not feasible for an application with considerable throughput. Thus, synapse characteristics of one image need to be comparable to synapse characteristics of another image. In the synapse image domain, however, it is often the case that even two samples labeled for the same protein can show quite some variation in their intensity profile. Hence, a normalization has to be carried out. Furthermore, synapse signals are disturbed by noise so that a noise reduction can be reasonable.

Before taking care of noise reduction and image normalization, image artifacts caused by



1. Image labeling: Images can be manually labeled to set up an expert reference or training set. An enlarged view enables an exact positioning.

2. SVM training: Parameter setting for SVM training. Input image, training samples and patch size have to be chosen. The training outcome can be visually inspected.

3. Image evaluation: New images are classified by a previously trained SVM. Confidence maps can be manually evaluated according to visual inspection or based on a defined expert reference. Statistics as PPV and SE are calculated.

Images to use: 2-8L179ai_filteredmedian.tiff
Width: 512 Height: 512
Crop Size: X: 512 Y: 512
Offset: X: 0 Y: 0
Lable files to use: [0] /vol/flumicro/SylvieImage
Use 100 % of labels for training
Lable size: 19 Min. label distance: 3
☒ Rotate samples ☒ Shuffle samples
Fusion distance: 3
○ Combination 0
● Present in at least in at lest 1 files
0%
● FMeasure ○ PPV ○ Sensitivity ○ Accuracy
Save graph as...
Output file for SVM: s/3366-8L79ai/3366-8L79ai
What else to save?
☒ perform PCA projection
8 # eigenvectors to use

Current Value
Found: 283
TP: 266
FP: 17
FN: 24
PPV: 0.939929
SENS: 0.917241

Best Value
Found: 283
TP: 266
FP: 17
FN: 24
Threshold: 0
PPV: 0.939929
SENS: 0.917241

Label size: 9 Threshold: 0.000
☒ Save Browse: □ .txt □ .png

Figure 4.6: Graphical user interface for the i3S detection system. It allows the user to label objects to obtain training sets and expert references (1). Furthermore, it provides an interface for the training of the i3S (2). The detection result can visually be inspected, including SE and PPV measures, and the threshold can be interactively adjusted (3). Images of the pancreas data set stained with DAPI are shown.

defective CCD elements need to be eliminated. Here, single pixels in the originally recorded images exhibit the maximum intensity value, referred to as *hotspots*. As a result of the shifting and bleach correction process (see section 2.1), however, the intensity values of the hotspots are modified and additionally *negative hotspots* are introduced as depicted in figure 4.7 (a) labeled with white boxes. The broken pixels show intensity values in the range of true signals and an elimination of broken pixels by locating maximum intensity pixels is thus not possible. For few hotspots, they can be manually eliminated which was done for some images in this work. However, if the number of images and/or hotspots increases, it is reasonable to perform an automated hotspot correction. Therefore, an adaptive median filter strategy is applied which takes into account the local intensity characteristic to decide if a pixel has to be adapted. Formally, the filter can be described as

$$\hat{f}(\mathbf{p}) = \begin{cases} \mathrm{median}(\Omega_3(\mathbf{p})) & \text{if} \quad \left|\frac{\mathrm{median}(\Omega_3(\mathbf{p}))-f(\mathbf{p})}{f(\mathbf{p})}\right| > 0.3 \\ f(\mathbf{p}) & \text{else} \end{cases} . \tag{4.41}$$

Here, $\mathrm{median}(\Omega_3(\mathbf{p}))$ is the median intensity value of the 3×3 neighborhood Ω_3 around $\mathbf{p}$ and $\hat{f}(\mathbf{p})$ is the filtered intensity value of $\mathbf{p}$. An efficient elimination of CCD-camera caused errors can thereby be obtained while leaving almost all other pixels untouched. As can be seen in figure 4.7, the positive and negative hotspots were nicely removed by the filtering

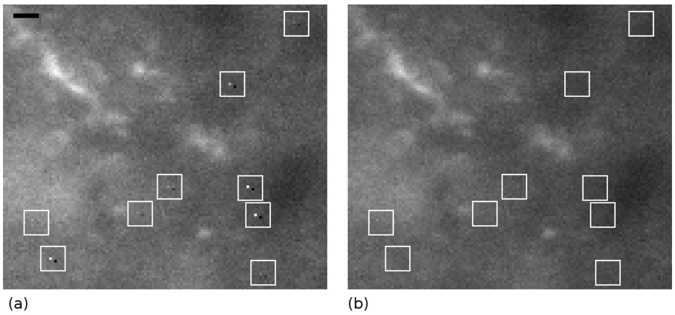


Figure 4.7: Example of the application of the image preprocessing pipeline. (a) Original image (for visualization purposes rescaled to 8bit). Image artifacts caused by defective CCD elements, termed *hotspots*, are highlighted with white rectangles. (b) Application of an adaptive median filter corrected positive and negative hotspots. Scale bar of 2.16μm refers to both images.

strategy. The chosen threshold of 0.3 was determined experimentally for a sub set of images.

If the noise level in the images is high, it is reasonable to carry out a noise reduction step. Here, it is of great importance to choose a filtering method which preserves the small synaptic structures (3×3 - 5×5 pixels). An edge preserving smoothing filter, termed *bilateral filtering*, has been introduced by Tomasi and Manduchi (1998). Its benefits over other edge preserving filters, such as anisotropic diffusion (Perona and Malik (1990)), lies in the effectiveness despite its non-iterative procedure. Basically, the filter is based upon the Gaussian filter (Gonzalez and Woods (2002)), however the weighting of the neighboring pixels is influenced by their distance to the center pixel in the spatial as well as in the intensity domain. The smoothing output for a center pixel $\mathbf{p}$ can be computed as

$$\hat{f}(\mathbf{p}) = \frac{1}{o(\mathbf{p})} \sum_{\mathbf{p}' \in \Omega_{M_b}} f(\mathbf{p}')g(\mathbf{p},\mathbf{p}')s(\mathbf{p},\mathbf{p}')) \ , \tag{4.42}$$

with $\mathbf{p}$' being a pixel in the $M_b \times M_b$ neighborhood Ω_{M_b} of center pixel $\mathbf{p}$, and f and $\hat{f}$ being the input and output intensities, respectively. The two terms

$$g(\mathbf{p},\mathbf{p}') = e^{-\frac{\|\mathbf{p}-\mathbf{p}'\|^2}{2\sigma_d^2}} \text{ and } s(f(\mathbf{p}), f(\mathbf{p}')) = e^{-\frac{\|F(\mathbf{p})-F(\mathbf{p}')\|^2}{2\sigma_r^2}} \tag{4.43}$$

account for the distance in the spatial domain and for the distance in the intensity domain, respectively. Here, σ_d defines the standard deviation in the spatial domain, known as geometric spread, and σ_r the standard deviation in the intensity domain, termed photometric spread. The term $o(\mathbf{p}) = \sum_{\mathbf{p}' \in \Omega_{M_b}} g(\mathbf{p},\mathbf{p}')s(\mathbf{p},\mathbf{p}'))$ is a normalizing term. In standard bilateral filtering, $F(\mathbf{p})$ is the intensity of a pixel $\mathbf{p}$, i.e. $F(\mathbf{p}) = f(\mathbf{p})$. However, also more complex characteristics of the intensity domain can be considered. For example, the sole intensity can be replaced by the median, mean or standard deviation of a neighborhood around $\mathbf{p}$. It is also possible to combine those characteristics, so that a feature vector instead of a

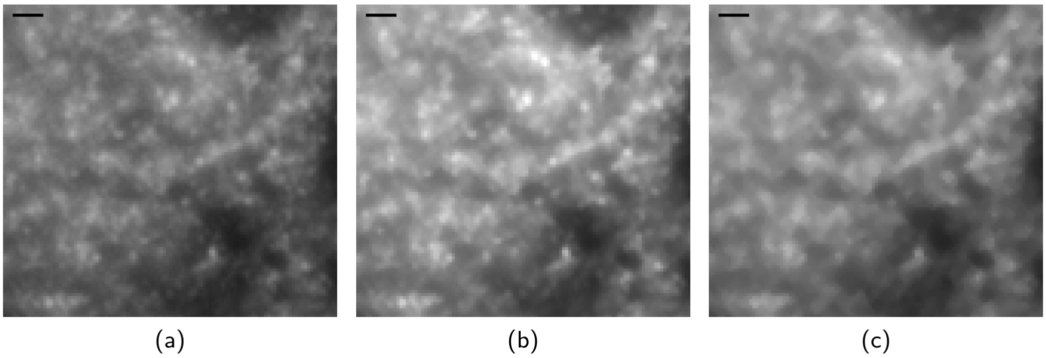


Figure 4.8: Preprocessed sub-image. (a) Original image eliminated for hotspots. (b) Bilateral filtered image. (c) Median filtered image. Scale bars are 1.8μm.

single feature is regarded. It has to be considered, however, that an additional parameter is introduced: the neighborhood size in the intensity domain.

Besides the standard bilateral filtering with $F(\mathbf{p}) = f(\mathbf{p})$, also this enhanced variant as well as standard filtering methods as Gaussian and median filtering were tested on the synapse images. For the Gaussian filtering, it could clearly be observed that synapses were largely blurred and also the median filtering tended to blur synaptic structures. Thus, they are not well suited for this image domain. Both the standard as well as the enhanced version of the bilateral filtering preserved synaptic structures if suitable parameters were chosen. No clear advantage of the enhanced version could be visually observed.

When applying filtering methods on an image as a preprocessing step, it is very difficult to judge if the cost of the filtering is justified by the obtained gain in image quality, and in this case, in synapse detection performance. One can visually inspect the obtained filtered outcome to estimate the noise elimination and, on the other hand, the artifacts introduced. Yet, it is only a visual impression which can vary for different observers and different settings of the computer monitor. On the other hand, one can analyze the obtained synapse detection performance with and without filtering. This has experimentally been done for the synapse data set and only slightly lower performances were observed for non filtered data. Thus, it can even be reasonable to omit the filtering step as the SVM seems to perform reasonably well due to its good generalization performance even if some noise is present in the image. Still I would suggest using standard bilateral filtering in the context of synapse detection with filter parameters $M_b = 3$, $\sigma_d = 0.5$ and $\sigma_r = 10$. The larger the geometric spread σ_d the more the filter blurs the image, as also image values from more distant locations are taken into account. The photometric spread σ_r controls the blurring of intensity edges. Thus, the smoothed output is mainly influenced by pixels which are spatially close and have a similar intensity value. By choosing small values for all parameters, the image is only slightly blurred and no severe artifacts are introduced. Figure 4.8 exemplarily shows one preprocessed sub-image, showing that synaptic structures are well preserved by bilateral filtering (see figure 4.8 (b)) while the 3×3 median filter tends to slightly more blur the image (see figure 4.8 (c)).

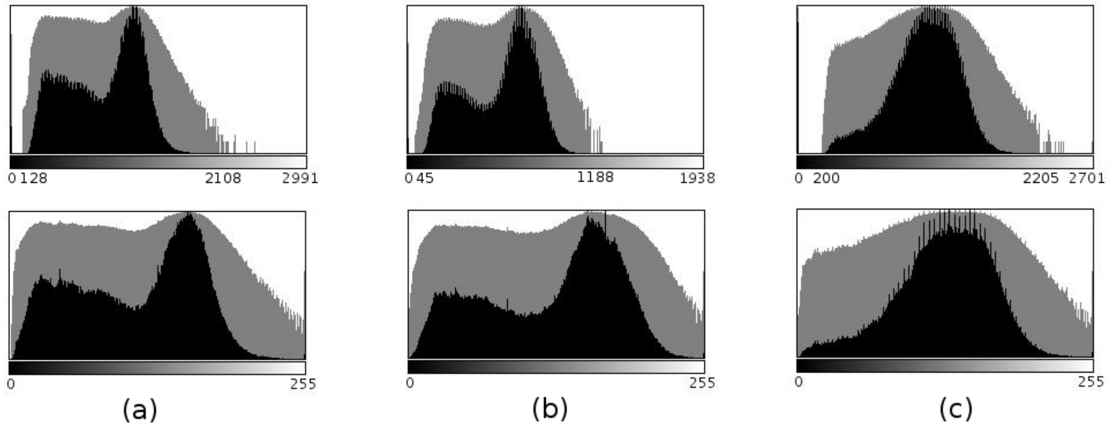


Figure 4.9: Histograms of three different images stained for synaptophysin. The top row displays the original 16bit histograms corrected for hotspots. The frequencies of each gray value is shown in black and the log of the frequencies in gray. Minimum and maximum intensity values are displayed as well as intensity values in between. The bottom row shows the preprocessed histograms. Intensity values between images are more comparable now.

The actual image normalization is carried out by the following contrast stretching procedure: The upper 0.01% of the gray value histogram were considered as outliers and were down shifted to the 99.99% gray value of the histogram. Subsequently, the images were normalized to cover the whole 8bit gray scale range of 0-255. To this end, the gray value interval in one image $[f_{min}, f_{max}]$ was linearly scaled to $[0, 255]$ (Gonzalez and Woods (2002)). Figure 4.9 shows image histograms for three different images stained for synaptophysin. The top row shows intensities in the 16bit gray scale range, with black being the frequencies and light gray the log of the frequencies. Intensity values for the minimum, maximum and values in between are displayed. It can be observed that intensities are not well comparable between images. The bottom row shows the histograms of the normalized images. It can nicely be seen that intensities are now more comparable.

4.5.2 Training Set Generation

The generation of a training set Γ_n^s for image I_n^s is based on an expert reference set ER$_n^s$. Therefore, for each labeled object position $\mathbf{p} \in \mathrm{ER}_n^s$, a feature vector $\mathbf{x}^{\mathbf{p}}$ is generated consisting of the M^2 gray values of an $M \times M$ neighborhood centered at $\mathbf{p}$. This set of feature vectors constitutes the set of positive items of Γ_n^s. Negative training items are obtained by randomly choosing image positions $\mathbf{p}'$ such that for any positive item position $\mathbf{p}$ the Euclidean distance $d(\mathbf{p}, \mathbf{p}') > (M-1)/2$. This strategy will guarantee negative items as the premise is a complete labeling of the (sub) image the expert reference ER$_n^s$ refers to. Thus, by requiring $d(\mathbf{p}, \mathbf{p}') > (M-1)/2$, the extracted negative patch will show object features at most at the rim of the patch but will never be centered on an object. Throughout this work, $M = 7$ was used for synapse detection resulting in 49-dimensional feature vectors.

This was motivated by the fact that synapses appear as 3×3 to 5×5 pixel sized objects. Thus, choosing image patches of size 7×7 will cover the synapse itself as well as some background region.

As the class of a patch is assumed to be invariant with respect to rotation, the training set size is increased by rotating each patch four times by an angle of $0, \pi/2, \pi, \text{and} \, 3\pi/2$.

4.5.3 i3S Training

In the training phase of the i3S system (see figure 4.5 step b), a SVM is trained with items of one training set Γ_n^s. The choice of the kernel and the number of training items which are used for training depend on the experimental setup and will be specified for each data set individually in section 4.6 and 4.7. A fully automated 10 fold cross validation is performed for estimating the SVM's soft margin regularization parameter C and, if required, the Gaussian kernel parameter σ. Here, those parameters are chosen which maximize the f-measure (see eq. 4.1). To obtain a probabilistic SVM output, a four fold cross validation is carried out on the same training set for parameter tuning of the sigmoid function (see eq. 4.38).

4.5.4 Computation of Object Positions

For object detection in one image I_n^s, the $M \times M$ neighborhood gray value vector $\mathbf{x}^{(\mathbf{p})}$ is extracted at each pixel location $\mathbf{p}$ in the image and the calibrated output $c(\mathbf{x}^{(\mathbf{p})})$ (cf. equation 4.39) is computed with the trained SVM. The value $c(\mathbf{x}^{(\mathbf{p})})$ is written to the corresponding position $\mathbf{p}$ of a new image matrix of the same size as the input fluorescence image I_n^s. Thereby, a new image called *confidence map* is created. The map is subsequently evaluated to obtain a set of object positions $\Lambda_n^s(t)$ according to the following rule

$$\Lambda_n^s(t) = \left\{ \mathbf{p} \middle| c(\mathbf{x}^{(\mathbf{p})}) \geq t \wedge c(\mathbf{x}^{(\mathbf{p})}) \geq c(\mathbf{x}^{(\mathbf{p}')}) \forall \left\{ \mathbf{p} | \mathbf{p}' \in \Omega_{M-2}(\mathbf{p}) \wedge \mathbf{p} \neq \mathbf{p}' \right\} \right\} \; . \tag{4.44}$$

The parameter t specifies the confidence threshold, $c(\mathbf{x}^{(\mathbf{p})})$ is the calibrated SVM output computed by equation 4.39, and $\Omega_{M-2}(\mathbf{p})$ defines the $(M-2) \times (M-2)$ neighborhood centered at position $\mathbf{p}$. If a local accumulation of object positions forms a region, i.e. this set of position constitute a connected set (see Gonzalez and Woods (2002), page 67), the accumulation of positions is replaced by the region center. The final set $\Lambda_n^s(t)$ contains the locations of all objects detected based on the confidence threshold t.

4.5.5 Estimating a Constant Confidence Threshold

Modification of the i3S confidence threshold t (see equation 4.44) results in different result sets $\Lambda_n^s(t)$ and thus in different detection performances (SE(t) and PPV(t), see equation 4.1). A confidence threshold tuned for a specific image can be obtained by selecting a confidence threshold $\tilde{t}_n^s$ for each analyzed image I_n^s based on the corresponding ER$_n^s$. Therefore, $\tilde{t}_n^s$ is chosen based on a SE(t) and PPV(t) specific criterion: the threshold $\tilde{t}_n^s$ for image I_n^s and expert reference ER$_n^s$ is determined according to

$$\tilde{t}_n^s = \arg\max_t \{\mathrm{SE}(t)|\mathrm{PPV}(t) \geq 0.8\} \,. \tag{4.45}$$

A cutoff value of PPV(t) ≥ 0.8 was chosen to allow for objects in $\mathrm{S}_{+,-}$. This was done to account for the non perfect human expert references, as has been discussed in section 4.2.

Although the selection of a threshold according to equation 4.45 results in a threshold tuned for a specific image, an application of this approach in practice would require a manual labeling for a sub-image in each image to be analyzed. For an application allowing for a significant image analysis throughput, this strategy is not feasible. Thus, a method is required which computes a confidence threshold t which can be kept constant for an entire sequence of images obtained under stable imaging conditions. Therefore, I developed a method to compute such a constant threshold, referred to as $t^{(c)}$ ($c =$ constant), based on ER_n^s-tuned thresholds $\tilde{t}_n^s$ (see equation 4.45) which were obtained for a sub set of manually labeled images. The constant threshold $t^{(c)}$ can be computed as follows.

$$\begin{aligned} &\text{if} \quad \lfloor 2\Theta \rfloor > 2\lfloor \Theta \rfloor \quad \text{then} \quad t^{(c)} = 10^{-3}\lceil \Theta \rceil \quad \text{else} \quad t^{(c)} = 10^{-3}\lfloor \Theta \rfloor \\ &\text{with} \quad \Theta = \frac{10^3}{S}\sum_s \tilde{t}_n^s \,, \end{aligned} \tag{4.46}$$

where S specifies the number of images contained in the sub set of manually labeled images, s is the running index of the stack the image belongs to, and n is the image number in that stack. Thus, each image which is manually labeled and used to estimate the constant confidence threshold belongs to a different stack but is labeled for the same protein.

4.6 Case study I: Brain Tissue

In the following, synapse detection results obtained with the i3S will be presented. Thereby, different questions will be addressed. First, it is investigated how the choice of the training set as well as the confidence threshold influences the quality of the detected synapses. Second, the influence on the detection performance of the labeling procedure as well as the application of a constant threshold $t^{(c)}$ is evaluated. This evaluation is performed on the image set $\mathcal{I}'$. Last, the performance of the i3S trained on one protein channel and tested on another protein channel is evaluated.

4.6.1 The Influence of Training Set and Threshold Choice

The aim of this first section is to get an impression how the choice of the confidence threshold as well as the choice of the training set influences the detection performance of the i3S. Therefore, the same sub-image $\hat{I}_n^s$ of image set $\mathcal{I}$ labeled for $\mathtt{syphys}$ as used to evaluate the human detection performance is chosen (see 4.2). This image was chosen as here labels with quality assignments exist and a quite good detection performance was obtained by the human experts.

Training and testing was performed on the same image so that the obtained detection accuracy could be ascribed to the chosen training set and threshold. Otherwise, it would not be clear if low detection performances are due to, for example, the chosen training set or an improper image normalization. Thus, to asses the accuracy and generalization performance of the i3S, a five fold cross validation strategy was performed. Therefore, an expert reference list $\overline{\mathsf{ER}}_n^s(X)$ (see section 4.2) was randomly split into five equally sized, disjoint parts

$$\overline{\mathsf{ER}}_n^s(X) = \bigcup_{i=1,\ldots,5} E_i(X)$$

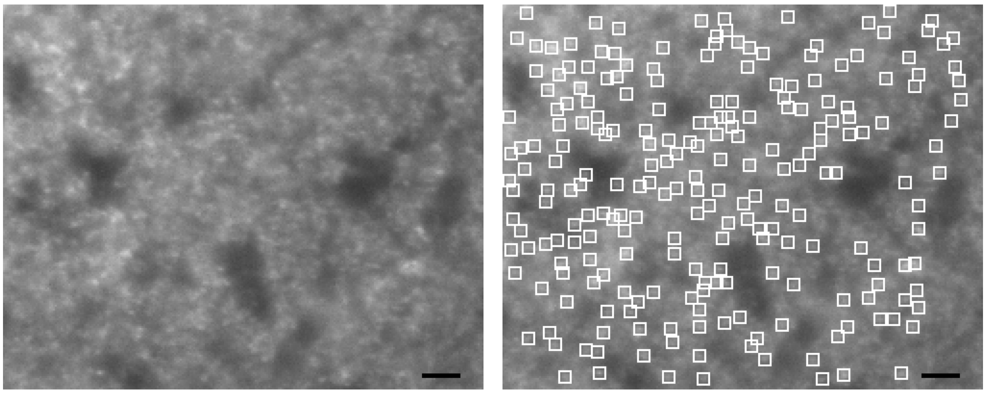
Iteratively, one $E_i(X)$ was chosen as a *test expert reference*, whereas the union of the remaining four was used to generate a training set. Thereby, five unique training sets were obtained for each expert reference $\overline{\mathsf{ER}}_n^s(X)$. Each training set constructed from $\overline{\mathsf{ER}}_n^s(A)$ contained 192 training items. The five training sets constructed from $\overline{\mathsf{ER}}_n^s(AB)$ each contained 332 training items. Each of the training sets was used to train one of five i3S for each quality group (A and AB), referred to as i3S$_i(X)$ (i indicates the corresponding test expert reference). Thus, in total 10 i3S were obtained. Training was carried out with a Gaussian kernel without output calibration. Training took around 4 hours on a Pentium 4 with 2.53GHz and 2GB RAM. Each obtained i3S$_i(X)$ was subsequently used for synapse detection on $\hat{I}_n^s$ as described in 4.5.4. Detection of synapses in one sub-image of size 250×200 pixel took around 5 seconds.

As training data was sampled from the whole sub-image $\hat{I}_n^s$, no meaningful PPV measures can be obtained if the detection result is compared solely against the test expert reference list $E_i(X)$. Already few synapses of $\mathcal{S}_{+,-}$ would decrease the PPV measure considerably, as only few test items are available (48 for $E_i(A)$ respectively 83 for $E_i(AB)$). To calculate a meaningful PPV, it is thus more reasonable to evaluate against $\overline{\mathsf{ER}}_n^s(X)$ which contains test as well as training items. While this is reasonable for the PPV measure, this strategy however might bias the SE measure as potentially only training synapses are detected. To obtain both reasonable SE and PPV measure, synapse detection performance was measured in two steps: First, synapse detection was evaluated against the whole $\overline{\mathsf{ER}}_n^s(X)$ to determine ER-tuned confidence thresholds $\tilde{t}_n^s$ for each i3S$_i(X)$ and meaningful PPV values. Second, SE was recalculated restricted to $E_i(X)$ for the determined confidence threshold $\tilde{t}_n^s$ to obtain unbiased SE measures.

Table 4.4 displays the mean SE values for ER-tuned thresholds $\tilde{t}_n^s$, referred to as SE$_\mu(\tilde{t}_n^s)$. The SE values were averaged over all five i3S$_i(X)$ trained on one of the two different training sets $\Gamma_n^s(A)$ and $\Gamma_n^s(AB)$. Evaluation was performed against expert references $\overline{\mathsf{ER}}_n^s(A)$, $\overline{\mathsf{ER}}_n^s(AB)$, and $\overline{\mathsf{ER}}_n^s(ABC)$. The mean PPV is not shown as it is 80% for each run due to the definition of $\tilde{t}_n^s$ (see equation 4.45). The unbiased SE$_\mu$ measure is shown in bold font. For the sake of completeness, in addition the biased SE$_\mu$ is shown in normal font. When comparing the unbiased SE$_\mu$ values, it is evident that the highest SE$_\mu$ is reached for that synapse quality which was used for i3S training. This behavior can not be observed for the biased SE$_\mu$. Here, the positions used for training are considered in the performance measure. Since it is likely that the i3S detects those synapses used for training, they positively influence the SE measures for evaluation against the reference list which contain all types of synapses,

		master expert reference		
		$\overline{\text{ER}}^s_n(A)$	$\overline{\text{ER}}^s_n(AB)$	$\overline{\text{ER}}^s_n(ABC)$
training	$\Gamma^s_n(A)$	78.28, **78.85**	78.66, **64.98**	80.61, **72.45**
set	$\Gamma^s_n(AB)$	71.23, **70.39**	80.43, **79.06**	83.09, **69.80**

Table 4.4: Average SE values obtained for i3S trained with $\Gamma^s_n(A)$ and $\Gamma^s_n(AB)$ for an ER-tuned confidence threshold $\tilde{t}^s_n$ and evaluated against expert references $\overline{\text{ER}}^s_n(A)$, $\overline{\text{ER}}^s_n(AB)$, and $\overline{\text{ER}}^s_n(ABC)$. SE values were averaged over all five runs of the cross validation. The biased SE$_\mu$, i.e. evaluated against test and training data, is displayed in normal font. The unbiased SE$_\mu$ is obtained by evaluation against only test items (bold font).



(a) (b)

Figure 4.10: Synapse detection result of one i3S$_i$ for image I^s_n. An ER-tuned threshold was applied, achieving an unbiased SE of 87% and a PPV of 80%. (a) Original image corrected for hotspots and rescaled to 8bit intensity range for display purposes. (b) Detection result. Detected synapses are highlighted with a white 7×7 rectangle. Scale bars are 4.3μm.

i.e. $\overline{\text{ER}}^s_n(ABC)$.

As an example, figure 4.10 displays a detection result obtained for an i3S$_i(A)$ with an ER-tuned threshold. A PPV of 80% and an unbiased SE of 87% were achieved. Here, 49 positions belong to the set $\mathcal{S}_{+,-}$, i.e. are detected by the i3S but not by the expert. When re-evaluating these patches against $\overline{\text{ER}}^s_n(AB)$, and $\overline{\text{ER}}^s_n(ABC)$, 37 and 46 of these false positives were found in these sets, respectively. Thus, they are most likely true synapses, but were not consistently labeled as one quality category by the different experts. The remaining three are labeled by one of the three experts as synapses, two of them category B, one category A. If one visually analyzes figure 4.10, it can be seen that the most prominent synapses were detected. However, still spot like structures are not detected.

Up to now, solely the average performance of the i3S for one specific threshold has been analyzed. Thereby, it could be shown that the choice of the training set influences the

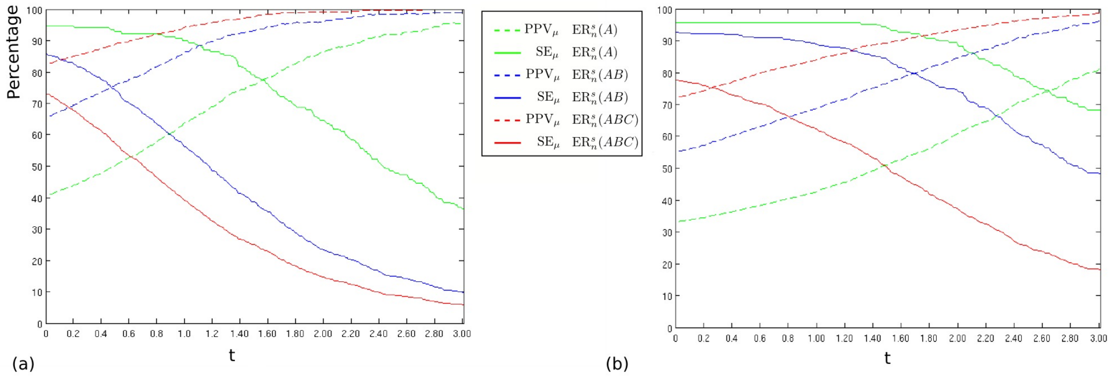
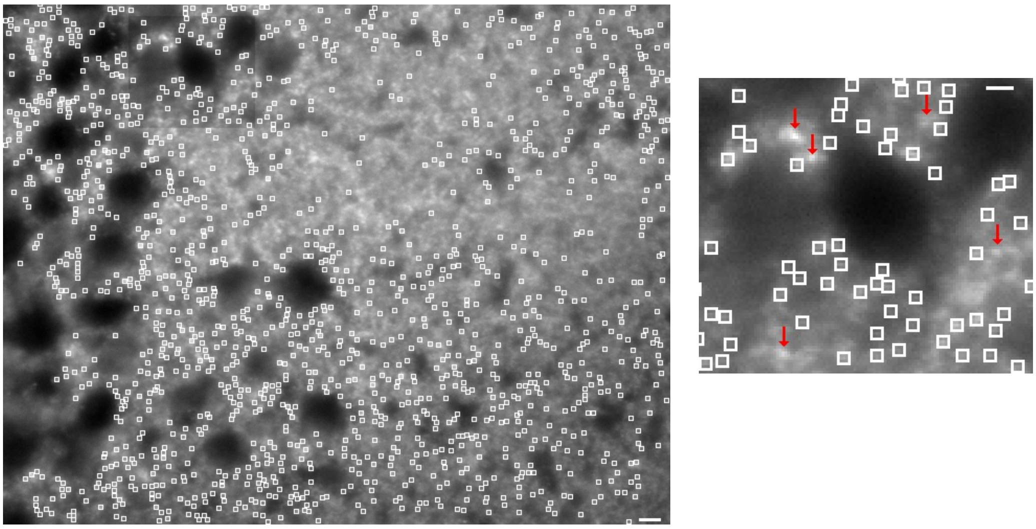


Figure 4.11: Mean unbiased SE (SE$_\mu$, solid lines) and PPV (PPV$_\mu$, dashed lines) obtained for different confidence thresholds t averaged over all five cross validation runs. (a) Training was performed on $\Gamma^s_n(A)$. (b) Training was performed on $\Gamma^s_n(AB)$. Performance was evaluated against $\overline{\text{ER}}^s_n(A)$ (green), $\overline{\text{ER}}^s_n(AB)$ (blue), and $\overline{\text{ER}}^s_n(ABC)$ (red). The variance of all results was less than 0.03 so error bars are omitted. The percentage refers to the SE as well as PPV measure.

detected synapse quality. It however also raised the question of, if and how the choice of the threshold influences the quality of the detected synapses. Therefore, SE and PPV were calculated for individual thresholds $t = \{0.0, 0.01, \ldots, 3.0\}$ and averaged over all five runs to obtain unbiased SE$_\mu(t)$ and PPV$_\mu(t)$ for each threshold. Figure 4.11 (a) displays the development of PPV$_\mu(t)$ and SE$_\mu(t)$ of the i3Ss trained with $\Gamma^s_n(A)$ for different confidence thresholds, and evaluation against different master expert references. Error bars are omitted as the variance was less then 0.03 for all results. The area under the curve of two associated SE and PPV lines is a good estimate of the i3S generalization performance. High SE values are obtained for evaluation against to $\overline{\text{ER}}^s_n(A)$, whereas SE values for $\overline{\text{ER}}^s_n(AB)$ and $\overline{\text{ER}}^s_n(ABC)$ are considerably lower. It is interesting to see that when specifying a high confidence threshold ($t \geq 2.28$), a PPV$_\mu(t) \geq 90\%$ (SE$_\mu(t) \leq 55\%$) was achieved with respect to $\overline{\text{ER}}^s_n(A)$. Thus, almost all detected synapses belong to the high certainty synapse set. Even higher PPV$_\mu(t)$ values are obtained for $\overline{\text{ER}}^s_n(AB)$ and $\overline{\text{ER}}^s_n(ABC)$. However, only around half of the synapses in $\overline{\text{ER}}^s_n(A)$ are detected. Decreasing the threshold results in a decrease in PPV$_\mu(t)$ but an increase in SE$_\mu(t)$ which is expected. If a threshold is chosen so that a SE$_\mu(t)$ of 90% is reached for $\overline{\text{ER}}^s_n(A)$, still a PPV$_\mu(t)$ of 86% and 94% with respect to $\overline{\text{ER}}^s_n(AB)$ and $\overline{\text{ER}}^s_n(ABC)$ was obtained. Thus, by decreasing the threshold, more lower quality synapses are included in the detection result. Similar behavior can be observed for the i3S trained with synapses of high and medium quality synapses, thus with $\overline{\text{ER}}^s_n(AB)$ (see figure 4.11 (b)). Compared to the training with $\overline{\text{ER}}^s_n(A)$, now high SE is reached for the evaluation against $\overline{\text{ER}}^s_n(A)$ *and* $\overline{\text{ER}}^s_n(AB)$. Also the SE values for $\overline{\text{ER}}^s_n(ABC)$, increased. However, there is still a larger gap between SE values of $\overline{\text{ER}}^s_n(AB)$ and $\overline{\text{ER}}^s_n(ABC)$. Thus, the i3S trained with high and medium synapses now detects well high and medium quality synapses but is not that sensitive to low quality synapses. Hence, the choice of the training



(a) (b)

Figure 4.12: Synapse detection on the whole image stained for `syphys` with an i3S trained on $\overline{\text{ER}}^s_n(A)$. (a) displays the whole image with detected synapses highlighted by white 5×5 boxes. (b) shows a sub-image of (a). Bright synapses which were not detected by the i3S are highlighted with a red arrow. Scale bars are (a) 4.32μm and (b) 2.16μm.

set as well as the choice of the confidence threshold are two possibilities to steer the quality of synapses detected by the i3S. As the experts agree quite well in their labeling of synapses of good quality (cf. section 4.2), it is reasonable to use synapses of good quality, and if required medium quality, to generate a training set. Fine tuning can then be achieved by tuning the threshold t.

Figure 4.12 (a) shows the detection result for the whole image I^s_n, manually contrast enhanced slightly for better visibility. Synapses are highlighted with white 5×5 pixel sized boxes. i3S training was performed on $\overline{\text{ER}}^s_n(A)$, extracted from the lower right part of the image. One can observe that synapses were detected across the image, however, in the upper middle part almost no synapses were detected, although punctuate structures are visible. Some of them are of low intensity, thus omitting these synapses shows that the i3S learned synapse characteristics of one quality class very well and thereby does not detect weaker synapses. However, it can also be observed that synapses with very high intensities were not detected which in the most cases is not a desirable result. Thus, the Gaussian Kernel tends to overfit the data. This is demonstrated by one example. Figure 4.12 (b) displays a sub-image of the whole image. Here, one can observe bright synapses, highlighted with a red arrow, which were not detected by the i3S.

4.6.2 The Influence of the Labeling Strategy and Constant Thresholds in an Inter-Image Detection Setup

In this section, i3S synapse detection performance is evaluated on images belonging to set $\mathcal{I}' = \{\mathbf{I}^{11}, \ldots, \mathbf{I}^{18}\}$. Recall that an individual image of $\mathcal{I}'$ is referred to by I^s, omitting the subscript as only one image is present in each stack. All other technical abbreviations are adapted accordingly to omit redundancy. The main questions addressed in this section are (i) the performance of the system in an inter-image detection setup, (ii) the influence of applying a constant confidence threshold, and (iii) the influence of the synapse labeling strategy (direct vs. indirect).

To generate expert references for i3S training and evaluation, for each image $I^s \in \mathcal{I}'$ an expert with three years of expertise in synapse labeling was asked to select a rectangular region, referred to as sub-image $\hat{I}^s$, which was representative for the entire image. The size of $\hat{I}^s$ was set to cover at least 120 synapses. In each of the eight $\hat{I}^s$, the expert manually marked *each* punctuate synaptophysin signal on the computer screen by selecting these signals, based on his visual expertise. Table 4.5 lists the size of the sub-images as well as the number of labeled synapses in that region for all eight images. For each of the obtained reference sets ER^s a training set Γ^s was extracted pursuant to section 4.5.2. An individual i3S, referred to as i3S$_s$, was trained for each image with a linear kernel, output calibration, and the corresponding training set Γ^s. Thus, eight trained i3S are obtained. A linear kernel was chosen instead of a Gaussian kernel, as the Gaussian kernel tended to overfit the data (see section 4.6.1). Output calibration was motivated by the fact that thresholds can thereby be chosen in a defined range of $[0, 1]$. Furthermore, the calibrated output provides a likelihood measure and are therefore easier to interpret, especially for biological experts. Still, intra-image performances similar to those observed in section 4.6.1 were achieved. Computational time for training of an i3S was mainly influenced by the size of the training set and the parameter range which had to be explored for optimization. Training in this setup, including output calibration, took on average two hours on a Pentium 4 with 2.53GHz and 2GB RAM. Synapse detection with a trained i3S on one of the 768×512 pixel sized images could then be achieved in approximately 20 seconds.

i3S Synapse Detection Performance

As has been discussed in section 3.2.1, to prove the real world applicability of the i3S approach, the accuracy as well as stability of the system need to be evaluated. First, it will now be focused on the accuracy or generalization performance of the system, i.e. how well it performs for varying image qualities. To this end, one sub-image $\hat{I}^{s'} \in \mathcal{I}'$ was used for i3S training. The trained i3S$_{s'}$ was then applied to the other sub-images $\hat{I}^s \in \mathcal{I}'$ with $s' \neq s$ from the same labeling procedure (directly or indirectly, see section 2.2.2) to detect synapses in these images. Accuracy measures SE and PPV were calculated based on the corresponding expert references ER^s for ER-tuned thresholds $\tilde{t}^s$, referred to as SE$(\tilde{t}^s)$ and PPV$(\tilde{t}^s)$. It has to be noted that this was done separately for each sub-image $\hat{I}^{s'}$ and separately for the directly labeled tissue sections and the indirectly labeled ones. As a consequence, each of the eight i3S$_{s'}$ was applied to three different images and an average SE$_\mu(\tilde{t}^s)$ for each

	sub-image size in pixel	# synapses
$\hat{I}^{11}$	768×128	558
$\hat{I}^{12}$	230×120	158
$\hat{I}^{13}$	272×160	128
$\hat{I}^{14}$	205×400	324
$\hat{I}^{15}$	200×111	159
$\hat{I}^{16}$	141×176	218
$\hat{I}^{17}$	275×100	225
$\hat{I}^{18}$	172×135	185

Table 4.5: Number of manually marked synapses in sub-images of four directly labeled sections ($\hat{I}^{11} - \hat{I}^{14}$) and four indirectly labeled sections ($\hat{I}^{15} - \hat{I}^{18}$). Sub-image size is given in pixels.

i3S$_{s'}$ could be computed. These values are displayed in the first column of table 4.6 where the first four rows correspond to directly labeled images $I^{11}, \ldots, I^{14}$ and the last four rows correspond to indirectly labeled synapses $I^{15}, \ldots, I^{18}$. The PPV$_\mu(\tilde{t}^s)$ are not shown as they are approximately 80% for each detection result as a consequence of the definition of $\tilde{t}^s$ (see eq. 4.45). Taking a closer look at the SE$_\mu(\tilde{t}^s)$ reveals that higher SE$_\mu$ were reached for indirectly labeled synapses (last four rows of table 4.6). On average, additionally lower standard deviations were observed for these images. Averaged over all i3S$_{s'}$ of one labeling type, an overall average SE$_\mu$ of 75% could be achieved for directly labeled synapses and 84% for indirectly labeled synapses.

As discussed in section 4.2, a human expert generated ER^s is error prone by nature because of the expert's limited ability to work on a constantly high cognitive level. Although an expert tries to label all synapses up to a certain minimum quality, it is often the case that he misses higher quality synapses but includes lower quality synapses. Furthermore, considerable variation can be observed between experts, especially between experts with different expertise. Therefore, a careful visual comparison of human expert-detected synapses with the synapse detected by i3S was performed. For an additional evaluation, a second expert (Walter Schubert) carefully inspected the resulting synapse positions computed by the i3S. The expert visually browsed through those image positions, which were marked as synapses by the i3S. Based on his experience he classified each of these positions as a true positive or false positive i3S result. A comparison of his results with the reference sets ER^s clearly showed that a significant number of synapses were missing in the ER^s. Hence, the quantity $|\mathcal{S}_{+,-}|$ was an overestimate for the real number of synapses falsely detected by the i3S, so that the actual PPV of the i3S must be considerably higher than 80%. Actually the expert measured the average PPV in the images to be 90%. This is demonstrated by two examples, one from the directly labeled set and one from the indirectly labeled set in figure 4.13. Figure 4.13 (a) and (c) show the i3S preprocessed sub-images of the input images (a: directly labeled,

	$\mathsf{SE}_\mu(\tilde{t}^s)$	$\mathsf{SE}_\mu(t^{(c)})$	$\mathsf{PPV}_\mu(t^{(c)})$
i3S$_{11}$	77 ± 6.24	82 ± 6.48	76 ± 6.85
i3S$_{12}$	76 ± 2.83	77 ± 6.13	78 ± 6.65
i3S$_{13}$	70 ± 8.18	68 ± 9.20	82 ± 1.70
i3S$_{14}$	78 ± 6.98	80 ± 7.41	76 ± 8.50
i3S$_{15}$	82 ± 3.86	83 ± 3.74	79 ± 4.97
i3S$_{16}$	83 ± 0.47	83 ± 3.40	78 ± 6.65
i3S$_{17}$	84 ± 3.77	86 ± 3.86	78 ± 2.83
i3S$_{18}$	86 ± 2.87	86 ± 4.99	78 ± 3.68

Table 4.6: The average SE and PPV obtained for each i3S$_{s'}$($s' = 11, \ldots, 18$) with an ER-tuned confidence threshold $\tilde{t}^s$ and a constant confidence threshold $t^{(c)}$. For each i3S$_{s'}$, the SE and PPV values were averaged over all images of one group (directly labeled ($s = 11, \ldots, 14$) or indirectly labeled ($s = 15, \ldots, 18$) which were not used for the i3S$_{s'}$ training, yielding SE$_\mu$ and PPV$_\mu$ in %. Additionally, the standard deviation is displayed. PPV$_\mu$ values are shown for the confidence threshold $t^{(c)}$ but no for $\tilde{t}^s$, as those were approximately 80% for all i3S$_{s'}$ (see eq. 4.45).

c: indirectly labeled), and figure 4.13 (b) and (d) highlight detected synapses by colored 5×5 pixel sized boxes. The same figure with non preprocessed images can be found in the appendix (figure A.1). Here, the box color encodes the agreement between ER and the i3S. Those synapses detected by both, i.e. belonging to $\mathcal{S}_{+,+}$, are highlighted in green. Synapses detected by i3S but not listed in ER (i.e. $\in \mathcal{S}_{+,-}$) are marked in orange and the vice versa cases $\mathcal{S}_{-,+}$ are marked in white. By reason of the computation of $\tilde{t}^s$ (see eq. 4.45), the PPV in these images is 80% so that 20% of the detected synapses ($\Lambda^s(\tilde{t}^s)$) belong to $\mathcal{S}_{+,-}$. But when having a closer look at these orange-marked synapses in figure 4.13(b) and (d), it can be observed that most of them actually show punctuate synapse like characteristics.

Figure 4.14(a) and (b) display the synapse detection in entire images of directly (a) and indirectly (b) labeled synapses. It can clearly be seen that synapses were detected across the entire image for both image types. As expected, fewer synapses were detected in the SP area (surrounding the perikaryal areas) than in the SR area, where synapses were detected across the whole field.

Application of a Constant Confidence Threshold

Calculation of ER-tuned confidence thresholds $\tilde{t}^s$ (see eq. 4.45) has the benefit that thresholds are found which result in high detection performances for the individual image. However, it requires the manual labeling of at least a sub-image of the whole image to be analyzed which is not feasible for an application in a setup with reasonable image throughput. Additionally, manually tuning a threshold t for each image is not reasonable. Therefore, a method has been proposed in this work for computing one confidence threshold $t^{(c)}$ (see eq. 4.46) which can be kept constant for all images of one group analyzed with the same i3S$_{s'}$.

Table 4.6, second and third column, shows the achieved PPV$_\mu(t^{(c)})$ and SE$_\mu(t^{(c)})$ values

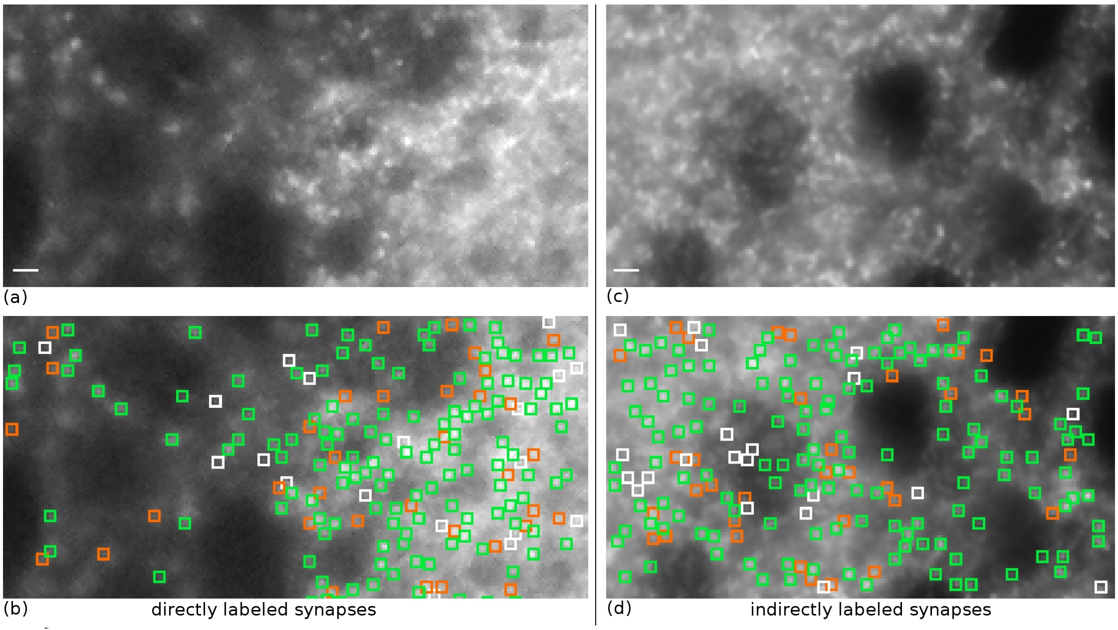
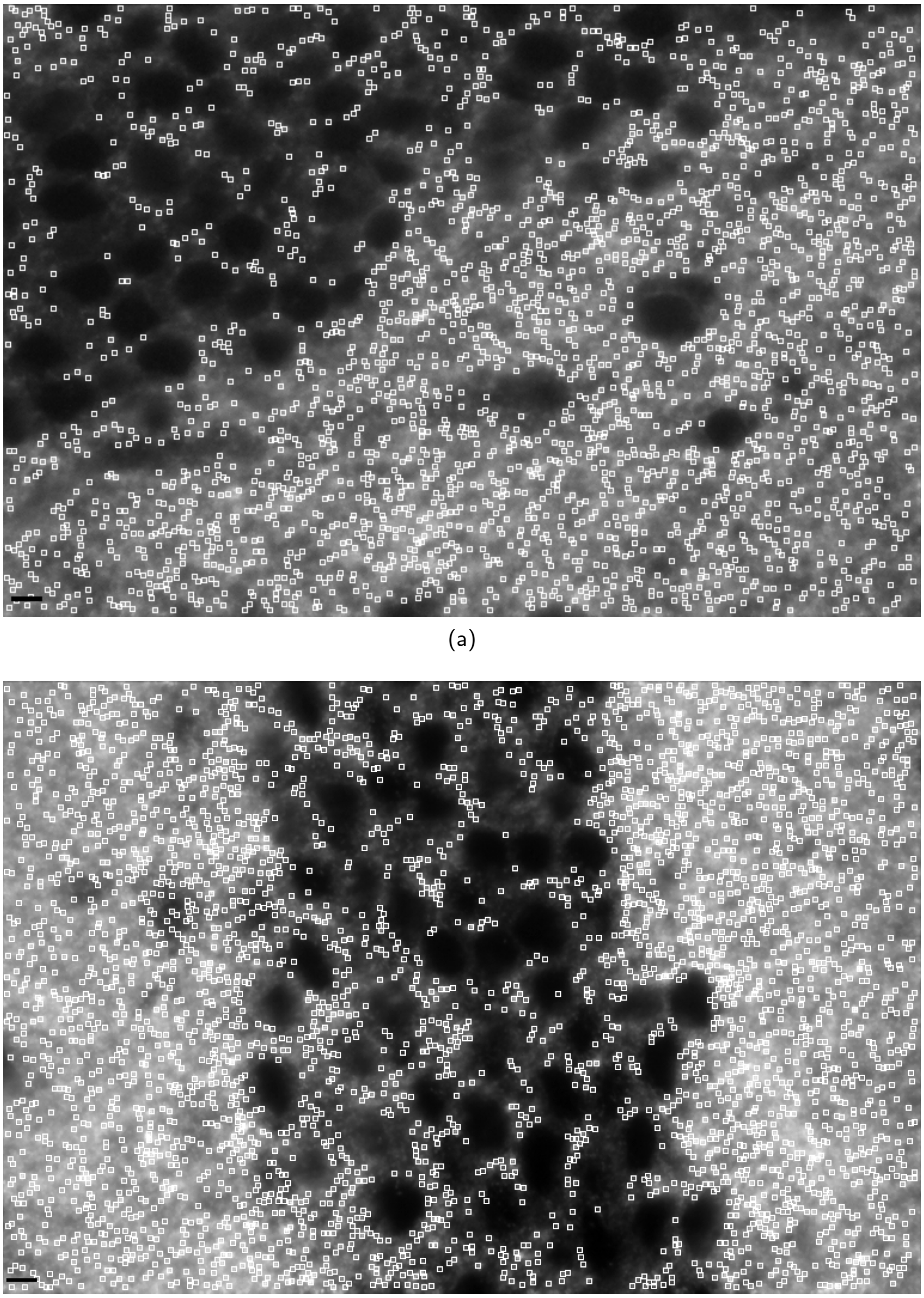


Figure 4.13: Synapse detection in directly (a, b) and indirectly (c,d) labeled tissue sections of the mouse hippocampus CA3 region compared to human expert labeling. The original images are shown in (a) and (c). Synapses detected either by the expert or the i3S are displayed on the original image in (b) and (d) with colored boxes (box size 5×5 pixels). Synapses detected both by i3S and the expert are marked with green boxes; Synapses only detected by i3S but not by the expert are denoted in orange; and synapses detected by the expert but not by i3S are shown in white boxes. An ER-tuned confidence threshold was applied on both images. Scale bars of 1.8μm refer to both images.

with their standard deviations for each i3S$_{s'}$ for the ER-tuned confidence threshold value. For each i3S$_{s'}$, the SE values were averaged over all images of one group (directly labeled ($s = 11, \ldots, 14$), or indirectly ($s = 15, \ldots, 18$)), which were not used for the corresponding i3S training. With respect to SE$_\mu(\tilde{t}^s)$, an increase in SE$_\mu$ was achieved while the PPV$_\mu$ slightly decreased. Again, slightly better performances were obtained for indirectly labeled synapses (last four rows), with an overall average SE$_\mu(t^{(c)})$ of 85%, an overall average PPV$_\mu(t^{(c)})$ of 78% and low standard deviations. For directly labeled synapses, an overall average SE$_\mu(t^{(c)})$ of 85% and an overall average PPV$_\mu(t^{(c)})$ of 78% could be obtained with higher standard deviations (first four rows table 4.6).

Depending of $t^{(c)} > \tilde{t}^s$ or $t^{(c)} < \tilde{t}^s$, obviously either a higher PPV or higher SE value was observed. In the case of $t^{(c)} < \tilde{t}^s$, a higher number of synapses was detected. Thus the question arises whether those additional detections are true synapse signals. Careful re-evaluation of those findings, i.e. belonging to the set $\mathcal{S}_{+,-}$, showed that most of those additional findings, caused by a lower $t^{(c)}$ are true synapse signals. This is exemplarily demonstrated in figure 4.15. The upper row displays results from an example image of directly labeled synapses. Corresponding results of an example image of indirectly labeled synapses are shown in the lower row. The preprocessed input images are shown in figure 4.15



(a)

(b)

Figure 4.14: Detection results for whole images of directly (a) and indirectly (b) labeled samples of size 768×512. Synapses detected by the i3S are marked with 5×5 pixel sized white boxes. Scale bars are 4.5μm.

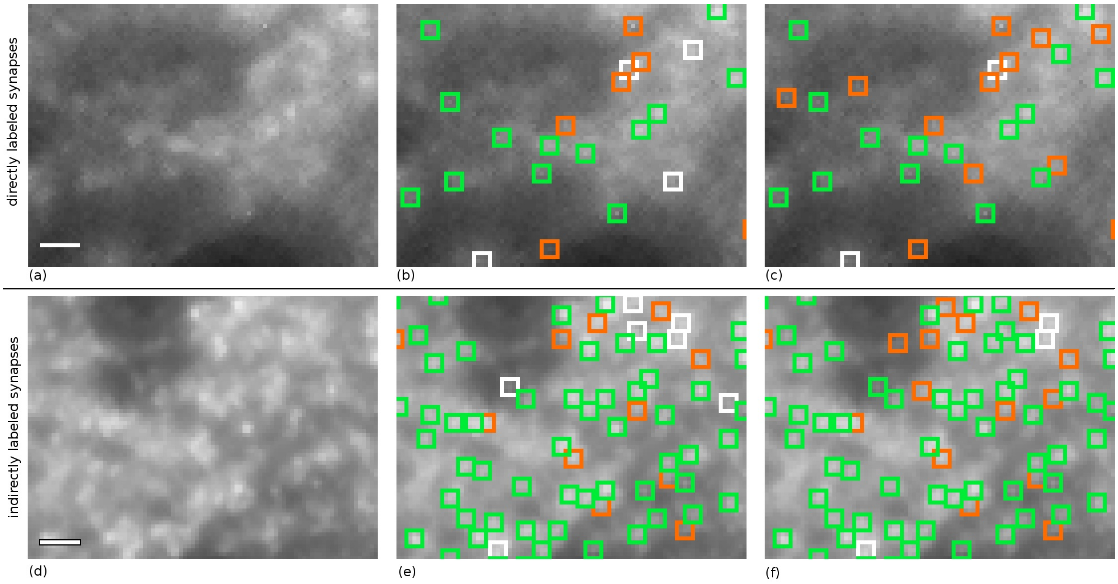


Figure 4.15: Detection results for directly (a,b,c) and indirectly (d,e,f) labeled synapses obtained by using the i3S approach compared to human expert markings. (a) and (d) show the original gray value images. (b,c) and (e,f) show the gray value image and detected synapses. Each punctuate synaptic signal is framed with a colored box (box size: 5×5 pixels). In (b) and (e) an ER-tuned confidence threshold $\tilde{t}^s$ (see eq. 4.45) was applied and compared to the human expert markings. The colored boxes encode the agreement or disagreement between the human expert markings and the i3S detection: Synapses detected by the i3S *and* the human expert are encoded in green. Synapses detected by the i3S but *not* by the expert are highlighted with orange boxes, and synapses not detected by the i3S *but* by the expert are marked in white. In (c) and (f) a constant confidence threshold $t^{(c)}$ (see eq. 4.46) was applied and compared to the same human expert markings as before using the same color code as in (b,e). Scale bars of 1.8μm refer to all images.

(a) and (d), the i3S detection result for $\tilde{t}^s$ are shown in figure 4.15(b) and (e) and figure 4.15 (c) and (f) displays the i3S results obtained for $t^{(c)}$ with green and orange boxes. The same figure with non preprocessed images can be found in supplementary figure A.2. The color encoding of synapses is identical to the one in figure 4.13. Again, most of the signals in $\mathcal{S}_{+,-}$ showed synapse characteristics, as can be seen in figure 4.15 (c) and (f), orange boxes. Therefore it can be concluded that these structures were correctly detected as synapses and that the PPV values of the i3S for a constant threshold $t^{(c)}$ were actually higher than those displayed in table 4.6.

Variation in Synapse Number for Different i3S$_{s'}$

It has been shown in section 4.6.1 that the choice of the training set can optimize the i3S for the detection of a certain synapse quality. Although in this study no quality label was assigned to the synapse positions in the expert references, it is still the question how strong

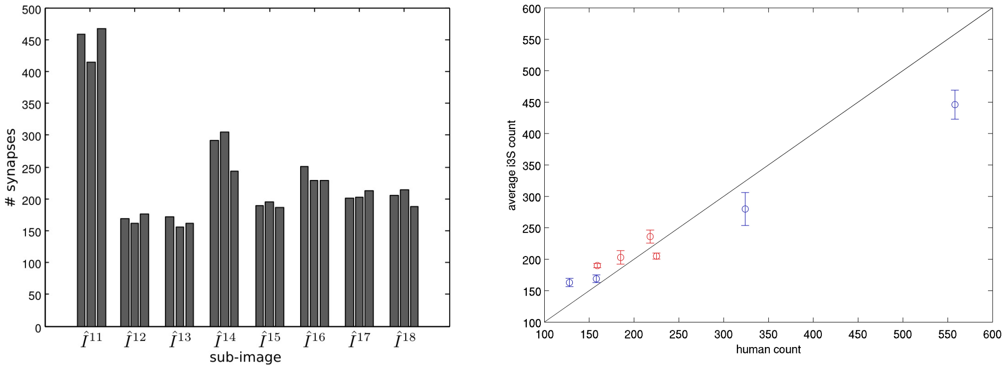
threshold	$\tilde{t}^s$		$t^{(c)}$	
i3S$_{s'}$ applied to sub-image or to whole image	sub-image	whole image	sub-image	whole image
I^{11}	503 ± 33.2	1987 ± 203.7	446 ± 23.2	1757 ± 158.0
I^{12}	167 ± 5.6	2175 ± 100.4	169 ± 6.1	2219 ± 93.7
I^{13}	113 ± 4.2	948 ± 36.0	163 ± 7.0	1339 ± 32.1
I^{14}	292 ± 30.9	2367 ± 165.2	280 ± 26.2	2317 ± 146.1
I^{15}	168 ± 3.8	2864 ± 103.8	190 ± 3.4	3246 ± 46.7
I^{16}	236 ± 6.6	3181 ± 293.5	236 ± 10.4	3274 ± 147.1
I^{17}	238 ± 4.0	3522 ± 24.1	205 ± 4.8	3034 ± 63.4
I^{18}	184 ± 5.4	2726 ± 205.6	203 ± 10.9	2996 ± 144.5

Table 4.7: For each image I^s and its sub-image $\hat{I}^s$ the average number of detected synapses and the standard deviation is computed from results obtained from different i3S$_{s'}$ with $I^s \neq I^{s'}$. For comparison, both confidence thresholds $\tilde{t}^s$ and $t^{(c)}$ were applied.

the selection of a training image I^s influences the detection outcome. Thus, in this section the stability of the i3S system is assessed, which is an important measure to prove the real world applicability of the system (cf. section 3.2.1). To this end, it was evaluated whether the number of detected synapses varies when different i3S$_{s'}$ are applied for synapse detection to the same image I^s (with $s \neq s'$). Table 4.7, first and second column, displays the average and standard deviation of the number of synapses detected with ER-tuned thresholds $\tilde{t}^s$ in the sub-images and whole images, respectively. For each individual image I^s, the values were averaged over all three i3S$_{s'}$ that were applied to it (with $s \neq s'$). For easier visual inspection, the synapse counts of each i3S$_{s'}$ and each sub-image $\hat{I}^s$ are additionally shown in a bar plot in figure 4.16 (a). On average, there was a slightly more stable detection performance for indirectly labeled images (i.e. $I^{15}, \ldots, I^{18}$), as indicated by the standard deviation in the first and second column of table 4.7. In addition, more synapses were detected in images with indirectly labeled synapses than in those with directly labeled synapses as can also be seen in figure 4.14 (a) and (b).

The third and fourth column of table 4.7 displays the average number of detected synapses in the sub-images and whole images for the application of $t^{(c)}$. It can be observed that for directly labeled synapses (first four rows) the application of the constant threshold had a positive effect and stabilized the number of detected synapses, i.e. it decreased the standard deviation. In the case of indirectly labeled synapses, mostly a slight increase of the standard deviation could be observed for the sub-image. However, a stabilizing effect was observed for the whole image (see last four rows of table 4.7).

Compared to the human expert labeling, the i3S as well as the human expert agreed quite well in their synapse count (cf. 4.5 and 4.6). Especially if ER-tuned confidence thresholds were applied, good agreement can be observed. For the constant confidence threshold $t^{(c)}$,



(a)

(b)

Figure 4.16: (a) Bar plot for the synapse counts obtained for each sub-image ($\hat{I}^{11}, \ldots, \hat{I}^{18}$) by three individual i3S$_{s'}$ with $s \neq s'$ of one group (directly or indirectly) and confidence threshold $t^{(c)}$. (b) Scatter plot of the number of synapses counted by the human expert in each sub-image ($\hat{I}^{11}, \ldots, \hat{I}^{18}$) against the average synapse number counted by the i3S and confidence threshold $t^{(c)}$. Synapse number was averaged for each image $\hat{I}^{s}$ over all i3S$_{s'}$ with $s \neq s'$ of one group (directly,blue, or indirectly,red). The standard deviation is shown for each count by error bars.

slightly weaker agreement were obtained. Figure 4.16 (b) displays a scatter plot of the number of synapses counted by the human expert in each sub-image ($\hat{I}^{11}, \ldots, \hat{I}^{18}$) against the average synapse number counted by the i3S. Synapse number was averaged for each image $\hat{I}^{s}$ over all i3S$_{s'}$ with $s \neq s'$ of one group (directly or indirectly). The standard deviation is shown as error bars.

4.6.3 i3S Performance in a Multi Protein Detection Setup

So far, the influence of the training set as well as the chosen confidence threshold has been evaluated in an intra- as well as inter-image detection setup. Especially the inter-image detection performance with constant thresholds is of great interest as only good performances in this setup allows for a real world application of the system. However, images were always labeled for the same protein. As it is of great interest to reduce the amount of user interaction to a minimum, it would be of great benefit if a i3S trained on, for example, an image stained for `syphys` could also be applied to the detection of synapses in images stained for `synap` or `nmdr1` which are chosen complementary to `syphys` to cover all synapses (cf. section 2.2.2). Furthermore, the number of required training labels should be considerably low, so that not too much time has to be spent for training set generation.

To test the performance of the i3S in a real world setup, three i3S$_{s'}$ were trained on images stained for `syphys`, each of a different image stack $\mathbf{I}^{s'} \in \mathcal{I}$. As it would be most likely the case in a routine setup, the training sets were constructed by choosing synapses of good quality sampled from the whole image. This is motivated by the fact that choosing high quality synapses for training also allows to detect lower quality synapses by setting an

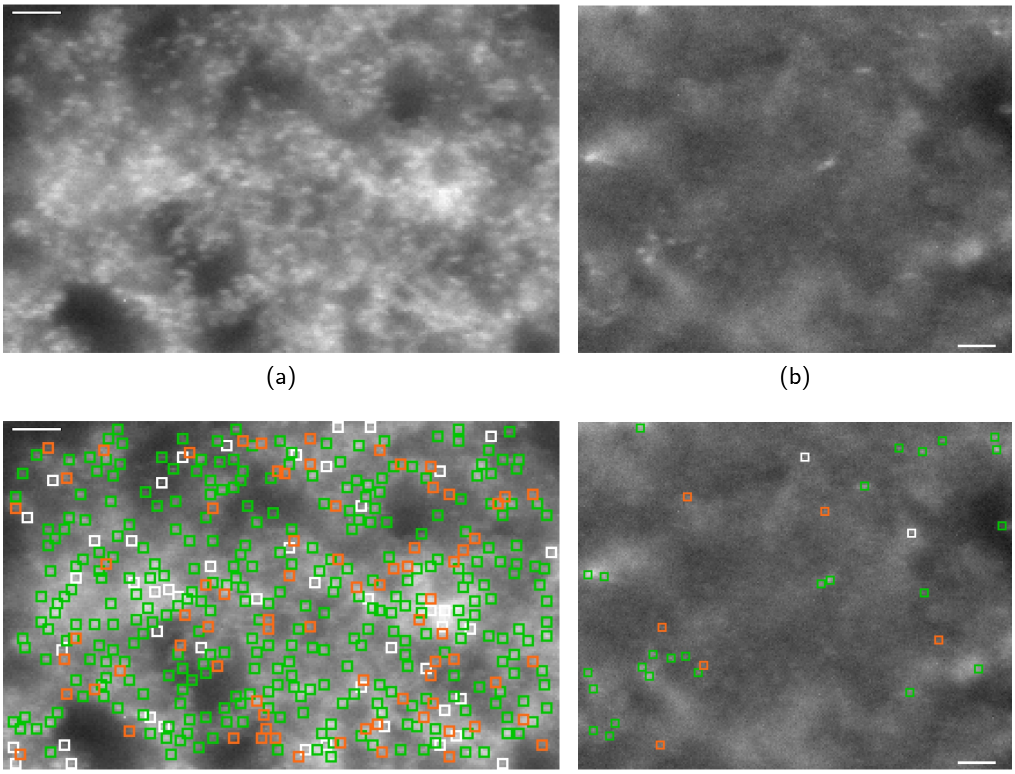
		$\text{SE}_\mu(\tilde{t}_n^s)$	$\text{SE}_\mu(t^{(c)})$	$\text{PPV}_\mu(t^{(c)})$
i3S_1	$\hat{I}_{\texttt{nmdr1}}$	84 ± 4.95	83 ± 7.78	79 ± 4.95
	$\hat{I}_{\texttt{synap}}$	89 ± 5.66	89 ± 5.66	80 ± 1.41
i3S_2	$\hat{I}_{\texttt{nmdr1}}$	84 ± 1.41	85 ± 1.41	81 ± 3.54
	$\hat{I}_{\texttt{synap}}$	88 ± 6.63	88 ± 6.36	81 ± 0.71
i3S_3	$\hat{I}_{\texttt{nmdr1}}$	83 ± 5.66	85 ± 0.00	78 ± 4.24
	$\hat{I}_{\texttt{synap}}$	84 ± 2.12	86 ± 9.90	62 ± 28.28

Table 4.8: Performance of individual $\text{i3S}_{s'}$ trained on images stained for `syphys` and applied to detect synapses in images of `synap` and `nmdr1` of two other image stacks. The mean SE and PPV value as well as the standard deviation is given. SE and PPV values were averaged over both detection results of `nmdr1` and `synap`. Mean SE (SE_μ) are given both for the ER-tuned threshold t_n^s and the constant threshold $t^{(c)}$. Mean PPV (PPV_μ) are only given for the constant thresholds as they are approximately 80% for each ER-tuned threshold.

appropriate confidence threshold (see section 4.6.1). Furthermore, choosing synapses from all across the image will capture varying synapse characteristics, in for example the SR and SP area. Relatively small training sets were constructed with on average 83 positions in $\text{ER}^{s'}$. Expert references were obtained by manual labeling of all synapses in a sub-image with a size of minimum 200×100 size in $\hat{I}^s_{\texttt{nmdr1}}$ and $\hat{I}^s_{\texttt{synap}}$ for each stack. The detection performances of each $\text{i3S}_{s'}$ was tested on sub-images stained for `synap` (referred to as $\hat{I}^s_{\texttt{synap}}$) and `nmdr1` (referred to as $\hat{I}^s_{\texttt{nmdr1}}$) extracted from the two additional image stacks $\mathbf{I}^s$ with $s \neq s'$.

Table 4.8 (first column) displays the obtained mean SE values obtained for each image with the ER-tuned thresholds. The SE values were averaged individually for $\hat{I}^s_{\texttt{synap}}$ and $\hat{I}^s_{\texttt{nmdr1}}$ over both detection results obtained with the two $\text{i3S}_{s'}$ with $s \neq s'$. PPV values are not shown as they are approximately 80% for each setup. It can be observed that quite high detection performances with reasonable standard deviations were achieved for both, the images stained for `synap` and the images stained for `nmdr1`. This can be possibly ascribed to the fact that synapses in images stained for `synap` and `nmdr1` mostly show synapses with high contrast, easing the detection process. Figure 4.17 exemplarily shows one detection result for the `nmdr1` and `synap` channel.

In general, it is not feasible to set an ER-tuned threshold for each new image encountered. This would require a human expert labeling for at least a sub-image of the image which would be too labor-intensive (see discussion in section 4.5.5). Therefore, a constant threshold was applied as has also been done in the previous section. To this end, constant thresholds were estimated separately for the `nmdr1` and `synap` channel for each $\text{i3S}_{s'}$. Table 4.8, second and third column, displays the obtained SE_μ and PPV_μ. Again, high SE_μ and PPV_μ values are obtained, with one exception. The i3S_3 applied to the images stained for `synap` achieves a lower $\text{PPV}_\mu(t^{(c)})$ with higher standard deviations (see last row of table 4.8). This low performance is caused by one $\hat{I}_{\texttt{synap}}$ showing only 28 synapses where $t^{(c)} < \tilde{t}_n^s$. Due to this low



(a)

(b)

(c)

(d)

Figure 4.17: Synapse detection results obtained for an image stained for `nmdr1` (a,c) and `synap` (b,d). The original images, corrected for hotspots and rescaled to 8bit intensity for visualization purpose are displayed in (a) and (b). The detection result obtained by an i3S trained on an image labeled for `syphys` are shown in (c) and (d). Synapses correctly detected by the i3S are highlighted with a green 5×5 pixel sized box. Synapses which were detected by the i3S but not contained in the ER are highlighted in orange. In the vice versa case, synapses contained in the ER but not detected by the i3S, are displayed in white. Scale bars are 4.3μm.

number of available synapses, only few additional synapses in $\mathcal{S}_{+,-}$ lead to a large decrease of $\mathrm{PPV}_{\mu}(t^{(c)})$. Overall, however, one can conclude that applying a constant threshold provides reasonable detection results even if i3S are applied which were trained on images labeled for a different protein.

4.6.4 Discussion

In this case study, the applicability of SVM-based synapse detection to fluorescently labeled brain tissue micrographs has been shown. Detection performances similar to that of an average human expert could be obtained, even if images were labeled for different proteins. Thereby, the i3S system allows for a high throughput screening of synapses and studies on a new statistical level which would not be possible by manual human counting. Automated detection of several thousand synapses in an area of 768×512 pixels took approximately 20 seconds. Thereby, the software overcomes the present limitation of a rapid and accurate detection of synapses in fluorescence images obtained by indirect or direct labeling. It could be shown that an expert is able to mark a limited number of synapses. However, an expert-based evaluation of an entire visual field, containing more than 1000 fluorescently labeled and densely packed synapses, was found to be impossible, because the number of mistakes increases dramatically over time. The automated analysis, however, features the benefit of being fast and 100% stable, i.e. multiple analyses with the same i3S setup results in the same detection outcome (see supplementary table A.2).

In the design of the i3S architecture it was intended to keep the number of adjustable parameters as low as possible. A low number of parameters, which are adjustable even for non-computer experts, is the prerequisite for the usability of automated image processing in bioimage labs. It could be shown that already few training items are sufficient to detect synapses in new images. The findings in section 4.6.1 indicate that it is even sufficient to label synapses of good quality to set up a suitable training set and steer the amount of synapses detected and their quality by the chosen threshold t. This eases the generation of training sets, as high quality synapses are easier to label. As it is not possible to compute an ER-tuned threshold $\tilde{t}_n^s$ for each new image encountered, a protocol was introduced which allows to compute a constant confidence threshold $t^{(c)}$. This threshold can be obtained from a set of hand labeled images $\{I_n^s\}$ and corresponding tuned confidence thresholds $\{\tilde{t}_n^s\}$. These thresholds were tuned so that a minimum PPV($\tilde{t}_n^s$) was reached. In this study, the minimum PPV($\tilde{t}_n^s$) was set to 80%. This was driven by the observation that the expert references ER were spoiled by a considerable number of missed synapses. As a consequence, the PPV($\tilde{t}_n^s$) is considerably underestimated. However, the minimum PPV($\tilde{t}_n^s$) is a value which can be adjusted by non experts if necessary in a straight forward way. Depending on the biological background considered, the sensitivity may be of particular of even smaller importance. Hence, a very conservative minimum PPV($\tilde{t}_n^s$) of 95% could also be applied. It could be shown that quite similar detection performances are achieved with the constant threshold compared to ER-tuned thresholds. It even stabilized the number of detected synapses for images of directly immunolabeled samples.

The i3S was developed to process microscopy images from tissue sections, so the subject matter of out-of-focus-signals must be discussed. If for example a ×40 water objective (numerical aperture 0.9) is used as for the acquisition of image set $\mathcal{I}'$, the depth of focus (depth of sharpness) in the 5μm thick cryo-tissue sections is approximately 0.5μm (= 500nm). For synaptic signals, imaged within this field, it is generally observed that similar synaptic signals outside this optical plane (above or below) are much lower in light intensity and moreover are out of focus. Nevertheless such out-of-focus signals are observable as grayish low level noise. If not too low thresholds are chosen, the i3S captures only those synaptic signals which are clear cut features of a light intense "mountain" with a decrease in signal intensity in the immediate area around that mountain (in the x/y direction). Thus, the i3S generally detects only synapses in the focus area, and ignores noisy synapses, which are out of focus above and below the corresponding focus area. This indicates that the measurements are adjusted to synapses in a focus depth of 500nm, while other synapses above and below that level (hence in both directions of the optic axis) are ignored. Chemical synapses have an average diameter of approximately 200-30nm. Hence, if two synapses are (i) immediately adjacent in the z-axis and (ii) located precisely within the focal depth area of the optical system, the approach will miss this information. It will count only one instead of two synapses. But in practice this error can be neglected, given the enormous number of synapses measured in a given visual field. In principle, it is upon to the discretion of the user to calculate an appropriate correction factor. For example one may consider that every synaptic signal inside the neuropil of the brain, captured with our above described optical parameters, should in fact be the result of two synapses (precisely superimposed within the optical plane), the number of synapses counted by the i3S will have to be doubled for every single image. However, comparative studies (e.g. comparing Alzheimers disease cases vs. normal brain areas) working with the standardized parameters described in this work, will clearly show significant differences in quantities because of the enormous numbers of detected synapses in every image, despite a possible optical resolution limit in the z-axis of the optical system.

Finally, the question was raised if reasonable detection results could be achieved for different labeling methods (direct and indirect). The comparison of detection results obtained from expert-based and i3S based analyses led to the conclusion that the difference in detection performance of the i3S between directly and indirectly labeled synapses (see table 4.6) can mainly be ascribed to the different ER qualities. Images of indirectly labeled synapses featured a higher contrast, which seemed to ease labeling for the human experts resulting in more consistent ER. In contrast, directly labeled synapses are of lower contrast so that it was more difficult for the expert to keep up a stable synapse detection performance. This phenomenon was observed when comparing synapses detected only by the i3S ($\mathcal{S}_{+,-}$) with synapses detected solely by the expert ($\mathcal{S}_{-,+}$). Surprisingly, some synapses of $\mathcal{S}_{+,-}$ featured clearer synapse signals. Thus, the question arises why those synapses were not marked by the human expert. One potential explanation can be the fact that the human visual system is obviously not suited for a stable synapse detection performance in fluorescence gray value images. It is a known fact that only a limited number of gray values, around 60-90, can be distinguished by the human visual system (Ware (2004)). Especially in low contrast images, this fact makes it difficult for the expert to perform a stable synapse detection. Although the expert appeared to "apply" thresholds in his visual assessment on different image characteristics, like intensity, local contrast, size or texture, she/he was not able to adjust these thresholds in a reproducible way to the image context. This phenomenon has also been discussed in section 4.2 and might be a reason for the contradictory results for similar synapse patterns.

4.7 Case study II: Pancreas Tissue

In this section, the application of the i3S to nucleus detection in images of pancreatic tissue samples will be shown. To this end, cell nuclei are detected in images stained for DAPI (see section 2.2.1). The DAPI images were preprocessed by converting the false color index images to 8bit gray value images. Images were bilateral filtered and normalized to the full

8bit intensity range. Filter parameters were manually fitted on a set of sample images and kept constant for all images.

A training set Γ was obtained by manual labeling of 59 positive and 59 negative sample positions by a human expert in one DAPI stained nucleus image. The neighborhood size for the feature vector computation was set to $M = 19$. Training set generation then followed the process outlined in section 4.5.2. To reduce the dimensionality of the feature vectors $\mathbf{x}^{(i)} \in \Gamma$, a *Principal Component Analysis* (PCA) (Pearson (1901); Hotelling (1933); Karhunen (1946)) projection onto the first eight eigenvectors was performed, resulting in eight-dimensional projected feature vectors $\tilde{\mathbf{x}}^{(i)}$. This dimensionality reduction, which was not performed for the synapse data, allowed for fewer training items and a faster computation. Due to the "curse of dimensionality", for higher dimensional data sets more training items are required. A more detailed introduction to the principles of PCA projection can be found in section 6.4.1. Training of the i3S was carried out with a Gaussian kernel. Parameters were automatically optimized using a ten fold cross validation strategy.

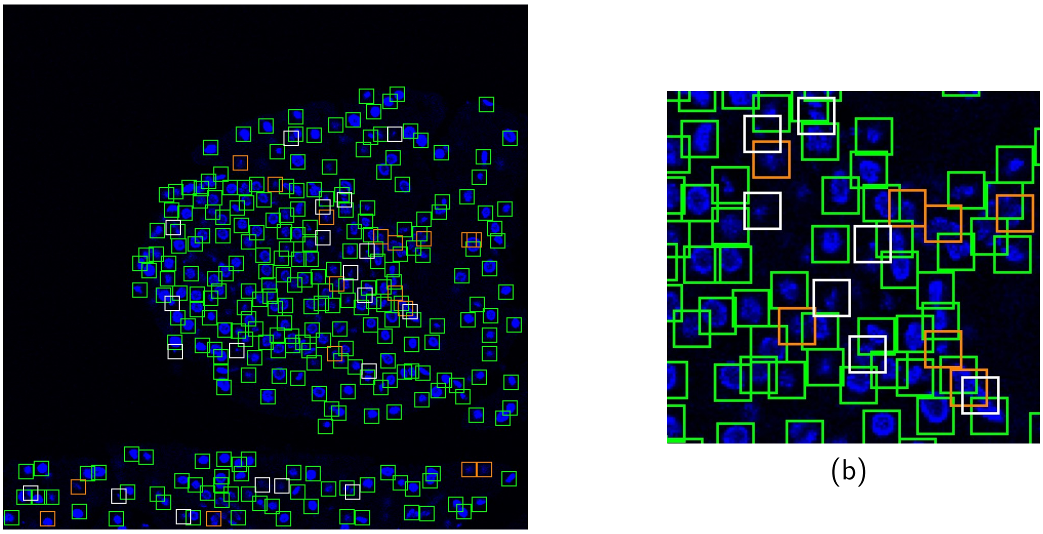
The performance of the i3S for nucleus detection was evaluated on two DAPI stained test images I_n^A and I_n^B of the pancreas data set which were not used for i3S training. To obtain expert references which can be used for the evaluation of the detection performance, an exhaustive labeling of all nuclei was carried out on these images. Thereby, expert references ER_n^A and ER_n^B were obtained.

4.7.1 Results

The performance of the i3S in nucleus detection was evaluated on the two test images I_n^A and I_n^B, which were DAPI stained for nucleus highlighting. A total of 283 and 517 nuclei were detected by the i3S in sample I_n^A and I_n^B, respectively. With respect to the expert references ER_n^A and ER_n^B, a SE_μ of 91% and a PPV_μ of 96% was achieved, averaged over both detection results. In both cases, a confidence threshold of $t = 0$ always lead to the best detection performance. Figure 4.18 presents the detection result for sample I_n^A overlaid on top of the original nucleus image. Here, nuclei belonging to the set $\mathcal{S}_{+,+}$ are highlighted with green boxes, those belonging to $\mathcal{S}_{+,-}$ are displayed in orange and nuclei of $\mathcal{S}_{-,+}$ are shown in white. Most of the nuclei were correctly detected, including overlapping and non circular shaped nuclei. FN detections were often associated to nuclei within nucleus accumulations or very faint and small nuclei as displayed in figure 4.18 (b). False positive detections were obtained for nuclei with lower intensities, therefore possibly missed by the expert. Furthermore, if nuclei show a very pronounced doughnut structure, with almost no intensity in the middle, the i3S often detects more than one nucleus.

4.7.2 Discussion

In this section, the application of the i3S to nucleus detection has been demonstrated. Reasonable detection performances could be achieved with only slight modifications of the i3S. For both images, a confidence threshold of $t = 0$ achieved best detection results, i.e. all patches classified as nuclei centers were included in the detection result. Nuclei of different sizes and shapes as well as overlapping nuclei were well detected. Nucleus detection in the



(a)

(b)

Figure 4.18: Cell nucleus detection result for a whole image (a) and an enlarged sub-image (b). Nuclei detected by both the i3S and the human expert are displayed in green, nuclei detected solely by the i3S in orange and nuclei not detected by the i3S but by the expert in white.

whole image could be achieved in around 20 seconds on an Intel Core 2 Duo CPU with 3 GHz and 2GB RAM.

As only one trained i3S is applied to two previously unseen images, these results are not intended to be a full study of i3S performance on nucleus detection. It, however, gives an impression of the applicability of the i3S to other microscopy object detection problems.

4.8 Summary

In this chapter a learning-based image processing software, termed i3S, has been introduced which enables the investigator to perform high throughput screening of synapses in the brain. The findings indicate that studies will be possible quantifying synapses in thousands of tissue sections on a new statistical level which would not be feasible by manual labeling.

Given the present lack of algorithms for automatic quantification of synapses which do not require a considerable amount of human interaction and/or special imaging setups (for example Micheva and Smith (2007); Silver and Stryker (2000)) the proposed approach is the first which can provide a fast and good approximation to the neurobiological reality with a minimum of human interaction and based on standard fluorescence images. As the system is 100% stable when applied to the same image, it furthermore does not suffer from variability which is always introduced when human observers need to fine tune the detection pipeline.

To show the applicability of the i3S to object detection tasks in different image domains, the i3S was additionally applied to nucleus detection in pancreas tissue samples. In this setup, reasonable detection performances could be achieved as well.

With the proposed system, it is possible to efficiently filter high-content image data on a morphological level to obtain a reduction of the data complexity. Furthermore, through the object detection process, a semantic annotation of the image is obtained and object specific multivariate image features can be extracted.

Chapter 5

A Direct Visual Data Mining Tool for Three-Channel High-Content Micrographs

In this chapter, the potential of combining image processing and visual data mining concepts for an object-based analysis of multivariate image data will be explored. Therefore, the low-dimensional pancreas data set is used which consists only of three image channels. This features the great benefit that images can directly be explored by visual analysis which would not be possible for higher dimensional data sets. Thus, the applied image processing and visualization concepts can be evaluated with respect to their performance in high-content image analysis. It is evident that visualizations which can be used for such a low-dimensional data set can not directly be transfered and used for the analysis of d-dimensional TIS data but more sophisticated visualizations are required. However, experiences of the low-dimensional data domain can also be considered for the design of approaches tailored to the need of high-dimensional TIS data exploration.

I will introduce a first strategy for the interactive, visual analysis of multivariate images for cell type classification and cell counting in pancreas tissue samples. In a first step, a semantic annotation is carried out in order to locate biological significant objects, i.e. cell nuclei. Second, an information visualization approach is used to interactively explore and classify the multichannel features extracted for each object. Here, the feature domain as well as the image domain can simultaneously be analyzed to classify α- and β-cells. This classification allows to calculate image characteristics such as the number of α- and β-cells, the percentage of α- and β-cells with respect to all cells, and the α- to β-cell ratio. This ratio is relevant for diabetes studies and testing of new anti-diabetes treatments (see section 2.2.1).

Since the principle of nucleus detection has already been described in section 4.7, this chapter will focus on the process of feature extraction and interactive, visual data exploration. The performance of the proposed system is evaluated on a set of test images. The whole analysis pipeline has been realized as a software tool termed *PancreasAnalyzer* which is heavily used by my cooperation partners at the University of Warwick.

5.1 Image Preprocessing

As images are provided as false colored index images, they are first converted to 8bit gray value images. Second, each image is filtered by a 3×3 median filter for noise reduction. For comparison, bilateral filtering (Tomasi and Manduchi (1998)) was also tested. However, the

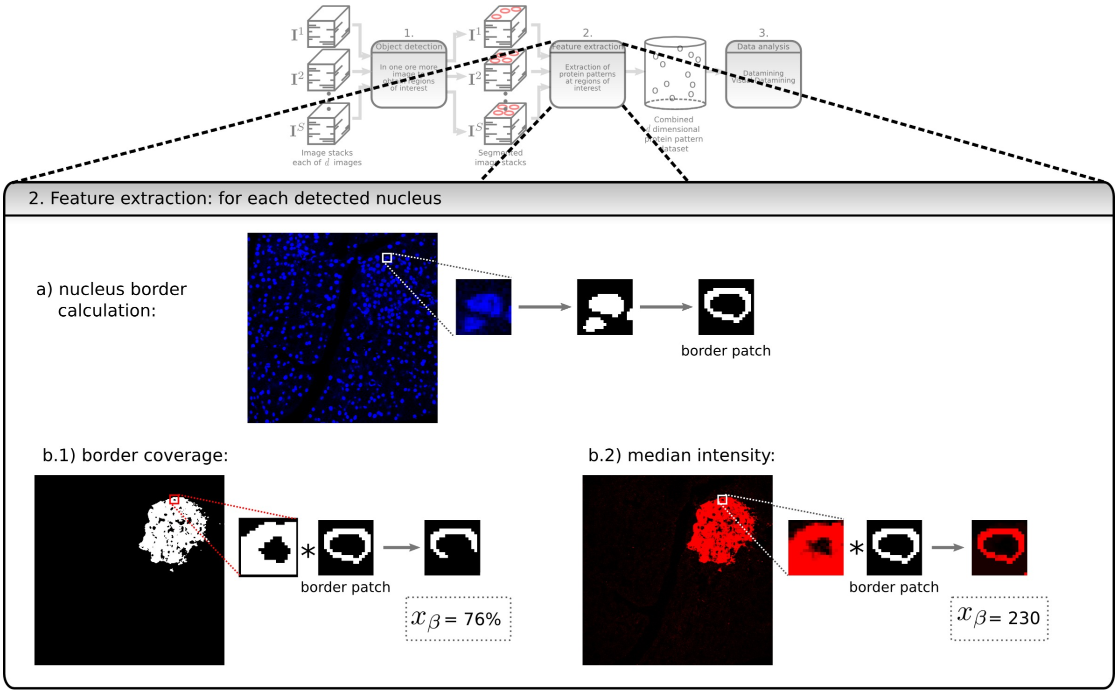

$\mathbf{I}^1$
$\mathbf{I}^2$
$\mathbf{I}^S$
Image stacks each of d images
1.
Object detection
In one ore more image to obtain regions of interest
$\mathbf{I}^1$
$\mathbf{I}^2$
$\mathbf{I}^S$
Segmented image stacks
2.
Feature extraction
Extraction of protein patterns at regions of interest
Combined d dimensional protein pattern dataset
3.
Data analysis
Datamining Visual Datamining
2. Feature extraction: for each detected nucleus
a) nucleus border calculation:
border patch
b.1) border coverage:
*
border patch
x_β= 76%
b.2) median intensity:
*
border patch
x_β= 230

Figure 5.1: Extraction of object-specific multi channel features for cell type classification in pancreas tissue data. (a) For each nucleus detected in the image through the strategy described in chapter 4, a nucleus border is extracted by Otsu thresholding, and morphological and logical operations. Two strategies can then be applied to extract object-specific multichannel features x_α and x_β. (b.1) Each protein channel is separately thresholded via Otsu thresholding (only the insulin channel is shown). A border coverage can then be calculated, defined as the percentage of border pixels extracted in step (a) which are object pixels in the thresholded protein image. (b.2) For each protein channel a median intensity can be calculated based on the pixels associated to the border of the nucleus.

slight gain in accuracy did not justify the introduction of two additional parameters. In this study, it is not required to compare the extracted feature vectors between stacks, as each stack is interpreted and explored separately. Hence, an image normalization step was not necessary.

5.2 Feature Extraction

Extraction of object-specific features is modeled according to manual human evaluation of such images for cell type identification and cell counting. The human expert analyzes the glucagon and insulin staining characteristics around each nucleus in an RGB overlay image (as depicted in figure 2.3 in chapter 2) to decide whether it is an α- or β-cell. This procedure was considered in the design of an object-specific multichannel feature vector.

In the first step of the feature extraction, the DAPI stained image was applied for nucleus detection as described in chapter 4. Based on this semantic image annotation, for each detected nucleus position $\mathbf{p}$, a nucleus border is extracted to obtain ROIs. To this end, a $M-2 \times M-2$ image patch (M corresponds to the maximum object size see chapter 4), centered at $\mathbf{p}$, is extracted. The patch is subsequently binarized via Otsu thresholding which allows for an automated setting of the threshold without human interaction (Otsu (1979)). If additional nuclei partially extend in the extracted image patch, only that connected component, i.e. thresholded nucleus, is used for further studies and kept in the binary image which contains the center pixel $\mathbf{p}$. The nucleus border is then obtained by a two fold dilation of the thresholded image with a binary, 3×3 structured element, and subsequent subtraction of the original thresholded image from the dilated (Gonzalez and Woods (2002)). Figure 5.1 (a) shows the border extraction for one sample nucleus position. To obtain thicker nucleus borders, a larger number of dilations could be carried out consecutively. This parameter can be specified by the user in the *PancreasAnalyzer* if required. Throughout this study, however, it was continuously set to a two fold dilation, as it has proven to be an appropriate setup in the experimental studies. Only minor differences in cell type classification were obtained if the number of dilations was varied between one and three.

Based on the extracted nucleus borders, object-specific features are calculated based on the two additional channels, stained for glucagon and insulin. In this study, two different feature extraction strategies are applied. For the first strategy, the insulin and glucagon image channels are classified into background and object pixels via Otsu thresholding. For each channel, the percentage of nucleus border pixels (*border coverage* (BC)) is calculated which are also object pixels in the thresholded channel as depicted in figure 5.1 (b.1). This strategy holds the advantage that the extracted feature corresponds to the user's visual interpretation of the images, where a nucleus has to be surrounded by stained cytoplasm to a minimum amount in order to be classified as an α- or β-cell. Thresholding the protein channels additionally provides information about the percentage of α- and β-cell mass with respect to the whole islet cell mass, a measure relevant for diabetes study. To obtain the whole islet mass, the thresholded insulin and glucagon channels are merged and all object pixels are counted. However, calculating the BC always requires a binarization of the images, which can lead to non desirable results, for example if the staining is non perfect. Therefore, additionally a second strategy is presented which does not require a thresholding of the images. Here, a *median intensity* (MI) is calculated for each channel based on the pixels associated to the nucleus border as depicted in figure 5.1 (b.2). Both of the feature extraction methods, BC and MI, produce two-dimensional feature vectors $\mathbf{x} = (x_\alpha, x_\beta)$ for each nucleus. Based on these feature vectors, cell nuclei can be classified as α-, β or non-islet cell nuclei. If the MI or BC, depending on the chosen feature extraction strategy, exceeds a user defined threshold, i.e. $x_\alpha > t_\alpha$ or $x_\beta > t_\beta$, the nucleus is classified as α- or β-cell nucleus, respectively. Any nucleus neither classified as α- nor as β-cell nucleus is classified as non-islet cell nucleus. It is obvious that if $x_\alpha > t_\alpha$ and $x_\beta > t_\beta$, the nucleus is classified both as α- and β-cell nucleus. Here, it should be up to the visual impression of the biological expert to decide to which class this nucleus shall be assigned to. To this end, the PancreasAnalyzer provides an interactive manipulation of the classification outcome. However, it is additionally reasonable to perform an automated decision if it is known, for example, that one protein channel is more dominant than the other.

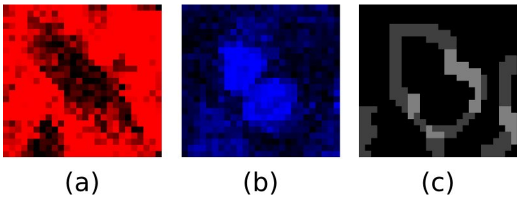


Figure 5.2: Effect of overlapping nuclei. Sub-image of (a) insulin channel, (b) nucleus channel and (c) extracted nucleus border. Overlapping nuclei can result in connected low intensity regions in the protein channel (a).

In general, other segmentation strategies for nucleus border extraction could be applied, as for example active contours (Hu et al. (2004)) or watershed segmentation (Wählby et al. (2004)). The benefit of these approaches is that each nucleus, even overlapping ones if the method is properly initialized, has a closed, continuous border which is not the case for the method proposed in this work. However, usually a large set of parameters (for example five in Wählby et al. (2004) and 11 in Hu et al. (2004)) have to be specified to obtain good results. Yet, in this work a closed contour is not crucial for nucleus type classification, but might even be disadvantageous. The presence of a nucleus can result in low intensity regions in the protein channels in many cases as can be seen in figure 5.2 (a). This does not affect the feature calculation if only a single nucleus is present as the border will mostly not reach into these low intensity regions. However, if nuclei overlap, the extracted features would be biased by these low intensities if closed contours are extracted. The extracted border of the nucleus would span across the low intensity regions which would result in lower border coverages or lower median intensities. Hence, partial borders are sufficient for this type of study and the proposed, straight forward but efficient, segmentation strategy can be applied. Figure 5.2 (c) shows the nucleus borders extracted by the proposed method for the nuclei shown in figure 5.2 (b). Pixels with higher intensity indicate that borders extracted for the nuclei overlap. This information could be used to find an appropriate separation point if required, but has not been applied for this study.

5.3 Interactive Visual Data Exploration and Cell Type Classification

For an efficient multivariate image interpretation, appropriate visualization techniques which allow to simultaneously analyze the feature as well as the image domain are crucial, as has been discussed in detail in chapter 3. Data exploration in the image domain is especially important as biological experts are very familiar in analyzing data in this domain. Even more important, the spatial information in the image domain represents the special advantage of imaging in system biology complementary to non-spatial techniques in standard proteomics as has been outlined in chapter 1.

To analyze multiple variables in different domains, the field of information visualization (Ware (2004); Spence (2007)) has proposed several methods. One powerful and often used

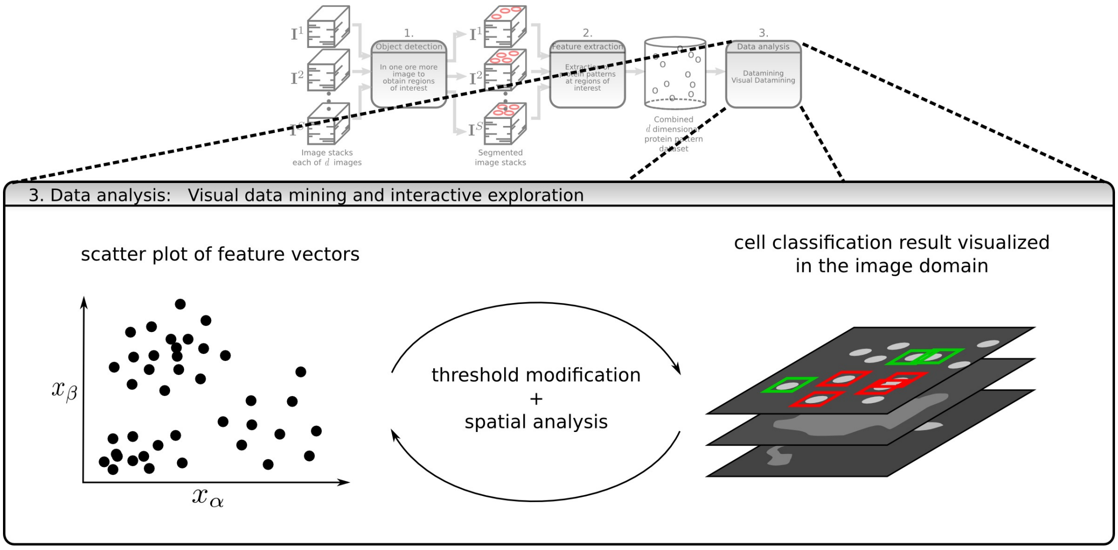
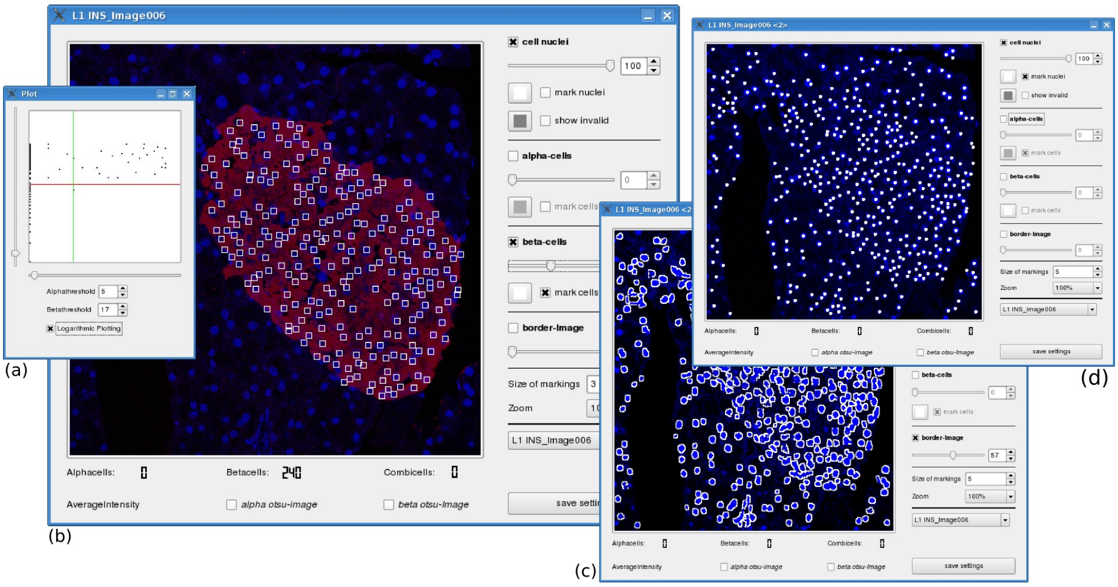

$\mathbf{I}^1$
$\mathbf{I}^2$
$\mathbf{I}^S$
Image stacks each of d images
1.
Object detection
In one ore more image to obtain regions of interest
$\mathbf{I}^1$
$\mathbf{I}^2$
$\mathbf{I}^S$
Segmented image stacks
2.
Feature extraction
Extraction of protein patterns at regions of interest
Combined d dimensional protein pattern dataset
3.
Data analysis
Datamining Visual Datamining
3. Data analysis: Visual data mining and interactive exploration
scatter plot of feature vectors
x_β
x_α
threshold modification + spatial analysis
cell classification result visualized in the image domain

Figure 5.3: Schematic illustration of the interactive visual data exploration for cell classification. Both the feature and the image domain are visualized simultaneously to allow for efficient data interpretation. The feature domain is visualized as a scatter plot where thresholds can be set interactively. In the linked image domain visualization, cell type classification results are highlighted accordingly with colored rectangles.

technique is the concept of *link & brush*. In a link & brush data exploration, the data is displayed in at least two different displays. In an interactive fashion, the user can select a sub set of the data in one display (brushing), which is automatically highlighted in the other display(s) (linking). This strategy allows the user to explore even complex data sets in an highly interactive manner.

Throughout this study, two channels in addition to the channel for semantic annotation are regarded. Therefore, it is sufficient to visualize the two-dimensional cellular feature vectors (x_α, x_β) in a basic 2D scatter plot, as shown in figure 5.3 (left) or figure 5.4 (a). In the PancreasAnalyzer software, the user can choose between linear or logarithmic axis scaling and two sliders can be used to set thresholds t_α and t_β. Each adaptation of a threshold triggers an update in the linked image domain visualization. All cells with feature values above the threshold are highlighted with different colored rectangles (see figure 5.3) and the number of detected cells is displayed. For convenience, α- and β-cell highlighting can be turned on and off to allow to focus on one cell type at a time. Furthermore, individual channels or overlays of multiple channels can be selected as a background image. In order to allow for an insight in the feature calculation, all detected nuclei can be highlighted (figure 5.4 (d)), nucleus borders can be superimposed (figure 5.4 (c)) and thresholded protein channels can be displayed. Furthermore, the size and coloring of the cell markings can be adapted individually and, if required, classification results can manually be corrected.



Plot

Alphathreshold 5

Betathreshold 17

☒ Logarithmic Plotting

L1 INS_Image006

☒ cell nuclei 100

□ mark nuclei

□ show invalid

□ alpha-cells 0

□ mark cells

☒ beta-cells

☒ mark cells

□ border-Image

Size of markings 3

Zoom 10

L1 INS_Image006

Alphacells: 0 Betacells: 240 Combicells: 0

AverageIntensity □ alpha otsu-image □ beta otsu-Image

L1 INS_Image006 <2

□ beta-cells 0

☒ mark cells

☒ border-Image 57

Size of markings 5

Zoom 100%

L1 INS_Image006

Alphacells: 0 Betacells: 0 Combicells: 0

AverageIntensity □ alpha otsu-image □ beta otsu-Image

L1 INS_Image006 <2>

☒ cell nuclei 100

☒ mark nuclei

□ show invalid

□ alpha-cells 0

☒ mark cells

□ beta-cells 0

□ mark cells

□ border-Image 0

Size of markings 5

Zoom 100%

L1 INS_Image006

Alphacells: 0 Betacells: 0 Combicells: 0

AverageIntensity □ alpha otsu-image □ beta otsu-Image

Figure 5.4: Screenshots of the PancreasAnalyzer software. (a) Scatter plot visualization, which is used for interactive setting of thresholds. The display mode can be set to logarithmic or linear scaling. (b) Image display showing cells classified as β-cells with a white marking on top of an overlay of DAPI and insulin channels. The number of detected cells is shown at the bottom of the display. The size and color of the markings can be adapted individually. (c) Overlay of the DAPI channel and the computed nucleus borders. (d) The nucleus detection result can be displayed. All detected nuclei are highlighted with white circles overlaid on top of the DAPI channel.

5.4 Results

In this section, I will present the cell classification and cell counting results obtained for four pancreas tissue image stacks, referred to as $\mathbf{I}^{A1}, \mathbf{I}^{A2}, \mathbf{I}^{B1}$ and $\mathbf{I}^{B2}$. For the sections referred to as A, c-Myc was switched off. c-Myc was switched on inducing β-cell death for sections referred to with B (see section 2.2.1). To obtain a gold standard for the α- and β-cell classification, image stacks were manually evaluated three times by a biological expert working on these images in daily laboratory work. The average of the α- and β-cell count can be used as an expert reference. For each human count, additionally the time required for manual analysis was recorded.

For the semi-automated analysis, the maximum object size M required for nucleus detection and feature extraction was set to $M = 19$ (see section 5.2). Nucleus detection was carried out as described in chapter 4. Figure 5.5 (a) and (b) shows the extracted boundaries for sample $\mathbf{I}^{A1}$. Nucleus boundaries were properly extracted especially for separated nuclei. Overlapping nuclei mostly featured slightly overlapping boundaries at their intersection point, as displayed in figure 5.5 (b) with a white color.

For each image set analyzed, the BC as well as the MI feature extraction strategy was applied. For each feature, a threshold was first roughly estimated based on the scatter plot

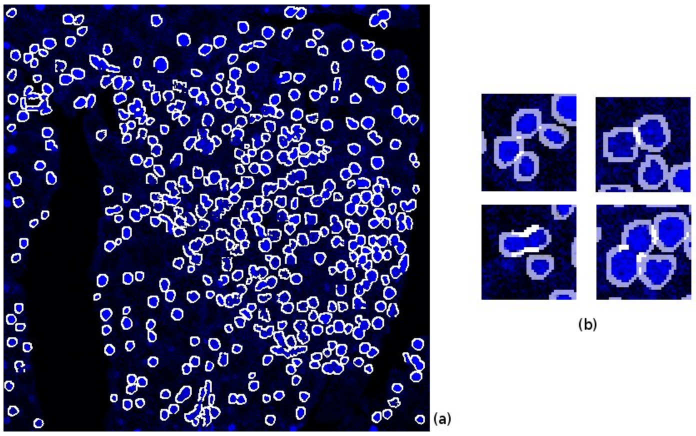


Figure 5.5: Extracted nuclei borders, shown in white in the DAPI image of sample $\mathbf{I}^{A1}$. Image patches on the right (b) show sub-images displaying overlapping nuclei. Intersection of the borders are visible through a white color.

visualization. Through the link & brush principle, thresholds could then conveniently be fine tuned. Figure 5.6 (a) and (d) displays the scatter plot visualization for image sets $\mathbf{I}^{A1}$ and $\mathbf{I}^{B2}$ for the MI and BC features, respectively. Thresholds were set to five and 17 for the glucagon and insulin channel for set $\mathbf{I}^{A1}$ and to 32 and 46 for the glucagon and insulin channel in image set $\mathbf{I}^{B2}$. The resulting cell classification for α- and β-cells is shown in figure 5.6 (b) and (c) for set $\mathbf{I}^{A1}$ and in (e) and (f) for set $\mathbf{I}^{B2}$. Regarding the thresholds set, it can be observed that they vary between images $\mathbf{I}^{A1}$ and $\mathbf{I}^{B2}$. However, this fact is not prejudicial as no automated threshold setting procedure is intended but images are manually explored. With the proposed strategy, the effect of a set threshold can be visually inspected and adapted directly.

Table 5.1 displays the number of nuclei classified as α- and β-cell nuclei in all four samples for both feature extraction strategies, BC and MI, as well as the three counts of the human observer (H1-H3). Besides the number of classified cells, also the α- to β-cell ratio for each analysis was computed, as this is a valuable information for diabetes study. For obvious reasons, the ratio is computed more accurately for larger number of cells. As table 5.1 shows, the cell counts obtained by the proposed semi-automated approach agree very well with the human experts counts. Slightly better results were obtained through the MI approach compared to the BC approach. However, there was a mismatch in the counting of the β-cell nuclei for samples $\mathbf{I}^{B1}$ and $\mathbf{I}^{B2}$ for both feature approaches MI and BC. Yet, the values displayed in table 5.1 are not corrected for cells counted both as α- and β-cells as this requires knowledge about the staining characteristics of the different channels and should be decided based on the visual impression of the expert. Hence, the counts presented are slightly biased. If the double counted cells are solely assigned to the α-cell class, agreement between the β-cell counts of the semi-automated analysis and the manual evaluation is higher. This suggests that the presented results even underestimate the capabilities of the system.

	α-cells					β-cells					α / β		
	MI	BC	H1	H2	H3	MI	BC	H1	H2	H3	m	MI	BC
$\mathbf{I}^{A1}$	26	50	26	25	26	240	243	286	286	272	0.09	0.11	0.21
$\mathbf{I}^{A2}$	11	18	8	8	8	82	78	68	69	67	0.11	0.13	0.26
$\mathbf{I}^{B1}$	14	15	10	10	9	22	20	9	9	9	1.11	0.64	0.75
$\mathbf{I}^{B2}$	4	4	3	3	5	11	8	6	6	6	0.5	0.36	0.5

Table 5.1: Number of nuclei classified as α- and β-cell nuclei using the proposed software system with both feature approaches (MI = median intensity, BC = border coverage) and obtained by three manual human counts (H1, H2, H3) are displayed for all four samples $\mathbf{I}^{A1}$,$\mathbf{I}^{A2}$, $\mathbf{I}^{B1}$ and $\mathbf{I}^{B2}$. In the last columns, the ratios of α/β-quantities are given (m = human count averages).

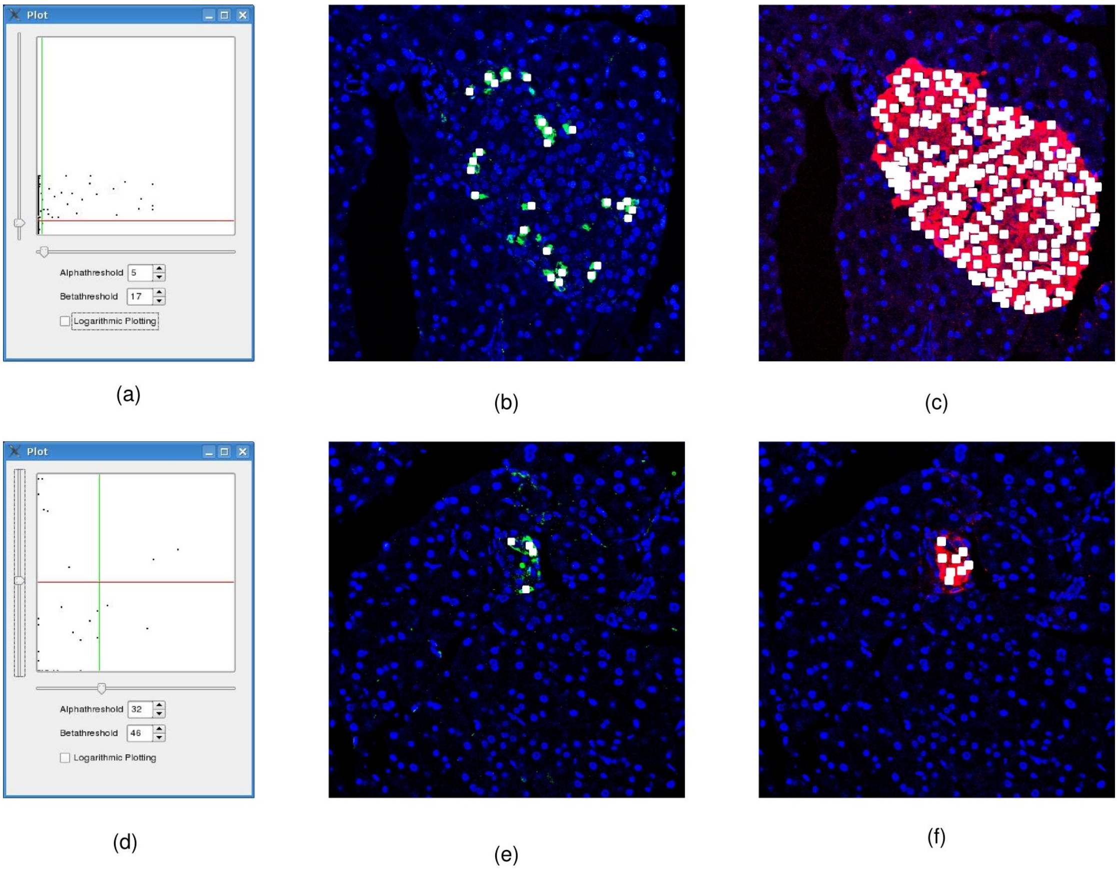
However, also without this correction, in the light of (a) the gain in evaluation speed and (b) the fact that the rank of the β-cell quantities is preserved, the mismatch is acceptable. The time a human expert needs to fully evaluate one image depends on the number of visible α- or β-cell signals. To evaluate image $\mathbf{I}^{A1}$, an expert needs on average 5 minutes, whereas evaluation of images with only few nuclei can be performed faster, for example $\mathbf{I}^{B2}$ with only 3 α- and 6 β-cells in around 30s. The proposed semi-automatic evaluation only requires around 20 seconds for nucleus detection and approximately 50 seconds were required for the evaluation of an image, including feature calculation. As the nucleus detection and the calculation of object features can be performed sequentially for multiple sets of images without human interaction once images are loaded in the system, the actual evaluation can be carried out in few seconds.

With the BC approach, it was additionally possible to obtain information about the cell masses. Therefore, the thresholded β- and α-cell images are evaluated. For image stacks $\mathbf{I}^{A1}$ and $\mathbf{I}^{A2}$, α-cell masses of 5% and 6% were obtained, respectively (rounded to full integer value). For samples $\mathbf{I}^{B1}$ and $\mathbf{I}^{B2}$, higher α-cell mass, with 40% and 30% was observed. Thus, for the samples where c-Myc was switched off, the β-cell death is reflected both by a decreasing number of β-cells as well as by a decrease in the β-cell mass with respect to the whole islet mass.

5.5 Discussion

In this chapter, a system has been proposed for the interactive, visual exploration of high-content image data tailored to the needs of pancreas cell type classification. Based on a preceding automated semantic image annotation, object-specific multichannel feature vectors were extracted. Concepts from the field of information visualization were subsequently applied to provide an interactive exploration tool.

The proposed system allows for a rapid extraction of quantitative information almost irrespective of the number of cells in the image. Semantic annotation, i.e. nucleus detection, and feature calculation can be achieved within a few seconds without the requirement of human interaction. Also the interactive analysis of the multichannel information can be solved



Alphathreshold 5
Betathreshold 17
□ Logarithmic Plotting

Alphathreshold 32
Betathreshold 46
□ Logarithmic Plotting

(a) (b) (c)

(d) (e) (f)

Figure 5.6: Interactive feature exploration: In a scatter plot visualization of the extracted feature vectors $\mathbf{x}$ (x-axis: x_α, y-axis: x_β), the user can select the threshold for the α-channel and the $\beta-$channel (see windows (a) and (d) on the left). Those cells with a x_α value above threshold t_α are marked simultaneously in the image display as α-cells in the green channel (see (b) and (e)). Those cells with a x_β above the threshold t_β are marked in the other image display as $\beta-$cell (see (c) and (f)). The top row (a)-(c) shows results obtained for sample $\mathbf{I}^{A1}$, the bottom row (d)-(f) shows results obtained for sample $\mathbf{I}^{B2}$.

within seconds. However, the time requirement for the interactive cell type classification always depends on how much fine tuning of the thresholds is required. Compared to the interactive setup, the disadvantage of manual cell counting in a high-throughput application is obvious. In a standard laboratory setup, a three-fold or four-fold counting has to be carried out for each image to account for intra-observer variability. As the time required to evaluate an image scales with the number of cells to be analyzed, such a multiple counting leads to long evaluation times especially for large islets. It has to be considered that the spatial attention of a human observer rapidly declines, and thus labeling can only be performed for a short time if reliable results are to be obtained (Jagoe et al. (1991)). With the proposed interactive analysis, a larger number of images can be evaluated in this time period.

Another advantage of the proposed method is that only very few parameters are required for the whole analysis setup, which I believe is the prerequisite for an application which is supposed to be used by non computer experts. As the nucleus detection is an inherent part of the analysis system, additionally the parameters required for this step have to be considered. Hence, for nucleus identification the object size M needs to be specified, which is directly linked to object characteristics. Further required parameters are derived from this object size. Additionally the PCA target dimension need to be determined, which can easily be deduced from a visualization of the eigenvalues (see chapter 4). Once an i3S has been trained, no further parameters are required to detect nuclei on a new image, provided that images are taken under stable conditions. For feature calculation, only one additional parameter, the width of the nucleus border, can optionally be specified. Experimental studies have shown, however, that the border width does not considerably affect the cell counting outcome and it is suggested to set it to the default value of two dilations. Hence, only two parameters are required for nucleus detection and for the calculation of object specific features. The process of threshold setting is guided by the visualizations in the image as well as feature domain and by the principles of link & brush. Customized image visualizations can be generated by overlays of different image channels and, for example, the classification results easing the visual analysis. Furthermore, a manual re-modification of the results is possible in case cells are wrongly marked or missed.

In this work it could be shown that results similar to a manual expert classification were obtained by applying the proposed semantic-based semi-automatic cell type classification approach. In daily laboratory work, this strategy is heavily used and the results are well comparable to the human expert classifications (Zhou and Epstein (2009)). The cell type classification obtained with the proposed method allows the user to compute diabetes relevant information as for example the α- to β-cell ratio. One may argue that a segmentation-based analysis of the image may not be necessary to extract such ratios since it could be sufficient to compute the cell mass in the insulin and glucagon channel. The simplest way to estimate this mass would be to apply a threshold to both images and compute the mass based on the resulting binarization. However, the cell mass can vary considerably due to disease, the individual cell states and their position related to the optical plane. Thus, analyzing only cell masses would not be sufficient but the number of cells has to be regarded as has also been done in other works such as Avril et al. (2002) or Huang et al. (2009). With the border coverage and the median intensity, two features are provided which well represent the characteristics of the analyzed cells and hence allow for a cell-based analysis of the data. The feature extraction step is crucial for the cell-based study. Solely regarding the image intensities of each channel at each pixel location does not allow to distinguish between different cell types as shown in Herold et al. (2009). Besides providing information of the α- to β-cell ratio, the border coverage feature additionally allows to obtain information about cell type specific cell masses. However, it has to be regarded that thresholding is required for this computation which can sometimes lead to unwanted results especially if staining is not perfect.

Besides implementing a tool for diabetes study, the main focus of this work was to explore how to incorporate visual data mining concepts in the field of high-content bioimage analysis. Already quite basic image processing and visualization strategies were sufficient to provide valuable information for diabetes study, showing the potential of combining image processing and visual data mining concepts for multivariate image analysis. Especially linking visualizations of both, the feature and the image domain, has proven to be a useful concept aiding the user in image analysis. The low-dimensional feature set allowed to directly compare the obtained results with manual evaluations, which would not be possible for high-dimensional data sets. The mentioned interactive analysis strategy, however, could also be applied to image sets with a larger number of channels, i.e. labeled proteins. To this end, different data processing and visualization methods for d-dimensional data sets, as scatter plot matrices or dimensionality reduction methods, would need to be applied. Strategies for the evaluation of high-dimensional multivariate image data will be presented in the following chapter using the example of synaptic protein colocation.

Chapter 6

An Unsupervised Learning Approach for Data Mining d-dimensional Fluorescence Images

In the previous chapter, the potential of combining image processing and information visualization concepts for the analysis of low-dimensional multichannel micrographs has been shown. Especially interactively linking visualizations in the image domain with visualizations in the feature domain has proven to be of great value in this analysis. I will now focus on how to provide suitable strategies for the evaluation of high-dimensional TIS data for colocation analysis at synapses. To this end, several characteristics of the TIS data have to be considered to set up a suitable analysis strategy. First, the dimensionality of the extracted feature vectors is now highly increased. Hence, the basic scatter plot visualization, which was sufficient for the analysis of low-dimensional feature vectors, needs to be replaced by more sophisticated visualizations tailored to the needs of high-dimensional protein colocation analysis. Furthermore, unlike for the pancreas data set, feature vectors now need to be comparable within and between image stacks. Spatial ordering of protein colocation patterns as well as those patterns which separate between, for example, treated and non treated samples can then be identified. Additionally, when analyzing multiple image stacks at a time, the number of protein patterns which needs to be explored significantly increases. Even if tuned visualizations are provided for the feature and image domain, an exclusive visual interpretation of the whole data set is not feasible. Therefore, clustering is incorporated in the analysis process to reveal hidden regularities and structures and obtain a further reduction of the data complexity. Since no a priori knowledge is available about which protein patterns can be expected and in how many different groups those patterns separate, such an unsupervised learning approach is well suited.

In the following, I will introduce the proposed feature extraction strategy for protein colocation analysis at synapses, as well as the clustering algorithms applied throughout this study. Subsequently, visualization strategies for the feature as well as the image domain are presented which aid the user in exploring the complex domain of TIS image data. Following the Shneiderman mantra (see section 3.2.3), overviews as well as in detail visualizations are provided.

6.1 Feature Calculation

A fundamental aspect in analyzing synapse protein profiles in images of brain tissue is the extraction of meaningful feature vectors. First, the feature vector has to reflect the underlying protein characteristic of a synapse in each channel. Second, in order to find groupings and regularities hidden in the data, feature vectors extracted at different spatial locations or from different image stacks have to be comparable.

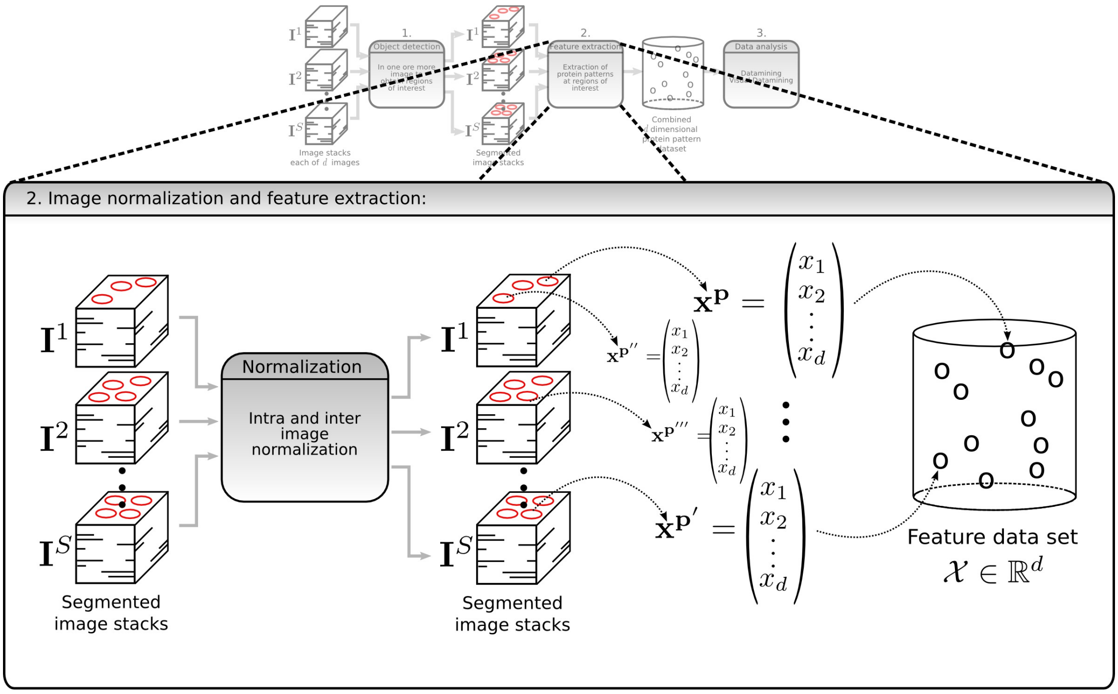
A direct extraction of the image intensities to obtain feature vectors of synapses is not feasible. Images of one stack can show considerable variation in their intensities between channels and between the same channel of different stacks (see for example figure 6.2 (a) vs (c)). Additionally, there often exists a considerable amount of background intensity variation within one image. A specific gray value intensity can be a signal in one region of the image but background in another. Hence, feature vectors based solely on pixel intensities would represent the intensity variation in the images rather then the synapse specific protein pattern. Thus, a strategy is required which adapts signals within and between image stacks to each other and considers the variation of signals within one image in the feature extraction process.

In the following, the steps necessary for feature extraction, as described in figure 6.1 are presented. First, inter- and intra-image normalization is carried out, which allows to subsequently extract feature vectors at each synapse position obtained through object detection as described in chapter 4. A feature data set $\mathcal{X}$ is obtained which holds the feature vectors of one or more image stacks.

6.1.1 Image Enhancement and Normalization

Due to hotspots caused by broken CCD elements, which occur in each of the recorded images of a stack, first of all hotspot elimination is carried out as described in section 4.5.1. Additionally, bilateral filtering was applied for noise reduction (see section 4.5.1). Subsequently, image normalization within and between stacks is performed. In section 4.5.1, a normalization strategy in the context of SVM based synapse detection has already been introduced. There, images of different samples labeled with a stable synaptic marker were very well adapted to each other and allowed to achieve good detection results. Now, however, a total of 22 different images has to be normalized within and especially between stacks which can show quite varying intensities especially between stacks. Figure 6.2 (a) and (c) show the nnos channel of two different image stacks, normalized to 0-255 gray values, as well as their gray value histogram for 16bit and 8bit. One can see that figure 6.2 (c) features few very high signals resulting in a low gray value range for the remaining tissue region when images are normalized to 0-255. Figure 6.2 (a) lacks these high intensities and thus the whole image is much brighter when rescaled to 0-255. Thus, histograms are poorly normalized to each other if solely a rescaling to 8bit is performed (see the right histograms of figure 6.2 (a) and (c)). Also the 16bit histograms are not well comparable (see the left histograms of figure 6.2 (a) and (c)). To enhance weaker signals, a $\tanh$-function was applied to the image. The new intensity value $\hat{f}(\mathbf{p})$ of a pixel $\mathbf{p}$ of image I_n^s is calculated as

$$\hat{f}(\mathbf{p}) = \tanh(1/(2 \cdot \mathrm{mean}(I_n^s)) \cdot f(\mathbf{p})) \; . \tag{6.1}$$



$\mathbf{I}^1$
$\mathbf{I}^2$
$\mathbf{I}^S$
Image stacks
each of d images
1.
Object detection
In one ore more image to obtain regions of interest
Segmented image stacks
2.
Feature extraction
Extraction of protein patterns at regions of interest
Combined d dimensional protein pattern dataset
3.
Data analysis
Datamining
Visual Datamining
2. Image normalization and feature extraction:
Segmented image stacks
Normalization
Intra and inter image normalization
$\mathbf{x}^{\mathbf{p}} = (x_1, x_2, \ldots, x_d)$
$\mathbf{x}^{\mathbf{p}''} = (x_1, x_2, \ldots, x_d)$
$\mathbf{x}^{\mathbf{p}'''} = (x_1, x_2, \ldots, x_d)$
$\mathbf{x}^{\mathbf{p}'} = (x_1, x_2, \ldots, x_d)$
Feature data set
$\mathcal{X} \in \mathbb{R}^d$

Figure 6.1: Short overview of the steps required for feature extraction at synaptic sites. First, images are normalized so that extracted features are comparable within and between stacks. Next, at each synapse position obtained via i3S classification (see chapter 4), a synapse specific feature vector is extracted. A feature data set $\mathcal{X}$ is obtained, holding the feature vectors of one or more image stacks $\mathbf{I}^s$.

In a subsequent step, the intensity values are scaled to the 8bit intensity range of 0-255. Normalized versions of figure 6.2 (a) and (c) are shown in figure 6.2 (b) and (d). It can be seen that the intensity ranges are better adapted to each other by enhancing weaker signals. Also the results of other normalizations as the z-normalization and histogram equalization were compared. However, with the $\tanh$ the most reasonable analysis results were obtained.

6.1.2 Synapse Specific Feature Calculation

As has been discussed, extracting solely the intensity value at a pixel to obtain a feature vector would reflect the intensity variation within an image rather than the synapse specific protein characteristics. To correct for intensity variations within an image, methods from the field of retrospective shading correction could be applied (Tomaževič et al. (2002)). However, as images of tissue sections are analyzed which show two highly varying morphological structures (SR and SP), this can be quite challenging. Furthermore, synaptic signals are very small and might get lost through the process of image correction. I therefore developed a strategy which takes into account the local background characteristic for feature calculation. Hence, a feature which represents the signal to background difference instead of the exact intensity value is obtained. This feature calculation is highly tuned for synapse signals which show

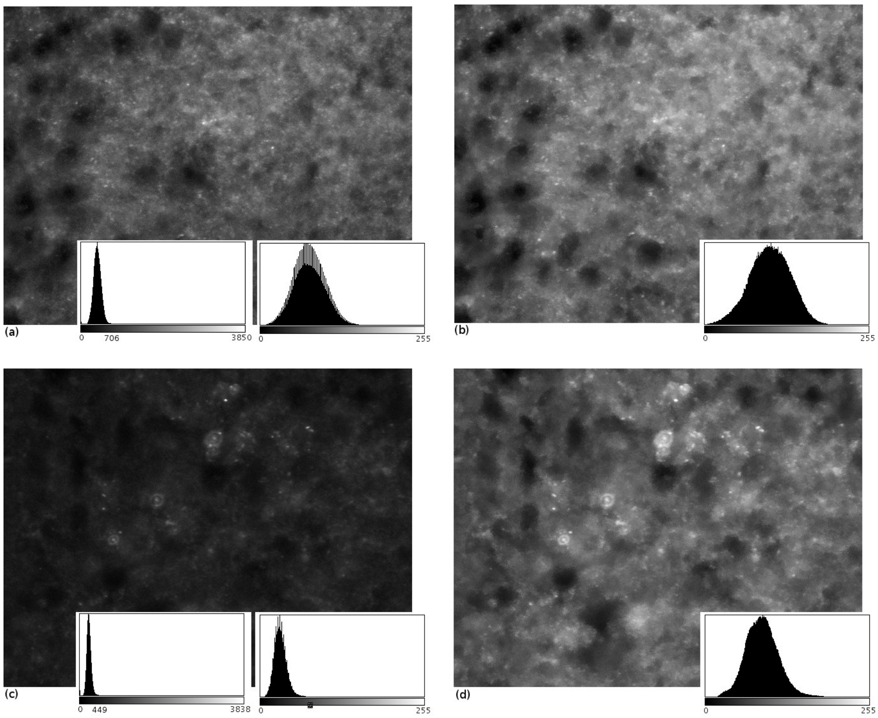


Figure 6.2: Comparison of normalization strategies for the $\mathtt{nnos}$ channel of image set $\mathbf{I}^1$ (a,b) and $\mathbf{I}^5$ (c,d). The original, hotspot corrected image is shown in (a) and (c) linearly rescaled to 8 bit for visualization purpose. The histogram of the 16bit and 8bit are shown as inlays to the image with the 16bit histogram left and 8bit histogram right. It is evident that a rescaling solely to 8 bit normalizes the images poorly to each other. In (b) and (d) the results of the proposed normalization is shown with the obtained histogram as an inlay. Although the histograms do not perfectly match, intensities are now better comparable across the images.

spot like structures.

For each synapse position $\mathbf{p} \in \Lambda^s$ and each image I_n^s $(n = 1, \ldots, d)$, the following steps are carried out to obtain feature vectors $\mathbf{x}^\mathbf{p} = (x_1^\mathbf{p}, x_2^\mathbf{p}, \ldots, x_d^\mathbf{p})^T$ with one feature for each channel. The set Λ^s represents all synapse positions detected for image stack $\mathbf{I}^s$, either by synapse detection in one or few images of the stack. Figure 6.3 schematically shows the feature calculation for one position $\mathbf{p}$ and one channel of the image.

1. The maximum intensity value within a 3×3 neighborhood of $\mathbf{p}$ is determined and becomes the new center point $\mathbf{p}^*$. This step is necessary as, despite the image registration, a shift of ± 1 pixel can still be present. By searching the maximum value, the center is shifted so that it represents the synapse center in the corresponding channel (see figure 6.3 (b)).

2. As a synapse appears as a 3×3 object, the intensity value of $\mathbf{p}^*$ is filtered by a 3×3 Gaussian filter mask. The Gaussian filtered intensity value $\hat{f}(\mathbf{p}^*)$ is used as the signal value (see figure 6.3 (b)).

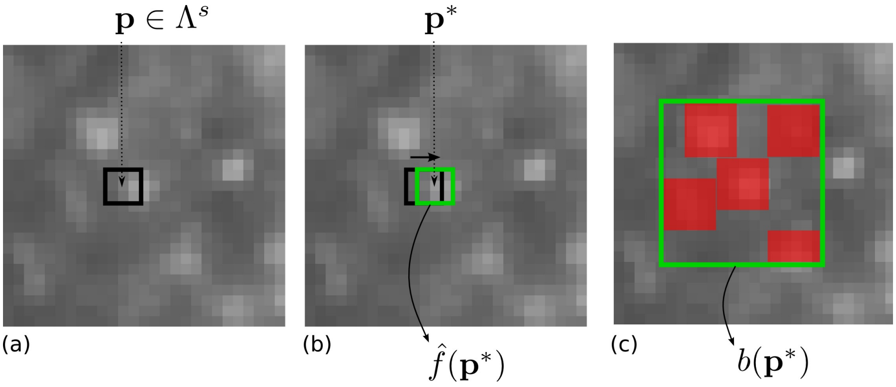


Figure 6.3: Schematic display of the synapse specific feature extraction. (a) For each synapse position $\mathbf{p} \in \Lambda^s$ the maximum intensity value within a 3×3 neighborhood around $\mathbf{p}$ is determined and becomes the new center point $\mathbf{p}^*$. (b) A 3×3 Gaussian filtering is carried out for $\mathbf{p}^*$ to obtain the filtered intensity value $\hat{f}(\mathbf{p}^*)$. (c) The local background intensity $b(\mathbf{p}^*)$ is calculated as the median intensity of a 15×15 neighborhood centered at $\mathbf{p}^*$. Pixels $\mathbf{p}'$ are omitted which satisfy $d(\mathbf{p}', \mathbf{p}) \leq 2 \, \forall \, \mathbf{p} \in \Lambda^s$, here depicted with a red colored overlay. The feature value is calculated as $x_n^{\mathbf{p}} = \hat{f}(\mathbf{p}^*) - b(\mathbf{p}^*)$. This calculation is done for each protein channel of the stack to obtain a feature vector $\mathbf{x}^{\mathbf{p}} = (x_1^{\mathbf{p}}, x_2^{\mathbf{p}}, \ldots, x_d^{\mathbf{p}})^T$.

3. The local background intensity estimate $b(\mathbf{p}^*)$ is calculated based on a $M \times M$ neighborhood $\Omega_M(\mathbf{p}^*)$ centered at $\mathbf{p}^*$. However, only neighbors $\mathbf{p}' \in \Omega_M(\mathbf{p}^*)$ which have a minimum distance to other synapse centers are used for background calculation, resulting in a modified neighborhood $\Omega'_M(\mathbf{p}^*)$ (see figure 6.3 (c)). Thus, $b(\mathbf{p}^*)$ is calculated as

$$b(\mathbf{p}^*) = \text{median}(\Omega'_M(\mathbf{p}^*)) \text{ with } \Omega'_M(\mathbf{p}^*) = \left\{\mathbf{p}' | d(\mathbf{p}', \mathbf{p}) \geq 3 \, \forall \, \mathbf{p} \in \Lambda^s\right\} . \tag{6.2}$$

   Omitting pixels with a distance < 3 is motivated by the fact that synaptic regions should not be considered for background calculation, as they potentially show high intensity signals.

4. The feature value for the currently analyzed channel is set to $x_n^{\mathbf{p}} = \hat{f}(\mathbf{p}^*) - b(\mathbf{p}^*)$.

With the proposed method, a feature data set $\mathcal{X} = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^L\}$ with $L = |\Lambda^s|$ and $\mathbf{x} \in \mathbb{R}^d$ is obtained. Throughout this study, $M = 15$ is used which is based on experimental results.

6.2 Cluster Analysis

It has been pointed out previously that no a priori knowledge is available on the grouping of synapses based on their protein patterns. Therefore, supervised learning approaches are not applicable and unsupervised learning is applied to find hidden regularities in the data.

Clustering is an unsupervised learning method whose main goal is to find a partitioning of a high-dimensional data set $\mathcal{X} = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^L\}, \mathbf{x} \in \mathbb{R}^d$ into K $(K \ll L)$ clusters so that the similarity between members of one cluster is high and the similarity between members of different clusters is low. With clustering, natural grouping inherent in the data set can be revealed.

The clustering methods available can broadly be split into two main groups: hierarchical and partitional clustering. Hierarchical clustering methods either iteratively merge data points to larger clusters or split large data clusters into smaller ones. The obtained result is a tree, showing the relation between clusters, with the individual data items at the leaves. Cutting the tree at a specific levels, results in a partitioning of the data into disjoint clusters where the number of clusters depends on the chosen level. Partitional clustering, on the other hand, directly decomposes the data set into a set of clusters by minimizing some kind of objective function. Partitional clustering is directly related to vector quantization, an application in signal processing. In vector quantization, clustering is used to find a set of K representative vectors $\mathcal{W} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K\}$ with $\mathbf{w} \in \mathbb{R}^d$, referred to as *codebook vectors* or, as done throughout this study, *prototype vectors*. These prototype vectors can be used as a representative for each cluster, so that a high rate of data reduction can be achieved.

When analyzing synapse protein patterns, protein patterns of up to 3500 synapses have to be analyzed for one image stack. Clustering thus provides a valuable method for partitioning the data into groups of similar protein patterns and to achieve a data reduction on the combinatorial level.

In the following, three strategies for partitional clustering for vector quantization are presented.

6.2.1 Online K-means

One of the most popular partitional clustering methods is the *K-means* approach (Forgy (1965); Lloyd (1982)). The objective is to find a set of K prototypes so that the distances between each data item and its closest prototype is minimized. Following Lloyds algorithm (Lloyd (1982)), the complete training set is iteratively used for adapting the prototype vectors to reduce the total cluster variance until cluster assignments do not change anymore. Although the obtained clustering result may represent only a locally optimal solution rather than the global optimum, the strategy is often used due to its algorithmic simplicity and efficiency and has been recently voted in the list of top ten algorithms in data mining (Wu et al. (2008)). For large data sets, however, the consideration of each data item for prototype adaptation can be computationally expensive and thus has motivated an online K-means version which applies the *winner-takes-all* (WTA) rule. The algorithm can be described by the following steps:

1. **Initialization:** The prototype set is initialized by randomly choosing K points from $\mathcal{X}$. The iteration counter is initialized to $a = 0$.
2. **Learning:** Randomly a point $\mathbf{x} \in \mathcal{X}$ is drawn. The closest prototype $\mathbf{w}_{k^*}$ is determined according to $k^* = \arg\min_{1 \leq k \leq K} d(\mathbf{x}, \mathbf{w}_k)$. The winner prototype $\mathbf{w}_{k^*}$ is moved

towards $\mathbf{x}$ by

$$\Delta \mathbf{w}_{k^*} = \epsilon(\mathbf{x} - \mathbf{w}_{k^*}) \ , \tag{6.3}$$

with $\epsilon \in [0, 1]$. The counter is incremented to $a = a + 1$.

3. **Iteration:** The learning step is repeated until a predefined number of iterations is reached.

The learning rate ϵ specifies how much $\mathbf{w}_{k^*}$ is adapted towards the data item $\mathbf{x}$. This learning rate can be chosen in different ways, e.g. as being inversely proportional to the number of data points for which the prototype has been winner so far (MacQueen (1967)) or as an exponentially decaying rate as suggested by Ritter et al. (1991) which is defined as

$$\epsilon(a) = \epsilon_i \left(\frac{\epsilon_f}{\epsilon_i} \right)^{a/a_{max}} . \tag{6.4}$$

Here, ϵ_i and ϵ_f define the initial and final learning rate, which need to be specified in advance, and a_{max} is the total number of iterations. Throughout this study, an exponentially decaying learning rate as specified in equation 6.4 is applied.

A drawback of the K-means algorithm is its sensitivity to the initialization of the prototypes which can lead to quite different clustering outcomes, often being only bad local minima of the solution. Therefore it is common to repeatedly run the algorithm with different initializations and take that solution as the clustering result which shows the smallest intra-class variance. However, for very large data sets, this can be quite time consuming.

6.2.2 Neural Gas

The *Neural Gas* (NG) clustering algorithm was introduced by Martinetz and Schulten in 1991 (Martinetz and Schulten (1991)). During the learning phase, not only is the winner prototype adapted, as it is done in the K-Means algorithm, but also some more distant prototypes are adapted, to some extent, to the training point $\mathbf{x} \in \mathcal{X}$. Martinetz et al. describe the resulting adaptation rule as a *winner-takes-most* instead of a *winner-takes-all* rule (Martinetz and Schulten (1991)). The NG clustering thus can overcome bad initializations better than the K-means approach especially in the beginning of the learning, where the prototype vectors "move around" in the input space very fast.

Again, the algorithm can be described by an initialization, learning and iteration step:

1. **Initialization:** The set of prototype vectors $\mathcal{W}$ is initialized by randomly choosing K points from $\mathcal{X}$. The iteration counter is set to $a = 0$.

2. **Learning:** A random input point $\mathbf{x}$ is drawn from $\mathcal{X}$. The elements of $\mathcal{W}$ are sorted in ascending order according to their distance to $\mathbf{x}$. Each prototype $\mathbf{w}_k$ receives a rank denoted by

$$r_k(\mathbf{x}, \mathcal{W}) \ , \tag{6.5}$$

following the definition of Martinetz and Schulten (1991). The prototypes are subsequently adapted by the following rule:

$$\Delta(\mathbf{w}_k) = \epsilon(a) \cdot h(r_k(\mathbf{x}, \mathcal{W})) \cdot (\mathbf{x} - \mathbf{w}_k) \;, \tag{6.6}$$

where

$$h(r_k) = \exp(-r_k/\lambda(a)) \tag{6.7}$$

$$\lambda(a) = \lambda_i(\lambda_f/\lambda_i)^{a/a_{max}} \;. \tag{6.8}$$

The counter is increased to $a = a + 1$.

3. **Iteration:** If $a < a_{max}$ go to step 2.

The learning rate $\epsilon(a)$ is again defined as an exponentially decaying learning rate, as in 6.4. The term $h(k_r)$ specifies the strength of the adaptation based on the prototype's rank where $\lambda(a)$ determines the neighborhood width, which is decaying with increasing number of iterations. As in the learning rate, λ_i and λ_f are the initial and final neighborhood widths which have to be chosen in advance. Thus, in the early iterations a large number of prototypes are adapted which rapidly distributes the prototypes according to the distribution of the input data. In later steps, the specialization of each prototype is increased by defining decreasing neighborhood widths.

6.2.3 Hierarchically Growing Hyperbolic Self Organizing Maps (H^{2}SOM)

In the following, the concepts of clustering with Hierarchically Growing Hyperbolic Self Organizing Maps (H^{2}SOMs) will be described. However, before going into detail of the H^{2}SOM, the general concepts of Self Organizing Maps and Hyperbolic Self Organizing Maps will be given.

Self Organizing Map A clustering strategy similar to the Neural Gas algorithm discussed in 6.2.2 is the *Self Organizing Map* (SOM) introduced by Kohonen (1982). Likewise, not only the winning but also neighboring prototypes are adapted during the learning phase. However, a striking difference is the way of how the SOM defines the neighborhood of a prototype. Instead of the ranking function $r_k(\mathbf{x}, \mathcal{W})$ (see eq. 6.5), a low-dimensional regular lattice structure $\mathcal{L}$ is used to order the prototypes. Therefore, each node of $\mathcal{L}$ is associated with one of the K prototype vectors. A common choice is a two-dimensional rectangular lattice as depicted in figure 6.4. By applying this strategy, a mapping from a high-dimensional input space to a low-dimensional mapping space is obtained. Therefore, the SOM algorithm can not only be applied for the purpose of clustering or vector quantization but it is also very well suited for dimensionality reduction and, in the case of a 2D or 3D mapping, for visualization purposes. The application of SOMs for dimensionality reduction and visualization will be discussed in detail in section 6.3.

Similar to the NG algorithm, the SOM algorithm can be described as follows:

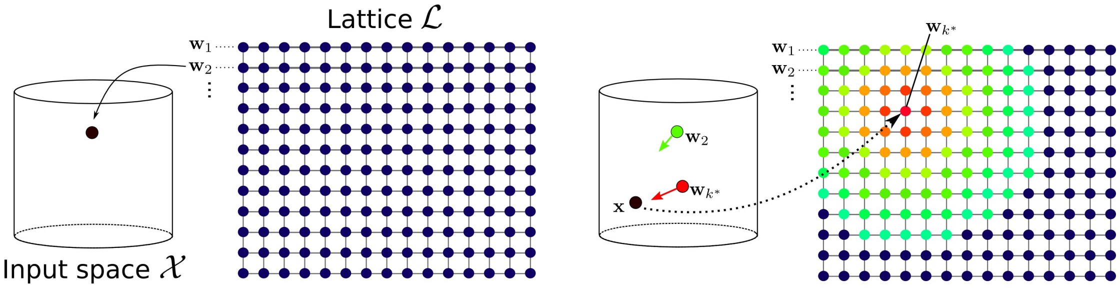


Figure 6.4: Schematic description of the SOM learning. (left) For each node of the regular lattice $\mathcal{L}$, a prototype vector $\mathbf{w}_k$ is initialized in the input space $\mathcal{X}$. (right) For a given data item $\mathbf{x} \in \mathcal{X}$, the best matching prototype $\mathbf{w}_{k^*}$ is computed. Depending on the value of the neighborhood function $h(k, k^*)$, which is encoded in the figure by different colors, the prototype vectors $\mathbf{w}_k$ are adapted towards $\mathbf{x}$.

1. **Initialization:** A prototype vector $\mathbf{w}_k$ is attached to each of the K nodes of the lattice $\mathcal{L}$ (see figure 6.4 left). The prototype vectors can be either initialized by randomly choosing data points from $\mathcal{X}$ or, if a rectangular 2D grid is given, by initializing them along the two principal components of the input training data. This results in a roughly ordered map before training starts and can reduce the required number of iterations (Kohonen (2001) p. 142). The iteration counter is initialized to $a = 0$.

2. **Learning:** A data item $\mathbf{x}$ is randomly chosen from $\mathcal{X}$. The best matching prototype $\mathbf{w}_{k^*}$ is computed according to $k^* = \arg\min_{1 \leq k \leq K} d(\mathbf{x}, \mathbf{w}_k)$.
   All prototype vectors are updated according to

$$\Delta \mathbf{w}_k = \epsilon(a) \cdot h(k, k^*) \cdot (\mathbf{x} - \mathbf{w_k}) \ . \tag{6.9}$$

   The learning step a is incremented.

3. **Iteration:** If $a < a_{max}$ continue with the learning step.

Here, $h(k, k^*)$ represents the neighborhood function, centered at k^*, which decays with increasing distance $d_{\mathcal{L}}(k, k^*)$ on the lattice $\mathcal{L}$. This is exemplarily shown in figure 6.4 (right), where the color encodes the value of the neighborhood function. Different choices for $h(k, k^*)$ are possible. A common neighborhood function used throughout this study is the Gaussian function, which is defined as

$$h(k, k^*) = \exp\left(\frac{-d_{\mathcal{L}}(k, k^*)^2}{2\sigma^2(a)}\right) \ . \tag{6.10}$$

The distance of two nodes on the lattice, $d_{\mathcal{L}}(k, k^*)$, can be the Euclidean distance or the length of the connecting path on the lattice depending on the chosen lattice topology. Here, the Euclidean distance is chosen. As in the NG algorithm, the radius $\sigma(a)$ of the neighborhood as well as the learning rate $\epsilon(a)$ are decaying over time and the initial and final values have to be chosen. Again, an exponential decaying function is applied for $\epsilon(a)$ and $\sigma^2(a)$.

Hyperbolic SOM The extension of the SOM to the Hyperbolic SOM (HSOM) was introduced by Ritter in 1999 (Ritter (1999)). Here, the regular lattice structure is not embedded in $\mathbb{R}^2$ but in the hyperbolic space $\mathbb{H}^2$. This space is characterized by a uniform negative curvature, so that the neighborhood around a point increases asymptotically exponentially with the radius. This characteristic overcomes the problem of Euclidean space that a quite restricted neighborhood "fits" around a point and makes it especially well suited for hierarchically organized data structures or large data sets (Ritter (1999)). The question at hand is which regular grid structure, i.e. which tessellation with congruent polygons, can be used in hyperbolic space so that each point in the grid has the same number of neighbors. It has been shown that there is an infinite number of such tessellations for $\mathbb{H}^2$ (Magnus (1974)). The geometrically simplest tessellation is a tiling with triangles, which is applied for the HSOM approach. A detailed explanation on how to compute such a tessellation can be found in Ritter (1999) .

Training of the HSOM follows the same steps as training of the SOM described before. The only difference is the neighborhood function $h(k, k^*)$ where the distance measure in $\mathbb{R}^2$ has to be adapted to a distance function in $\mathbb{H}^2$. The hyperbolic distance can be described by the following function:

$$\delta_{\mathcal{L}}(k, k*) = 2\,\mathrm{arctanh}\left(\left|\frac{z_k - z_{k^*}}{1 - \bar{z}_k z_{k^*}}\right|\right) \ , \tag{6.11}$$

where $z_k, z_{k^*} \in \mathbb{C}$ are complex values representing the 2D position of node k in the Poincaré model. The Poincaré model allows to map the infinite $\mathbb{H}^2$ entirely into the Euclidean unit disk. This mapping preserves angles but distorts distances greatly, resulting in a "fish eye" effect. The center of $\mathbb{H}^2$ is almost correctly mapped, whereas more distant regions get exponentially squeezed as can be seen in figure 6.5. However, shapes on $\mathbb{H}^2$ are not deformed, only their size shrinks with increasing distance from the origin (Ritter (1999)). It is a consequence of the $\mathbb{H}^2$ properties that there exists no perfect embedding in the $\mathbb{R}^2$. A more detailed introduction to the Poincaré model and the calculation of the distances can be found e.g. in Ritter (1999) or Ontrup (2008).

Hierarchically Growing HSOM As the HSOM, as well as the SOM algorithm, scale linearly with the number of nodes (Kohonen (2001), Ontrup (2008)), Ontrup and Ritter have proposed the Hierarchically Growing Hyperbolic SOM (H^{2}SOM) (Ontrup and Ritter (2006)). The main idea is to use the same lattice structure as presented for the HSOM algorithm but the learning is carried out in hierarchical manner. In the following, the main steps of the H^{2}SOM algorithm are sketched:

1. **Initialization:** The root node of the grid is placed at the origin of $\mathbb{H}^2$. This node is initialized with the center of mass of the training data and does not change during the learning process. The *branching factor* b is chosen which specifies how many children a node can have. This factor has a lower limit of $b > 6$ but no upper limit (Ontrup and Ritter (2006)). The nodes of the first hierarchical level are placed around the root node by positioning them at the vertices of the equilateral triangles which span the

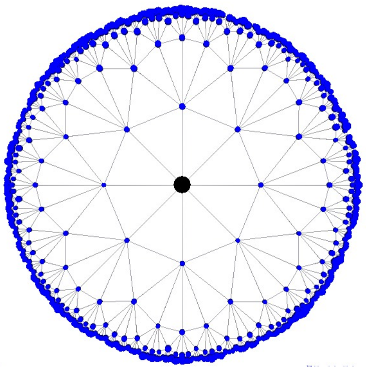


Figure 6.5: Embedding of the regular lattice $\mathcal{L}$ in $\mathbb{H}^2$ into the Euclidean unit disk via the Poincaré model. The origin of $\mathbb{H}^2$ is visualized by a black circle, whereas the nodes of $\mathcal{L}$ are represented by blue circles. The lattice structure is almost correctly mapped near the origin of $\mathbb{H}^2$ whereas more distant regions are exponentially squeezed, resulting in a strong fish eye effect.

initial grid. The first level nodes are initialized by small random variations of the root node.

2. **Learning:** The learning takes place in the standard way: The best matching node k^*, i.e. the node which is associated to the prototype vector $\mathbf{w}_{k^*}$ closest to the input signal $\mathbf{x}$, is determined. All prototypes are updated by the learning rule specified in equation 6.9 with the neighborhood function defined in eq. 6.10 and the hyperbolic distance as defined in eq. 6.11. The learning rate $\epsilon(a)$ and the neighborhood width $\sigma(a)$ again decrease with increasing iteration number.

3. **Growing:** After a fixed number of iterations, the quantization error is evaluated for each node of the current hierarchical level. The quantization error is defined as $EQ(k) = 1/|\mathcal{C}_k| \sum_{\mathbf{x}\in\mathcal{C}_k} \parallel \mathbf{x} - \mathbf{w}_k \parallel$ where $\mathcal{C}_k$ is the set of items belonging to the k-th cluster. If $EQ(k)$ exceeds a given threshold, the node k is marked for growing. Each marked node is surrounded by $b-3$ new children nodes (a parent node and two sibling nodes already exist). For an algorithmic description on how to position these nodes, please refer to Ontrup and Ritter (2006).

4. **Iteration:** After expansion of all marked nodes, the prototype vectors of the previous hierarchical level are fixed and steps 2 and 3 are carried out for the newly generated level. This iteration can be repeated until a predefined depth is reached or no nodes get marked for extension.

A very time consuming factor in training the H^{2}SOM is the search of the best matching prototype. Therefore, a *beam search* can be applied to approximate the global search for the best matching node (Ontrup (2008), Ontrup et al. (2009)). As a starting point for the search, the root node of the grid is chosen. Recursively, the r $(1 < r \leq b)$ best matching nodes among the b neighbors are determined until the periphery is reached. For small r, this strategy can reduce the computational complexity considerably with respect to the global optimum search. In Ontrup (2008), it has been shown that for $r = 2$ and $r = 3$ this strategy finds the direct or neighboring node of the global winner node in 96.6% and 97.7%, respectively. For the following studies, a beam search parameter of $r = 2$ is chosen.

In the case of the H^{2}SOM, several hierarchical levels are obtained which represent the data with increasing number of prototypes, and thereby with increasing granularity. Thus, to obtain a final cluster result, one level of the H^{2}SOM has to be chosen. Depending on how the grid topology is defined, the prototype number in each level varies. For example, if each node has eight neighbors, the first level will have 8 prototypes, the second 32 and the third 120.

6.2.4 Cluster Validation

One of the most fundamental problems in cluster analysis is to find the appropriate number of clusters for a given data set if no a priori knowledge is available. Several validity indices and techniques have been proposed to measure how suitable a chosen cluster number is. Generally, they can be split into three categories: External, internal and relative criteria (Halkidi et al. (2001)). For the external indices, the structure generated by a clustering algorithm is evaluated against a predefined structure which reflects external knowledge about the data. Internal indices evaluate the generated structure solely based on the underlying data. The third approach of evaluating cluster results is based on relative criteria. Here, clustering results of the same algorithm but initialized with different parameters are compared.

In this study, there is no knowledge about the underlying data structure, so external indices are not applicable. Hence, validity indices which evaluate relative criteria are chosen in this work, as they are computationally less time consuming than internal indices. Four of them, which are used throughout this study, are presented in the following.

Calinski Harabasz (CH) Index: This index introduced by Calinski and Harabasz in 1974 analyses the within and between cluster scatter of the obtained cluster result (Calinski and Harabasz (1974)). For K clusters and L data points, the index can be computed as

$$CH(K) = \frac{BSS(K)/(K-1)}{WSS(K)/(L-K)}\,, \tag{6.12}$$

where $WSS(K) = \sum_{k=1}^{K} \sum_{\mathbf{x}\in\mathcal{C}_k} d(\mathbf{x}, \boldsymbol{\mu}_k)^2$ is the sum of squared distances within data items and its cluster centers $\boldsymbol{\mu}_k$, and $BSS(K) = \sum_{k=1}^{K} |\mathcal{C}_k|\, d(\boldsymbol{\mu}_k, \boldsymbol{\mu})^2$ is the sum of squared distances between cluster centers and the center of the total data set $\boldsymbol{\mu}$. The term $\mathcal{C}_k$ denotes the set of items belonging to the k-th cluster with $|\mathcal{C}_k|$ being the cardinality of that set. Larger values of $CH(K)$ correspond to good cluster configurations as the within cluster variation is minimized whereas the between cluster variation is maximized.

Index $\mathcal{I}$: The index $\mathcal{I}$ has been proposed by Maulik and Bandyopadhyay (2002). To be consistent with the notation of Maulik and Bandyopadhyay (2002), here $\mathcal{I}$ denotes the validity index, as opposed to the rest of the thesis where $\mathcal{I}$ refers to a specific set of image stacks. The index $\mathcal{I}$ for K clusters can be calculated as

$$\mathcal{I}(K) = \left(\frac{1}{K} * \frac{E_1}{E_K} * D_K\right)^v \ , \tag{6.13}$$

where E_K measures the sum of distances of each point to the associated cluster center. It can be written as

$$E_K = \sum_{k=1}^{K} \sum_{\mathbf{x} \in \mathcal{C}_k} d(\mathbf{x}, \boldsymbol{\mu}_k) \ , \tag{6.14}$$

where $\boldsymbol{\mu}_k$ is the center of the k-th cluster. The constant E_1 measures the distances of all data points to the center of the data set. D_k accounts for the maximum separation between all pairs of clusters and is defined as

$$D_k = \max_{i,j=1}^{k} d(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) \ . \tag{6.15}$$

The index $\mathcal{I}$ is thus composed of three factors which compete with each other. The factor $\frac{E_1}{E_k}$ and D_k increase with increasing number of K resulting in an increase in the index whereas $\frac{1}{K}$ decreases with increasing K and thus reduces the index. However, D_k is bounded by the maximum distance between two points in the data set. The exponent v is used to control the contrast between different cluster configurations and was set to $v = 1$ throughout this study. The value K which maximizes the index $\mathcal{I}$ is assumed to be the proper number of clusters.

Davies-Boudlin (DB) Index: The DB Index analyses the sum of within cluster scatter and between cluster separation (Davies and Bouldin (1979)). The within cluster scatter in cluster $\mathcal{C}_k$ can be computed as

$$WCS_k = \frac{1}{|\mathcal{C}_k|} \sum_{\mathbf{x} \in \mathcal{C}_k} d(\mathbf{x}, \boldsymbol{\mu}_k) \ . \tag{6.16}$$

The distance between two clusters $\mathcal{C}_k$ and $\mathcal{C}_{k'}$ is defined as $d_{kk'} = d(\boldsymbol{\mu}_k, \boldsymbol{\mu}_{k'})$. Based on these measures, the DB index is defined as

$$DB(K) = \frac{1}{K} \sum_{k=1}^{K} R_k \ , \tag{6.17}$$

with $R_k = \max_{1 \le k' \le K, k' \ne k} \left\{\frac{WCS_k + WCS_{k'}}{d_{kk'}}\right\}$. Lower $DB(K)$ values reflect better cluster configurations and the value K which minimizes $DB(K)$ is chosen as the best number of clusters.

Dunn Index: The Dunn index (Dunn (1973)) favors cluster configurations where the diameter of the clusters are small, so the clusters are compact, and the distance between the clusters is large, so they are well separated. The diameter of a cluster $\mathcal{C}_k$ can be computed as $\mathrm{diam}(\mathcal{C}_k) = \max_{\mathbf{x^p},\mathbf{x^{p'}} \in \mathcal{C}_k} d(\mathbf{x^p}, \mathbf{x^{p'}})$ where $d(\mathbf{x^p}, \mathbf{x^{p'}})$ is the distance between the two data items $\mathbf{x^p}$ and $\mathbf{x^{p'}}$. The distance between two clusters $\mathcal{C}_k$ and $\mathcal{C}_{k'}$ is defined as $d(\mathcal{C}_k, \mathcal{C}_{k'}) = \min_{\mathbf{x^p} \in \mathcal{C}_k, \mathbf{x^{p'}} \in \mathcal{C}_{k'}} d(\mathbf{x^p}, \mathbf{x^{p'}})$. The Dunn index can thus be defined as

$$DN(K) = \min_{1 \leq k \leq K} \left\{ \min_{1 \leq k' \leq K, i \neq j} \left\{ \frac{d(\mathcal{C}_k, \mathcal{C}_{k'})}{\max_{1 \leq i \leq K}(\mathrm{diam}(\mathcal{C}_i))} \right\} \right\} . \tag{6.18}$$

Larger values of $DN(K)$ reflect better cluster configurations and the maximum of $DN(K)$ is considered as the correct cluster number.

6.2.5 Selecting a Distance Function

The cluster algorithms as well as the validity indices all depend on the specification of a distance measure $d(\cdot, \cdot)$ between two vectors. Generally, the Euclidean or Squared Euclidean distance is chosen, however also other distance measures are possible, potentially with some adaptation of the algorithm.

For protein colocation analysis, the characteristic profile of a protein is of higher interest than the actual intensities of each protein. If one feature vector is a multiple of another, the distance between both shall be rather small. Thus, the Euclidean distance is not well suited. Therefore, the dot product or scalar product was chosen as a distance measure. If two vectors $\mathbf{x^p}$ and $\mathbf{x^{p'}}$ of unit length are considered, the similarity of $\mathbf{x^p}$ and $\mathbf{x^{p'}}$ is given as the cosine angle calculated via the dot product $\mathbf{x^p} \cdot \mathbf{x^{p'}} = \cos(\alpha)$. The distance between the two vectors is thus given as $d(\mathbf{x^p}, \mathbf{x^{p'}}) = 1 - \cos(\alpha)$. Generally, $d(\mathbf{x^p}, \mathbf{x^{p'}})$ could take values in $[0, 2]$. However, the data set only shows data items whose features are ≥ 0, thus only angles in $[0, \pi/4]$ are obtained and therefore $0 \leq d(\mathbf{x^p}, \mathbf{x^{p'}}) \leq 1$.

To be able to use the dot product, all data items have to be normalized to unit length, thus $\hat{\mathbf{x}} = \mathbf{x}/ \parallel \mathbf{x} \parallel$. Here, $\parallel \mathbf{x} \parallel$ is the Euclidean length of vector $\mathbf{x}$. The adaptation rule of the clustering algorithm (see equation 6.3, 6.6, 6.9), as well as the validity indices also have to reflect the distance measure used. Therefore, a renormalization of the prototype vectors to unit length has to be carried out after each adaptation as well as a normalization of the cluster centers calculated for the validity indices.

Although the height of the pattern is not as important as the profile itself, it shall still be omitted that patterns with very low intensities, i.e. probable no true signals, are too similar to that pattern with high intensities. Therefore, a bias term $x_{d+1} = 255$ is introduced prior to the normalization to unit length.

6.3 Visualizations of the Feature Domain for TIS Data Exploration

Although clustering greatly aids in finding groupings inherent in the data, the success and efficiency of knowledge discovery mainly depends on suitable, linked visualizations of the

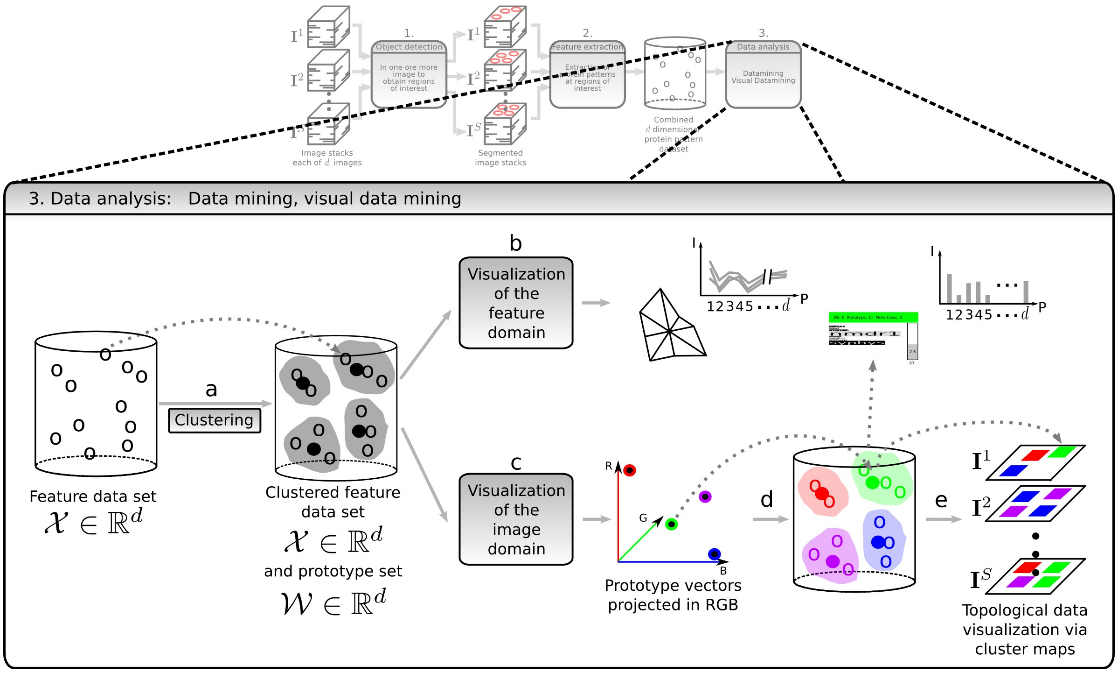

I^1
I^2
I^S
Image stacks each of d images
1.
Object detection
In one ore more image to obtain regions of interest
I^1
I^2
I^S
Segmented image stacks
2.
Feature extraction
Extraction of protein patterns at regions of interest
Combined d dimensional protein pattern dataset
3.
Data analysis
Datamining Visual Datamining
3. Data analysis: Data mining, visual data mining
Feature data set
$\mathcal{X} \in \mathbb{R}^d$
a
Clustering
Clustered feature data set
$\mathcal{X} \in \mathbb{R}^d$
and prototype set
$\mathcal{W} \in \mathbb{R}^d$
b
Visualization of the feature domain
I
1 2 3 4 5 ... d
P
I
1 2 3 4 5 ... d
P
c
Visualization of the image domain
R
G
B
Prototype vectors projected in RGB
d
e
I^1
I^2
I^S
Topological data visualization via cluster maps

Figure 6.6: Schematic display of the proposed visual data mining of TIS image stacks. (a) The feature data set $\mathcal{X}$ is clustered to reveal hidden regularities in the data. Different visualizations of the feature domain (b) as well as the image domain (c) are provided to allow for an interactive, visual exploration of the data. Visualizations in the image domain can be rendered by assigning colors to each cluster and thus to each object location (d),(e).

feature domain, i.e. clusters and prototypes, as well as the image domain, i.e. the topological ordering of the data items. The process of clustering and subsequent visualization in the feature and image domain is displayed in 6.6.

In this section I will focus on visualizations of the feature domain. To this end, it has to be considered that a large number of high-dimensional feature vectors have to be displayed. It is evident that not just one view of the feature domain is sufficient, but several linked views have to be provided for an efficient exploration (see figure 6.6 (b)). Furthermore, suitable visualizations for the display of prototypes and for the visualization of the cluster results and individual data items need to be provided.

One simple strategy for the display of multivariate data is an extension of the scatter plot to a *generalized drafter's plot* (Chambers et al. (1983)), also referred to as a scatter plot matrix. Here, scatter plots for all possible pairs of features are displayed. A related technique, termed *dimensional stacking* (LeBlanc et al. (1990)), embeds one coordinate system into another and bins the data. Another popular way to display multivariate data are *glyph* or *icon* displays. According to Colin Ware " A glyph is a graphical object designed to convey multiple data values" (Ware (2004), p.145). Each data feature is mapped to a different graphical attribute of the glyph such as size, shape or color. For example *Chernoff faces* (Chernoff (1973)), *star glyphs* (Chambers et al. (1983)), *color icons* (Levkovitz (1991)), or *stick figures* (Pickett and

Grinstein (1988)) belong to these types of displays. In general, the perception of the glyphs depends upon the assignment of the data features to the glyph parameters and often only a limited number of attributes, around eight, can be displayed as many graphical attributes are not independent from another (Ware (2004)). A strategy which is often combined with scatter plot visualizations or iconic displays is parameter selection and reduction. Different approaches from the field of pattern recognition and machine learning can be applied. One can try, for example, to find that sub set of features that most contribute in partitioning the data set into different clusters. Other approaches try to find a projection of the data into lower dimensional subspaces, such as *Principal Component Analysis* (Pearson (1901); Hotelling (1933); Karhunen (1946)). An overview of visualization techniques for multivariate data can be found for example in Keim (2002), Wong and Bergeron (1997) or Ware (2004).

I will now introduce multivariate visualization techniques used in this work in more detail and present a glyph display, the **C***ombinatorial* **I***ntensity* **Pr***ofile* **A***rchetype* (CIPRA), which has been developed in this work to suite the needs of non-binarized protein colocation analysis.

6.3.1 Prototype Visualization

By focusing on the cluster representation only, i.e. prototypes, the concept of vector quantization is used to significantly reduce the data complexity. The main protein colocation characteristics of the data can be explored without the need of analyzing each single data item which eases the knowledge discovery process. If interesting prototypes are found, the associated data items can be analyzed in a subsequent step following the Shneiderman mantra of “Overview first, zoom in and filter, details on demand”. However, suitable visualization strategies are still required for the prototype display. As stated above, several approaches have been proposed for multivariate data display. Although generalized drafter's plot and dimensional stacking are a straightforward extension of lower dimensional displays, these types of displays are often hard to interpret with increasing dimensionality. This holds especially if a combination of more than two features contribute to an interesting pattern, as it is likely the case in protein colocation studies. Dimensionality reduction strategies, as stated before, could be applied to select a sub set of parameters or find lower dimensional projections of the data. However, the former discards some of the features, i.e. proteins, so that not the whole combinatorial power can be analyzed. In the latter case, it is often not possible to interpret the data with respect to combinations of proteins as the direct link of feature to protein is lost.

I will therefore present two strategies which display all features of the prototypes at once and have influenced the novel visualization strategy introduced in this work. One of the most simple visualizations which achieves this is the *bar graph* or *bar chart*. For each data attribute, a bar is plotted where the height of the bar is proportional to the value of the data attribute. The associated data attribute is usually written at the x-axis of the plot. Figure 6.7 (a) displays an eight dimensional prototype of the synapse data set. Although it is a quite basic visualization strategy, it is well suited to analyze the quantities of the different data features for one prototype. According to Clevland and McGill as well as Mackinlay, length is one of the most accurate encodings for quantitative data (Spence (2007)).

Another straight forward way of displaying all features in one visualization are the previously

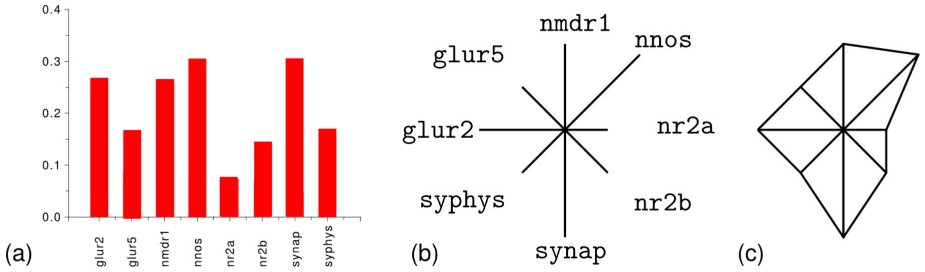


Figure 6.7: Visualization strategies for prototype display. (a) Bar plot (b) Whisker plot and (c) star glyph.

mentioned glyphs. As stated, there exists a vast number of different glyphs. In this work, *star glyphs* (Chambers et al. (1983)) are applied to present protein colocation. The basis of a star glyph is a *whisker plot* which is designed so that each data feature is represented by a straight line (axis) originating from one central point. The length of the axis reflects the value of the feature. Figure 6.7 (b) shows a whisker plot for an eight dimensional data point. The star glyph is an extension to the whisker plot, where the ends of the axes are connected by lines (see fig. 6.7) (c). Thereby, a characteristic shape is generated for each prototype. The benefit of whisker or star glyphs is that each data feature is visualized the same way and therefore is perceived the same. To increase the number of rapidly distinguishable attributes, the luminance polarity of half of the axis can be inverted or the width as well as the length of each axis can be changed (Ware (2004), p. 184).

6.3.2 Prototype Visualization via Combinatorial Intensity Profile Archetypes

So far two standard visualization strategies were discussed which are applied for prototype visualization. They are both capable of visualizing multivariate data, however, with increasing dimensionality it becomes more and more difficult to compare individual plots with each other and to associate data features, i.e. proteins, to graphical attributes. For prototype analysis, however, these two features are of high importance. If the association of proteins to graphical attributes is not clear at first sight, a rapid identification of prototypes which display an interesting protein combination is not possible.

Therefore, I have developed a new prototype visualization strategy, the **C**ombinatorial **I**ntensity **Pr**ofile **A**rchetype (CIPRA). It combines visualization aspects mentioned in the context of the bar chart and star glyphs and was inspired by the *sequence logo* display, which represents patterns in nucleotide or amino acid sequences (Schneider and Stephens (1990)). In a sequence logo, for each position of a set of aligned sequences, e.g. nucleotide sequences, the four nucleotides are arranged on top of each other sorted according to their frequency at that position. The character height represents the frequency of the corresponding nucleotide. Figure 6.8 displays a sequence logo for a nucleotide sequence alignment of length 10. Through this visualization, a rapid identification of prominent sequence patterns can be achieved as high frequent nucleotides can directly be "read" from the logo. In the given example, the pattern ATG starting at position five is characteristic for most sequences. This reading of

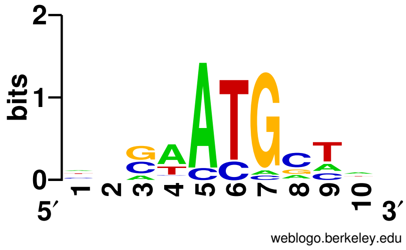


Figure 6.8: Sequence logo for sequences of 10 nucleotide length. The nucleotide conservation ATG starting at position 5 can be rapidly identified. (Logo generated with WebLogo (Crooks et al. (2004)) and randomly created sequences).

interesting patterns is a desirable feature, which is adapted in the CIPRA visualization.

To construct a CIPRA for one prototype, a horizontal box is drawn for each data feature (see figure 6.9). The height, as well as the length, of each box is scaled according to the feature's value. To increase differentiation between neighboring boxes, they are alternately colored black and light shaded gray. This follows Ware's suggestion for star glyphs or whisker plots to increase the number of dimensions by changing length and width of the bars as well as using different luminance levels. Furthermore, by employing length as an attribute for data representation, a graphical parameter well suited to encode quantitative data has been chosen, as has already been discussed for the bar plot. To allow for a fast identification of prominent proteins, the protein names are directly incorporated into the visualization. To this end, the associated protein name is written in each bar and scaled in height and length analog to the bar itself. With this strategy, prominent protein colocalization can easily be identified by "reading" the CIPRA analog to the reading of a sequence logo.

Figure 6.9 gives an overview of the construction of the CIPRA display (left) and shows two CIPRA examples for 12 dimensional synapse prototypes (right). In addition to the box plot display of the feature vector components, the top bar of a CIPRA shows the color which is assigned to a CIPRA, and meta information as the associated data set and the prototype number. How color will be assigned to each prototype will be discussed in section 6.4. On the right, the CIPRA displays information about the abundance of the feature combination. This abundance can be related to an entire set of TIS image stacks or to one particular image stack as in this example. The gray box with the overlaid number shows the relative abundance, below the absolute abundance of pixels is displayed, which are assigned to this prototype according to the best match criterion. This meta information is very valuable for prototype analysis.

In the CIPRA example of figure 6.9 (top right), it can be clearly seen that the proteins termed `nmdr1` and `syphys` both have high values whereas the other proteins are rather poorly represented. Colocalization of the two proteins `nmdr1` and `syphys` can thus be expected for those data points associated to the corresponding prototype. The bottom CIPRA shows higher values for `nmdr1`, `nr2b` and `syphys`. Both prototypes can be compared very easily and the major characters can be readily extracted.

If the CIPRA visualization is compared to the bar graph and star glyph (see figure 6.30

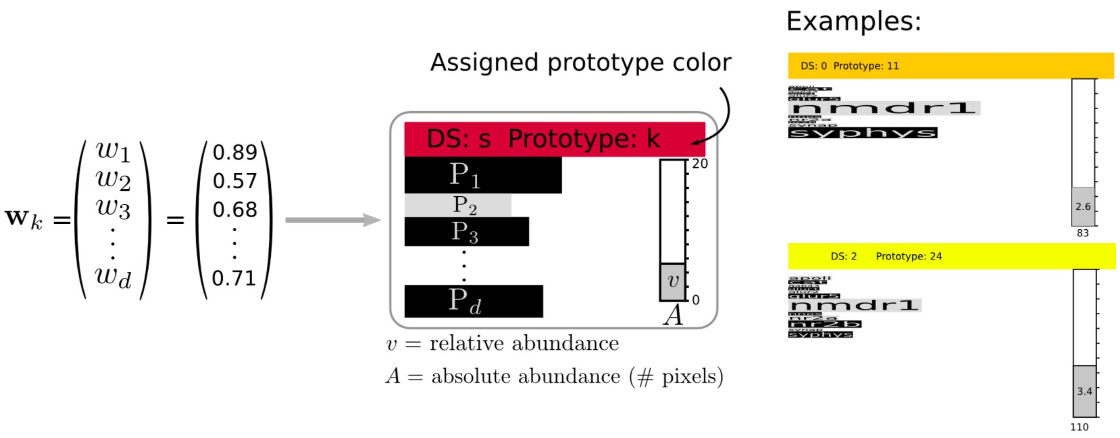

Examples:
Assigned prototype color
$\mathbf{w}_k = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_d \end{pmatrix} = \begin{pmatrix} 0.89 \\ 0.57 \\ 0.68 \\ \vdots \\ 0.71 \end{pmatrix}$
DS: s Prototype: k
P_1
P_2
P_3
P_d
20
0
v
A
v = relative abundance
A = absolute abundance (# pixels)
DS: 0 Prototype: 11
nmdr1
syphys
2.6
83
DS: 2 Prototype: 24
nmdr1
nr2b
syphys
3.4
110

Figure 6.9: Generation of a CIPRA for one prototype $\mathbf{w}_k$. Each box is scaled in height and width according to the features' values. The prototype names $(P_1, \ldots, P_d)$ are written in the boxes for fast association of the proteins to the boxes. The relative and absolute abundance of the prototype, i.e. how many synapses are associated to that prototype, as well as the assigned prototype color are given. Exemplarily two CIPRAs are displayed at the right. Quickly, the main characteristics of both prototypes can be perceived and compared. Both feature higher levels of `nmdr1` and `syphys`, the second one additionally has signal for `nr2b`.

and 6.7), it is evident that in the CIPRA display the association of proteins to individual graphical attributes is much easier. Furthermore, besides being able to rapidly identify the dominant proteins, an advantage of the CIPRA display is that only features with high values allocate space, whereas low value features are squeezed. Thereby, space is only allocated proportional to the importance of the protein and the total size of the CIPRA reflects the amount of information provided by the prototype. In some applications, this might not be a desirable feature so that bar graphs, or CIPRAs with constant bar width would be better suited.

6.3.3 Cluster Visualization

I will now focus on the aspect of how to visualize clusters, i.e. their associated data items, to provide for an in detail analysis of the data domain. While the analysis of prototype visualizations gives an overview of the main protein colocation characteristics in the data set, the analysis of cluster visualizations allows to obtain a deeper insight in the protein patterns of individual data items. Furthermore, the cluster quality can visually be inspected and outliers can be identified.

In principle, the visualization methods presented for prototype display can also be applied for the visualization of individual data items of a cluster. However, in this work a large number of feature vectors have to be analyzed. For one image stack, feature vectors are extracted at around 2500 synapse positions. Even if solely the items of one cluster are analyzed, this can imply that hundreds of, for example, CIPRAs need to be inspected. For an efficient analysis, these types of displays are not suited and more compact visualizations

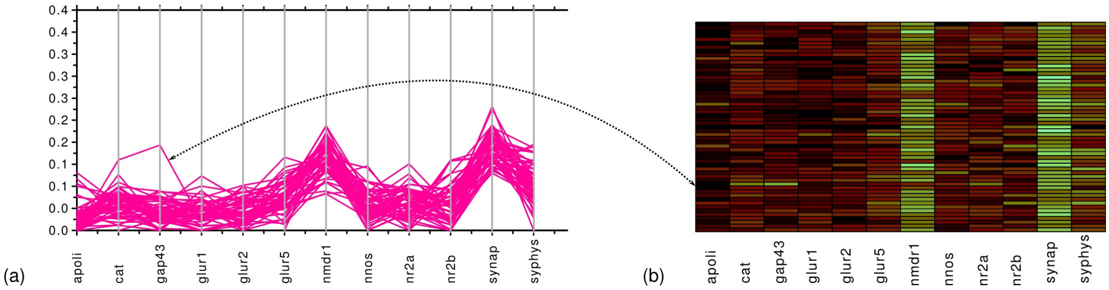


Figure 6.10: (a) Parallel coordinate visualization of a set of 12-dimensional data items. (b) Color lines view visualization of the same items as in (a). Both visualizations allow to identify the overall profile trend of the items with high values for `nmdr1` and `synap`. The characteristics of Individual data items can be better followed in the color lines view whereas the trend and compactness of the data is better perceivable in the parallel coordinates plot.

are required.

A popular method to represent sets of multivariate data in a compact way are *parallel coordinate* plots (Inselberg and Dimsdale (1990)). Here, parallel coordinate axes, i.e. vertical lines, are used for the display of the data features. Data points are represented by lines running through the axis at a height proportional to the value of that attribute (see fig. 6.10 (a)). With the parallel coordinate plot, correlations or anti-correlations of data items can be well perceived. For large data sets, however, this can result in quite dense views so that the line of an individual data point can not be followed visually. Therefore, interactive *brushing* is often applied (Becker and Cleveland (1987)) to highlight a single point or a range of points. Interaction with the visualization itself is also required for the ordering of axes. In the case of the proteins, there is no natural order of the data features as it is given, for example, for time series. A reordering of the axes can result in quite different visual impressions of the data.

A technique strongly related to parallel coordinates has been introduced by Matković et al. (2007). The proposed *color lines view* arranges the data set in a $(L \times d)$ matrix (with L being the total number of data items, i.e. synapses, and d being the number of protein channels). Each row (line) represents a data item $\mathbf{x}^\mathbf{p}$ and each column represents one data feature x_i. The matrix elements are colored according to their data features' values. Figure 6.10 (b) shows a color lines view of the same cluster as shown in 6.10 (a). It can be seen that, similar to the parallel coordinates view, the proteins termed `nmdr1` and `synap` can be identified as featuring a high signal in all data items. In the parallel coordinates plot, the compactness and overall trend of the data items can better be perceived. The color lines view, however, can be more helpful to analyze the pattern of individual data items even when no brushing functionality is provided. For example, for the data item marked with an arrow in figure 6.10, it can easily be seen in the color lines view that this item features high levels of `cat`, `gap43`, `nr2a` and `synap`. In the parallel coordinates plot, however, following this item is not possible so that the information about high `nr2a` and `synap` levels for that item gets lost. Still, efficiently working with the color lines view requires manual interaction. Matković

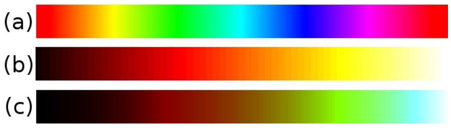


Figure 6.11: Color Scales. (a) Rainbow scale. (b) Heated object scale. (c) Linearized optimal color scale.

et al. have proposed various line sorting techniques and brushing for the visual exploration of the views. However, for the purpose of protein colocation analysis column sorting should also be provided, as there exists no predefined order of the proteins as stated before.

Another aspect of this visualization is the choice of an appropriate color scale so that the user can map the features' values to the color code. Since the features are of quantitative type and thus exhibit an ordering, color scales have to be chosen which are perceived as preserving the order and distance of the features' values (Levkovitz and Herman (1992)). A very simple color scale is the gray scale, which is naturally perceived as ordered. However, it has the disadvantage that only 60 to 90 colors are perceived as distinct (just noticeable differences (JND)) (Levkovitz and Herman (1992)). A color scale that increases the perceived differences is the rainbow scale, traversing through all the hues of the rainbow (see figure 6.11 (a)). While it has more JND than the gray scale, it has the drawback that the colors are not intuitively perceived as ordered and bright colors stand out. Other color scales are motivated by the human visual system as the *heated object scale* and the *magenta scale*. The heated object scale traverses from black over red and yellow to white (see figure 6.11 (b)), motivated by the fact that the human visual system is very sensitive to changes in the luminance for orange-yellow hue (Levkovitz (1991)). The magenta scale is motivated by the observation that the visual system is additionally very sensitive to hue changes for the magenta hue (Levkovitz and Herman (1992)). These color scales are perceived as ordered, however only a limited color range is used. All these color scales can be applied for the color lines view and are implemented in a color lines view generator implemented at the Biodata Mining & Applied Neuroinformatics Group, University of Bielefeld. However, results are presented with the linearized optimal color scale proposed by Levkovitz and Herman (1992) (see figure 6.11 (c)). Here, the number of JND are maximized (optimal) and additional colors are inserted in the color scale so that the perceived differences between adjacent colors are as uniform as possible (linear). For the line view display of the TIS data and visualization in this printed thesis, this color scale provided the best visualizations.

6.4 Visualizations in the Image Domain

The great advantage of imaging in biology is the availability of topological information. Not only protein profiles can be analyzed but additionally their spatial origin provides valuable information. Furthermore, the image domain is a natural source of information for the biologists which they are familiar in interpreting. Thus, providing a visualization in the image domain which links to the previously discussed visualizations of prototypes and clusters is of great importance.

Generally, all glyph representations mentioned previously for the prototype visualization could be employed for this purpose. To this end, a CIPRA or star glyph representation of the item's protein pattern or of its associated prototype could be depicted at the spatial position of each data item. Such an approach has been proposed by Pickett and Grinstein (1988), where stick figures were applied to plot five dimensional weather data at their spatial origin, or by Weigle et al. (2000) who developed a technique termed *oriented sliver textures*. Here, each parameter image is represented by slivers each having the same orientation. The individual sliver images were then combined to one sliver image. In both approaches, the individual item representations produce a visual texture so that regional trends in the data can be perceived as a distinct texture. For synapse colocation analysis, however, these displays are not well suited. First, the stick figures, for example, are interpreted primarily based on their total orientation rather than on the individual angles of the stick. Thus, misinterpretation of the data can easily occur. Furthermore, synapses are not evenly distributed across the image, so that continuous regions will not be generated and a textural interpretation is not possible. One main argument against the glyph based approaches for visualization of the image domain, however, is that well suited visualizations for prototype and item display are already provided. Rather than depicting the attributes of each individual data item in the image, an overview shall be generated so that the topological distribution of items belonging to one cluster can easily be perceived. Interesting regions need to be quickly identifiable whose individual in detail patterns can then be identified in the visualizations of the feature domain through link & brush techniques.

Therefore, in this work color is chosen to encode cluster membership of each synapse (see figure 6.6 (c)). One may argue that color is not well suited for this purpose as only five to ten colors should be used if a rapid identification is desired (Ware (2004), p. 123, p. 125). In this application, however, most often more than ten categories, i.e. clusters, need to be encoded. Nevertheless, the visualizations in this work are not used as static displays but interaction is provided by the link & brush concept. Additionally, in this application similar prototypes, i.e. clusters, are intended to be encoded with similar colors. This follows the assumption that protein patterns which differ only in few proteins contribute to more similar function than patterns which differ in a larger number of proteins. Hence, one can easily obtain an impression if clusters with similar colors and thus with similar functions form compact sub-regions in an image or are spread over the whole image. A rapid visual discrimination between colors is thus not that essential. Furthermore, encoding individual clusters with similar colors is also motivated by the fact that in the case of non-binary data there exists no such clear separation of protein patterns as it is the case for binary CMP data. In non-binary data analysis, continuous features are analyzed instead of a protein present-absent code. Although clustering aids in finding groups of similar patterns in the data, the separation will never be as sharp as in the case of CMPs. Thus, coloring similar prototypes with similar colors accounts for the continuity of the data. It is, however, reasonable to additionally provide color encodings which increase the separability of the colors if individual clusters need to be visually distinguished. These visualizations are then comparable to the random coloring for toponome map generation as descried in section 3.1.

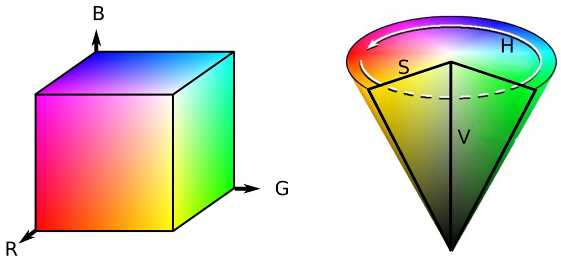


Figure 6.12: Display of the RGB and HSV color space.

It is now the question how to assign colors to prototypes and thus to individual synapses so that similar colors are mapped to similar prototypes. In general, coloring can be interpreted as a projection into some kind of color space. The most common color space is the RGB space, where color is produced by additively mixing red, green and blue primaries as can be seen in figure 6.12 (left). Another widely used space is the HSV space, where color is described by means of hue (H), saturation (S) and value (V). A HSV color cone is displayed in figure 6.12 (right). A problem of both spaces is that they are device dependent, i.e. different computer displays for example may show different colors for the same mixture of primaries. Furthermore, the spaces are non uniform, i.e. the perceived color difference of two colors in one region of the space can be considerably different from the perceived color differences in another region of the space. Therefore, the Commission Internationale de l'Éclairage (CIE) has proposed two device independent, uniform color spaces in 1978 (Ware (2004)). One may thus argue that these spaces are best suited if prototype similarity should be reflected by color similarity. However, regardless of the choice of the color space, they are three dimensional spaces. Thus, in order to specify similar colors for similar prototypes, the similarity between the prototypes in the d-dimensional space ($d > 3$) have to be mapped to similarities in the three dimensional color space. It is evident that such a mapping will not preserve all similarities faithfully. Thus, similarity distortion is already introduced through the mapping process and RGB and HSV spaces are sufficient for color encoding.

In the following I will present dimensionality reduction techniques which can be used to project the high-dimensional prototypes into a low-dimensional color space. Furthermore, different strategies for color assignment are presented.

6.4.1 Dimensionality Reduction Methods

In the following section, some dimensionality reduction methods are described which can be applied to map the high-dimensional prototypes to a low-dimensional space. This low-dimensional encoding can subsequently be used for color assignment to each prototype. Each projection method introduces two types of errors (Venna and Kaski (2001)). First, data items which are projected close to each other in the low-dimensional space might originate from distant regions in the high-dimensional space. Second, close neighbors in the high-dimensional space might be mapped to distant regions in the map space. Both errors misleads the observer in the interpretation of the data. Venna and Kaski (2001) have therefore proposed two measures, *trustworthiness* and *continuity*, to quantify the two errors. These measures can be applied to assess the quality of a data mapping. Additionally, the visualizations proposed in this work provided a means to assess these errors. If CIPRA visualizations are sorted according to the assigned prototype color, discontinuities and non trustworthy mappings can be identified.

This section is not intended to be an exhaustive survey of dimensionality reduction methods, but shall only point out methods which were applied in this study. More detailed surveys of dimensionality reduction methods can be found e.g. in Krzanowski (2000), Hastie et al. (2001) or Camastra and Vinciarelli (2008).

Principle Component Analysis A popular method to obtain a linear projection from a high-dimensional input space to a low-dimensional output space is the *Principal Component Analysis* (PCA) (Pearson (1901); Hotelling (1933); Karhunen (1946)). The objective is to find a projection that preserves as much of the data variance as possible. Let $\mathcal{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K)$ be the initial prototype set where the variables of an individual prototype are defined as $\mathbf{w}_k = (w_1^k, w_2^k, \ldots, w_d^k)^T$. The empirical covariance matrix $\mathbf{C}$ can be calculated as

$$\mathbf{C} = \frac{1}{n-1} \sum_{k=1} K(\mathbf{w}_k - \boldsymbol{\mu})^T (\mathbf{w}_k - \boldsymbol{\mu}) \ , \tag{6.19}$$

where $\boldsymbol{\mu}$ is the mean vector of the prototypes. Through the PCA, a set of q $(q \leq d)$ new variables $y_i^k (i = 1, \ldots, q)$, termed *principal components*, can be derived as a linear combination of the initial variables. For an individual prototype $\mathbf{w}_k$ of the original data set, the value of the ith principal component is computed as

$$y_i^k = \mathbf{e}_i^T \mathbf{w}_k = e_{i,1} w_1^k + e_{i,2} w_2^k + \cdots + e_{i,d} w_d^k \ . \tag{6.20}$$

Here, $\mathbf{e}_i^T = (e_{i,1}, e_{i,2}, \ldots, e_{i,d})$ is the eigenvector of $\mathbf{C}$ corresponding to the ith largest eigenvalue λ_i of $\mathbf{C}$. The variance covered by the principal component y_i is $\mathrm{var}(y_i) = \lambda_i = \mathbf{e}^T C \mathbf{e}$. As the variance of y_i depends on the length of $\mathbf{e}_i$, the eigenvectors are restricted to unit length vectors, thus $\mathbf{e}_i^T \mathbf{e}_i = 1$ (Krzanowski (2000)). Furthermore, the eigenvectors have to fulfill the condition of orthogonality, i.e. $\mathbf{e}_j^T \mathbf{e}_i = 0 (i < j)$. For further details on the principles of PCA, the reader is referred to, for example, Krzanowski (2000) or Hastie et al. (2001).

In a geometrical sense, the PCA can be interpreted as a projection of the data into a q-dimensional space which is spanned by the q eigenvectors $\mathbf{e}_i (i = 1, \ldots, q)$. A two dimensional example is given in figure 6.13. Thereby, as much of the initial data variance as possible is covered. However, as usually $q < d$, some data variance gets lost. For purpose of visualization, q is mostly chosen as $q = 2$ or $q = 3$. For other applications, a threshold for the minimum of data variance which needs to be preserved can be set.

Although the actual similarities are not mapped through the PCA projection but a mapping is performed to preserve as much data variance as possible, PCA mapping can be well suited for visualization purposes if there is a large variation in some variables of the prototypes. Thus, more emphasis is put on proteins which have a larger feature range, thus show very high signals in some data points and very low in others.

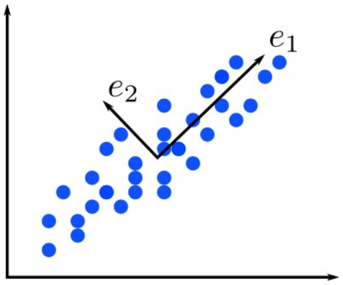


Figure 6.13: Two dimensional example for the PCA projection. Here, $\mathbf{e}_1$ is the first eigenvector and $\mathbf{e}_2$ the second eigenvector of the data cloud. If the data is mapped on $\mathbf{e}_1$, the largest amount of data variance is preserved

Sammon's Mapping Sammon's Mapping is a method to obtain a non-linear mapping from a high-dimensional input space to a low-dimensional output space (Sammon (1969)). Therefore, the data points are arranged in the output space in such a way that their pairwise distances reflect the original pairwise distances in the input space as faithfully as possible. The error or cost function which is optimized to find a good mapping is defined as

$$E = \frac{1}{\sum_{k<k'}^{K} d(k,k')} \sum_{k<k'}^{K} \frac{d(k,k') - d_{\mathcal{M}}(k,k')}{d(k,k')} \,. \tag{6.21}$$

Here, $d(k,k')$ is the distance between prototype vectors $\mathbf{w}_k$ and $\mathbf{w}_{k'}$ in the input space and $d_{\mathcal{M}}(k,k')$ defines the distance between $\mathbf{w}_k$ and $\mathbf{w}_{k'}$ in the map space, following the HSOM notation of section 6.2.3. For visualization purposes, the map space is usually chosen as $\mathbb{R}^2$ or $\mathbb{R}^3$. By normalizing the difference between $d(k,k')$ and $d_{\mathcal{M}}(k,k')$ with the original distance $d(k,k')$, an emphasis is placed on preserving small distances rather than large distances. Minimization of E can be realized by gradient descent methods, as described in Sammon (1969). A disadvantage of gradient descent is the fact that it can get caught in local minima. Thus, different initializations should be tried out. Alternatively, mapping configurations should be changed although E increases to overcome local minima. Often, a good initialization is a mapping onto the first principal components determined with a PCA projection. This initialization strategy was applied throughout this study.

H^{2}SOM As has been discussed in section 6.2.3, the SOM as well as the HSOM and H^{2}SOM can not only be applied for clustering purposes but also for dimensionality reduction by a non-linear projection of the data onto a lower-dimensional lattice structure. The mapping onto the lattice structure together with the neighborhood learning strategy of the H^{2}SOM has the convenient effect that prototypes which are similar in the input space lie close to each other on the lattice structure. Therefore, a neighborhood preserving mapping to a low-dimensional structure is directly obtained through the clustering approach.

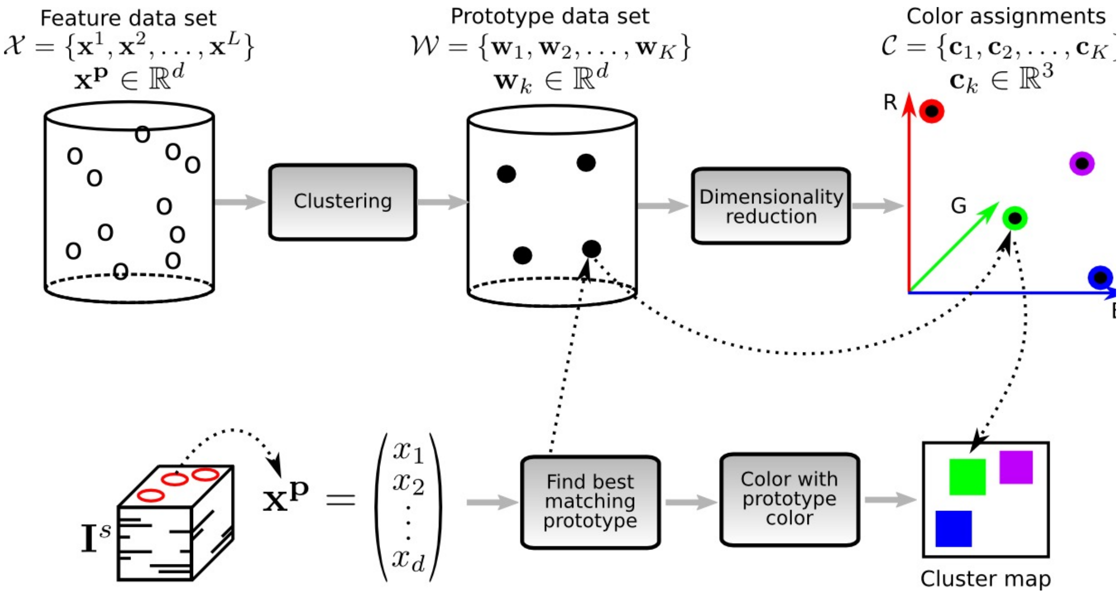

Feature data set
$\mathcal{X} = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^L\}$
$\mathbf{x}^\mathbf{p} \in \mathbb{R}^d$
Clustering
Prototype data set
$\mathcal{W} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K\}$
$\mathbf{w}_k \in \mathbb{R}^d$
Dimensionality reduction
Color assignments
$\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K$
$\mathbf{c}_k \in \mathbb{R}^3$
R
G
$\mathbf{I}^s$
$\mathbf{x}^\mathbf{p} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix}$
Find best matching prototype
Color with prototype color
Cluster map

Figure 6.14: Schematic visualization of the color assignment for cluster map generation. The feature data set is clustered to obtain a set of representative prototypes. Through dimensionality reduction, prototypes are projected into a lower dimensional color space and a three dimensional color vector is assigned to each prototype. For a given data item $\mathbf{x}^p$, the best matching prototype is searched and the corresponding image location is colored according to the assigned prototype color.

6.4.2 Color Assignment

Different strategies to assign color to the prototypes were applied in this study. Common to all coloring routines is the strategy displayed in figure 6.14. A feature data set $\mathcal{X}$ is clustered to obtain a set of prototypes $\mathcal{W}$. Via a dimensionality reduction method, Sammon, PCA or H^{2}SOM, the prototypes are mapped into a color space where each prototype is associated to a three dimensional color vector $\mathbf{c} = (c_1, c_2, c_3)$. For a given feature vector $\mathbf{x}^\mathbf{p}$, the best matching prototype is searched and the associated synapse position is encoded in the image domain by a 5×5 square colored with the according prototype color. Images as depicted in figure 6.15 are obtained. Although both images show the same clustering, it is clear to see that depending on the coloring strategy used, different visual impressions of the same data can be obtained. Thus, it is most convenient if different coloring strategies can be applied or even dynamically edited. In the following, different coloring strategies based on the RGB and HSV color space are described.

Mapping in the RGB Color Space The simplest coloring strategy for assigning color to three dimensional data is a mapping into the RGB color space. Here, the three primary axes are usually bounded by 8bit, i.e. each axis ranges from 0-255. By specifying a point in this 3D color cube, a distinct color is addressed.

By applying a PCA or Sammon's mapping, a projection of the high-dimensional input

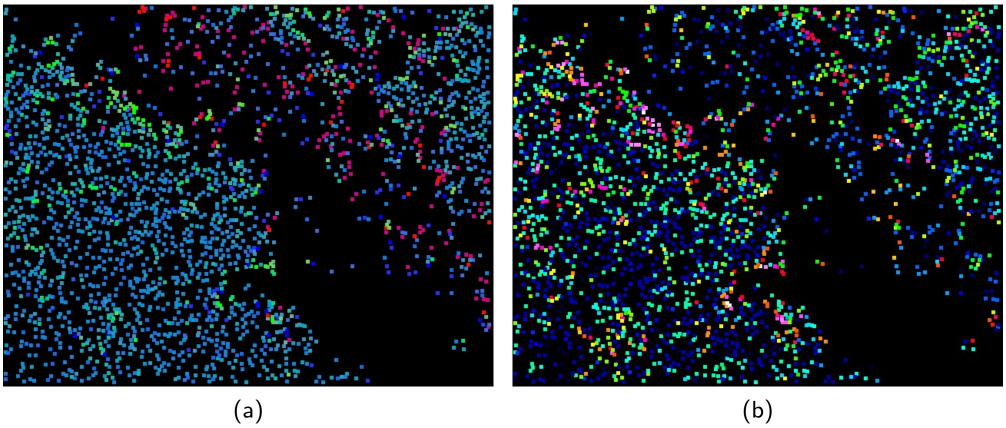


Figure 6.15: Cluster maps obtained through different projection strategies. (a) Sammon's mapping, (b) PCA projection.

space to a 3 dimensional output space can be obtained. This does not necessarily mean that this projection entirely fits within or utilizes the RGB color cube. Thus, a normalization step is required which scales the features of the prototypes to fit within 0-255. Thus, the color variable c_r of the three dimensional color vector $\mathbf{c}_k = (c_1, c_2, c_3)$ for the 3D mapped prototype $\tilde{\mathbf{w}}_k$ is defined as:

$$c_r(\tilde{\mathbf{w}}_k) = \frac{\tilde{w}_r - \min(\tilde{\mathcal{W}}_r)}{\mathrm{range}(\tilde{\mathcal{W}}_r)} \cdot 255 \;. \tag{6.22}$$

Here, $\min(\tilde{\mathcal{W}}_r)$ defines minimum entry and $\mathrm{range}(\tilde{\mathcal{W}}_r)$ the range of the first variable over all $\tilde{\mathbf{w}} \in \tilde{\mathcal{W}}$ with $\tilde{\mathcal{W}}$ being the set of mapped prototypes.

Mapping Onto One Dimensional Color Scales Especially in the field of coloring univariate data, quite some effort has been put on defining color scales which are perceived as ordered and maximize the number of just noticeable differences, as has been discussed in section 6.3.3. For the application of encoding prototype similarities, mapping onto a one dimensional color scale has some advantages and disadvantages. The main disadvantage is that a mapping from a high-dimensional space to a one dimensional space will, by no means, faithfully represent the structure of the high-dimensional data. However, providing a one dimensional, ordered color scale simplifies the interpretation of the colors as only order in one dimension has to be evaluated instead of in three dimensions. Furthermore, if a color scale such as the rainbow scale is used, clusters can often be distinguished more easily. The color scales mentioned in section 6.3.3 and some additional ones are implemented in the TIS visualization tool and can be applied for data display. In this work, results are presented which result from mapping on the rainbow scale.

Mapping in the HSV Space via H^{2}SOM Projection As discussed briefly in section 6.2.3, the hyperbolic lattice structure of the HSOM and H^{2}SOM can be mapped entirely onto the

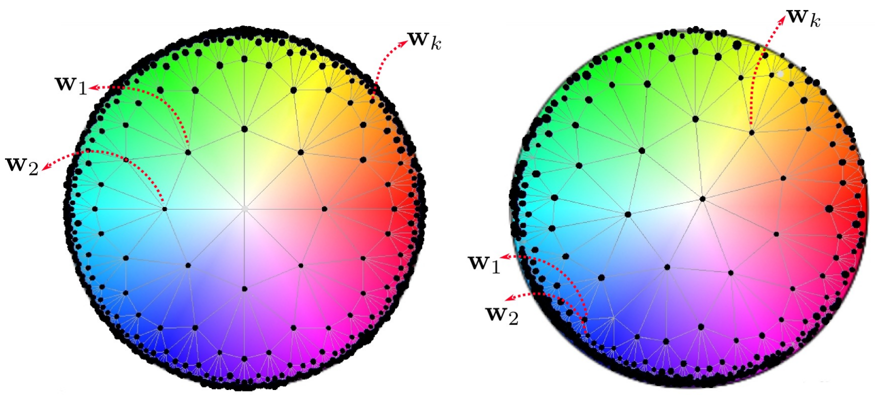


Figure 6.16: Projection of the hyperbolic lattice of the H^{2}SOM onto the HSV disc. (left) the origin of the lattice is projected onto the white point of the HSV disc. (right) Through a Möbius transformation the center of the lattice can be moved so that nodes lying at the rim of the disc are stretched and obtain more discriminable colors.

unit disc by means of the Poincaré model . A visualization which features a fish eye effect is obtained, faithfully representing the center of the map but squeezing the more distant regions. This representation of the lattice structure in the Poincaré disc offers a convenient possibility for a mapping in the HSV color space as has been proposed by Saalbach et al. (2005). Here, the hue and saturation values of the HSV color model are represented as a disc like structure, as depicted in figure 6.12. The Poincaré coordinates of each node of the lattice $\mathcal{L}$ can then be associated to a point on the HSV disc, assigning a HSV color to each lattice node, thus to each prototype, as can be seen in figure 6.16. This HSV color can easily be transformed to the primary RGB colors of the display.

It is, however, the nature of the mapping onto the Poincaré disc that the rim of the lattice, thus the outer levels, is squeezed and only the origin of $\mathbb{H}^2$ is faithfully represented. Thus, there is a large difference between colors assigned to prototypes of the first levels, whereas prototypes on the outer levels will receive quite similar colors which will not be easily distinguishable. Therefore, Saalbach et al. (2005) has proposed an interactive exploration and modification of the color. The focus of the lattice representation can be continuously adjusted by moving the center of $\mathbb{H}^2$, as depicted in figure 6.16 (right). Such a transformation can be achieved by means of a Möbius transformation (cf. Saalbach (2006); Ontrup (2008)). By moving the focus to a specific region of the lattice, the color contrast for the prototypes associated to these nodes will be increased while compressing other regions of the lattice. Although one level of the H^{2}SOM has to be chosen for clustering, in figure 6.16 the whole lattice is displayed so that the focus adaptation can better be perceived.

6.5 Statistical Comparison of Image Stacks

Through the visualizations introduced for the feature and the image domain, it can be analyzed which protein colocations are present in one image stack and what topological ordering

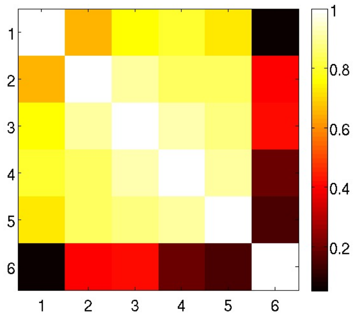


Figure 6.17: Correlation map: Visualization of correlation matrices via color encoding through a heated object scale.

they reveal. Furthermore, if the feature vectors extracted from different image stacks are clustered together, prototypes and topological ordering can be directly compared for different image stacks. Samples of diseased individuals may reveal different prototypes than samples of healthy individuals. They might, however, as well show the same prototypes, but with different abundances, i.e. number of associated data items, or different topological ordering. These differences can be analyzed by visual inspection of the clusters results, however, an additional statistical measure of similarity would be beneficial as it is not influenced by the humans' perceptual skills or varying expertise. To this end, image stack correlation is computed based on the abundance of prototypes in different stacks. Therefore, two statistical correlation measures can be applied: the *Pearson's Correlation Coefficient* and the *Spearman's Rank Correlation*. They allow to obtain an impression whether a linear correlation between prototype abundances of different image stacks exists.

6.5.1 Pearson's Correlation Coefficient

The most common measure to analyze linear correlations between two samples is the *Pearson's Correlation Coefficient* (Dalgaard (2008)). Let $\mathbf{v}^s = (v_1^s, v_2^s, \ldots, v_K^s)$ denote the vector of length K which holds the relative abundance for each prototype for image stack $\mathbf{I}^s$, i.e. the percentage of data items associated to that prototype with respect to all items of the image. The Pearson's Correlation Coefficient can then be calculated for two image stacks $\mathbf{I}^s$ and $\mathbf{I}^{s'}$ as

$$P(s,s') = \frac{\sum_{k=1}^{K}(v_k^s - \bar{\mathbf{v}}^s) * (v_k^{s'} - \bar{\mathbf{v}}^s)}{\sqrt{\sum_{k=1}^{K}(v_k^s - \bar{\mathbf{v}}^s)^2}\sqrt{\sum_{k=1}^{K}(v_k^{s'} - \bar{\mathbf{v}}^s)^2}} \ , \qquad (6.23)$$

where $\bar{\mathbf{v}}^s$ is the mean of $\mathbf{v}^s$. The coefficient $P(s,s')$ takes values between -1 and $+1$, where -1 indicates a perfect anti correlation, $+1$ a perfect correlation and 0 that no linear correlation is given. However, non linear correlation can be present. If correlations between multiple images stacks are computed, the correlation matrix obtained can be nicely displayed via a correlation map, as depicted in figure 6.17, where the correlation values are encoded through a heated object color scale.

6.5.2 Spearman's Rank Correlation Coefficient

A more robust correlation measure is the *Spearman's Rank Correlation*. Instead of using the concrete values v_k^s, their ranks r_k^s are considered. In case of *ties*, i.e. two values v_k^s receive the same rank place, their values are slightly blurred and the averaged rank place is assigned to both values. Similar to the Pearson's Coefficient the Spearman's Rank is defined as:

$$SP(s, s') = \frac{\sum_{k=1}^{K}(r_k^s - \bar{\mathbf{r}}^s) * (r_k^{s'} - \bar{\mathbf{r}}^{s'})}{\sqrt{\sum_{k=1}^{K}(r_k^s - \bar{\mathbf{r}}^s))^2}\sqrt{\sum_{k=1}^{K}(r_k^{s'} - \bar{\mathbf{r}}^{s'})^2}}\ , \tag{6.24}$$

with $\bar{\mathbf{r}}^s$ being the mean of the rank vector $\mathbf{r} = (r_1^s, r_2^s, \dots, r_K^s)$ holding the ranks of each prototype vector. Again, the coefficient takes values between -1 and $+1$, with $+1$ indicating perfect correlation, -1 perfect anti correlation and 0 no linear correlation. Applying ranks instead of concrete values makes the measure more robust against outliers and measurement errors.

6.6 Results

In the following I will present results obtained for the TIS image set $\mathcal{I} = \{\mathbf{I}^1, \mathbf{I}^2, \dots, \mathbf{I}^5\}$ achieved by the data mining and visualization approaches presented in this chapter.

For semantic image annotation, synapse detection was carried out as described in chapter 4. To obtain a nearly complete list of synapse positions, synaptic regions were detected in the syphys image of each stack and in two additional images, labeled for synap and nmdr1. This was motivated, as already described in section 2.2.2, by the fact that syphys is a stable marker for mature synapses but certain stages of synaptic plasticity might not be detected. Combining the detections of syphys, synap and nmdr1 to one master detection list Λ^s for each image stack $\mathbf{I}^s$ will cover almost all synaptic regions present (Schubert (2006)). As described in chapter 4, an i3S was trained on the syphys image of one stack $\mathbf{I}^s$. The trained i3S was then applied to the syphys, synap and nmdr1 channel of each stack in $\mathcal{I}$. Constant confidence thresholds were applied, however, each detection result was visually inspected and threshold parameters adapted if required to provide for the best detection result. The three detection lists $\Lambda^s_{\texttt{synap}}(t)$, $\Lambda^s_{\texttt{syphys}}(t)$, $\Lambda^s_{\texttt{nmdr1}}(t)$ obtained for each image stack $\mathbf{I}^s \in \mathcal{I}$ were combined to master detection lists Λ^s. Two positions $\mathbf{p} \in \Lambda^s_n(t)$ and $\mathbf{p}' \in \Lambda^s_m(t)$ were considered the same if their Euclidean distance $d(\mathbf{p}, \mathbf{p}') < 2$.

The image stacks $\mathbf{I}^1, \dots, \mathbf{I}^5$ were obtained from samples of different mice, however, each of the mice was treated the same way. Thus, one should expect to see quite similar protein patterns in each of the samples. To construct a sample which is supposed to be non-similar to the rest of the stacks, a detection list of non synaptic regions was constructed by randomly choosing $|\Lambda^3|$ positions $\mathbf{p}$ in image set $\mathbf{I}^3$ so that $d(\mathbf{p}, \mathbf{p}') > 3$ for $\mathbf{p}' \in \Lambda^3$. A set Λ^6 was obtained which subsumed non synaptic regions. Although these positions actually correspond to image set $\mathbf{I}^3$, it is referred to as the set of image $\mathbf{I}^6$ for clarity.

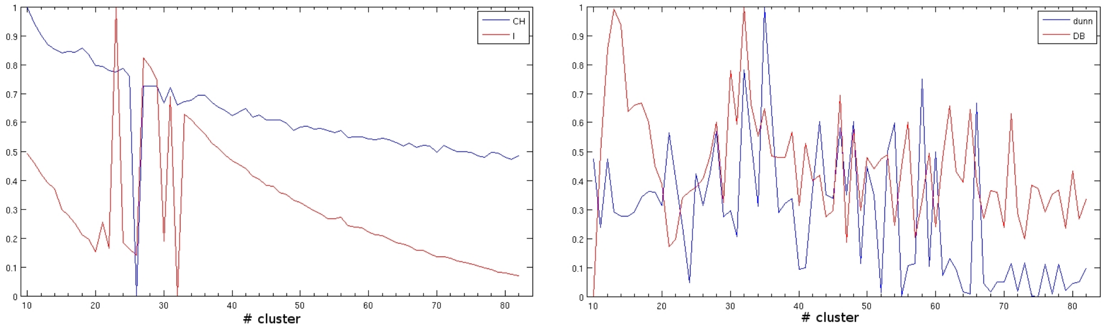
Image normalization and feature calculation was performed as described in section 6.1.1 and 6.1.2. In total six data sets were obtained, where $\mathcal{X}^1, \dots, \mathcal{X}^5$ correspond to data sets for synaptic regions and $\mathcal{X}^6$ belongs to the negative examples, i.e. non synaptic regions. In a first step, only 12 channels with high quality signals were used for feature calculation to ensure that meaningful feature vectors were obtained. This selection contained two proteins with non synaptic localization (`apoli, cat`) and 10 proteins with synaptic localization (`glur1, glur2, glur5, gap43, nmdr1, nnos, nr2a, nr2b, synap, syphys`). In a later study, all channels but the `prop` channel were used for feature calculation. The `prop` channel is used in this imaging setup solely as an orientation marker.

The data sets $\mathcal{X}^1, \ldots, \mathcal{X}^6$ were joined together into one data set $\mathcal{X}$ for clustering. Thus, clustering was performed conjointly for all six sets and the obtained clusters and prototypes could directly be compared across samples. In general, also other cluster setups are possible. For example, each data set could be clustered separately while colors are assigned conjointly to all data sets. However, this would not allow to directly compare prototype abundances between image stacks and thus correlation measures could not be computed. Alternatively, the prototypes obtained by individual clustered of each image could be used as a data basis for a "meta" clustering. Hence, first each data set would be approximated by its own set of prototypes which would be merged in the second meta clustering step. Prototype abundances would then be comparable for all images. Both alternative strategies were tested for the data sets. However, no major differences between the obtained clusters and prototypes could be observed so that the straight forward and reasonable strategy of jointly clustering all data sets is presented in this work. Clustering was performed by K-means, with an exponentially decaying learning rate with $\epsilon_i = 0.9, \epsilon_f = 0.01$ and 40 times as many iterations as items in the data set. Clustering with K-means took approximately 10 seconds on an Intel Pentium 4, 2.53 GHz, 2GB RAM. The same parameters were used for the NG clustering, with the additional neighborhood parameters $\lambda_i = K/2, \lambda_f = 0.01$. With this strategy, half of all prototypes are adapted at the beginning of the clustering, resulting in a fast movement of the prototypes in the feature space. NG clustering took approximately 30 seconds for the given data set. Additionally, a H^{2}SOM clustering was performed. It was again parameterized with an exponentially decaying learning rate with $\epsilon_i = 0.9, \epsilon_f = 0.01$, and $\sigma_i = 9.171425, \sigma_f = 0.764285$. Training took approximately 1 minute. In the following, the results obtained for each cluster approach will be presented.

6.6.1 Estimating the Appropriate Number of Clusters

To estimate how many clusters should be used for the given data set, the CH, $\mathcal{I}$, DB and Dunn validity indices were computed for a range of 10 to 80 clusters. This was done both for the K-means as well as the NG approach, however, not for the H^{2}SOM since the hierarchical level implicitly specifies the number of clusters. Figure 6.18 presents the indices obtained for the K-means clustering with (a) showing the CH and $\mathcal{I}$ indices and (b) the DB and Dunn indices. The CH and $\mathcal{I}$ indices for the NG clustering are displayed in figure 6.19 (a) and the Dunn and DB indices are shown in figure 6.19 (b).

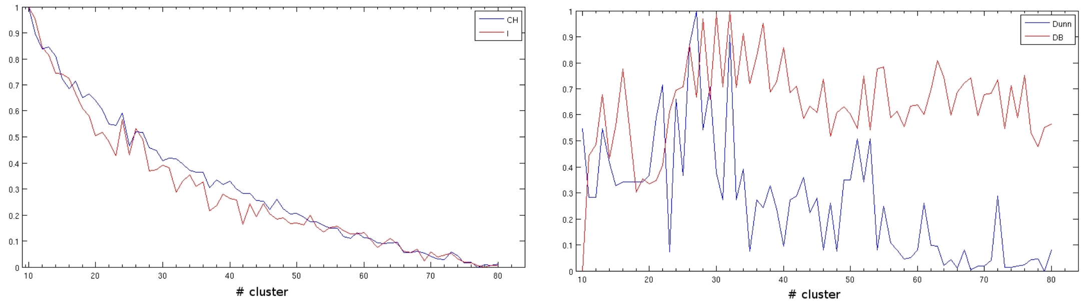
When analyzing the indices obtained for K-means clustering, it is clear to see that for the CH and DB indices, the best cluster configuration would be 10 (see figure 6.18(a) blue line and (b) red line). Recall that, contrary to the other indices, a low DB index represents a good cluster configuration. The indices $\mathcal{I}$ and Dunn have higher variations, with peaks at



(a)

(b)

Figure 6.18: Cluster validity indices for 10 to 80 clusters obtained through K-means clustering. (a) CH and $\mathcal{I}$, (b) Dunn and DB.



(a)

(b)

Figure 6.19: Cluster validity indices for 10 to 80 clusters obtained through NG clustering. (a) CH and $\mathcal{I}$, (b) Dunn and DB.

23, 27 and 31 clusters for $\mathcal{I}$ (figure 6.18(a) red line) and for 36 clusters for Dunn (figure 6.18(b) blue line). For the NG clustering, DB, CH as well as $\mathcal{I}$ show the best values for 10 clusters. However, $\mathcal{I}$ shows slight peaks at 24 and 26 clusters (see figure 6.19(a) red line). The Dunn index shows peaks at 27 and 32 clusters (figure 6.19(b) blue line).

Comparing the indices obtained for one clustering method with each other, it can be seen that they do not agree in what is the best cluster number. Especially for K-means clustering and index $\mathcal{I}$, there are several configurations which could be suitable and the index varies considerably between individual cluster configurations in the range of 20 to 35. This can be due to the fact that K-means is fragile with respect to the cluster initialization. Therefore, the indices of multiple initializations of K-means were compared to each other, however, there was no overall consensus between the indices obtained. Yet, good values ranged between 27 and 35 clusters.

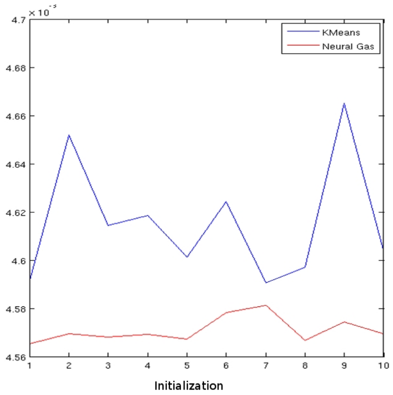


Figure 6.20: Within cluster point scatter obtained for K-means (blue) and NG clustering (red) for ten random initializations.

I therefore have chosen a cluster number in the range of the number of clusters which were identified as optimal by the different indices. In the following, clustering was performed with 30 clusters. It furthermore has the advantage over higher cluster numbers that fewer visualizations of prototypes and clusters need to be analyzed which eases the interpretation. It is, however, straight forward to increase or decrease the number of clusters if the visual interpretation of the cluster results reveals that too few clusters were chosen and highly different feature vectors are grouped.

6.6.2 Stability of Cluster Results

It is known that the K-means algorithm can be highly influenced by its cluster initialization. Therefore, multiple initializations are usually carried out and the one with the best value for the objective function is chosen as the clustering result (Hastie et al. (2001), p. 463). To study the stability of the K-means clustering, the algorithm was initialized ten times with a randomly chosen sub set of the data set. The stability was tested in terms of the objective function, i.e. within cluster point scatter (wps) $\sum_{k=1}^{K} |C_k| \sum_{\mathbf{x} \in C_k} d(\mathbf{x}, \boldsymbol{\mu}_k)$ (Hastie et al. (2001), p. 462), as well as in terms of the image stack correlations and obtained prototypes. Although the NG algorithm better compensates bad initializations, its performance was also tested for ten runs.

Figure 6.20 shows the wps for each of the ten initializations for K-means and NG. It is evident that the NG is less influenced by the choice of the initialization,i.e. the wps shows lower variation, and a lower overall wps is obtained than for the K-means algorithm (see 6.20(b) vs. (a)). Calculation of the wps is only one way to estimate the cluster stability and several other measures, as the validity indices described above, could be used. Yet, the wps is a common choice (Hastie et al. (2001)).

To get an impression of the stability of the cluster results with respect to the obtained image stack correlations, the Spearman's rank correlation was calculated for each run. For a stable clustering it would be expected that the calculated image correlations are also of stable

	$\mathbf{I}^1$	$\mathbf{I}^2$	$\mathbf{I}^3$	$\mathbf{I}^4$	$\mathbf{I}^5$	$\mathbf{I}^6$
$\mathbf{I}^1$	1.00±0.000	**0.65±0.040**	**0.58±0.055**	**0.64±0.066**	**0.55±0.046**	**0.29±0.049**
$\mathbf{I}^2$	0.65±0.040	1.00±0.000	**0.95±0.014**	**0.90±0.017**	**0.88±0.023**	**0.53±0.029**
$\mathbf{I}^3$	0.58±0.055	0.95±0.014	1.00±0.000	**0.94±0.014**	**0.94±0.012**	**0.54±0.031**
$\mathbf{I}^4$	0.64±0.066	0.90±0.017	0.94±0.014	1.00±0.000	**0.93±0.015**	**0.39±0.048**
$\mathbf{I}^5$	0.55±0.046	0.88±0.023	0.94±0.012	0.93±0.015	1.00±0.000	**0.45±0.043**
$\mathbf{I}^6$	0.29±0.049	0.53±0.029	0.54±0.031	0.39±0.048	0.45±0.043	1.00±0.000

Table 6.1: Average rank correlation and standard deviation obtained for K-means clustering over 10 random initializations.

	$\mathbf{I}^1$	$\mathbf{I}^2$	$\mathbf{I}^3$	$\mathbf{I}^4$	$\mathbf{I}^5$	$\mathbf{I}^6$
$\mathbf{I}^1$	1.00±0.000	**0.58±0.074**	**0.52±0.056**	**0.56±0.075**	**0.48±0.060**	**0.22±0.051**
$\mathbf{I}^2$	0.58±0.074	1.00±0.000	**0.94±0.017**	**0.88±0.030**	**0.86±0.031**	**0.48±0.003**
$\mathbf{I}^3$	0.52±0.056	0.94±0.017	1.00±0.000	**0.92±0.015**	**0.92±0.015**	**0.50±0.035**
$\mathbf{I}^4$	0.56±0.075	0.88±0.030	0.92±0.015	1.00±0.000	**0.91±0.020**	**0.32±0.029**
$\mathbf{I}^5$	0.48±0.060	0.86±0.031	0.92±0.015	0.91±0.020	1.00±0.000	**0.40±0.039**
$\mathbf{I}^6$	0.22±0.051	0.48±0.033	0.50±0.035	0.32±0.029	0.30±0.039	1.00±0.000

Table 6.2: Average rank correlation and standard deviation obtained for NG clustering over 10 random initializations.

nature and change only slightly. Table 6.1 and 6.2 display the average correlations over all ten initializations obtained for K-means and NG as well as their standard deviation. When comparing table 6.1 and 6.2, one can see that both methods are quite stable with respect to the correlations and feature quite low standard deviations, with slightly lower values for K-means.

In addition, the obtained prototypes were visually inspected and compared across the ten runs. Although they slightly change in the overall profile height, i.e. protein signals, only few differences could be observed for the prototypes obtained.

6.6.3 Analysis of Protein Colocation and Inter Image Correlation

In this section I will focus on the visual analysis of the obtained clustering results and feature vectors. In general, each visualization can be used as a starting point for the investigation of TIS data. However, for presentation of the results, I will follow the Shneiderman mantra of "Overview first, zoom in and filter, details on demand" which has also proven to be an adequate way to analyze TIS data. Figure 6.21 schematically displays the analysis approach. (a) First cluster maps are inspected to study the distribution of clusters within and between clusters and potentially observe interesting topological orderings. (b) To verify findings made in the inter image analysis or to get an impression on the image stack correlations, correlation maps and pie charts can be analyzed. (c) Furthermore, prototypes showing

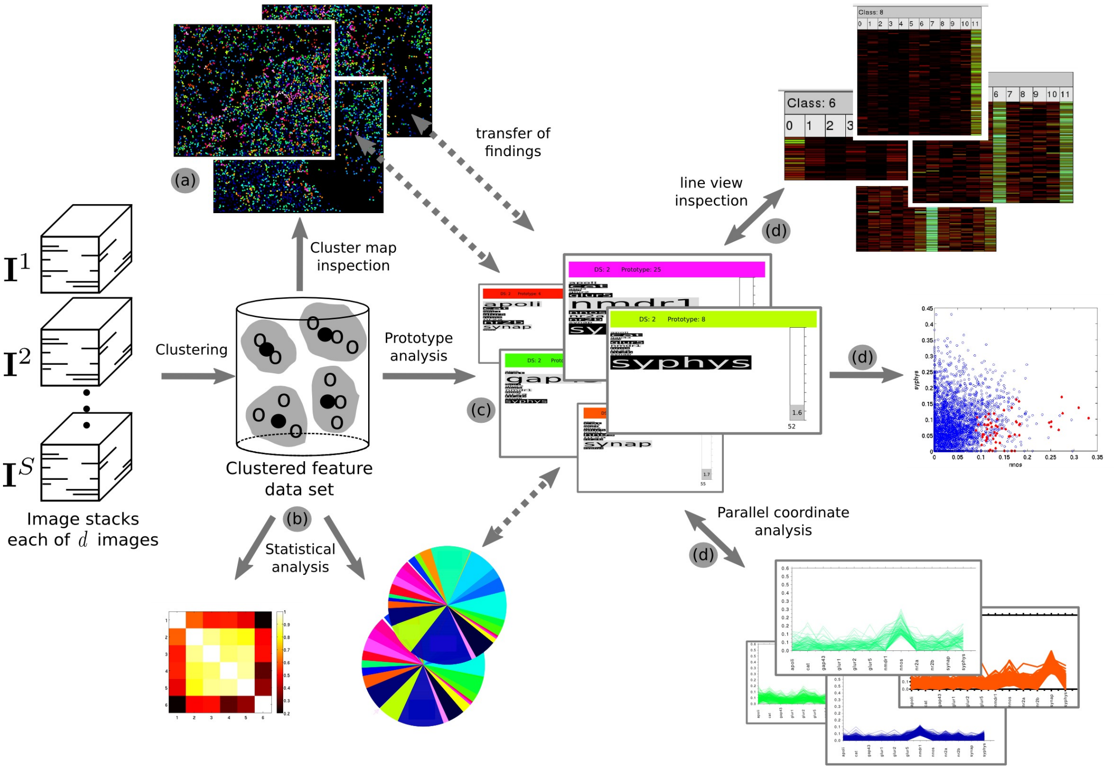
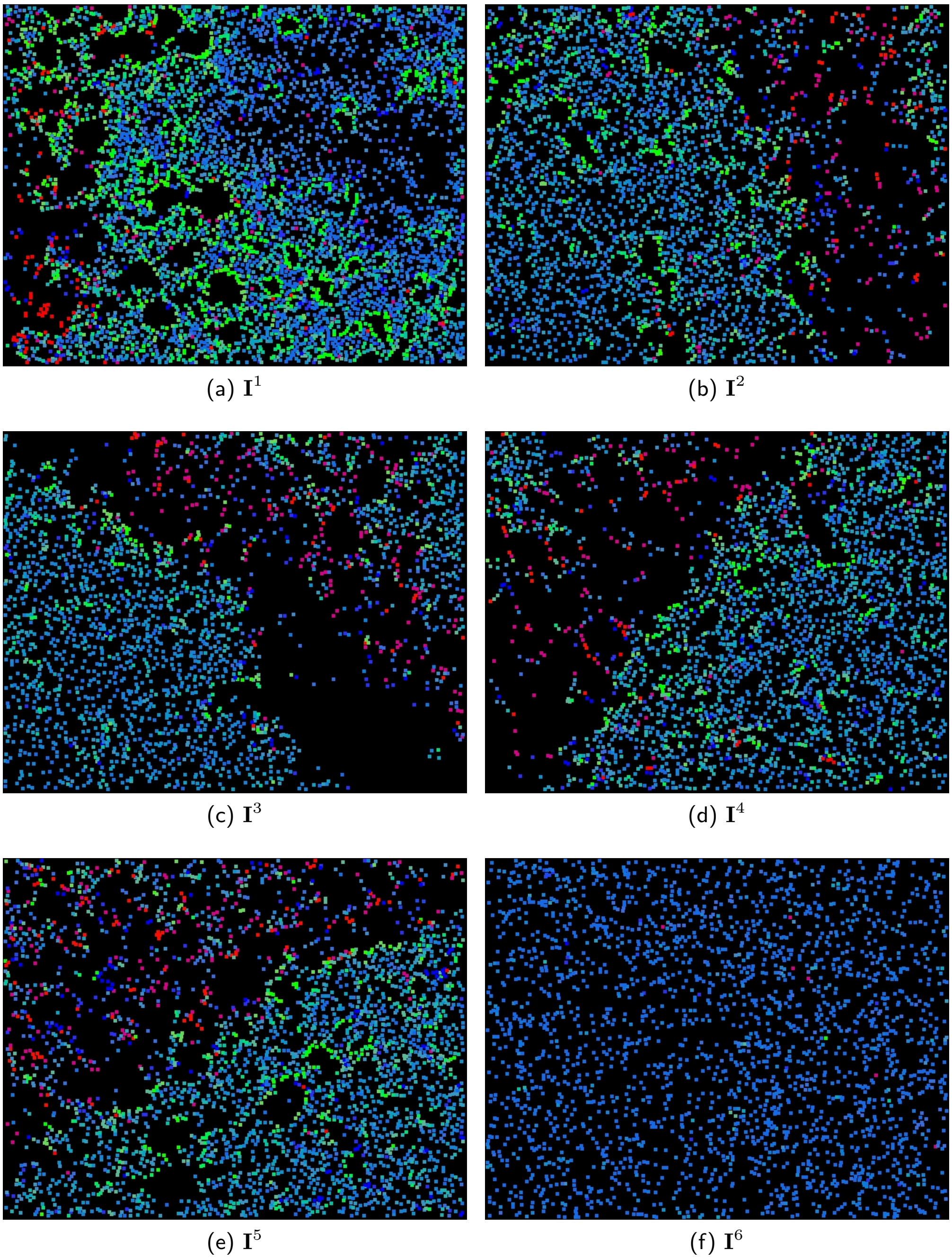


Figure 6.21: Overview of the analysis workflow for TIS images as applied in this study. The data set is clustered and the result is visualized as cluster maps (a), pie charts and correlation maps (b), prototype visualizations (c) or in detail visualizations as line views, parallel coordinates and scatter plots (d). For TIS analysis, it is appropriate to investigate overviews in a first step and then findings can be verified by analyzing detailed visualizations.

interesting topological ordering or causing differences between image stacks can then be observed in more detail in the prototype visualization. (d) To get an in detail impression of the underlying data items themselves and the cluster configurations, parallel coordinate plots, line views or scatter plots of selected proteins can be analyzed. In the following, I will exemplarily analyze the clustering result achieved through NG clustering in more detail following the analysis pipeline mentioned earlier.

Figure 6.22 shows the rendered cluster maps for all six image stacks, including the negative example. Coloring was obtained by Sammon's mapping of the prototype vectors (omitting the bias term) into the three-dimensional RGB color space as described in section 6.4 with a scalar product as distance metric. With this color encoding, it can very well be seen that the cluster maps belonging to image sets $\mathbf{I}^1, \ldots, \mathbf{I}^5$ decompose into two different regions (see figure 6.22 (a)-(e)). One mainly subsumes clusters encoded in red color whereas the other region features clusters encoded in bluish and greenish colors. This separation can also be seen, although not as clear, in the cluster maps generated by PCA projection (see supplementary figure A.3 (a)-(e)), where the two regions fall into bluish colored clusters



(a) $\mathbf{I}^1$

(b) $\mathbf{I}^2$

(c) $\mathbf{I}^3$

(d) $\mathbf{I}^4$

(e) $\mathbf{I}^5$

(f) $\mathbf{I}^6$

Figure 6.22: Cluster maps generated for each image through Sammon's mapping into the 3D RGB space. Each color encodes the cluster association of that synaptic region.

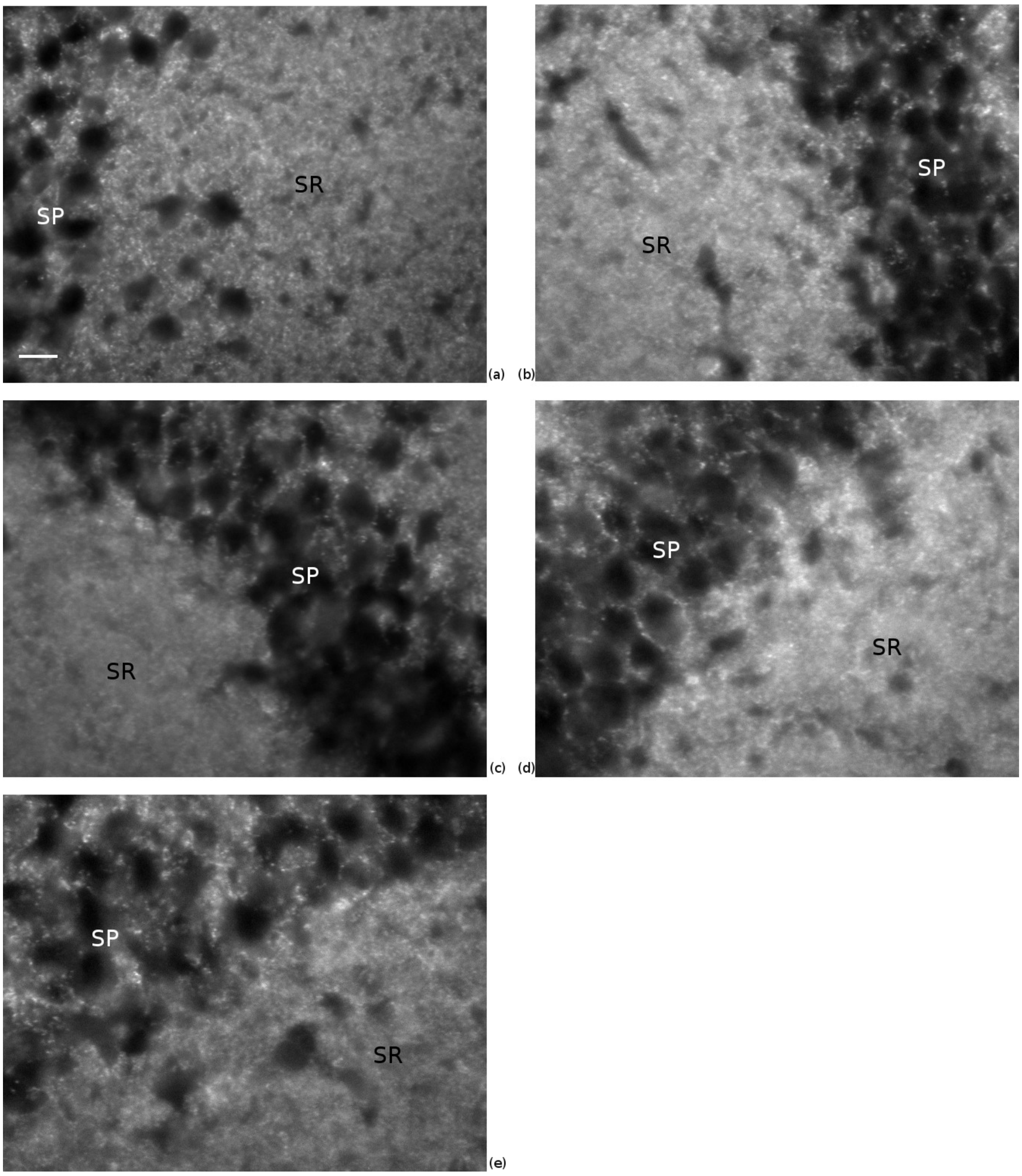
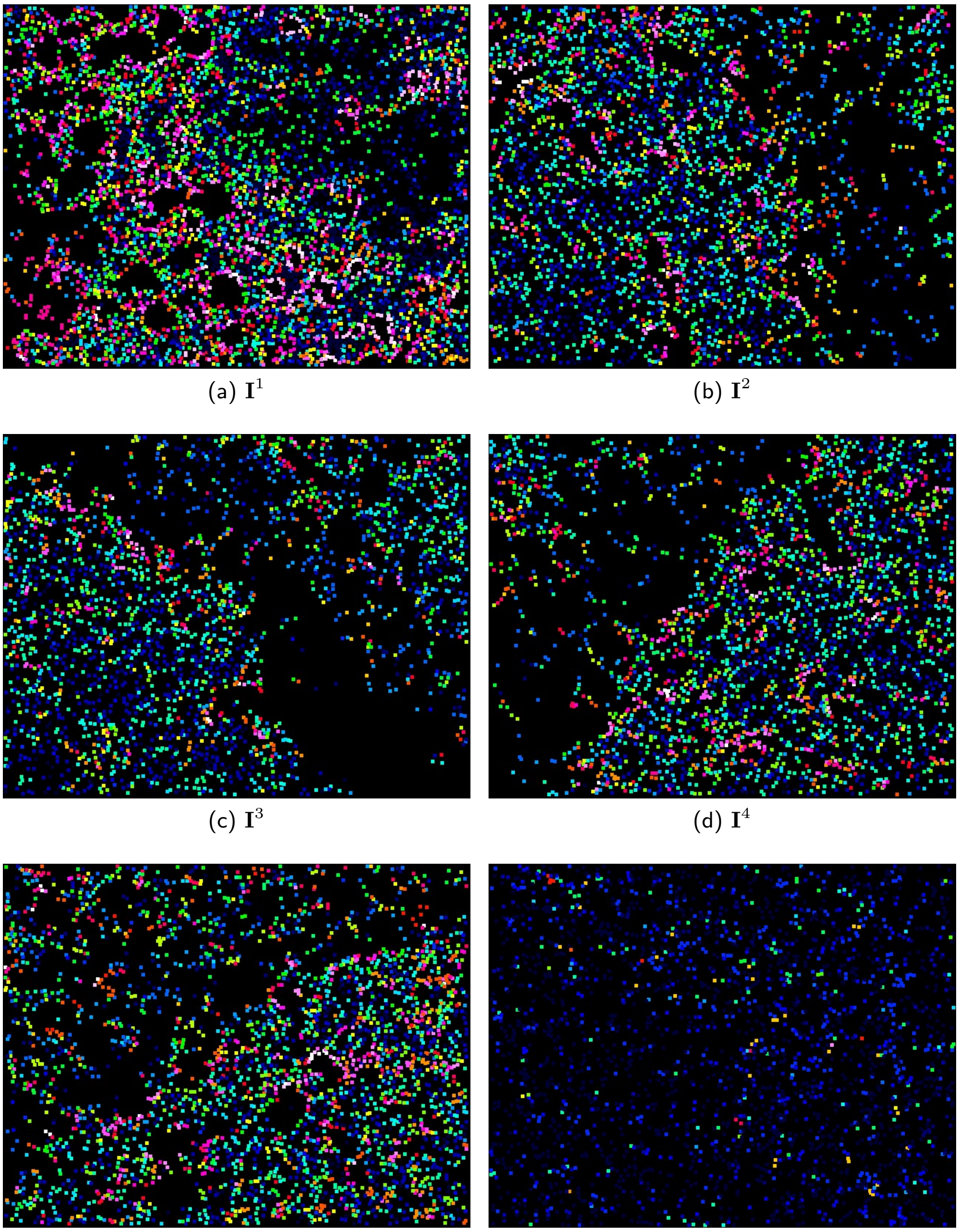


Figure 6.23: Images obtained by staining with syphys for each of the five image stacks (a) $\mathbf{I}^1$, (b) $\mathbf{I}^2$, (c) $\mathbf{I}^3$, (d) $\mathbf{I}^4$, (e) $\mathbf{I}^5$. An image is not shown for $\mathbf{I}^6$ as it is the same as $\mathbf{I}^3$. Scale bar of 10.8 μm refers to all images.



(a) $\mathbf{I}^1$

(b) $\mathbf{I}^2$

(c) $\mathbf{I}^3$

(d) $\mathbf{I}^4$

(e) $\mathbf{I}^5$

(f) $\mathbf{I}^6$

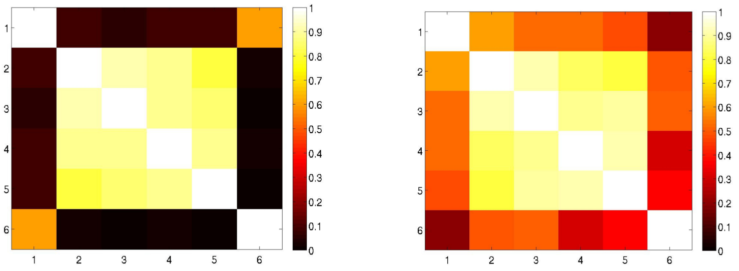
Figure 6.24: Cluster maps obtained for NG clustering with 12 proteins by mapping onto a 1D pseudo color scale.

versus purple and beige clusters. If one compares these separation with the general structure of the tissue samples, which can nicely be seen when observing the synaptophysin channel (figure 6.23), it becomes evident that these two regions seen in the cluster map coincide with the SR and SP regions. Although the negative data set $\mathcal{X}^6$ was extracted from the same image set $\mathbf{I}^3$ as data set $\mathcal{X}^3$, with the only difference that only non synaptic regions were used for feature calculation, red colored prototypes belonging to the SP region can not be observed (see figure 6.22 (f)). Thus, SP specific prototypes must represent protein colocation specific for synapses located in the SP region rather than the overall SP characteristics itself.

Because of the lack of SP prototypes in sample $\mathbf{I}^6$, one has the impression that image $\mathbf{I}^1, \ldots, \mathbf{I}^5$ are more equal to each other than to image $\mathbf{I}^6$. However, especially in the SR region it is very hard to judge the diversity of prototypes as they are all colored very similarly. Furthermore it is not clear if the blue colors of $\mathbf{I}^1, \ldots, \mathbf{I}^5$ encode the same prototypes as in image $\mathbf{I}^6$. This could be clarified by an interactive analysis, however, for purpose of representation and easier visual interpretation a mapping onto a 1D rainbow scale with equidistant spacing between the prototypes is applied. The cluster maps obtained are depicted in figure 6.24. It is evident that the separation between SR and SP tissue regions is not well visible anymore, however, it can now better be perceived that the SR region contains synapses belonging to several different prototypes. The separation between the negative case and the other five cases now becomes even more evident as almost only blue colors are present in the negative case (see figure 6.24(f) versus (a)-(e)).

This visual finding can now be verified through analysis of image stack correlations, as the result is not influenced by the chosen color encoding and visual impression but is solely based on the clustering result itself. Figure 6.25 (a) and (b) display the squared Pearson's correlation and the Spearman's Rank correlation, respectively, as a correlation map with a heated object color scale for display of the correlation values (corresponding tables can be found in the appendix. Table A.4 and A.4). In both cases, $\mathbf{I}^2, \mathbf{I}^3, \mathbf{I}^4$, and $\mathbf{I}^5$ are highly correlated, whereas $\mathbf{I}^1$ and $\mathbf{I}^6$ show lower correlations to those images. It can be observed that the rank correlations for $\mathbf{I}^1$ and $\mathbf{I}^6$ is higher than the Pearson's correlation, suggesting a non linear relationship between the data sets (Estelberger and Reibnegger (1995)). Correlation between $\mathbf{I}^1$ and images $\mathbf{I}^2, \ldots, \mathbf{I}^5$ in most cases is higher than the correlation between $\mathbf{I}^6$ and $\mathbf{I}^2, \ldots, \mathbf{I}^5$. It is interesting to find correlations of up to 0.92 as one has to remember that images of tissue samples are analyzed. Each sample shows individual variation in their tissue composition, especially as two different regions, SR and SP, are present in one sample. Still, synaptic protein colocation seems to be preserved and stable across images.

The correlation analysis gives a good impression on how similar the images are to each other based on the obtained clusters and thereby based on their protein colocation at synaptic sites. To get a more detailed insight in the differences between the clusters of different stacks, especially between $\mathbf{I}^1$ and images $\mathbf{I}^2, \ldots, \mathbf{I}^5$, pie charts were rendered. To this end, each pie chart displays the percentage of synapses belonging to each cluster for each stack. Figure 6.26 gives an overview of these distributions for all six stacks where each slice is colored according to the prototype color obtained via mapping on the 1D rainbow scale which corresponds to figure 6.24. This coloring eases the interpretation of the pie chart as different slices can be distinguished more easily from each other. Again, the correspondence between $\mathbf{I}^2, \ldots, \mathbf{I}^5$



(a) Squared Pearson's correlation

(b) Spearman's rank correlation

Figure 6.25: Correlation between the images $\mathbf{I}^1, \dots, \mathbf{I}^6$ based on their cluster frequency. (a) Squared Pearson's correlation. (b) Spearman's rank correlation. $1 = \mathbf{I}^1, 2 = \mathbf{I}^2, 3 = \mathbf{I}^3, 4 = \mathbf{I}^4, 5 = \mathbf{I}^5, 6 = \mathbf{I}^6$.

can be observed very well, which show almost identical pie charts (see figure 6.26). This has also been reflected in the high image stack correlations calculated for those image stacks, both with the Pearson as well as the Spearman's rank. With the pie chart display, however, now the reasons for low correlations in stack $\mathbf{I}^1$ and $\mathbf{I}^6$ can be explored. It can be seen that $\mathbf{I}^1$ has a larger amount of synapses belonging to prototype 23, but fewer belonging to 21 than the other stacks. Furthermore, a higher percentage of synapses belongs to prototypes 12, 14 and 28 whereas fewer synapses are observed for prototypes 1, 3 and 5. There exist more such differences between $\mathbf{I}^2, \dots, \mathbf{I}^5$ and $\mathbf{I}^1$, however, these are the most obvious ones. For set $\mathbf{I}^6$ it can clearly be seen that the majority (82.6%) of synapses belong to prototype 23, 16 and 9.

Up to now, by analyzing the cluster maps, correlation maps and pie charts which all give a good overview of the data, several interesting observations were made. First, prototypes were identified which separate synapses of SP and SR. Second, high correlations between images $\mathbf{I}^2, \dots, \mathbf{I}^5$ were observed and prototypes were identified which lead to differences between image stacks. Now, these observations can be studied in more detail by inspecting the prototype visualizations.

I will start with the analysis of prototypes belonging to SR and SP regions to see how these regions differ with respect to their protein colocation profiles. Figure 6.27 displays the CIPRA visualizations for those prototypes which belong almost exclusively either to the SP or SR image region. Overlays of the cluster maps and the `nmdr1` channel of image $\mathbf{I}^3$ (figure 6.27 left) and $\mathbf{I}^5$ (figure 6.27 right) are provided for topological analysis of prototype distribution. Colors were assigned to the prototypes so that different colors can easily be distinguished from each other. Hence, the individual profiles are treated more like nominal variables as it is done for the generation of toponome maps (see sec. 3.1). However, it was taken care that colors of prototypes belonging to one tissue region were similar to each other with respect to their color encoding. The abundance information provided in the CIPRAs of figure 6.27 correspond to image $\mathbf{I}^5$. When analyzing the CIPRA visualizations of those prototypes whose associated synapses are located mostly in the SP area (see figure 6.27 top CIPRAs), it is interesting to note that those synapses feature high levels of `syphys` or `synap`,

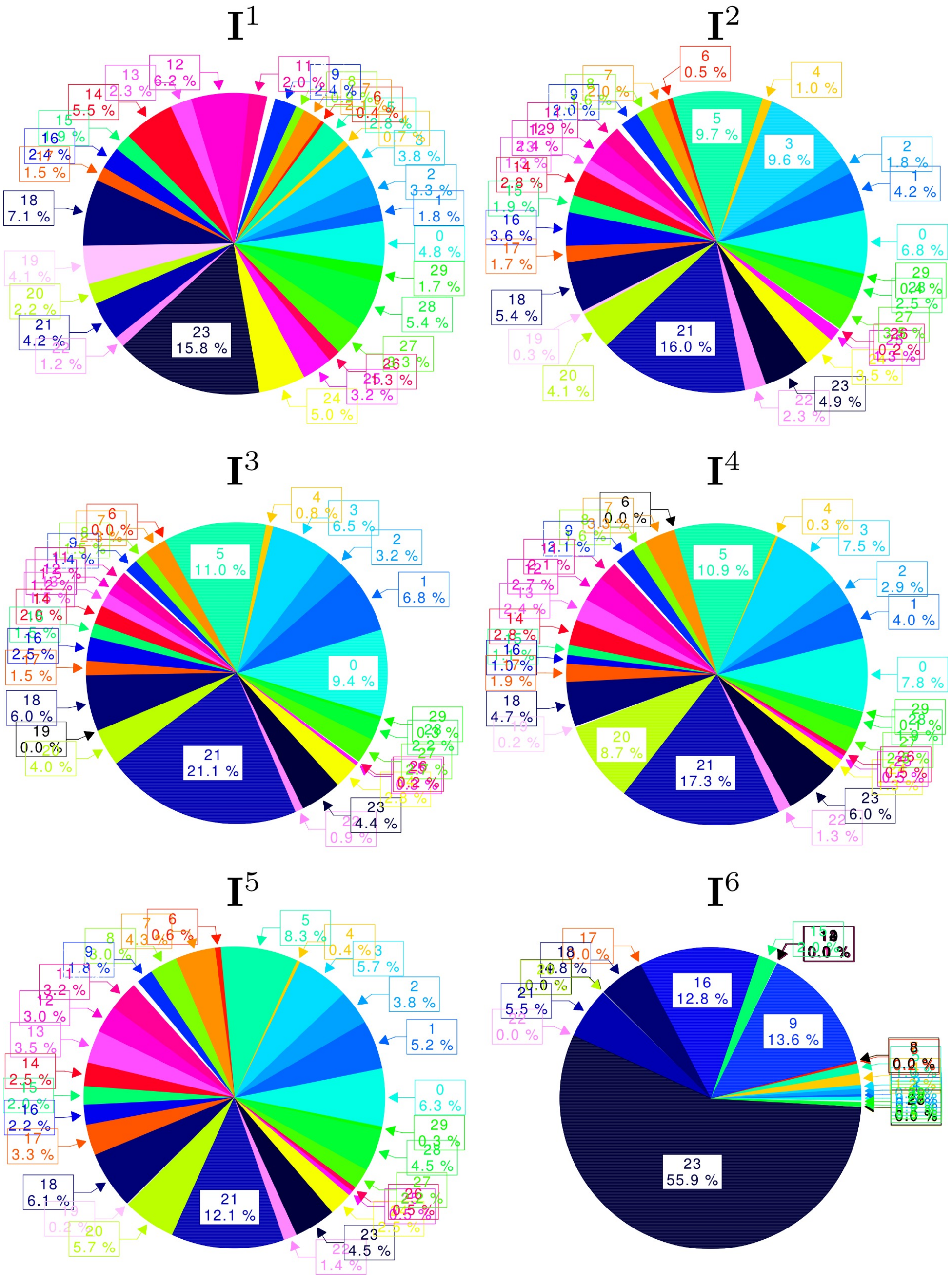


Figure 6.26: Visualization of the relative abundances of prototypes for each image stack via pie charts. Individual slices of the pie were colored according to the mapping onto a 1D rainbow scale.

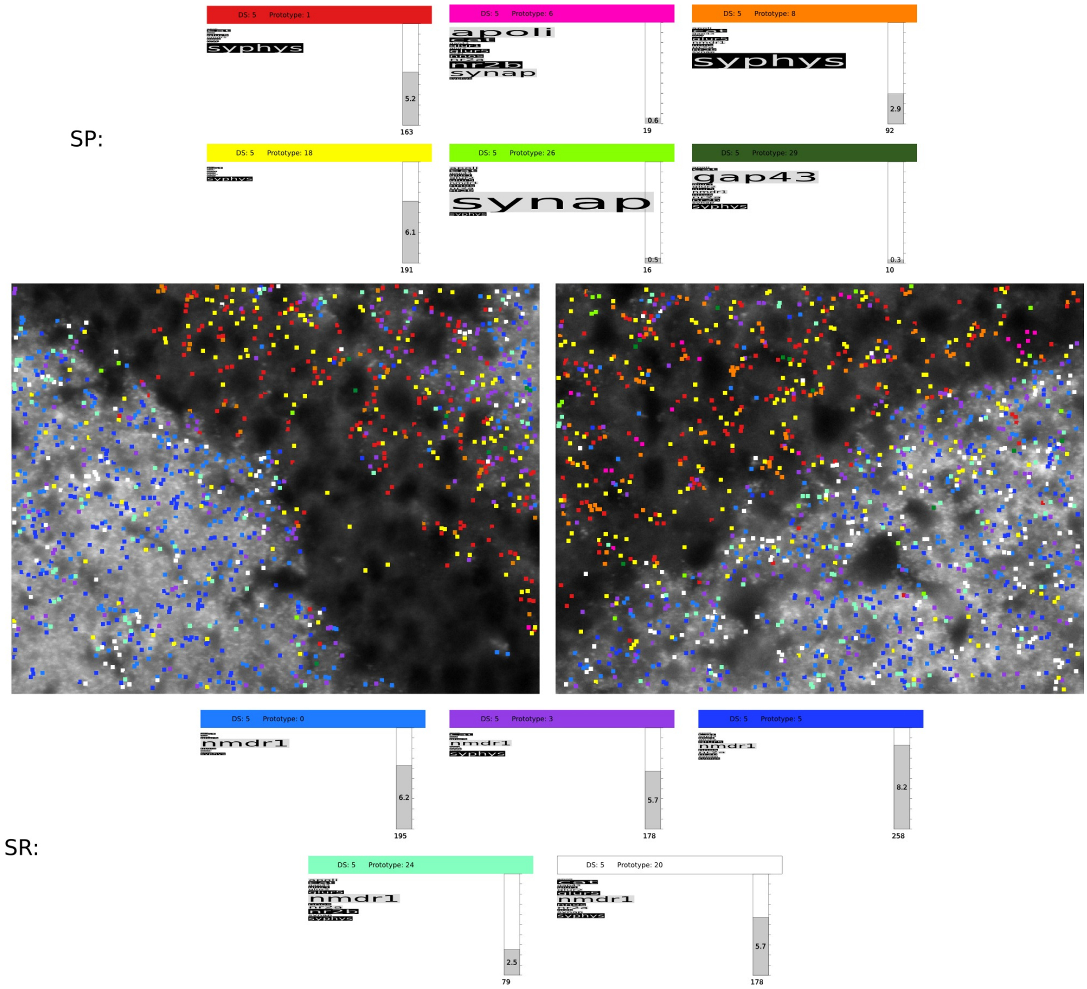


Figure 6.27: CIPRA and cluster map visualization of those prototypes located solely in the SP or SR area. Colors were assigned so that discrimination between prototypes can easily be achieved.

combined with other proteins such as `apoli`, `nr2b`, `gap43` and `glur5`. Prototypes of the SR region (see figure 6.27 bottom CIPRAs) all show high levels of `nmdr1` in combination with lower levels of `syphys`, `nr2b`, `cat` and `glur5`. Thus, absence of `nmdr1` and high levels of `syphys` or `synap` seem to be characteristics of purely SP synapses.

Now I would like to analyze in more detail the profiles of prototypes leading to differences between individual image stacks. To this end, figure 6.28 gives an CIPRA overview of the 30 prototypes obtained. CIPRAs are sorted according to prototype number for easier association of prototype numbers. It is, however, also reasonable to sort CIPRAs according to their color so that similar CIPRAs are located close to each other. To this end, the hue of the color can be applied for sorting. Thereby, discontinuities also introduced by the mapping in the low-dimensional color space can be more easily perceived. The first observation made was an almost inverted abundance between prototype 23 and 21 for $\mathbf{I}^1$ versus $\mathbf{I}^2, \ldots, \mathbf{I}^5$. By

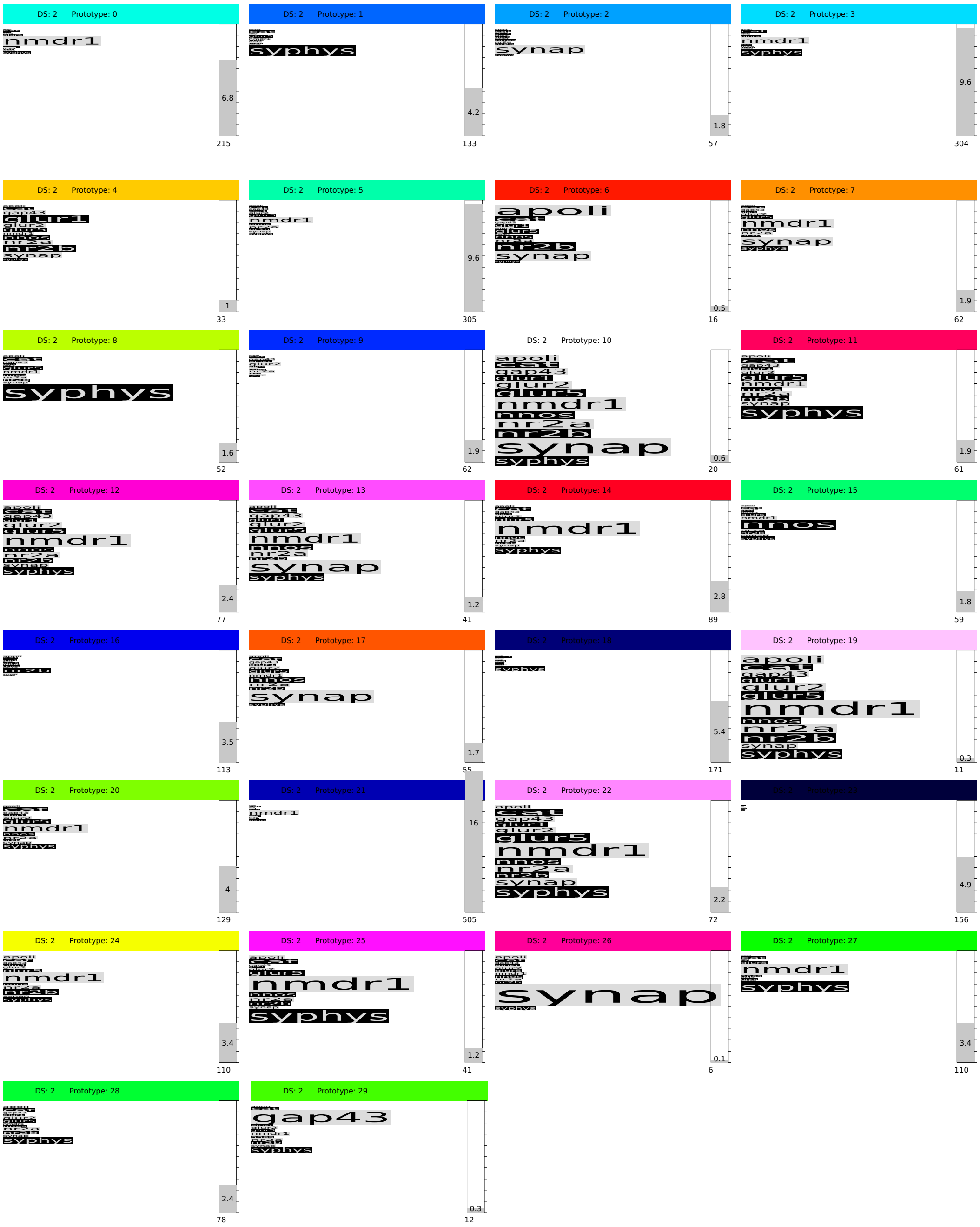


Figure 6.28: CIPRA visualization of NG clustering with 30 prototypes.

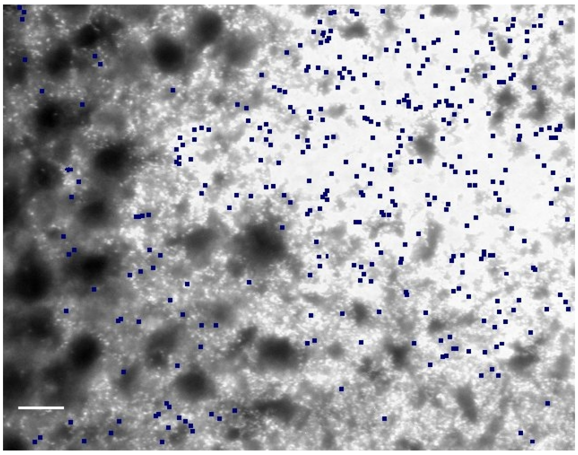


Figure 6.29: Highlighting of synaptic regions associated to prototype 23 overlaid on the `nmdr1` channel of $\mathbf{I}^1$. Scale bar is 10.8 μm.

analyzing prototypes 21 and 23 in figure 6.28 it can be seen that prototype 21, similar to prototype 5 which was also identified as a separating prototype with low abundance in $\mathbf{I}^1$, has signal almost solely for `nmdr1` whereas prototype 23 has signals rarely in any of the channels. Hence, $\mathbf{I}^1$ seems to lack signal for `nmdr1` in many synapses. This phenomenon can now be inspected by analyzing the spatial locations of synapses belonging to prototype 23 as an overlay to the `nmdr1` channel of $\mathbf{I}^1$ (see figure 6.29). It can be observed that these synapses mostly fall within a region where the signal of the channel is too high and not trustworthy. Hence, no meaningful features can be extracted at these sites as the background signal is almost as high as the synapse signal itself. This results in a higher percentage of synapses with very low intensities in their feature vectors. Another observation made for image $\mathbf{I}^1$, has been the high abundance of prototypes 12, 14 and 28, and the low abundance of prototypes 1 and 3 compared to $\mathbf{I}^2, \ldots, \mathbf{I}^5$. It can be observed that prototypes 3, 12 and 14 all show signals for `nmdr1` and `syphys`, however with differing intensity. Prototype 1 and 28 both show high signal only for `syphys` with varying signal heights. It might therefore be the case that differences between those images are mainly caused by the fine nuances inherent in the non-binary feature vectors and by the fact that image $\mathbf{I}^1$ features a smaller SP region than the remaining samples. Concerning image $\mathbf{I}^6$, it has been observed previously that high levels of prototype 23, 16 and 9 and the lack or low level of other prototypes causes the separation of $\mathbf{I}^6$ from the other samples. Similar to prototype 23, prototype 9 only shows very weak intensities (see figure 6.28), thus no meaningful protein colocation can be analyzed. Interestingly, prototype 16 shows signal for `nr2b`, a marker located at synaptic sites. This prototype is very rare in the other stacks, suggesting that synapses featuring `nr2b` were not captured by synapse detection in `synap, syphys` and `nmdr1`. When analyzing the image intensities at prototype 16 locations in the `nr2b` channel, many of them are true signals however some are also caused by the high levels of noise inherent in this channel.

Up to now, the analysis of individual prototypes was triggered by findings in the image

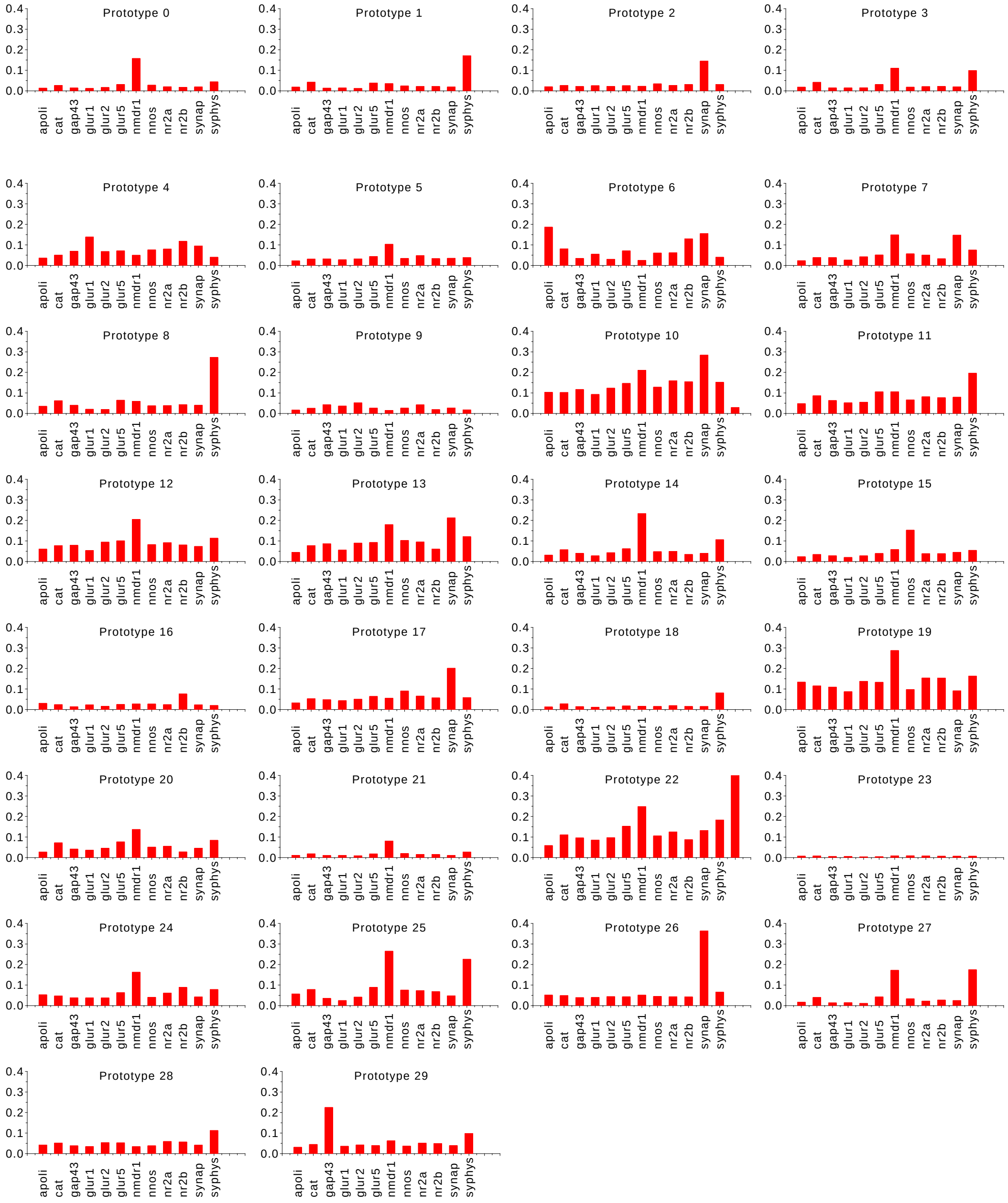


Figure 6.30: Prototype visualization via bar graphs.

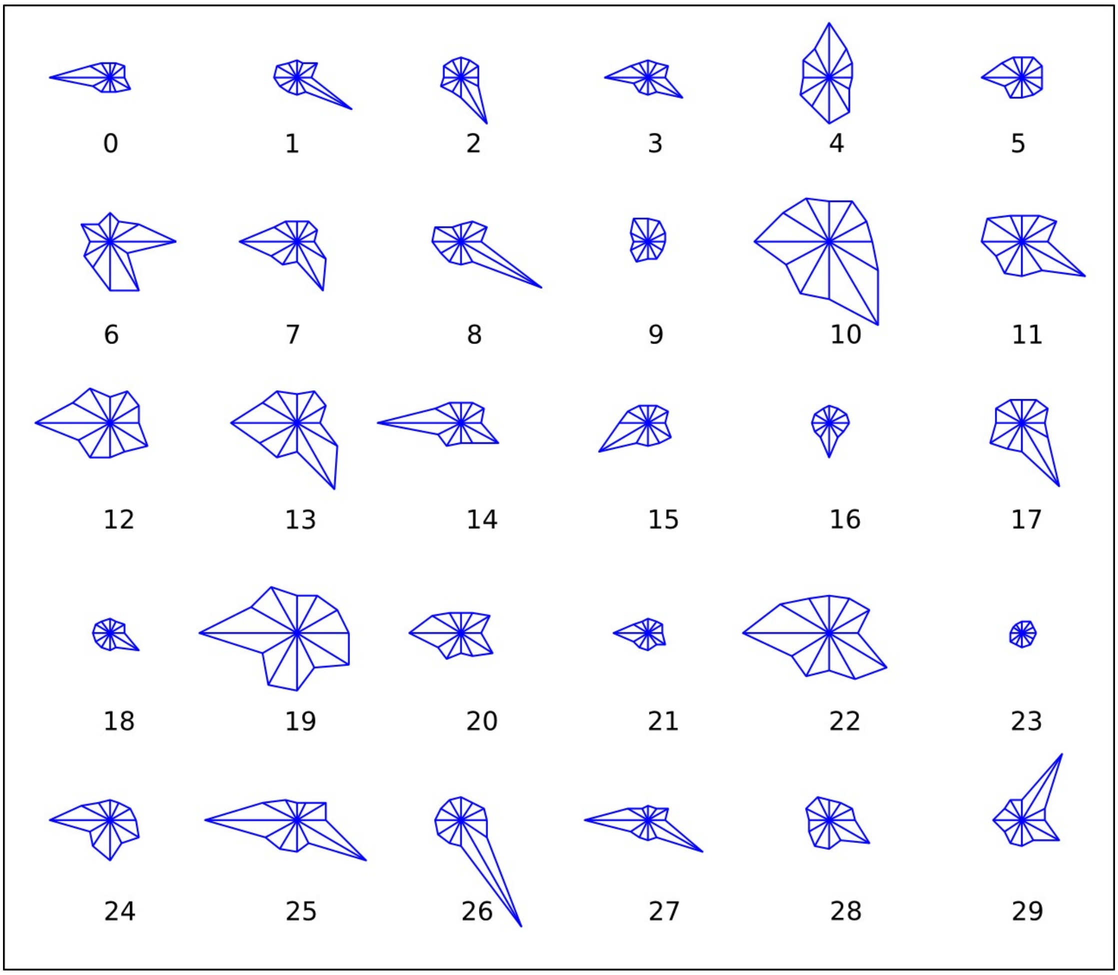


Figure 6.31: Prototype visualization via star glyphs.

domain. However, hypotheses or interesting observations can also be made when analyzing prototype visualizations, i.e. the feature domain. Therefore, not only CIPRA visualizations but also star glyphs and bar graphs can be applied to visualize the characteristics of a prototype's protein profile. For comparison, the same prototypes as displayed in figure 6.28 are displayed in figure 6.30 as bar graphs and in figure 6.31 as star glyphs. It can be seen that all of these prototype visualizations show advantages and disadvantages for the analysis of protein colocation. The characteristic of the CIPRA view is that the bars are scaled according to the channels' intensities. For very low intensities, this can result in bars which are not readable anymore and the corresponding proteins vanish from the display. While this is a desired feature in many cases since the observer's perception is not distracted by irrelevant information, in some cases it might be more desired to see all channels alike. In this case, a bar graph display is more suited, however, it is evident that matching between individual prototypes becomes more difficult as the assignment of a protein to a bar is not as clear as in the CIPRA view (see figure 6.30). The star glyphs, on the other hand, are a very compact display and shapes can well be matched. However, for a quite small number of protein channels, linking the individual tines of the star to a protein already becomes very hard and can only be achieved by interactive analysis since writing the prototype names at each tine is not feasible for such compact displays (see figure 6.31). Throughout the work with TIS data, I have therefore found the CIPRA visualization most suitable for prototype comparison and analysis of prototype characteristics.

By visually inspecting the CIPRA or other prototype visualizations, interesting insights in the protein colocations can be obtained. When analyzing figure 6.28, most of the prototypes show high levels of `nmdr1`, `syphys` and `synap`. However, other proteins also have high levels for some prototypes, such as prototype 4 which shows higher signals for `glur1`, `nr2b` and `synap` or prototype 15 and 17 which show a signal for the `nnos` channel.

Having identified interesting protein profiles by visual inspection of the prototype displays, it would now be reasonable to analyze the spatial distribution of those prototypes in the image domain. By applying principles of link & brush techniques, this analysis is straight forward. However, it is often also important to get a deeper insight in the individual patterns of synapses forming the cluster of interest. Therefore, one can analyze the proposed parallel coordinate plots or color lines views. Figure 6.32 exemplarily shows some parallel coordinate plots and color lines views for selected CIPRAs for image $\mathbf{I}^5$. Here, the homogeneity of each cluster can be visually evaluated and outliers can be identified. For example, for cluster 23 the CIPRA shows almost no signal which can be verified by the analysis of the parallel coordinate plot and color lines view (see figure 6.32 fourth row). Prototype 14, on the other hand, features high levels of `nmdr1` and medium levels of `syphys` (see figure 6.32 first row). Through the analysis of the line view, one can nicely observe that there exists one item which shows additionally high levels for `cat` and `glur5`, which also peak out in the parallel coordinates view. In the parallel coordinate plot, however, one can not see that the peak at `cat` coincides with the peak at `glur5`. Another prototype showing interesting protein combinations is prototype 22, showing high signals for `glur5`, `nmdr1` and `syphys` (see figure 6.32 third row). To get another view on the data of that prototype, selected scatter plots can now be analyzed. Figure 6.33, for example, shows scatter plots for the feature vectors of image $\mathbf{I}^5$. In figure 6.33 (a), `nmdr1` is plotted against `glur5` and those items belonging to prototype 22 are highlighted in red. In figure 6.33 (b) the same items are highlighted in red, however in a `nmdr1` vs. `syphys` plot. It can be observed that several other items not belonging to cluster 22 surround the red items in both plots. It is now possible to raise the question to which clusters those surrounding items belong to. Therefore, in green, synapses of prototype 25 and in orange synapses of prototype 27 are highlighted (see figure 6.33 (b)). Analysis of the prototypes' protein profiles reveals that they show high levels for `nmdr1` and `syphys`, however only low or no signal for `glur5`. Therefore they are not associated to cluster 22. When evaluating these scatter plots, it becomes obvious that an analysis based solely on a scatter plot matrix evaluation would not lead to the desired results as one would not be able to delineate groups of synapses which show similar high-dimensional protein profiles.

So far, the analysis of the image data was based on the extracted feature vectors and

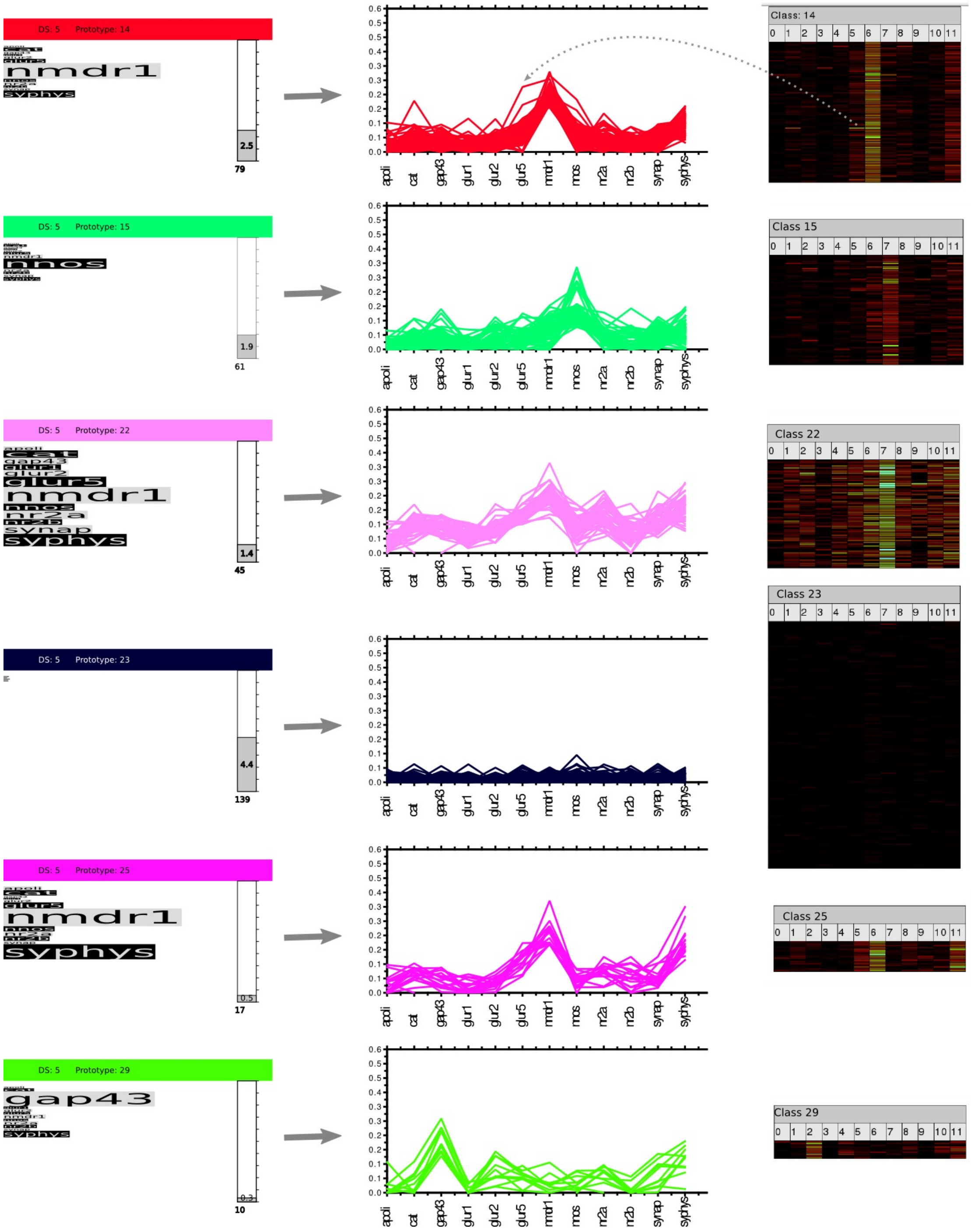


Figure 6.32: Exemplary in detail exploration of some prototypes. Parallel coordinates plots and line views are displayed for image $\mathbf{I}^5$.

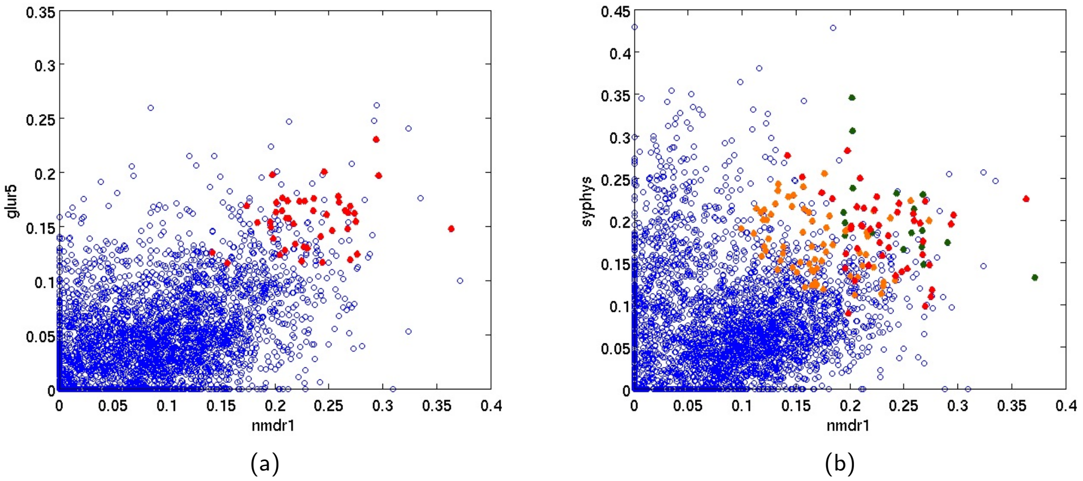


Figure 6.33: Scatter plot of selected proteins for image $\mathbf{I}^5$. In (a) nmdr1 is plotted versus glur5 and synapses belonging to cluster 22 are highlighted in red. In (b), the same synapses are highlighted in red in a nmdr1 vs. syphys plot. In orange, synapses of cluster 27 and in green synapses of cluster 25 are depicted.

analyzing prototypes, cluster maps and selected clusters itself. The most in-detail analysis would now be to visually inspect the individual protein channels for a selected item to see if the feature extraction process yielded reasonable feature vectors. Figure 6.34 exemplarily shows sub-images for some protein channels and a synaptic region is highlighted with a red box. The associated prototype is shown on the left. It can be seen that the protein channels of the first three examples correspond very well to their associated prototype. Prototype 27 shows high signals for syphys and nmdr1 which is reflected in the protein images (see figure 6.34 first row). Similarly, prototype 17 has high signals for nnos and synap which can be verified for the selected data item (see figure 6.34 second row). However, artifacts also occur through the feature extraction process. Prototype 19 features higher signals for apoli, nmdr1, nr2a, nr2b and syphys (see figure 6.34 fourth row). While this can be verified for a data item for syphys and nmdr1, nr2a, nr2b and glur5 show no true signals at that position. High values for those channels are artificially introduced as a border between a dark and a light region can be observed. If a pixel $\mathbf{p}$ identified as synapse positions lies on the light region but the neighborhood of that pixel covers more of the dark region, artificially a high signal values is introduced.

6.6.4 Comparison of Cluster Results of K-means, NG and H^{2}SOM

Up to now, only the results of the NG clustering have been analyzed. As it would be lengthy to illustrate each clustering result in such a detail, I will focus on comparing the most important aspects of the analysis. First of all, it is interesting to compare the cluster maps obtained by each of the results to see if a separation into SR and SP, as has been

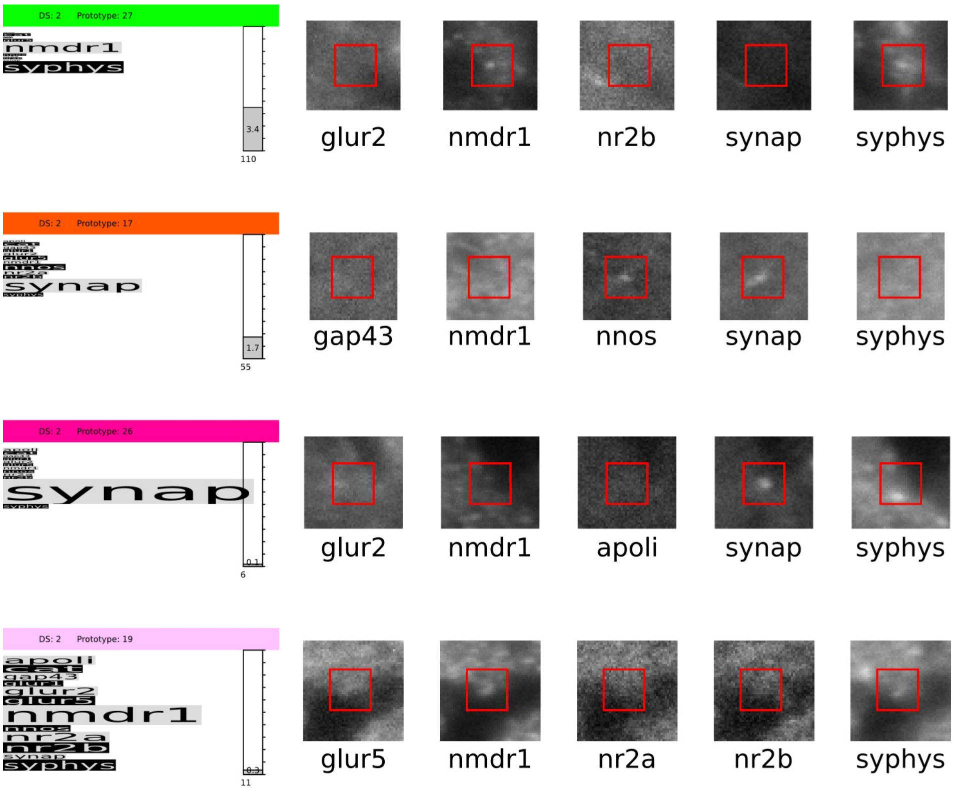
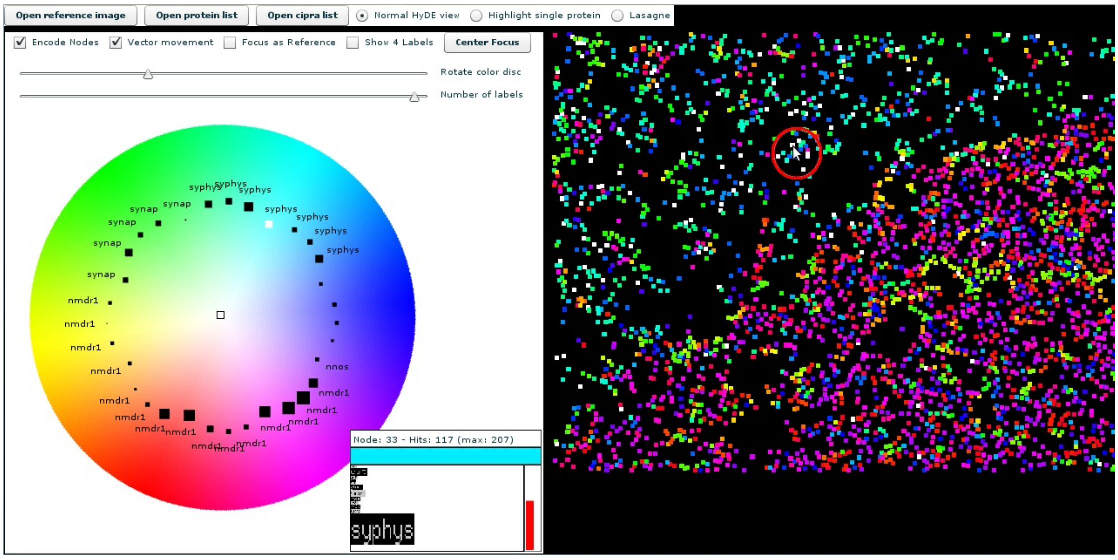


Figure 6.34: In detail analysis of selected data items. The associated protein is displayed and a selection of protein channels.

seen for the NG approach (see section 6.6.1), can be observed for each of the other results. K-means clustering was performed with 30 prototypes similar to the NG approach. Clustering was repeated with 10 random initializations and the result with the lowest wps was chosen. For the H^{2}SOM clustering, the second level of the hyperbolic lattice was chosen. This is motivated by the fact that in this level 32 nodes are present which well fits to the 30 clusters chosen for K-means and NG and thus allows for a good comparison. Since the H^{2}SOM itself can be used for color assignment, each prototype was colored according to its location in the HSV disc. This also induces a linear ordering of the prototypes along the rainbow scale, which eases the interpretation of the CIPRA display. In figure 6.35, a screenshot is shown of the interactive, web-based exploration tool for H^{2}SOM clustering results, developed in the Biodata Mining & Applied Neuroinformatics Group, University of Bielefeld. The distribution of the prototypes on the HSV disc (see figure 6.35 left) along with the cluster map encoding (see figure 6.35 right) and a CIPRA visualizations of a selected prototype (see encircled mouse pointer) can be observed. The most prominent protein is shown at the node of each prototype which eases the analysis of the prototypes. Interactively, the color assignment can be changed by rotating the HSV disc or adjusting the focus of the nodes as described in section 6.4. Thereby, that configuration can be chosen very intuitively, which is best suited



Open reference image | Open protein list | Open cipra list | ● Normal HyDE view ○ Highlight single protein ○ Lasagne

☑ Encode Nodes ☑ Vector movement ☐ Focus as Reference ☐ Show 4 Labels | Center Focus

Rotate color disc

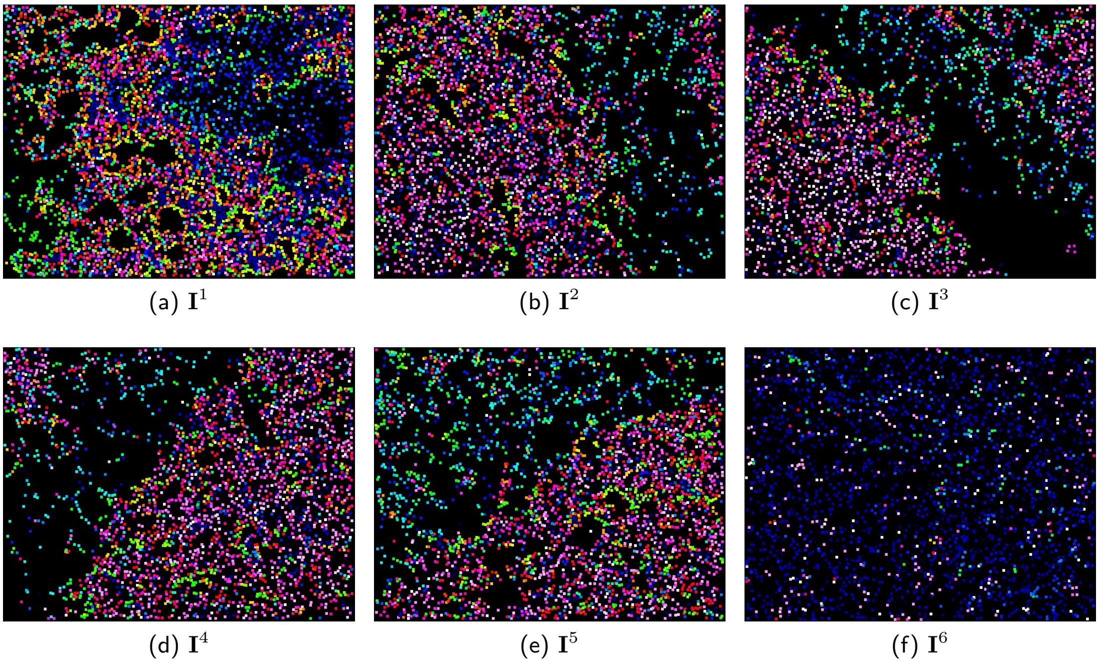
Number of labels

Node: 33 - Hits: 117 (max: 207)

Figure 6.35: Interactive, web-based exploration tool for the analysis of H^{2}SOM clustering results. Prototypes mapped on the HSV disc are shown at the left side with the corresponding cluster map on the right. Color assignment can easily be changed by rotation of the HSV disc or movement of the nodes' focus point, depicted with a rectangle. A CIPRA visualization is provided for a selected synapse/prototype which eases the interpretation of the data.

for the current analysis. Yet, already the standard coloring without the requirement of node movement results in very meaningful color assignments. The obtained cluster maps for all six image stacks are depicted in figure 6.36. The coloring again very nicely reflects the separation of prototypes into SP and SR regions for $\mathbf{I}^1, \ldots, \mathbf{I}^5$ and a lack of this separation for $\mathbf{I}^6$. Similar to the NG clustering, SP specific prototypes show syphys and synap signals but lack nmdr1 signals whereas SR specific prototypes all feature nmdr1 signal (see figure 6.37 red and green marked prototypes, respectively). Analog observations can be made for the K-means clustering (see figure 6.38 red and green marked prototypes). Thus, as it was the case for the NG clustering, protein networks can be identified which are present solely in the SR or SP region and are specific for synaptic regions as they are missing in image set $\mathbf{I}^6$. Image set $\mathbf{I}^6$ again mostly shows feature vectors with very low intensities, such as prototypes 37 (11%), 38 (38%), and 39 (15%) for the H^{2}SOM clustering (see figure 6.37) or prototypes 7 (57%), 11 (11%), and 20 (14%) in the K-means clustering (see figure 6.38).

With respect to the obtained prototypes, the results of the K-means and the NG approach can very well be matched as can be seen by comparing figure 6.28, which displays the CIPRAs for the NG clustering, with figure 6.38 displaying the CIPRAs for the prototypes obtained by K-means clustering. Note that only 29 prototypes are shown in figure 6.38 as one prototype (19) only occurs in one single data item of image set $\mathbf{I}^4$. For example, prototype 6 of the NG clustering can very well be matched to prototype 15 of K-means (see figure 6.39). Other matches are 29 to 26 or 15 to 1 (first refers to the NG number, second to the K-means number) as displayed in figure 6.39. As the cluster frequencies for the K-means prototypes



(a) $\mathbf{I}^1$ (b) $\mathbf{I}^2$ (c) $\mathbf{I}^3$

(d) $\mathbf{I}^4$ (e) $\mathbf{I}^5$ (f) $\mathbf{I}^6$

Figure 6.36: Cluster maps obtained for H^{2}SOM clustering with 32 prototypes (second level) and mapping of the prototypes onto the HSV disc for color assignment.

as well as the NG prototypes refer to the same image $\mathbf{I}^2$, even the number of synapses belonging to the cluster can be compared. Again, K-means and NG are well comparable for the mentioned prototypes. Comparing the CIPRAs of the H^{2}SOM clustering (see figure 6.37) to the ones obtained via NG and K-means (figure 6.28 and 6.38), it can be observed that the H^{2}SOM prototypes all show high levels for `syphys, synap` and `nmdr1`. However, prototypes showing higher levels for other proteins as `gap43` or `apoli`, or interesting combinations of proteins as `syphys` and `glur5`, which was observed in the NG and K-means clustering and could be verified as true signals, are missing. It seems that the H^{2}SOM approach, through its hierarchical learning, does not represent rarely occurring items well in the second level but represents the most frequent prototypes. Therefore, the third level of the lattice was also analyzed. Here, 120 nodes are present, thus the data set is split into 120 clusters. Again, most of the prototypes show `syphys, synap` and `nmdr1`, however, now also prototypes with high signals for `gap43, nr2b, apoli, cat, glur5` and `nnos` are present (data not shown).

By analyzing the obtained cluster maps of the K-means (see supplementary figure A.4) and H^{2}SOM clustering, a separation between image set $\mathbf{I}^6$ and $\mathbf{I}^1, \ldots, \mathbf{I}^5$ is suggested. To obtain a more objective criterion, figure 6.40 (a) displays the rank correlation obtained for the K-means and H^{2}SOM (figure 6.40(b) second and (c) third level) clustering approach. It is evident that the correlations obtained for K-means are very similar to that of the NG clustering (compare figure 6.25 (b)). The image stack correlations obtained for the second

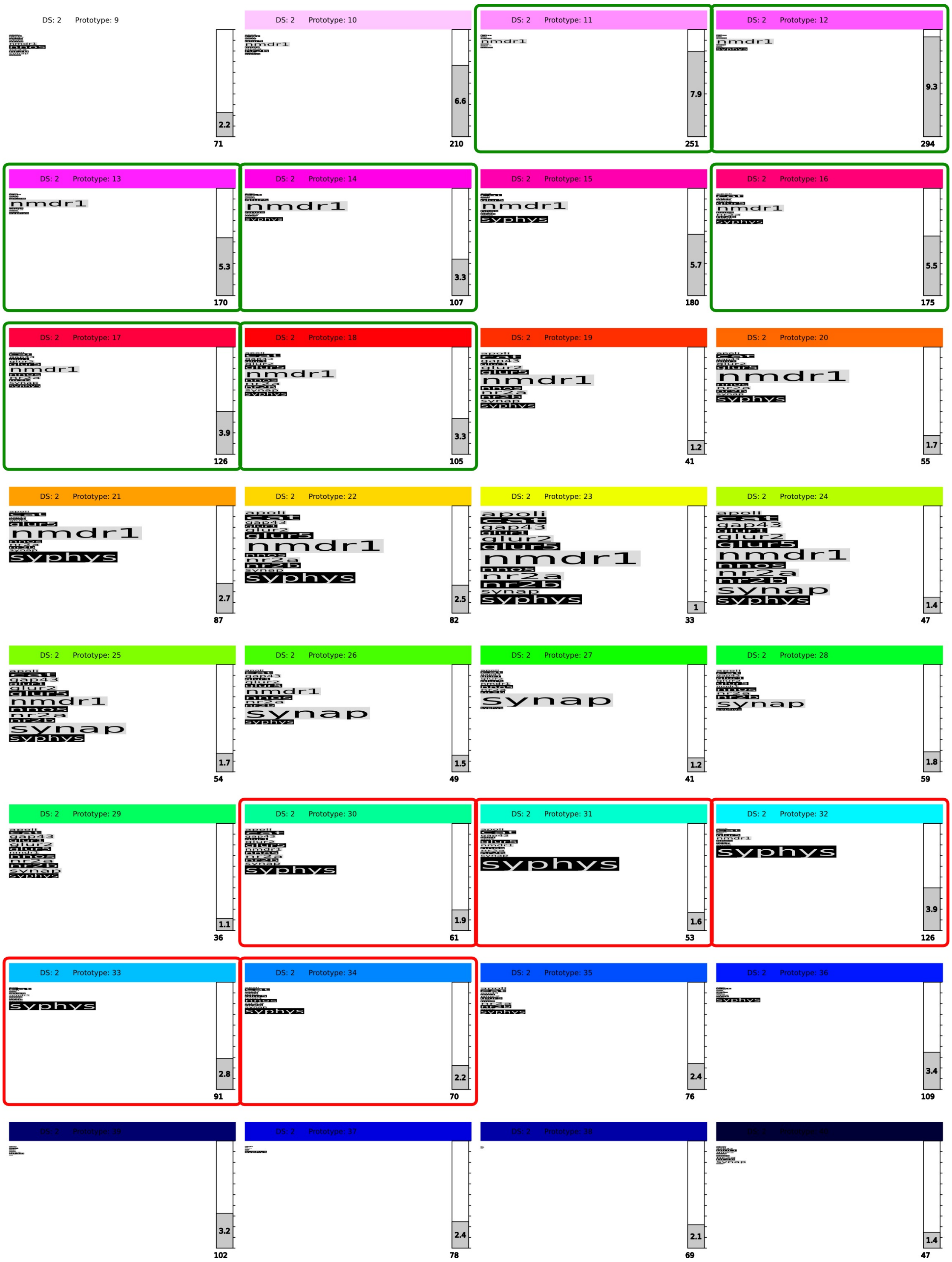


Figure 6.37: CIPRA visualization of the prototypes obtained in the second level of the H^{2}SOM. SP and SR specific prototypes are highlighted in red and green respectively. Prototype abundances refer to image set $\mathbf{I}^2$.

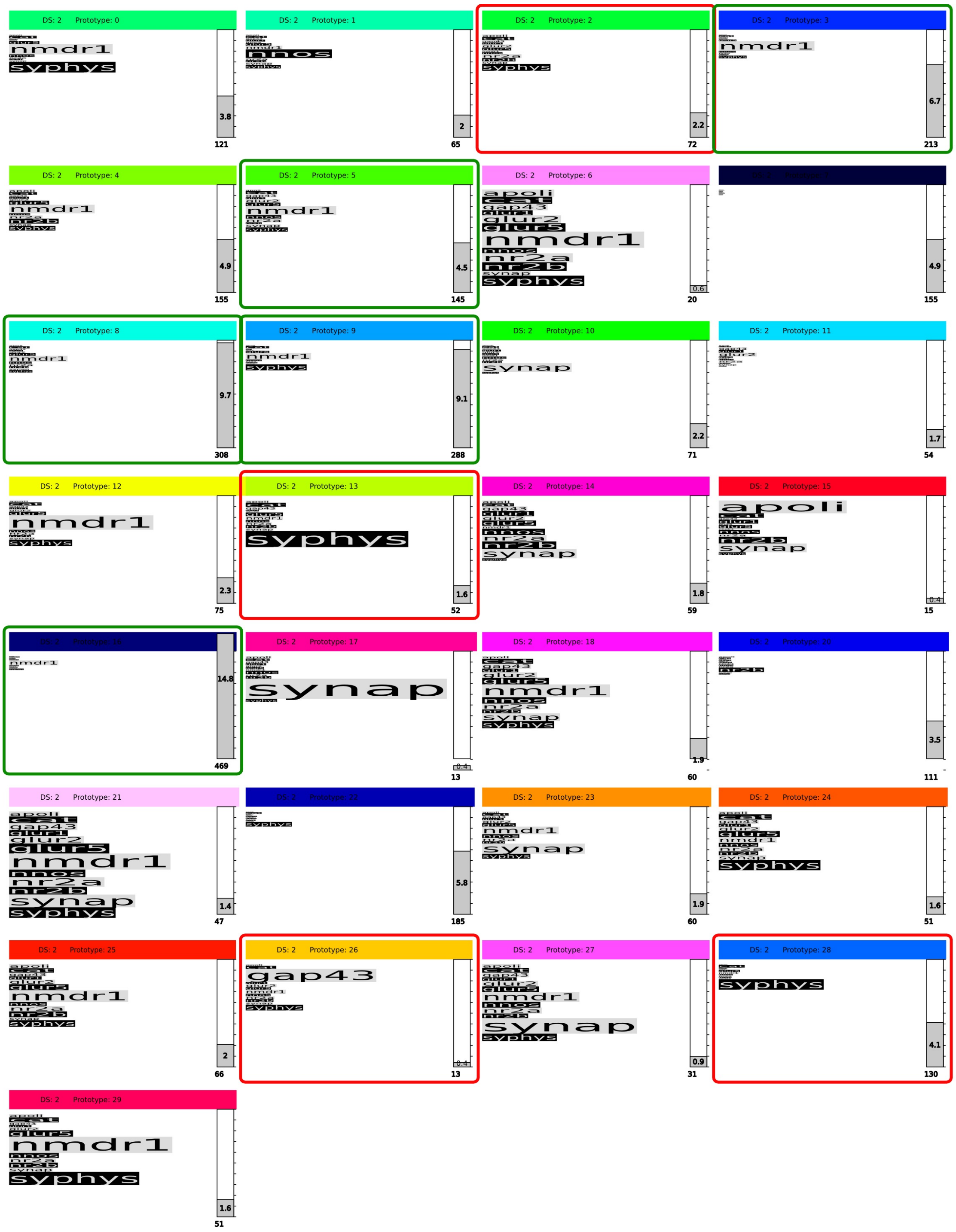


Figure 6.38: CIPRA visualization of the prototypes obtained for K-means clustering with 30 prototypes. SP and SR specific prototypes are highlighted in red and green respectively. Prototype abundances correspond to image set $\mathbf{I}^2$. Prototype 19 is not shown as it occurs only in one data point of image set $\mathbf{I}^4$.

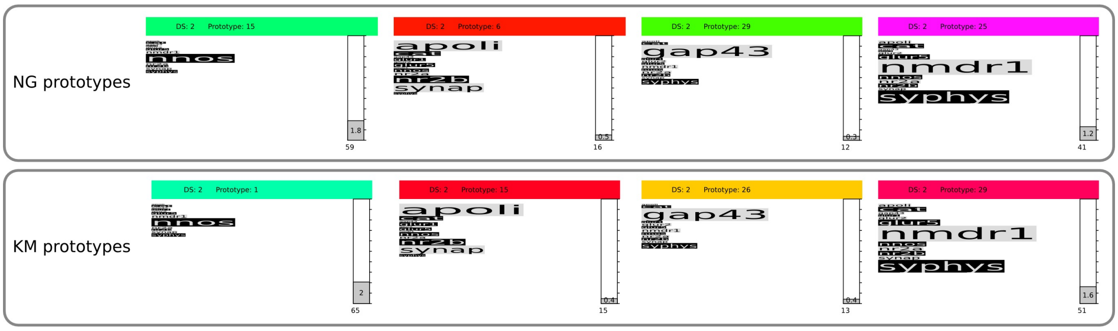
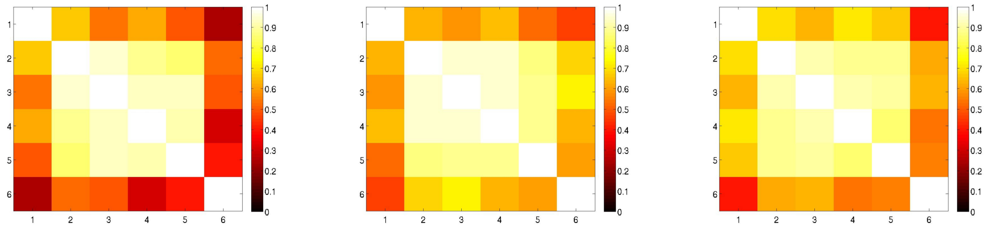


Figure 6.39: Comparison of selected prototypes obtained via NG clustering (top) and K-means clustering (bottom). In both cases, the protein abundance refers to image stack $\mathbf{I}^2$ so that abundances are comparable. Colors are not comparable as color assignment was carried out for each clustering (NG, K-means) separately.



(a) K-means

(b) H^{2}SOM level 2

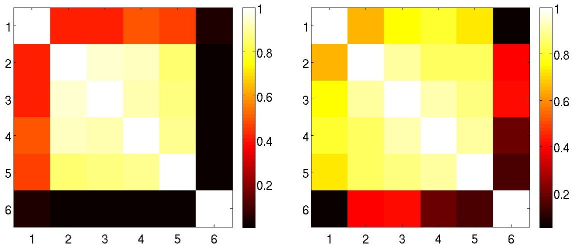
(c) H^{2}SOM level 3

Figure 6.40: Rank correlations obtained for the K-means clustering (a), the second level of the H^{2}SOM (b) and the third level of the H^{2}SOM.

level of the H^{2}SOM approach, however, are lower for $\mathbf{I}^1$ than for $\mathbf{I}^6$ which was not observed in the K-means and NG clustering. Yet, for the third level the obtained correlations are again more similar to that of K-means and NG with higher correlations for $\mathbf{I}^1$ than for $\mathbf{I}^6$.

6.6.5 Analysis of the Whole Protein Set

Up to now, a sub set of 12 proteins, which showed clear signals, were analyzed to ensure that meaningful feature vectors were extracted. Now, the whole set of 21 proteins, omitting the prop channel, is evaluated. Again, all six data sets are jointly clustered with the NG algorithm into 30 clusters and the obtained clusters are analyzed within and between image sets. In the following, the image stack correlations are evaluated. Figure 6.41 (a) and (b) display the Pearson's and Spearman's rank correlation matrix for the six data sets. Again, very high correlations can be observed between the images $\mathbf{I}^2$ to $\mathbf{I}^5$, and $\mathbf{I}^6$ separates very well from from the remaining images. Compared to the 12 channel setup, now higher correlations are achieved between $\mathbf{I}^1$ and $\mathbf{I}^2, \ldots, \mathbf{I}^5$, so that the synapse cases separate even better from the non synapse case $\mathbf{I}^6$.



(a) (b)

Figure 6.41: Squared Pearson's and Spearman's rank correlations for the six image sets $\mathbf{I}^1, \ldots, \mathbf{I}^6$. Feature vectors were calculated based on all 21 protein channels, omitting the `prop` channel, and clustering was performed with NG clustering and 30 prototypes.

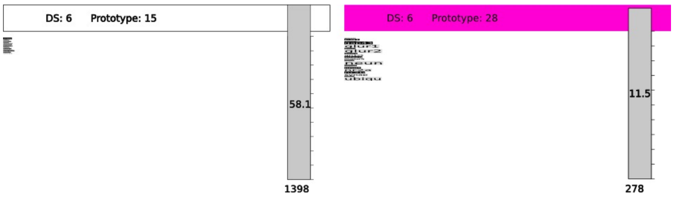


Figure 6.42: Prototype 15 and 28 which subsume almost 70% of all synapses of image $\mathbf{I}^6$ in the setup with 21 proteins. Both prototypes show only weak signals for all proteins.

This separation can also be observed when analyzing the pie chart visualization (supplementary figure A.5). Here, almost 70% of the synapses of stack $\mathbf{I}^6$ belong to cluster 15 and 28, which are of very low abundance in the other stacks. A detailed analysis of the CIPRA visualizations (see figure 6.42) belonging to these two clusters reveals that prototype 15 (58.1%) again shows almost no signal in any of the protein channels. Prototype 15 can therefore be matched very well to prototype 23, which was observed in 55.9% of the synapses of $\mathbf{I}^6$ in the 12 protein channel setup. Prototype 28 shows very weak signals for `glur2, neun` and `ubiqu`.

Next, it is analyzed if a separation between the SR and SP, which was observed for the 12 channel setup, can also be revealed for a setup with 21 prototypes. A coloring reflecting this separation very well, as in figure 6.22, could not be obtained for this setup, neither with the Sammon's nor with the PCA projection. H^{2}SOM clustering was not carried out so that a projection on the HSV disk was not obtained. With the interactive approach of the H^{2}SOM display, however, such a separation could presumably be found. Nevertheless, when interactively analyzing the distribution of individual prototypes in the cluster maps obtained through mapping onto a 1D color scale (shown in supplementary figure A.6), several SR and SP specific prototypes could be observed. Figure 6.43 displays those prototypes found almost exclusively in the SP and SR region. Again, SP prototypes show no `nmdr1` signal whereas it is present in all SR specific signals. The SP and SR specific prototypes can very well be matched to the SR and SP specific prototypes of the 12 channel setup according to their profiles and spatial location in the image. However, for the 21 channel setup, signals

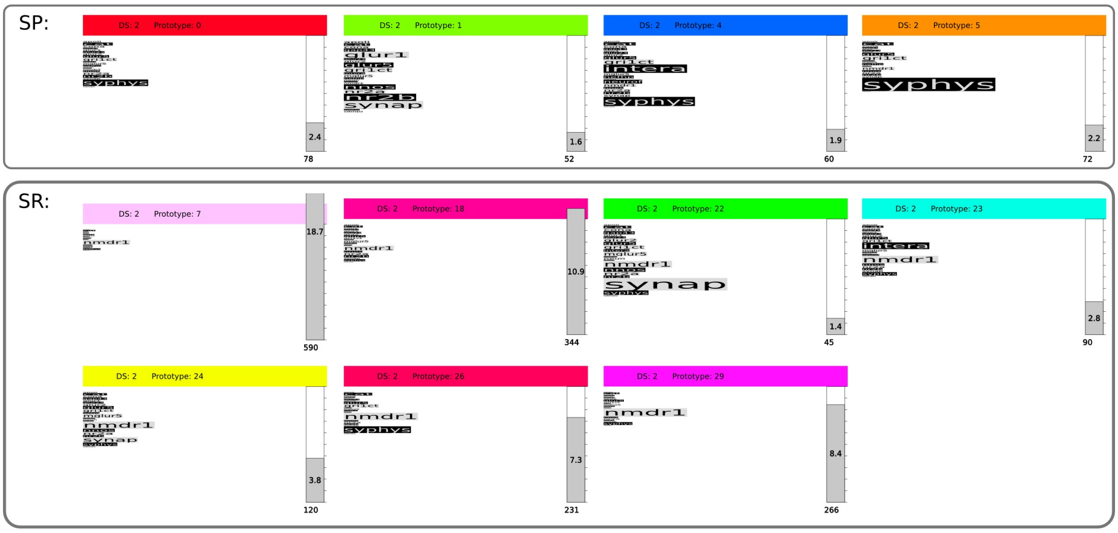


Figure 6.43: CIPRA visualization of prototypes whose associated data items occur almost exclusively in the SP or SR region of the samples. All 21 proteins were used as a data basis. Prototype abundances refer to image stack $\mathbf{I}^2$.

for several additional proteins as intera, gri1ct and mglur5 can be observed in these prototypes.

For the sake of completeness, figure 6.44 gives an overview of the 30 prototypes which approximate the data. They can very well be compared to the prototypes obtained for the 12 protein setup as most of the prototypes show high levels for nmdr1, syphys and/or synap. Again, one prototype shows very high levels for gap43 (see figure 6.44 (prototype 16)) and can very well be matched to prototype 29 of figure 6.28, even with respect to the abundance of that prototype. Also prototypes featuring higher levels of nnos can again be observed in this setup (prototypes 1 and 3). From the set of 9 proteins added in this study, especially intera, nefhp, neurof and gri1ct are present in many prototypes.

6.6.6 The Impact of Feature Calculation

In this section, I would like to demonstrate the importance of a sophisticated feature calculation for the analysis of protein colocation at synapses. Therefore, data sets $\mathcal{X}^s$ were constructed for each image $\mathbf{I}^s$ as described in section 6.1.2, however, synapse specific feature calculation was omitted. The Gaussian filtered intensity values ($\hat{f}(\mathbf{p}^*)$) of each channel were extracted as image features only, omitting the process of local background correction.

Clustering was performed with the NG approach as it is more robust against the initialization of the algorithm and thus has not to be rerun several times. For comparison purposes, the same parameters were chosen for clustering as in section 6.6.1 with 30 prototypes and 40 times as many update steps as data items. Figure 6.45 displays the cluster maps obtained through mapping onto a 1D color scale. It can be seen that synapses with the same color form large regions which spread in a wavelike manner. Furthermore, the negative example

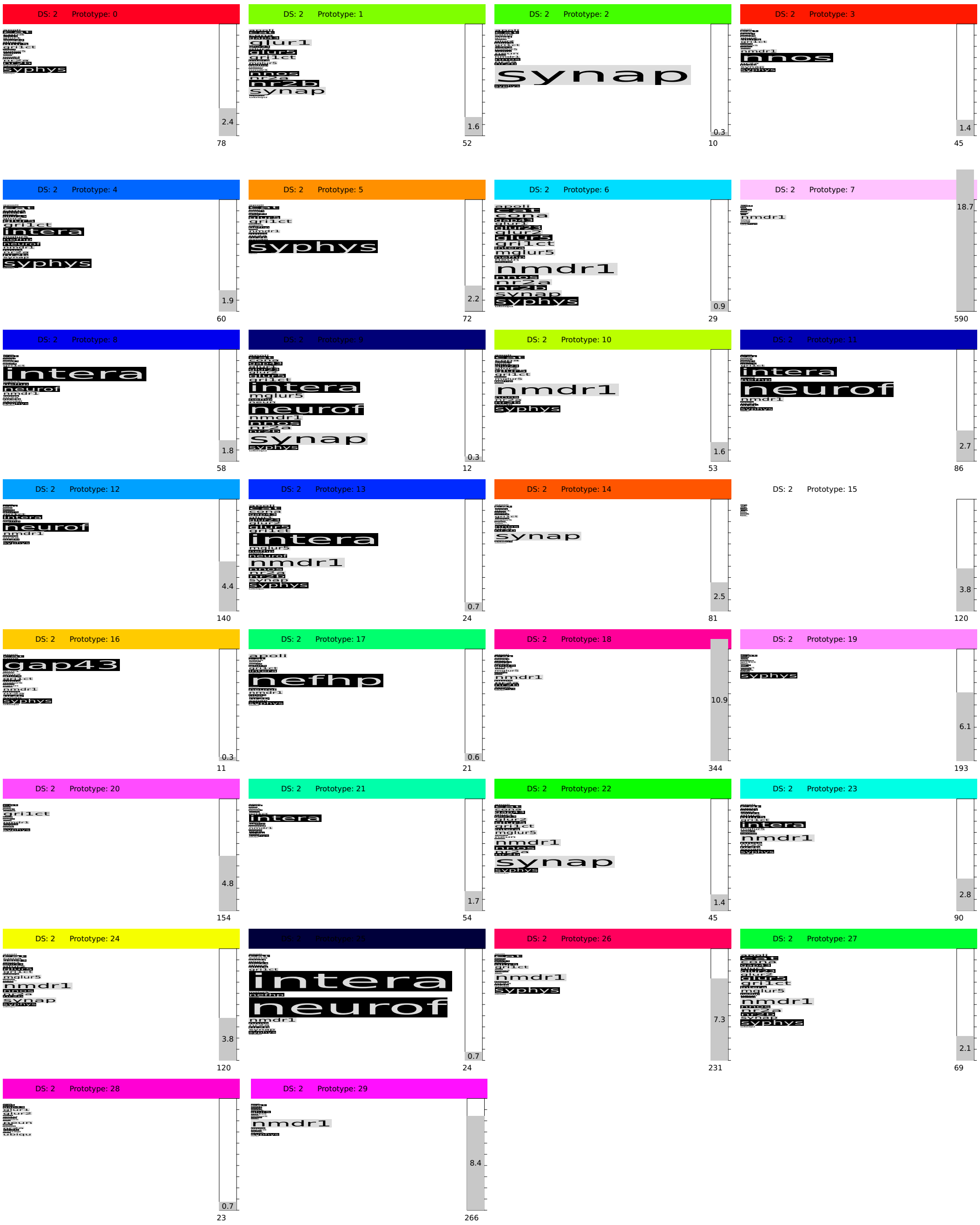
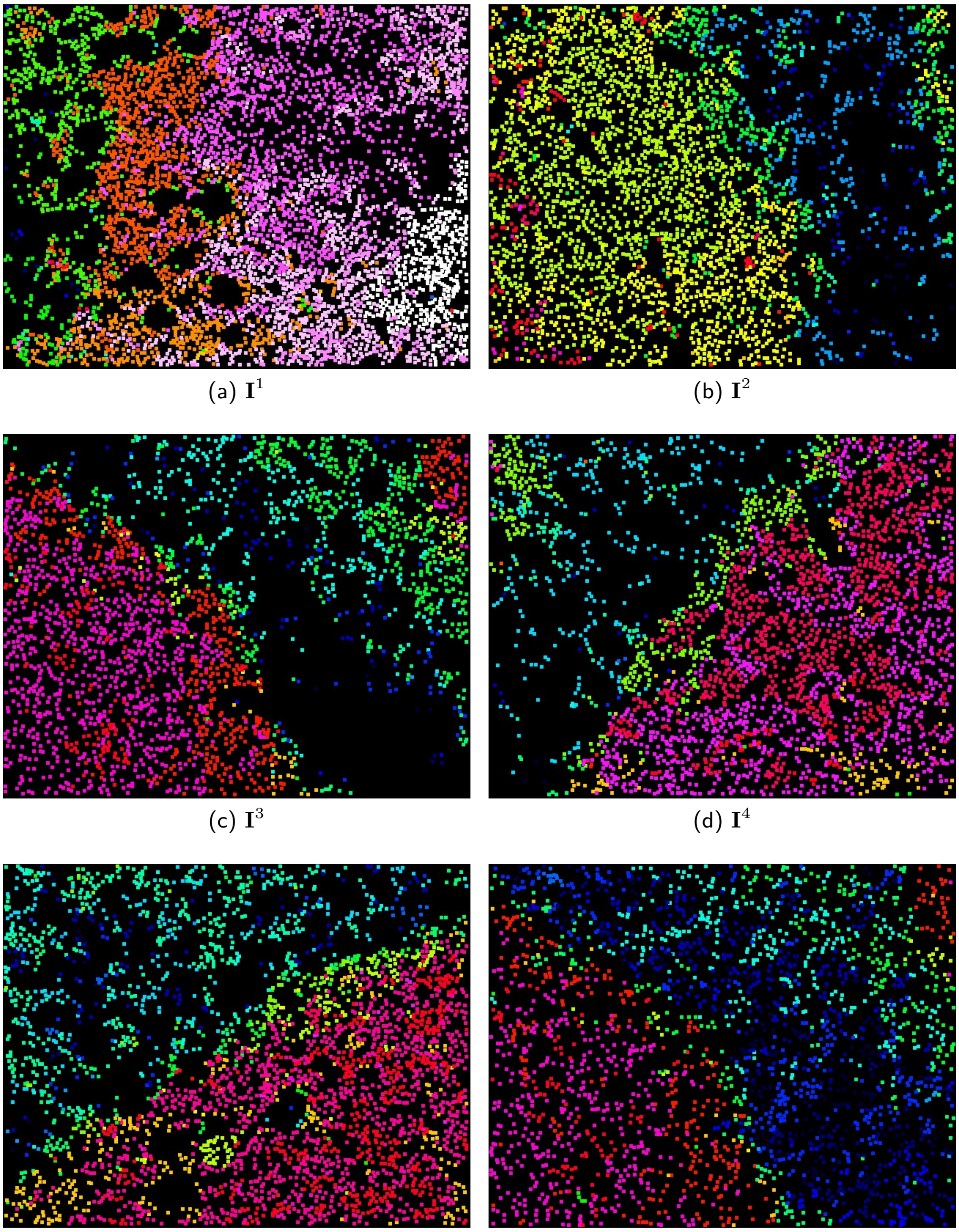


Figure 6.44: CIPRA visualization of the obtained prototypes for the NG clustering of the 21 dimensional synapse data set. Absolute and relative abundances reference to image set $\mathbf{I}^2$.



(a) $\mathbf{I}^1$

(b) $\mathbf{I}^2$

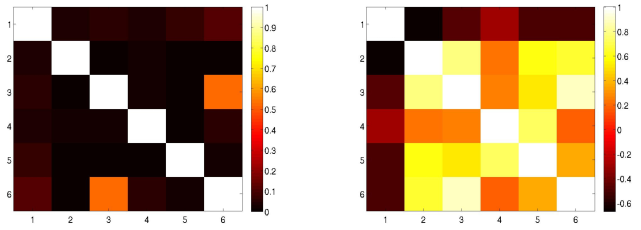
(c) $\mathbf{I}^3$

(d) $\mathbf{I}^4$

(e) $\mathbf{I}^5$

(f) $\mathbf{I}^6$

Figure 6.45: Cluster maps obtained by mapping onto a 1D pseudo color scale for NG clustering of feature vectors based solely on the extracted intensity values.



(a) Squared Pearson's correlation (b) Spearman's rank correlation

Figure 6.46: (a) Squared Pearson's and (b) Spearman's rank correlations obtained for NG clustering of feature vectors obtained through extraction of the pure intensity values. Negative correlations occur which has to be taken into account when mapping the colors to correlation values.

case $\mathbf{I}^6$ shows similar colors and partitioning and is thus not separated from the other images as it was the case when features were properly extracted (see figure 6.45 (f) compared to 6.24 (f) or 6.22 (f) in section 6.6.1). One has the visual impression that the correlation between images, especially $\mathbf{I}^1$ and $\mathbf{I}^2$ is low. This can be verified when analyzing the Pearson's and Spearman's correlations, which are very low in the most cases and furthermore do not separate between case $\mathbf{I}^6$ and the remaining cases (see figure 6.46(a) and (b)). Now even negative correlations occur, which has to be taken into account when interpreting the correlation maps in figure 6.46 as now the meaning of the color has changed. Yet, a high Pearson's and Spearman's correlation can be observed for image $\mathbf{I}^3$ and $\mathbf{I}^6$. When analyzing the corresponding pie charts in figure 6.47, one can see that clusters are almost mutually exclusive between image stacks again with one exception, image $\mathbf{I}^3$ and $\mathbf{I}^6$. With exception of one prototype, which occurs in image $\mathbf{I}^5$ in 0.08% of the data items, both images show the same 20 prototypes but lack the remaining 10. It has to be recalled that the data set $\mathcal{X}^6$ has actually been extracted from image stack $\mathbf{I}^3$ and referring to it as $\mathbf{I}^6$ has been introduced for clarity. Hence, finding the same sub set of prototypes in both images and almost mutually exclusive prototypes for the other stacks suggests that the feature vectors, and thus the prototypes, represent the image characteristic rather than the protein colocation at synapses. This assumption is also supported by the high correlation between these two data sets, which could not be observed when feature vectors were calculated.

The mutually exclusive characteristic of prototypes of different image stacks can also be seen in more detail when analyzing the prototypes of the clustering result. Figure 6.48 exemplarily shows some prototypes for images $\mathbf{I}^1, \mathbf{I}^2$ and $\mathbf{I}^4$. It can be observed that prototypes of one image stack are very similar to each other whereas prototypes between images show some variation in different protein channels. Only few prototypes approximate the main portion of the synapse data set of one image stack.

6.6.7 The Impact of Semantic Image Annotation

One might argue that semantic annotation is not crucial for the analysis of synaptic signals as they can also be analyzed if feature vectors of the whole data set are evaluated. Thus, the

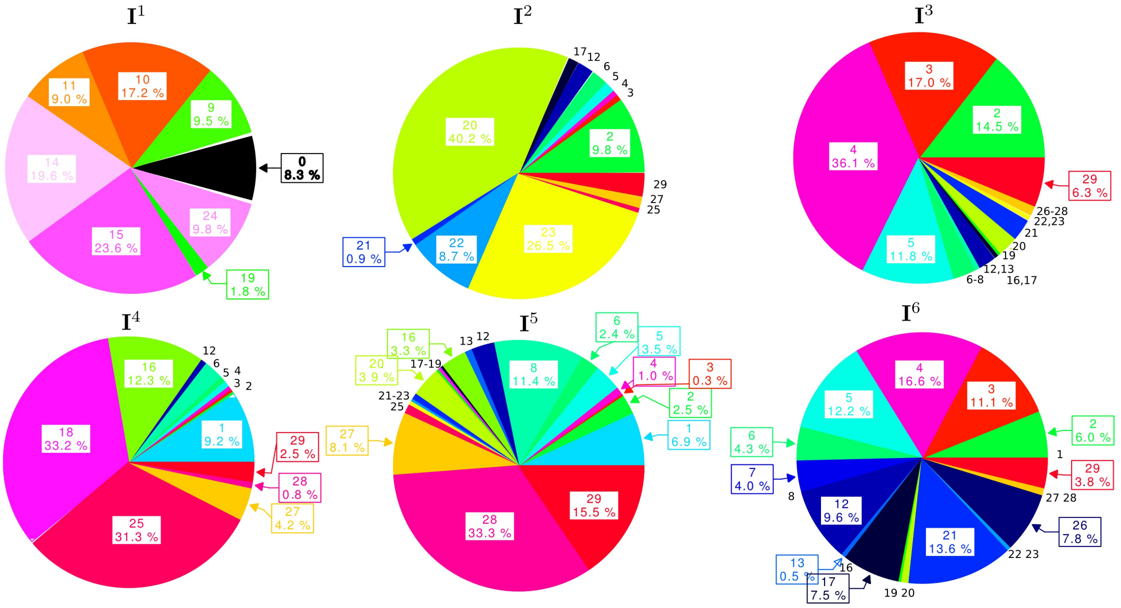


Figure 6.47: Pie chart visualizations of the relative abundances of prototypes in the individual stacks. Feature vectors only consisting of the images' intensity values were used as a data basis.

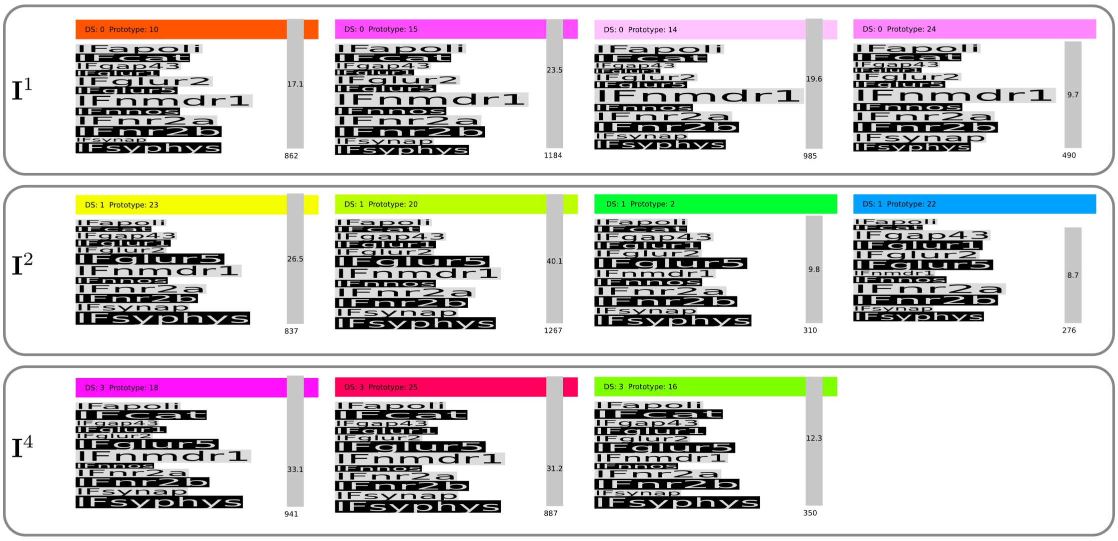
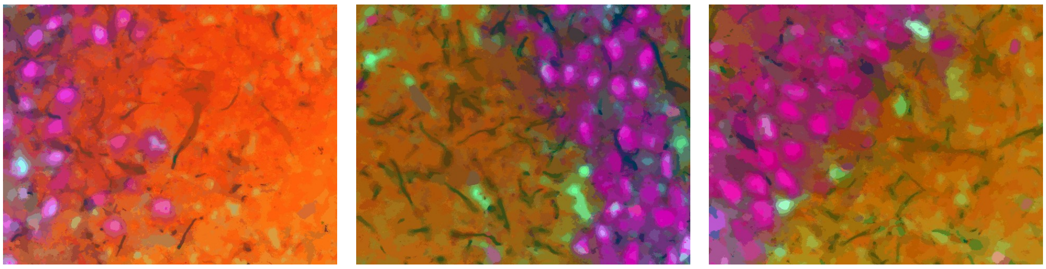
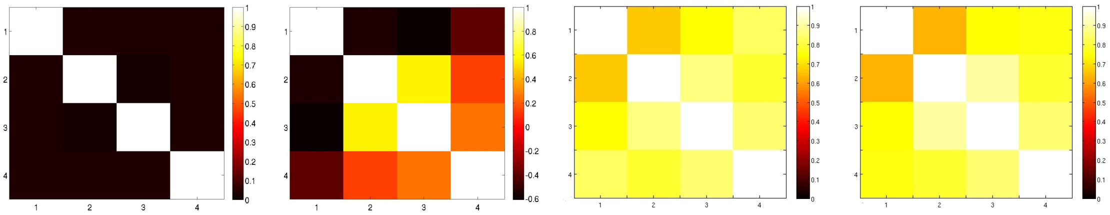


Figure 6.48: Selection of CIPRAs of image $\mathbf{I}^1$, $\mathbf{I}^2$ and $\mathbf{I}^4$. Only few very similar prototypes approximate the main portion of the synapse data set of one image stack. Prototypes between images show variations in their profiles.



(a) $\mathbf{I}^1$ (b) $\mathbf{I}^2$ (c) $\mathbf{I}^4$

Figure 6.49: Cluster maps obtained for whole image clustering with the pure intensity values as features.



(a) (b) (c) (d)

Figure 6.50: Squared Pearson's (a,c) and Spearman's rank (b,d) correlation for whole image clustering without (a,b) and with feature extraction (c,d).

potentially error prone synapse detection can be omitted and prototypes showing synaptic signals are extracted from the whole image data set. To show how difficult and tedious this analysis would be, exemplarily clustering on the whole image data set, for 21 proteins, is performed.

The straightforward way of such an analysis would be to extract the image intensities at each pixel location for each protein channel. However, as has already been shown for the analysis including semantic annotation, this results in feature vectors which show the varying intensities within and between the stacks. Figure 6.49 exemplarily shows three cluster maps for the whole image data sets of $\mathbf{I}^1, \mathbf{I}^2$, and $\mathbf{I}^4$. Neural Gas clustering was performed jointly for four image sets $\mathbf{I}^1, \ldots, \mathbf{I}^4$ with 300 prototypes. Each pixel in the cluster map was colored according to the color of its associated prototype. Color assignment was obtained by PCA mapping in the RGB space. Through visual comparison of the cluster maps of figure 6.49, it is evident that again there exist high variations between the colors assigned to different image stacks, thus between their prototypes, especially for image $\mathbf{I}^1$. This impression can be verified when evaluating the obtained Pearson's and Spearman's rank correlations (see figure 6.50 (a) + (b)). Also looking at the distribution of the individual prototypes across images reveals that mostly they are only present in one of the image stacks and show no or low abundance in the remaining stacks.

As the sole extraction of intensity values at the individual pixel locations show inter stack

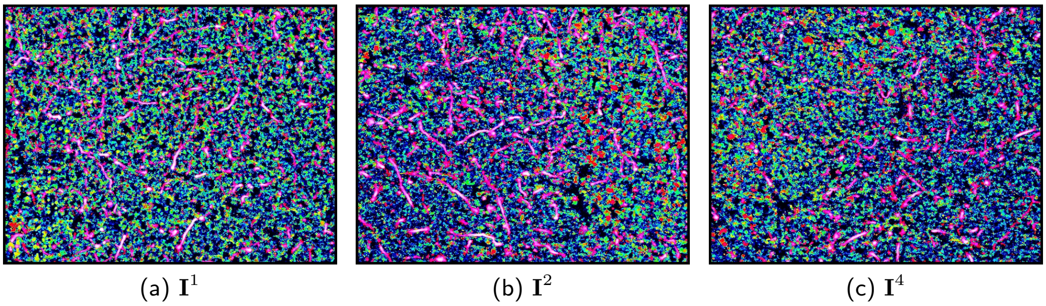

(a) $\mathbf{I}^1$ (b) $\mathbf{I}^2$ (c) $\mathbf{I}^4$

Figure 6.51: Cluster maps obtained for whole image clustering with adapted feature extraction.

variation rather than synapse specific signals, a feature extraction was performed similar to that described in section 6.1.2. Therefore, at each pixel the local background intensity of a 11×11 neighborhood around that pixel was calculated. The difference between the 3×3 Gaussian filtered pixel intensity and the background intensity was used as the feature value. Here, a smaller neighborhood size was chosen for the calculation of the background intensity since no pixels were filtered out as it was the case in the synapse specific feature calculation where synaptic regions were omitted. Figure 6.51 displays the obtained cluster maps for $\mathbf{I}^1, \mathbf{I}^2$, and $\mathbf{I}^4$. It can be seen that now similar colors are assigned to each image stack and higher image stacks correlations are obtained (see figure 6.50 (c) and (d)). Thus, features extracted from different image stacks are better comparable. However, synaptic regions can not be visually delineated in any of the cluster maps and a separation between SR and SP is not visible in this display. In order to analyze synaptic protein colocation, synaptic regions have to be manually found in the image display itself, for example through the analysis of `syphys`, `synap` and `nmdr1` channels, which is very time consuming and not feasible for an efficient analysis. Alternatively, the set of 300 prototypes needs to be manually evaluated to find synapse specific protein colocation profiles. However, it is not known which protein colocations can be found at synapses so an efficient analysis of the obtained prototypes is not possible. For each prototype, one would need to check if it coincides with a synaptic region in the image domain which is not feasible and overburdens the perceptual skills of a human observer. Yet, through the previously performed segmentation-based analysis there already exists knowledge of the protein colocations occurring at synaptic sites. Therefore, it can be evaluated if prototypes can be observed in the set of 300 prototypes which show synapse specific protein patterns. Prototypes were found which were similar to those obtained through the segmentation-based analysis and were found at similar image regions. As expected, the synapse specific protein colocation information is contained in the whole image data set. However, extracting this information is much more difficult than in the segmentation-based setup. Furthermore, image stack correlation is not solely based upon synaptic information but on the whole image content and is therefore not meaningful.

6.7 Discussion

In the last section, the proposed feature extraction, clustering and visualization techniques were applied to a set of five tissue samples and one additional negative data set. It was demonstrated that the analysis strategy proposed in this work is well suited for the evaluation of TIS protein colocations and reasonable results were obtained.

It has been shown that a feature extraction strategy tailored to the needs of the problem at hand is highly needed. Without sophisticated feature extraction, image specific intensity variations are detected rather than synapse specific differences within and between images. Although the proposed feature extraction is not perfect and potentially introduces artifacts at image regions where a border between high an low intensities can be observed, reasonable feature vectors could be extracted for the majority of the synapses. Certainly it is not known if the extracted feature vectors represent the true biology as it is not known a priori which protein colocations can be observed. However, the finding made in this study that there exist synapses specific for the SR and SP region is consistent with the observations made in a CMP-based synapse colocation study (Bode and Krusche (2007)). Similar the observations presented in this work, Bode et al. have found SR and SP specific synapses where SR synapses featured high levels of `nmdr1`. Furthermore, clustering of the feature vectors separated the synapse cases $\mathbf{I}^1, \ldots, \mathbf{I}^5$ from the non synapse cases $\mathbf{I}^6$ very well especially if all 21 proteins are used. Additionally, clusters are nicely distributed across all five images stacks $\mathbf{I}^1, \ldots, \mathbf{I}^5$.

Furthermore, the significance of a semantic image annotation could be demonstrated. Although synapse specific colocations are implicitly contained in the whole image data set, their analysis is quite laborious. Either the image domain has to be visually interpreted to find synapses and hence synapse specific protein patterns. Or the prototype visualizations have to be evaluated to delineate prototypes associated to synaptic regions. However, without the knowledge synapse specific protein combinations, filtering the large set of prototypes is not feasible.

Three clustering methods were applied to the problem of feature vector grouping. It is evident that without the presence of a gold standard, it is hard to define which of those three algorithms reveals the best results or how many clusters should actually be chosen to represent the data. To get a better impression if a cluster result might be feasible, a biological expert would need to perform an in detail analysis and visual exploration of the data with the proposed analysis strategies. Yet, it can be stated that the results obtained through the different clustering strategies are well comparable, especially between NG and K-means. Through its hierarchical nature, the H^{2}SOM could not represent rarely occurring protein patterns well at the second level. However, the third level with 120 prototypes already showed a variety of protein combinations. This hierarchical nature might even be advantageous in the analysis of TIS data as it might support the idea of a lead protein. In the lead protein hypothesis, it is assumed that one protein is essential for many protein networks and deactivating a lead protein results in the loss of all associated protein networks and thus cellular function. With the H^{2}SOM approach, one might be able to observe lead proteins in the first levels and the individual networks can then be delineated in higher levels. However, this hypotheses has not been analyzed in this study. Irrespective of the clustering method, all results revealed SR and SP specific synapses and separated the synapse from the non synapse case. This separation could also be observed when analyzing the obtained image stack correlations. Here, especially $\mathbf{I}^2, \ldots, \mathbf{I}^5$ showed very high correlations whereas the correlations observed for $\mathbf{I}^1$ were surprisingly lower. Yet, as has been discussed in the result section, this low correlation could be caused by a smaller SP tissue region and prototypes which are lower in their overall protein signals. It might therefore be reasonable for the future, to take into account the individual tissue types separately and to consider prototype similarity for correlation calculation.

Concerning the provided visualizations, especially the CIPRA visualization combined with the cluster maps were well suited for the analysis of the TIS data. Interesting, and reasonable findings could therewith be made for the given TIS image sets. Yet, their full potential can only be exploited in an interactive and linked approach. As has been experienced in this thesis, static displays are a poor choice for communicating the findings made and especially for generation of new knowledge and hypotheses. One has to switch between different visualizations, link between findings made in the feature and image domain, and change the color assignments according to the current focus of investigation. Therefore, approaches as discussed for the H^{2}SOM are especially valuable, as the color mapping can be changed interactively and intuitively.

6.8 Summary

In this chapter, I have proposed a strategy for the protein colocation analysis in high-dimensional TIS data. Semantic image annotation and sophisticated feature extraction were combined to extract synapse specific protein colocation information. It has been shown that the integration of unsupervised learning and interactive visual data mining combined with sophisticated visualization strategies greatly aids in the knowledge discovery process. It allows for a purely data driven exploration of multivariate image data. A new visualization strategy, the CIPRA, was introduced as a graphical representation of cluster prototypes. With the CIPRA, a graphical display tailored to the needs of protein colocation analysis has been introduced which was found to be very valuable in the TIS analysis process. Prototypes could easily be compared and the main protein profile could be extracted. With the proposed strategy, findings were made which are in accordance to CMP-based results suggesting that the proposed segmentation-based TIS analysis strategy is well suited for the analysis of non-binary TIS data.

Chapter 7

Statistical Synapse Distribution Analysis

The spatial information inherent in the TIS image data is the special benefit of imaging in microscopy. Hence, topological ordering of protein colocation patterns, for example at synaptic regions, can be evaluated. In the previous chapter, the spatial distribution of synapse types, i.e. synapses belonging to one cluster, was evaluated purely by visual inspection. As has been shown and discussed in section 6.6, the visual impression of object distribution in the cluster maps, however, often largely depends on the assigned prototype colors. Hence, it would be beneficial to statistically assess the distribution of synapses in the image to obtain a more objective measure.

In ecology, various approaches have been proposed to statistically analyze spatial point patterns and deduce the processes which cause the observed patterns (for example Stoyan and Penttinen (2000); Ripley (1981); Diggle (1983)). A point pattern consists of a set of locations in a defined study region. This can be, for example, the locations of trees in a part of a forest. The spatial distribution of the trees might be influenced by several conditions as soil, seed distribution or competition between different plants which cause regularity or aggregation of plants at different spatial scales. It is of special interest to determine whether a point pattern is random, clustered or regular. In recent years, point pattern analysis has also been used to analyze microscopy image data, showing the wide field of applicability of these concepts. For example, the spatial location of dividing and non dividing nuclei in breast cancer has been studied by Mattfeldt et al. (2009), while Fleischer et al. (2006) investigated the distribution of centromers.

In this chapter, I will apply methods for statistical point pattern analysis to the synapse data set. Hence, the point pattern is the set of locations of synapses and the study region is the imaged tissue sample. Thus, the term point refers to a synapse location in the following. Analysis of the data was performed by the Programita tool[1], kindly provided by Thorsten Wiegand from the Helmholz Center for Environmental Research - UFZ. The description of the statistical point pattern analysis methods follow Wiegand and Moloney (2004).

7.1 Ripley's K and O-Ring Statistic

When analyzing point pattern distributions one has to distinguish between first and second order effects. First order effects describe the density or intensity λ of points in the study region, i.e. large scale variations of the point pattern. First order effects in ecology are

[1] http://www.oesa.ufz.de/towi/towi_programita.html

caused, for example, by obstacles as stones. Second order effects subsume measures which analyze the distribution of inter-point distances. They describe the small scale variations and correlations in the pattern, caused for example by the dispersal of seeds to continue with examples from the field of ecology. To faithfully analyze second order effects, first order effects have to be properly handled. Otherwise, they bias the interpretation of the second order statistics. In the following, I will shortly introduce the point pattern statistics applied in this study.

7.1.1 Ripley's K Statistic

A common choice to calculate second-order statistics on a point pattern is the Ripley's K-function (Ripley (1976, 1981)). It statistically assesses the characteristics of all inter-point distances over a range of distance scales by analyzing circular regions. Thereby, they provide more information on the scale of the pattern than for example methods which use only information of the nearest neighbor as in Diggle (1983).

The function $K(r)$ is the expected number of points within a circle with radius r divided by the intensity λ, i.e.

$$K(r) = \lambda^{-1} E[\text{ \# points within distance } r \text{ of a randomly chosen point}] \;, \tag{7.1}$$

with $E[\cdot]$ denoting the expectation operator and # stands for the "number of". λ specifies the intensity of the point pattern, which can be estimated as (# points)/(area of study region). For a bivariate point pattern, i.e. two types of points are given, equation 7.1 can be adapted to

$$\begin{aligned} K_{k,k'}(r) = \lambda_{k'}^{-1} E[&\text{ \# points of type } k' \text{ within distance } r \\ &\text{of a randomly chosen point of type } k] \;. \end{aligned} \tag{7.2}$$

In the case of protein colocation analysis, the type of a point is defined by its cluster assignment, so the point types are referred to by k and k' to be consistent with the notation of chapter 6.

7.1.2 O-ring Statistic

An alternative to Ripley's K function to analyze spatial point patterns is the *pair-correlation function* (Ripley (1981); Stoyan and Stoyan (1994); Wiegand and Moloney (2004))

$$g(r) = \frac{1}{(2\pi r)} \frac{dK(r)}{dr} \;, \tag{7.3}$$

which can be used to derive the O-ring statistic (Wiegand et al. (1999); Wiegand and Moloney (2004))

$$O(r) = \lambda g(r) \;. \tag{7.4}$$

With the O-ring statistic, rings of radius r with a predefined width w instead of circles are used to assess the topological order of a point pattern. Hence, specific distances can be analyzed instead of mixing effects at larger distance scales with effects at smaller distance scales as it is the case in the K-function. However, it requires a decision on the ring width. If rings are chosen too narrow, not enough points will fall within the ring so that no meaningful measures can be calculated. For too wide rings, the advantage of isolating order effects at specific distances will be lost.

Analogous to the K-function, the bivariate O-ring function can be defined as

$$O^{w}_{k,k'}(r) = \lambda_{k'} g_{k,k'}(r) = E[\ \#\ \text{points of type } k' \text{ at distance } r \text{ of a randomly chosen point of type } k]\ . \tag{7.5}$$

The univariate statistic is similarly calculated by treating all points as one point type.

7.1.3 Numerical Estimation

When estimating the $K(r)$ or $O(r)$ function from the data, it has to be considered that they are both defined under the assumption of homogeneous and isotropic point patterns, meaning that they can not directly be applied to heterogeneous patterns, i.e. first order effects are present (Goreaud and Pélissier (2003)). Furthermore, at the edges of the study region, the estimates will be biased if edge effects are not properly handled.

To estimate $K(r)$ and $O(r)$, basically two approaches can be applied. The analytical approach considers all inter point distances and uses geometric formulas to calculate weights that account for edge effects (Wiegand and Moloney (2004); Ripley (1976, 1981)). A more simple strategy which can easily cope with edge effects is the numerical approach, which uses an underlying grid of cells to estimate $K(r)$ and $O(r)$. While in ecology it is a crucial decision how to divide the study region into a regular grid, for microscopy image data a grid structure is implicitly given. In the imaging process the continuous space is already divided in to a grid, i.e. pixels, which can now readily be used.

The K-function is estimated by

$$\hat{K}_{k,k'}(r) = \lambda_{k'}^{-1} \pi r^2 \frac{\frac{1}{|\mathcal{C}_k|}\sum_{i=1}^{|\mathcal{C}_k|} \mathrm{Points}_{k'}[C_i(r)]}{\frac{1}{|\mathcal{C}_k|}\sum_{i=1}^{|\mathcal{C}_k|} \mathrm{Area}[C_i(r)]}\ . \tag{7.6}$$

$C_i(r)$ is a circle centered at point $\mathbf{p}^{(x_i,y_i)}$ with radius r. $\mathrm{Points}_{k'}[C_i(r)]$ counts the points of type k' lying within circle $C_i(r)$. $\mathrm{Area}[C_i(r)]$ is the area of the circle lying within the study region. $\lambda_{k'}$ is defined by $\lambda_{k'} = |\mathcal{C}_{k'}|/A$ with A being the area of the study region. The area is estimated by

$$\mathrm{Area}[C_i(r)] = \sum_x \sum_y S(x,y) I_r(x_i, y_i, x, y)\ , \tag{7.7}$$

with $I_r(x_i, y_i, x, y) = 1$ if $\sqrt{(x-x_i)^2 + (y-y_i)^2} \leq r$ otherwise $I_r(x_i, y_i, x, y) = 0$ and $S(x,y) = 1$ only if pixel $\mathbf{p}^{(x,y)}$ belongs to the study region, otherwise $S(x,y) = 0$. Thereby, only pixels within the study region, i.e. image, are counted which are within the radius of the circle. Without this strategy, the estimate would be biased at the edge of the study region.

The point count is defined similarly by

$$\mathrm{Point}[C_i(r)] = \sum_x \sum_y S(x,y) P_{k'}(x,y) I_r(x_i, y_i, x, y) \ , \tag{7.8}$$

where $P_{k'}(x,y) = 1$ if the pixel $\mathbf{p}^{(x,y)}$ contains a point, i.e. synapse, of type k' and $P_{k'}(x,y) = 0$ otherwise.

The O-ring function is defined similarly, with the difference that

$$I_r(x_i, y_i, x, y) = \begin{cases} 1 & \text{if } r - \frac{w}{2} \leq \sqrt{(x-x_i)^2 + (y-y_i)^2} \leq r + \frac{w}{2} \\ 0 & \text{otherwise} \end{cases} \ , \tag{7.9}$$

and

$$\hat{O}^w_{k,k'}(r) = \frac{\sum_{i=1}^{|\mathcal{C}_k|} \mathrm{Points}_{k'}[R_i^w(r)]}{\sum_{i=1}^{|\mathcal{C}_k|} \mathrm{Area}[R_i^w(r)]} \ , \tag{7.10}$$

where $R_i^w(r)$ is the ring with radius r and width w.

7.2 Null Models for Univariate and Bivariate Point Patterns

An important aspect in point pattern analysis is the selection of an appropriate null model, i.e. a hypothesis of the distribution of the data. Different null models exist which can be applied, however, in the following I will only introduce those models which are applied in this work. To analyze the significance of the departure of the observed data from the null model, Monte Carlo simulation is used to obtain confidence envelopes (Wiegand and Moloney (2004)). Therefore, $\hat{K}(t)$ and $\hat{O}(t)$ are determined for a number of simulated realizations of the stochastic process underlying the chosen null model. If the highest and lowest values of $\hat{K}(t)$ respectively $\hat{O}(t)$ are chosen to calculate the confidence envelopes, then $n/(n+1) * 100\%$ confidence envelopes are obtained (with n being the number of simulations). If the observed $\hat{K}(t)$ or $\hat{O}(t)$ has values outside the confidence envelope, it is considered to be a significant departure from the null model (Wiegand and Moloney (2004)).

7.2.1 Univariate Null Models

The simplest and widely used null model for univariate point patterns is *complete spatial randomness* (CSR), which can be used to test whether the observed point pattern is in consistence with a homogeneous Poisson process. Here, homogeneous refers to the fact that no first-order effects occur in the data and the intensity λ is almost constant in the entire study region. The Poisson process implies that the probability of finding l points in an area of size M follows a Poisson distribution, i.e. points are observed with equal probability at any position in the study region and the position of a point is not affected by the position of another point. Under CSR $K(t) = \pi r^2$ holds for all r. In practice it is easier to use $L(r) = \sqrt{\frac{K(r)}{\pi}} - r$ instead, since the variance is approximately constant under CSR and allows for easier visual inspection (Dixon (2002); Wiegand and Moloney (2004)). Hence, for

CSR $L(r) = 0$ holds. Values $L(r) > 0$ indicate aggregation of the pattern up to distance r, i.e. more points then expected, and $L(r) < r$ indicates regularity of the pattern. Similarly, under CSR $O(r) = \lambda$ holds and $O(r) > \lambda$ indicates aggregation while $O(r) < \lambda$ indicates regularity.

Evaluation against CSR requires that no first-order effects occur in the data, i.e. the point pattern is homogeneous. Only then, departures from the null model can be ascribed to true second order effects. If the point pattern is heterogeneous, i.e. there are larger gaps in the study region with no or very few points, evaluation against CSR can lead to misinterpretation of the data. In ecology, gaps often occur because of obstacles in the study region or specific soil conditions. However, also in the case of the synapse data, gaps occur due to biological constraints. Synapses can not be detected in the regions of cell nuclei and for example blood vessels. Thus, those regions have to be excluded from the study region. This can either be achieved by providing a mask which specifies all pixels which should be excluded or a null model has to be used which accounts for the first-order characteristics. The shape of the study region would be irregular and more edges are introduced, however, the numerical estimator properly handles these edge effects. The simplest alternative for the CSR for non homogeneous patterns which does not require a mask image is the *heterogeneous Poisson process*. Here, the intensity λ, which is constant for the whole study region for CSR, is replaced by space dependent function $\lambda(x, y)$ whose value varies with location (x, y). It is now the question how this varying intensity can be estimated from the data. For the numerical implementation of $K(r)$ and $O(r)$ Wiegand and Moloney (2004) propose an estimator $\hat{\lambda}^R(x, y)$ to calculate the non-constant intensity. It is defined as

$$\hat{\lambda}^R(x, y) = \frac{\text{Points}[C_{(x,y)}(R)]}{\text{Area}[C_{(x,y)}(R)]} , \tag{7.11}$$

where $C_{(x,y)}(R)$ is a circular moving window with fixed radius R. For each position $\mathbf{p}^{(x,y)}$ in the study area it is counted how many points fall within the window. To account for edge effects at the border of the study region, the value is again weighted with the area of the circle within the study region. It is evident that the estimate depends on the radius R which has to be chosen appropriately. The heterogeneous Poisson process can now be modeled by placing points at positions $\mathbf{p}^{(x,y)}$ in the study region but preserving each point only with a probability $\hat{\lambda}^R(x, y)$. An alternative to the heterogeneous Poisson process is to delineate homogeneous sub-regions in the data as described in Wiegand and Moloney (2004) and apply the CSR null model on the homogeneous sub regions. However, for study regions which show several gaps of different sizes, i.e. complex first-order heterogeneity, the heterogeneous Poisson null model is recommended (Wiegand and Moloney (2004)).

Another aspect which has to be considered for the choice of a null model is the case when objects can, by definition, not be closer to each other than a certain minimal distance δ. Especially for the L-function, which has a "memory effect" as it accumulates effects of different scales, this can lead to erroneous interpretation of the data. Therefore, it can be reasonable to combine a *hard-core process* with one of the other models. In a hard-core process, points are placed in the study region following the specified null model. However, a point $\mathbf{p}$ is only accepted if it has a distance $d(\mathbf{p}, \mathbf{p}') \geq \sigma$ to the closest point $\mathbf{p}'$.

7.2.2 Bivariate Null Models

Analysis of bivariate point patterns is much more difficult than the analysis of univariate patterns, as the spatial relationship of two patterns has to be analyzed, which themselves can have a complex spatial structure. Furthermore, visualization of the pattern mostly does not provide an intuitive idea of the underlying first and second order effects and there does not exist a simple null model as the CSR. One generally has to distinguish between two conceptually different null models: *independence* and *random labeling* (Goreaud and Pélissier (2003); Wiegand and Moloney (2004)). The hypothesis of independence assumes that the two patterns are *a priori* the result of two independent processes and the expected absence of interaction between the two patterns corresponds to the fact that the location of one pattern is independent from the location of the other pattern. Random labeling, on the other hand, assumes that both patterns were created by the same stochastic process and a posteriori the two types are assigned to the individuals of the population. In random labeling, it is not investigated if the processes interact with each other but if the type assignment is random within the given spatial structure of the joint pattern. The absence of interaction would mean that the probability of one type is the same for all points and does not depend on neighbors. These two models lead to the computation of different confidence intervals and if the wrong null model is chosen it can lead to erroneous interpretation of the data (Goreaud and Pélissier (2003)).

In this study, the random labeling null model is chosen, assuming that synapses are created by one stochastic process and their type, i.e. molecular network, is assigned a posteriori. Existing synapses will change their molecular networks, i.e. functions, to react to different conditions rather than removing synapses and creating new synapses.

Random labeling can be modeled by keeping the location of the joint pattern fixed and randomly assigning type labels to each location (Wiegand and Moloney (2004); Goreaud and Pélissier (2003)). To characterize departure from the null model of random labeling, the following has to be considered. Each of the patterns represents random "thinning" of the joint patterns, and the K-function as well as the g-function is invariant under random thinning (Wiegand and Moloney (2004)). Therefore, it is to be expected that $K_{k,k'}(r) = K_{k',k}(r) = K_{k,k}(r) = K_{k',k'}(r)$ (Wiegand and Moloney (2004); Dixon (2002)). Hence, $K_{k,k'}(r) > K(r)$ would indicate that more type k' points are around a type k point than would be expected, and thus type k and k' tend to be positively correlated (Goreaud and Pélissier (2003)). The difference $K_{k,k}(r) - K_{k,k'}(r)$ indicates if points of type k tend to be surrounded more by other points of type k than expected, and $K_{k,k}(r) - K_{k',k'}(r)$ evaluates if one type is more clustered than the other (Dixon (2002)). Only if both types are equally close to the border of the study region, $K_{k',k} = K_{k,k'}$ holds. For irregular shaped study regions, however, $K_{k,k'}(r) \neq K_{k',k}(r)$ but they are positively correlated (Dixon (2002)).

7.3 Sample Application

In this section, the presented methods for investigating point pattern distributions are applied to the stacks of the synapse data set $\mathcal{I}$. First, univariate point pattern distributions are

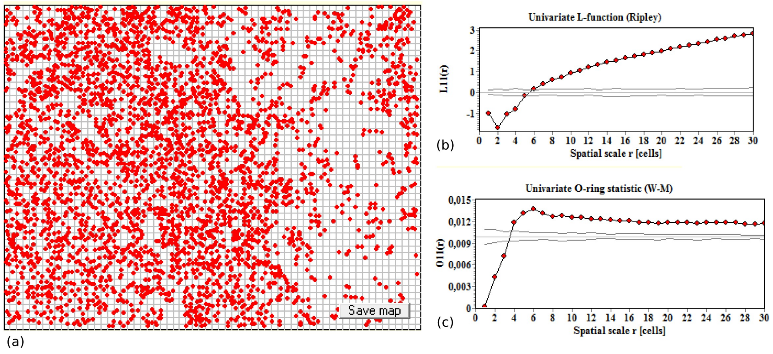


Figure 7.1: Univariate analysis of synapse distribution of image stack $\mathbf{I}^2$. (a) shows the observed point pattern with synapses marked with red dots. (b) displays the Ripley's L-function showing the local neighborhood density up to scale r (red circles) and the confidence envelopes (solid lines) around the straight line at 0. (c) shows the O-ring statistic which gives the local neighborhood density at scale r with a ring width $w = 3$. The solid straight line indicates the intensity λ of the pattern. Confidence envelopes were constructed through 99 simulations.

considered, i.e. the distribution of all synapses in the image stack is analyzed. Second, exemplarily bivariate analysis is performed for some selected synapse clusters. In each study, confidence envelopes are constructed through the highest and lowest values of 99 simulations, thus constructing 99% confidence intervals.

7.3.1 Analysis of Synapse Distribution

To study the distribution of synapses in the considered tissue sample, a univariate distribution analysis was performed with the O-ring and Ripley's K statistic. CSR was chosen as a null model in both cases and scales up to 30 pixel were considered. A ring width of $w = 3$ was set for the O-ring statistic to account for slight inaccuracies of the synapse location. Figure 7.1 exemplarily shows the results obtained for sample $\mathbf{I}^2$ with the spatial synapse distribution shown in (a) and the obtained $\hat{L}(r)$ and $\hat{O}(r)$ with red circles in (b) and (c), respectively. The confidence envelopes are shown as lines around a straight line which depicts the zero value in (b) and the intensity λ of the pattern in (c). Ripley's K suggests regularity for scales up to $r = 4$ and aggregation for scales $r > 6$ (see figure 7.1 (b)). Quite similar observations were made for the O-ring statistics with regularity for scales up to $r = 3$ and aggregation for scales $r > 4$ (see figure 7.1 (c)). However, these observations indicate an effect of virtual aggregation, i.e. the observed aggregation is due to first order effects rather than second order effects. Virtual aggregation is indicated by constant $O(r)$, well above the intensity λ, for a larger range of scales (Wiegand and Moloney (2004)) and by a linear increase in $\hat{L}(r)$. This is due to the fact that starting at some scale r_1 the rings or circles overlap the gaps in the study region. When comparing the distribution of synapses in figure 7.1 (a) with the gray value images of stack $\mathbf{I}^2$, it can be observed that these gaps are actually due to biological constraints, as cell nuclei and "holes" in the sample where no synapses can be observed. Thus, in order to account for these first order effects and only consider true second order

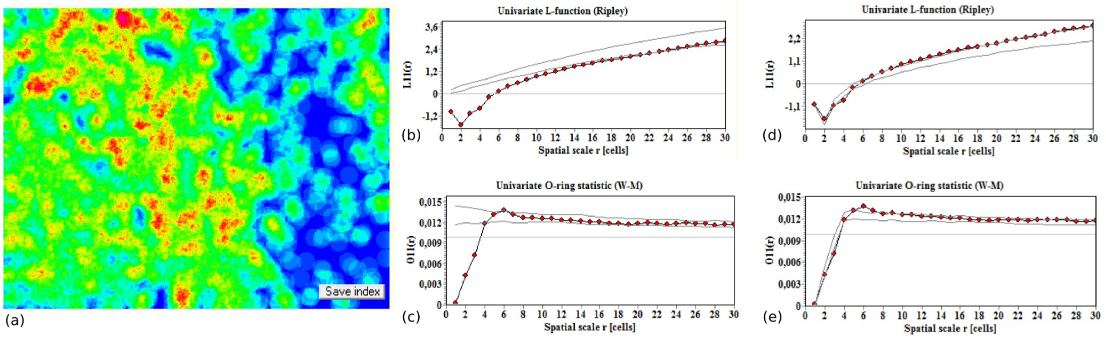


Figure 7.2: Univariate point pattern analysis for $\mathbf{I}^2$ with a heterogeneous Poisson process. (a) intensity map generated through the moving window estimate with $R = 15$ for the point pattern shown in figure 7.1(a). (b) and (c) display the L-function and O-ring statistic obtained for the point pattern with confidence envelopes constructed through the heterogeneous Poisson process. (d) and (e) confidence envelopes were constructed for a heterogeneous Poisson process with an additional hard-core process with minimum distance 3.

effects, these regions have to be excluded from the study.

Excluding those regions from the study area which can not show synapses due to biological constraints can be achieved by constructing a mask image, for example by thresholding and combining different images of the stack such as syphys, nmdr1 and prop (data not shown). Similar results, however, can be obtained by applying a heterogeneous Poisson process as a null model. Thus, the potentially tedious mask construction can be omitted and the pattern itself can be used to estimate the non-constant first order intensity. Figure 7.2 (a) displays the obtained intensity map with $R = 15$ (see equation 7.11) for the pattern shown in figure 7.1 (a) with blue regions indicating low intensity and red regions high intensity of points. Figure 7.2 (b) and (c) display the obtained $\hat{O}(r)$ and $\hat{L}(r)$ values. It has to be noted that $\hat{O}(r)$ and $\hat{L}(r)$ did not change but the confidence envelopes differ from those shown for the CSR null model (compare figure 7.1) as now the simulations are performed in accordance with the non homogeneous intensity λ. For the O-ring statistic, again a regularity up to scales $r = 3$ can be observed, a very weak aggregation at scale $r = 6$ and random distribution for the other scales. The regularity for scales up to $r = 3$ is well in accordance with the observed data. In the process of synapse detection, synapse positions were evaluated so that their distance is ≥ 3 (see equation 4.44). When merging positions obtained from different images in one stack, a minimum distance of $d(\mathbf{p}, \mathbf{p}') \geq 2$ between two position, i.e. synapses, was required (see section 6.6). Thus, at distances < 2 no synapses can be observed and at distances < 3 only very few synapses occur. Since rings of width $w = 3$ were chosen, scales up to $r = 3$ capture this required minimal distance very well (see equation 7.9). A similar observation can be made for $\hat{L}(r)$ where high regularity is observed for $r = 2$. However, the L-function shows a memory effect so that regularity is observed for all scales (see figure 7.2 (b)). To eliminate this memory effect, it is reasonable to introduce a hard-core process which accounts for the known regularity. Therefore, a hard-core distance of 3 is chosen, although it is evident that some points will occur in scales $2 \leq r \leq 3$. The confidence envelope for

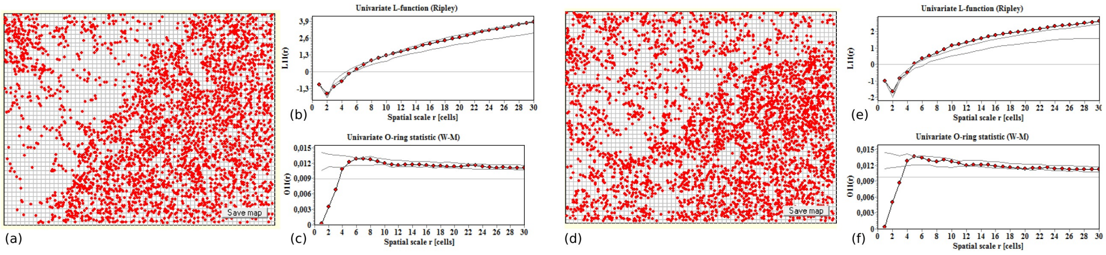


Figure 7.3: Univariate point pattern analysis for $\mathbf{I}^4$ and $\mathbf{I}^5$. (a) and (d) show the point patterns, (b) and (e) the obtained L-function and (c) and (f) the obtained O-ring function.

the L-function in figure 7.2 (d) now encloses the observed function, with a slight departure at $r = 2$ and $r = 4$. The O-ring statistics with a hard-core process are shown in figure 7.2 (e) for comparison. Again, the confidence envelope encloses the $O(r)$ estimate. However, as the O-ring function has no memory effect, larger scales are less influenced by small scale regularities. Summing up, it can be concluded that the synapses are distributed randomly across the sample, possibly with a slight aggregation at scale $r = 6$. Similar observation can be made for image set $\mathbf{I}^4$ and $\mathbf{I}^5$, shown in figure 7.3, and $\mathbf{I}^1$ and $\mathbf{I}^3$ and the negative case $\mathbf{I}^6$ (data not shown).

7.3.2 Class Specific Synapse Distribution

For a complete analysis of bivariate cluster specific synapse distribution in the images, all pairwise combinations of the 30 clusters in each image would need to be inspected. To illustrate the applicability of the statistics, I will however only show two examples for image $\mathbf{I}^2$ and the cluster result obtained via NG clustering with 30 prototypes and 12 proteins (see section 6.6.3).

First, a point pattern is inspected where it is evident from visual inspection that both types of patterns are clustered. Cluster 5, which is a SR specific cluster, is compared to cluster 8 which is a SP specific cluster with the pair-correlation function with width 3, since it does not suffer from memory effects. A mask image was applied to account for first order effects, however, similar results were obtained without this masking. Confidence envelopes were constructed by 99 simulations of the pattern under the null model. Figure 7.4 displays the point pattern in (a) where cluster 5 is displayed in red and cluster 8 in green. In (b) the pair-correlation function $g_{5,8}(r)$ for both patterns is shown. It can be observed that starting at scale $r = 4$, type 8 synapses are weaker correlated to type 5 synapses then expected and similar observations can be made for the opposite direction (see the low difference of $g_{5,8}(r) - g_{8,5}(r)$ in figure 7.4 (e)). Random labeling of scales up to $r = 3$ can be explained by the fact that no or only few points are found up to this distance, as a minimum distance was required in the synapse detection process. Furthermore, it can be observed that type 5 synapses are more surrounded by type 5 synapses then expected starting at scale $r = 4$ (see figure 7.4 (d)). Similarly, type 8 synapses are positively correlated with type 8 synapses at scales $r = 3, \ldots, 8$ and cluster more at those scales then type 5 synapses (see figure 7.4

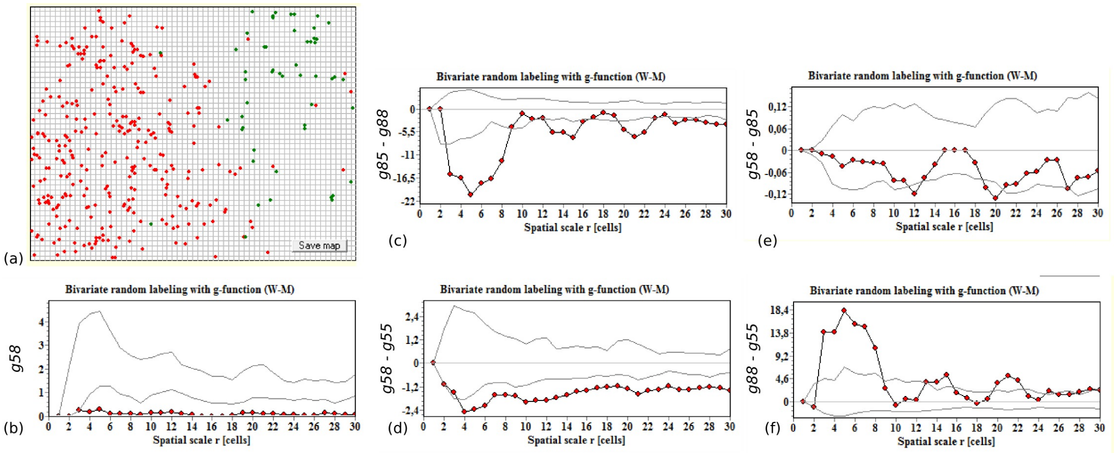


Figure 7.4: Bivariate point pattern analysis for image set $\mathbf{I}^2$ and clusters 5 (red) and 8 (green). The observed pattern is shown in (a) and the pair-correlation function under random labeling in (b). Derived measures are displayed in (c-f). (c) corresponds to the measure if type 8 synapses tend to be surrounded by type 8 synapses more than by type 5 synapses. (d) reflects the same measure but for type 5 points. With (e) it can be evaluated how similar the two bivariate measures $g_{5,8}(r)$ and $g_{8,5}(r)$ are. (f) evaluates whether type 8 points are more clustered than type 5 points conditional on the observed structure of the joint pattern.

(c) and (f)). Thus, the visual impression of the pattern that type 5 and type 8 synapses cluster with synapses of similar type but are separated from synapses of the different type could be verified by evaluating the pair-correlation function. Similar observations were made for the remaining four synapse data stacks (data not shown). Repulsion of patterns 5 and 8 was observed for all image stacks starting at scale 3 or 4. Clustering of type 5 synapses was observed for all images stacks for $r = 6, \dots, 30$ with one exception. Image $\mathbf{I}^1$ shows random labeling starting at $r = 13$. Clustering of type 8 synapses was observed for all stacks starting from $r = 3$ and ending, depending on the stack, between $r = 5$ and $r = 10$. Again, image $\mathbf{I}^1$ differs and shows clustering of type 8 synapses for ranges $r = 10, \dots, 12$. This different behavior of points of image stack $\mathbf{I}^1$ might arise from the fact that it shows a much smaller SP region and hence a lower number of synapses belonging to type 8. The negative sample case was not considered as it does not contain synapses of type 8 and only very few synapses of type 5.

A second study has been performed for clusters 0 and 3, which mostly occur in the SR region of the tissue sample. Figure 7.5 displays the observed point pattern with cluster 0 in red and cluster 3 in green and the observed statistical measures. It can be observed that type 0 and type 3 are correlated as expected conditioning the observed joint pattern (see figure 7.5 (a) and (e)). As the difference $g_{0,3}(r) - g_{0,0}(r)$ indicates, type 0 synapses are not relatively more frequent at distance r around type 0 points then type 3 synapses (see figure 7.5 (d)). However low positive correlation of type 3 synapses can be observed for scale 2 and 4 (see figure 7.5 (c)). Nevertheless, type 3 synapses are not significantly more clustered than type 0 synapses, conditional on the structure of the joint pattern (see figure 7.5 (f)).

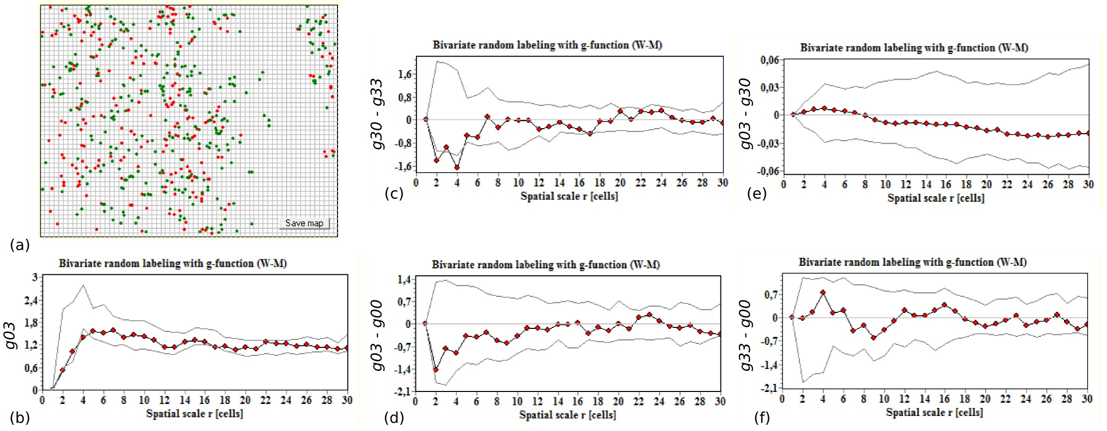


Figure 7.5: Bivariate point pattern analysis for image set $\mathbf{I}^2$ and clusters 0 (red) and 3 (green). (a) displays the observed pattern. (b-f) shows the computed statistical measures with confidence intervals obtained through 99 randomizations.

Summing up, it can be concluded that random labeling is given for cluster 0 and 3. Similar results were obtained for the remaining four image stacks.

7.4 Discussion

In this section, the applicability of point pattern analysis strategy to TIS image data has been demonstrated. It could be observed that synapses seem to be distributed following a heterogeneous Poisson process in each of the image stacks and the known regularity up to a distance of three pixels could be verified by the chosen methods. Also for the analysis of bivariate point patterns, visually observable structures could be verified. Thus, point pattern analysis by Ripley's K and the O-ring statistic can be a well suited complementary method to analyze the spatial distribution of synapses and synapse groups. They are less effected by visual perception and provide for a statistical measure of topological organization. To obtain meaningful statistics, first order effects have to be taken care of. This can be difficult if gaps in the study region can not easily be explained by biological factors. Furthermore, especially in the bivariate case the right null model has to be chosen so that meaningful conclusion can be drawn.

To perform a full evaluation of the bivariate structure of the synapse data set, each pairwise combination of synapse clusters would need to be analyzed for each image. This could be guided by visualizations similar to the color lines view. For each image, the obtained bivariate measures could be represented as a line, where each segment corresponds to a scale r. A bivariate color scale could be chosen to represent aggregation or repulsion of the data. Hence, if all image stacks feature the same characteristic for a bivariate pattern, it would easily be observable in the color lines view.

Chapter 8

Conclusion and Outlook

In this thesis, a novel analysis strategy for the exploration of non binarized protein colocation patterns in multivariate tissue micrographs has been proposed. Multivariate fluorescence imaging is a new field in microscopy for which only a very limited number of available analysis methods exist. This thesis therefore aims at providing approaches which help the biologists in getting an insight and understanding of this new data domain. The presented strategy consists of a SVM-based object detection system, termed i3S, and the subsequent actual data analysis which integrates concepts from the field of data mining, visualization and statistics.

Object detection has been introduced to reduce the number of data points which need to be analyzed. It additionally offers the benefit that object-specific protein information can be extracted and linked to semantic meaning. In the case of the pancreas data set, object detection was even the prerequisite to extract meaningful object features allowing the different cell types to be distinguished. A reasonable nucleus detection performance has been achieved with the i3S system. For the synapse data sets, it has been shown that object detection greatly aids in the subsequent analysis of protein colocation at synapses. Individual protein patterns can directly be associated to individual synapse locations. With this mapping, the spatial distribution of synapses with similar patterns has been explored. Although the same information is implicitly contained in a data set of the whole image, it is very laborious to find protein patterns of synapses in this large amount of data.

To prove the real world applicability of the i3S system, a careful evaluation of its accuracy and stability in synapse detection was performed. This analysis, however, could only be based on reference lists of manual human labeling as no true gold standard exists for this image domain. Still, the results have shown that synapse detection performances comparable to human detection performances were obtained by the i3S with respect to human expert reference lists. Thus, it can be stated that the majority of detected objects are true synapses and it is therefore possible to extract synapse specific protein patterns. The obtained results also indicate that studies will be possible quantifying synapses in thousands of tissue sections on a new statistical level which would not be possible by manual exploration. Hence, the i3S can not only be applied as a first step in multivariate image analysis but is also suited for obtaining a fast and good approximation to the neurobiological reality of synapse number with a minimum of human interaction and based on standard fluorescence microscopy.

To explore object-specific protein information, an interactive analysis strategy was proposed which integrates concepts from the field of data mining, statistics and information visualization. Special focus was put on providing visualizations tailored to the need of multivariate image analysis for the feature as well as the image domain, to allow for an efficient exploration of the data.

The potential of integrating concepts from visual data mining in the field of high-content bioimage analysis has been demonstrated for the pancreas image data set. The applied visualization strategies combined with the possibility to simultaneously explore the feature as well as the image domain have proven to be a useful concept for multivariate image analysis. With the presented feature extraction strategies, the individual cell characteristics were well captured and allowed to obtain results comparable to manual expert counting. Hence, valuable information for diabetes study can be obtained by this semi-automatic analysis approach in a fast and efficient way.

For the exploration of the higher dimensional and more challenging synapse data set, the interactive exploration of the image and feature domain was extended by methods for multivariate data visualizations and by a clustering step. A feature extraction strategy was proposed to obtain synapse specific protein colocation information. It has been shown that without this feature extraction, no reasonable protein information could be obtained but rather the intensity variation within and between images were captured. Although a reduction of the data complexity was already achieved through the object detection step, still a large number of protein patterns needed to be analyzed. Therefore, different cluster strategies were applied to further reduce the amount of data and to obtain representative prototypes. The representation of the data set through prototype vectors enables the user to explore the main characteristics of the underlying data in a first overview step. In a second step, interesting clusters or synapse patterns can then be analyzed in more detail. To support the biologists in this exploration process, several different visualizations were provided, offering overviews and in-detail access to the data. For example, the cluster maps could well be used as an overview to compare different image stacks based on their cluster occurrence, or to analyze the spatial distribution of synapses of different clusters. With the CIPRA visualization, a new way of displaying multivariate data has been introduced. This graphical display is highly tailored to the needs of multivariate imaging for protein colocation studies and was found to be very valuable in the analysis of protein colocation. Different patterns can be easily compared and the overall protein colocation can be observed. Linked with the cluster map and other visualizations, it provides a suitable way to explore the multivariate image domain.

To allow for a more objective analysis of visually made findings, different statistical measures were applied. The Spearman's rank and the Pearson's correlation were employed to measure the correlation of images based on their observed clusters. Similarity or dissimilarity of cluster maps can thus be evaluated on a more objective level than by pure visual inspection. It has been shown that the manual findings agreed very well with the obtained statistical measures. Furthermore, methods from the field of point pattern analysis were applied to statistically assess the synapse distribution and verify hypotheses generated by visual inspection. The application of these methods was possible as synapse locations were known through the introduced object detection step.

Evaluating the benefit of the proposed strategies for protein colocation analysis in multivariate image data is particularly challenging. There exists almost no a priori knowledge of the data so that the evaluation goal is very vague. Furthermore, only few other approaches are available for colocation analysis in multivariate image data which can be applied to verify the findings made. Thus, it is difficult to assess if one clustering is more reasonable then the other or if one visualization leads faster to a certain finding than the other. However, the applied approaches have shown that reasonable protein colocation could be extracted and findings were made which are in accordance to CMP-based results. Furthermore, the negative sample image stack was well separated from the positive sample cases by visual inspection as well as by statistical measures. The full potential of the system now has to be investigated in daily work.

In conclusion, it can be stated that protein colocation information of multivariate image data can well be explored in a fully data-driven manner by combining object detection with strategies from data mining, information visualization, and statistics. The interactive exploration of both the image as well as the feature domain has proven to be of great value in the exploration process and allows the biological researcher to form a mental model of the complex data domain. With the proposed object-based analysis strategies, meaningful results were obtained suggesting that such an interactive approach is well suited to extract new knowledge from multivariate tissue micrographs.

Future Research Perspectives

It has been shown that the analysis of multivariate microscopy image data for protein colocation study requires a very interdisciplinary research. Concepts from the field of image processing, data mining, visualization and statistics need to be applied to obtain a system which aids in the knowledge discovery process. Therefore, possible future work could lie in any of these fields.

One potential direction of further research is the extraction of synapse protein patterns. Although the proposed method has shown to provide reasonable results and without it no meaningful synapse protein information could be extracted, it tends to produce artifacts at image regions with high gradients as has been shown in figure 6.34. It would thus be reasonable to enhance this feature extraction. On the one hand, image regions with very low intensity could be excluded for feature calculation. On the other hand, a nucleus image channel could be used as a mask as the artifacts often occur at the rim of low intensity regions which are caused by the presence of a nucleus. Another possibility would be to consider different image enhancement and normalization strategies, so that the sole intensity value could be considered for protein colocation studies.

Further possible enhancements to the analysis strategy could be achieved in the clustering step. In this work, three different cluster techniques have been applied. They have all shown results which were in accordance with findings made through the CMP approach. However, it could be beneficial to apply additional clustering methods which take into account the characteristics of multivariate image data in more detail. For example, the applied cluster techniques represent rarely occurring protein patterns very poorly. However, these rare protein patterns might be of high importance in some applications. It would thus be reasonable to apply clustering methods which put more emphasis on rarely occurring patterns. This could be achieved by adapting the concepts of the WTA vector quantization with activity equalization proposed by Heidemann (2001). Additionally, the confidence values obtained through the object detection process could be considered for clustering. Thereby, more emphasis could be put on certain objects while uncertain objects affect the cluster outcome only to a lower extend. To better assess the applicability of cluster methods, it would be beneficial to work on synthetic multivariate image data with precise knowledge of the expected protein colocation patterns. Recently, a method to generate synthetic synapse data sets has been implemented by a student of the Biodata Mining & Applied Neuroinformatics Group, University Bielefeld, which could be extended by precisely defining the number of clusters which should be represented in the data set. Hence, it could be used as a basis to evaluate synapse clustering strategies.

Another aspect of multivariate image data analysis, which should be regarded in more detail in the future, is the statistical description of the obtained results. Right now, most of the results are obtained by visual analysis. As little is known of the underlying data, this visual analysis will most likely never be replaced by a fully automated procedure. However, it is influenced by inter- and intra- observer variability. Thus, it would be of great benefit to provide complementary statistical measures. In this work, the Ripley's K and O-ring statistic have been applied to statistically assess the distribution of synapses. Yet, the full potential of point pattern statistics in multivariate image data analysis has not been investigated and should therefore be considered in more detail in future research. Furthermore, the Pearson's or Spearman's correlation were applied to statistically describe similarity of image stacks based on their cluster result. Here no information about the similarity of cluster prototypes is regarded but each cluster is treated as being equally different. To obtain higher correlations for image stacks which show similar clusters, it would thus be reasonable to include prototype similarity information into the calculation of image correlation. Furthermore, correlation is influenced by the characteristics of the tissue sample, for example size of the SR and SP region. It could therefore be reasonable to regard different tissue regions separately. Other possibilities to statistically describe the protein colocation data could be to adapt concepts from traditional colocation analysis to the needs of multivariate colocation analysis, apply non binary frequent item set techniques to describe protein colocation (Gyenesei et al. (2006); Kuman et al. (2009)), or employ measures such as the Earth Movers Distance (Rubner et al. (1998)) to obtain a measure for the similarity of images based on the observed clusters.

Finally, the field of visualization and interactive data exploration has great potential. In protein colocation analysis, there is still a great demand for sophisticated visualization strategies which allow to easily perceive the data, find valid and useful patterns, or verify hypotheses. Furthermore, the way the proposed strategies are made available to the user is of great importance. In recent years, the world wide web has evolved. First, a growing amount of web applications allow users to generate and share digital content in an interactive way. Second, personal computers nowadays achieve high performance so that even computationally intensive web applications can be executed on the client side. This development is widely known by the term Web2.0. Providing multivariate image data analysis methods as web-based applications, also referred to as rich internet applications, has the benefit to allow for new ways to work collaboratively independent of location or computer platform. Researches can work together on the same image set, share ideas and findings. Following the concept of Web2.0, this idea is referred to as Science2.0 (Shneiderman (2008)). Currently, the Biodata Mining & Applied Neuroinformatics Group, University Bielefeld, is investigating the applicability of web-based multivariate image analysis. Different visualization and exploration methods have already been implemented showing great potential for the web-based exploration of multivariate image data.

Appendix

A.1 Acronyms

BC Border coverage

CCD Charge-coupled device

CH Calinski Harabasz index

CIPRA Combinatorial Intensity Profile Archetype

CMP Combinatorial molecular phenotype

CSR Complete spatial randomness

DAPI 4'-6Diamidino-2-phenylindol (Nucleus stain)

DB Davis Bouldin index

FITC Fluoresceinisothiocyanat

FN False negative

FP False positive

FRET Fluorescence resonance energy transfer

GFP Green fluorescent protein

HSOM Hyperbolic SOM

H^{2}SOM Hierarchical growing HSOM

i3S Intelligent synapse screening system

JND Just noticeable differences

KKT Karush-Kuhn-Tucker

MELC Multi-epitope ligand cartography method

MI Median Intensity

NG Neural Gas

PCA Principle component analysis

PPV Positive Predictive Value

RBF Radial basis function

ROC Receiver-operating characteristic

ROI Region of interest

SE Sensitivity

SOM Self organizing map

SP Stratum pyramidale

SR Stratum radiatum

SVM Support vector machine

TIS Toponome imaging system

TN True negative

TP True positive

WTA Winner takes all

A.2 Technical Abbreviations

$\mathcal{I}, \mathcal{I}'$:	sets of image stacks
$\mathbf{I}^s$:	stack of images for tissue sample s
I_n^s:	n^{th} image in stack $\mathbf{I}^s$
$\hat{I}_n^s$:	sub-image of image I_n^s
n, m:	image indices
s:	sample index
d:	number of images in one stack. Equals the number of used antibodies
$\mathbf{p}, \mathbf{p}'$:	pixels
$f(\mathbf{p})$:	gray value at pixel $\mathbf{p}$
θ_n^s:	gray value threshold for image I_n^s
S:	number of image stacks
N:	number of training samples
N_+, N_-:	number of positive/negative samples
Γ_n^s:	SVM training set for image I_n^s
$\mathbf{x}$:	feature vector
i:	index for training samples
y:	class label
H:	optimal hyperplane
H_1, H_2:	hyperplanes parallel to H
$\mathbf{w}, b$:	SVM parameters
γ:	optimal margin
$h(\mathbf{x})$:	classification function
ξ_i:	slack variable
C:	regularization parameter
L_P:	primal Lagrangian
L_D:	dual Lagrangian
α_i, r_i:	Lagrange multipliers
Φ:	non linear transformation function
$K(\cdot, \cdot)$:	kernel function
$\mathbb{F}$:	feature space obtained by kernel function
p_i:	target probabilities for output calibration
$c(\cdot)$:	calibrated output
ER_n^s:	expert reference for image I_n^s
$\overline{\mathsf{ER}}_n^s$:	master expert reference list for image I_n^s
t:	threshold for synapse detection
$\tilde{t}$:	threshold tuned for one image I_n^s
$t^{(c)}$:	constant threshold applied to all images
$\Lambda_n^s(t)$:	set of synapses detected by i3S in image I_n^s with threshold t

$SE(t)$:	sensitivity for threshold t
$PPV(t)$:	positive predictive value for threshold t
SE_μ:	average sensitivity for a set of i3S OR a set of images
PPV_μ:	average positive predictive value for a set of i3S OR a set of images
$\mathcal{S}_{+,+}$:	set of synapses detected by i3S AND the expert
$\mathcal{S}_{-,+}$:	set of synapses not detected by i3S BUT by the expert
$\mathcal{S}_{+,-}$:	set of synapses detected by i3S AND NOT the expert
Ω:	neighborhood
M:	neighborhood size
M_b:	neighborhood size for bilateral filtering
$g(\cdot,\cdot)$:	smoothing term
$s(\cdot,\cdot)$:	smoothing term
$o(\cdot,\cdot)$:	normalizing term
σ_r, σ_d:	bilateral filter parameters
$i3S_n$:	i3S trained with data from one image I_n^s
$f(\mathbf{p})$:	intensity at pixel $\mathbf{p}$
$\hat{f}(\mathbf{p})$:	filtered intensity at pixel $\mathbf{p}$
$\mathcal{X}$:	feature vector data set
$\mathcal{W}$:	prototype data set
L:	total number of items in $\mathcal{X}$
$\mathbf{w}$:	prototype
k, k', k^*:	cluster indices
K:	total number of clusters
a:	iteration counter for clustering
$\epsilon(\cdot)$:	exponential learning rate
ϵ_i, ϵ_f:	initial and final learning rate
λ:	neighborhood width for clustering. Intensity for point pattern analysis
λ_i, λ_f:	Initial and final neighborhood width
$r_k(\cdot)$:	rank of prototype $\mathbf{w}_k$
$\mathcal{L}$:	lattice structure
$\mathbb{H}^2$:	hyperbolic space
$EQ(\cdot)$:	quantization error
$\boldsymbol{\mu}_k$:	cluster center
$BSS(K)$:	between cluster sum of squares
$WSS(K)$:	within cluster sum of squares
$DB(K)$:	Davis Bouldin index
$\mathcal{C}_k$:	set of items belonging to cluster k
$\mathbf{c}$:	color vector
$\tilde{\mathbf{w}}$:	projected prototype vector
$\mathbf{v}^s$:	vector holding the relative abundances of prototypes for image stack $\mathbf{I}^s$
$\mathbf{r}^s$:	vector holding the ranks of each prototype for image stack $\mathbf{I}^s$
$K(r)$:	Ripley's K function
$L(r)$:	Alternative representation of Ripley's K

$O(r)$:	O-ring statistics
$g_{k,k'}(r)$:	pair correlation function
$C(r)$:	Circle of radius r
$\mathrm{Area}(C(r))$:	Area of circle $C(r)$ within study region
$\mathrm{Points}(C(r))$:	Points within circle $C(r)$

A.3 Supplementary Figures

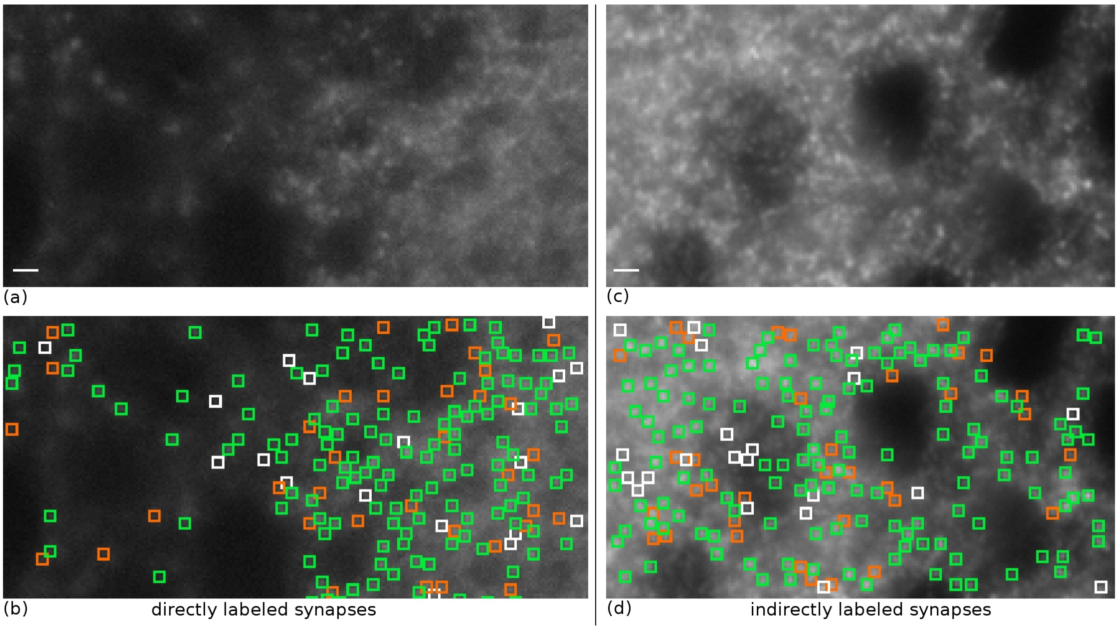


Figure A.1: This figure shows the same image regions and results as figure 4.13 but the original image data is shown, i.e. the i3S preprocessing has not been applied. Scale bar of 1.8μm refers to all images.

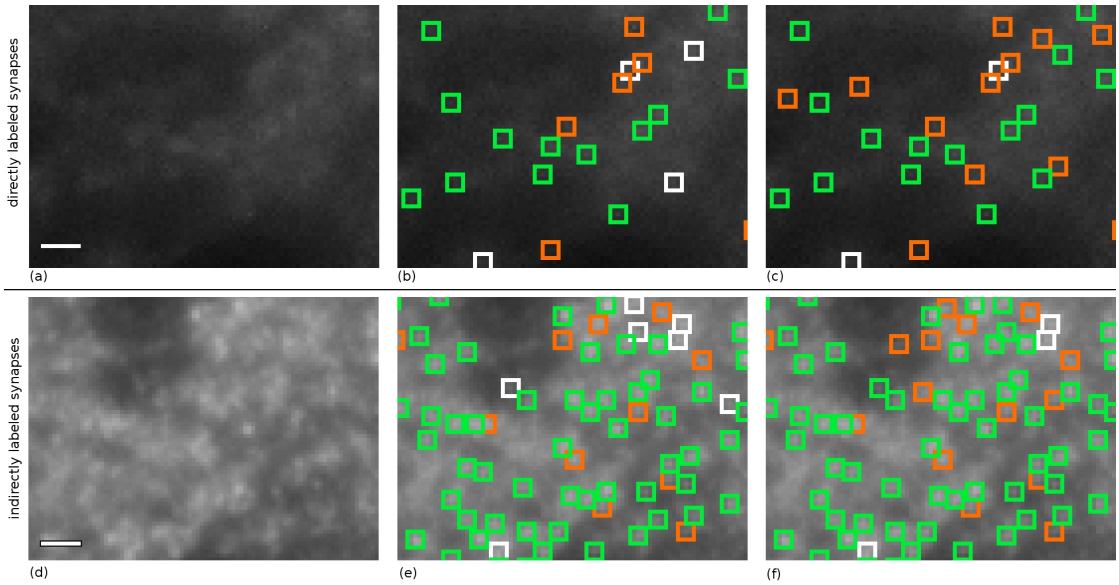
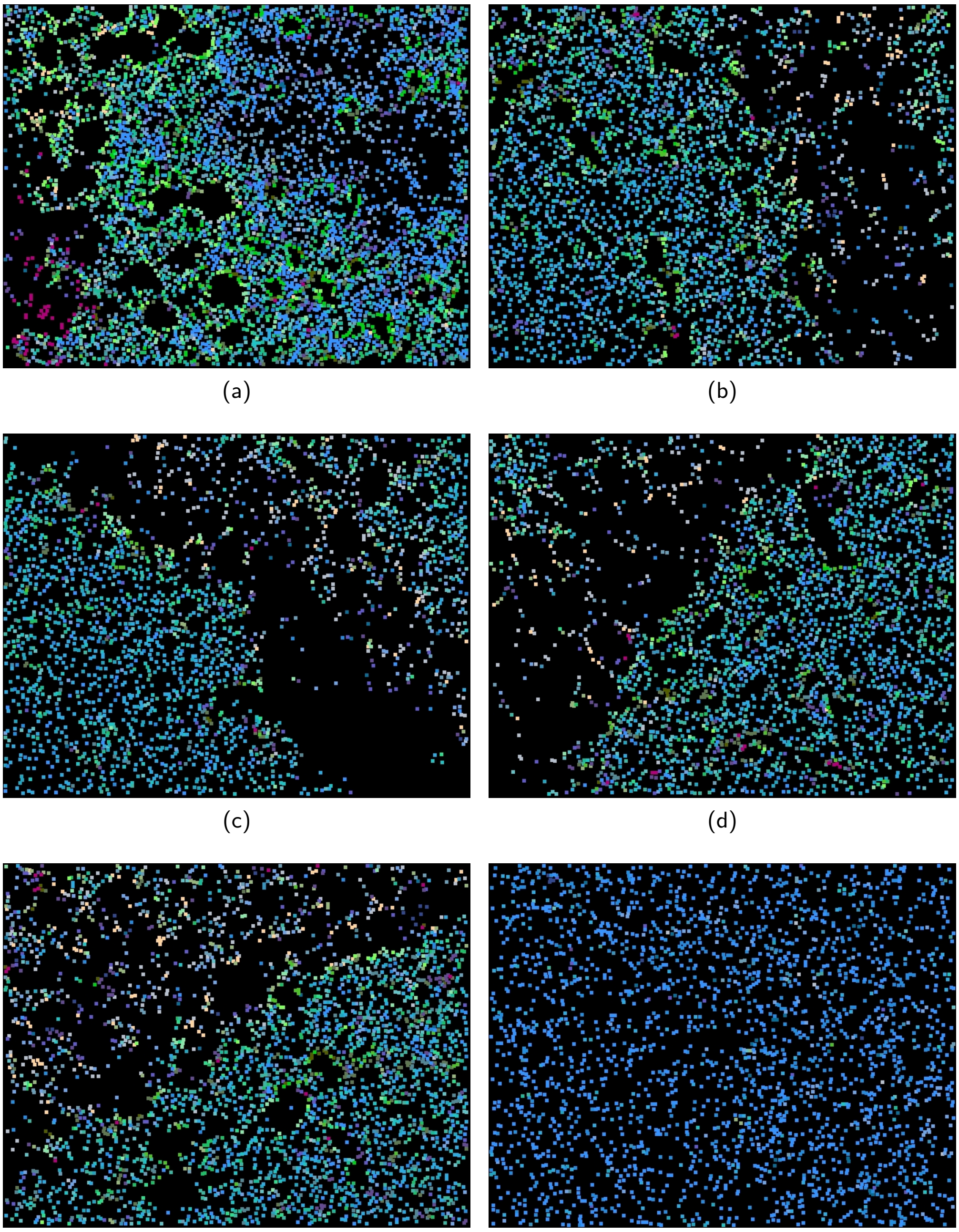


Figure A.2: This figure shows the same image regions and results as figure 4.15 but the original image data is shown, i.e. the i3S preprocessing has not been applied.Scale bar of 1.8μm refers to all images.



(a)

(b)

(c)

(d)

(e)

(f)

Figure A.3: Cluster maps generated for NG cluster results through PCA projection into the 3D RGB space.

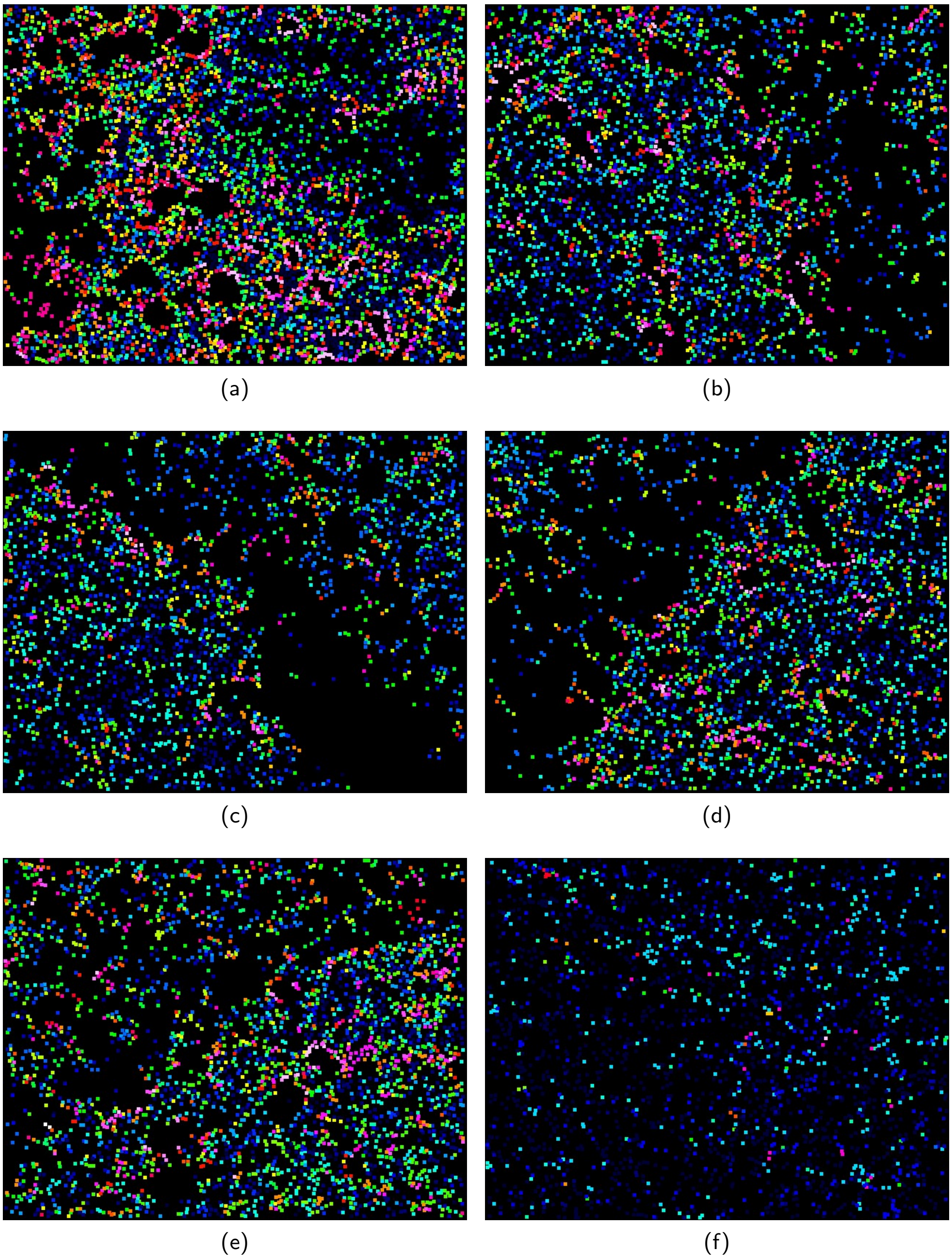


Figure A.4: Cluster maps generated for K-means cluster results through PCA projection into the 3D RGB space.

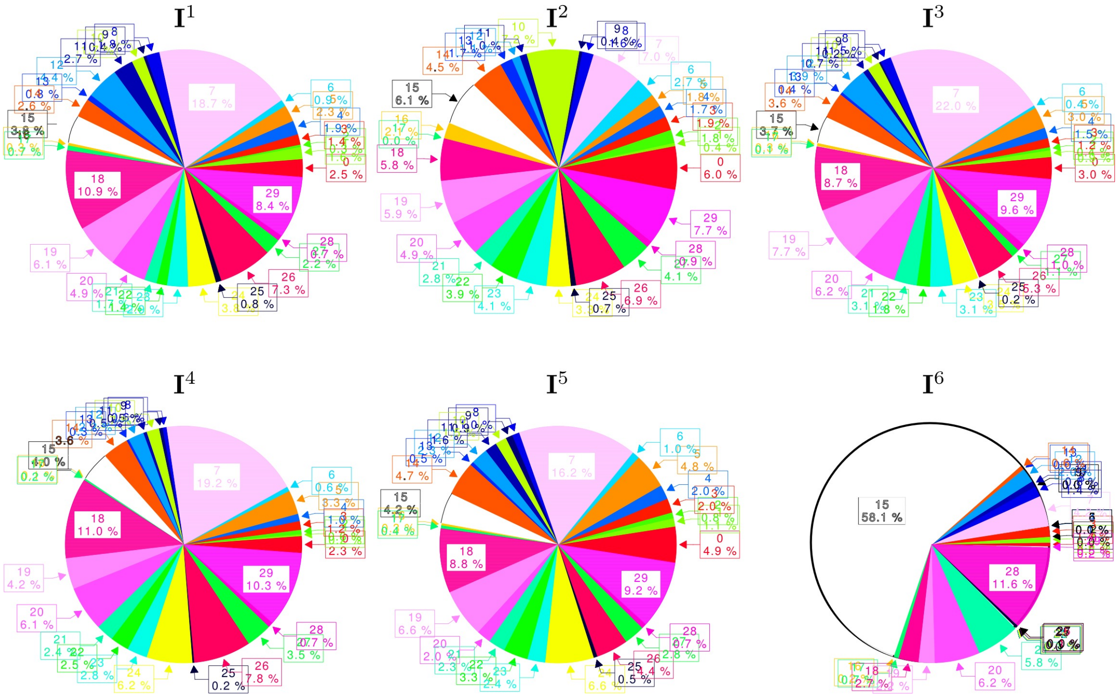


Figure A.5: Pie chart for the cluster distribution of the NG clustering with all protein channels.

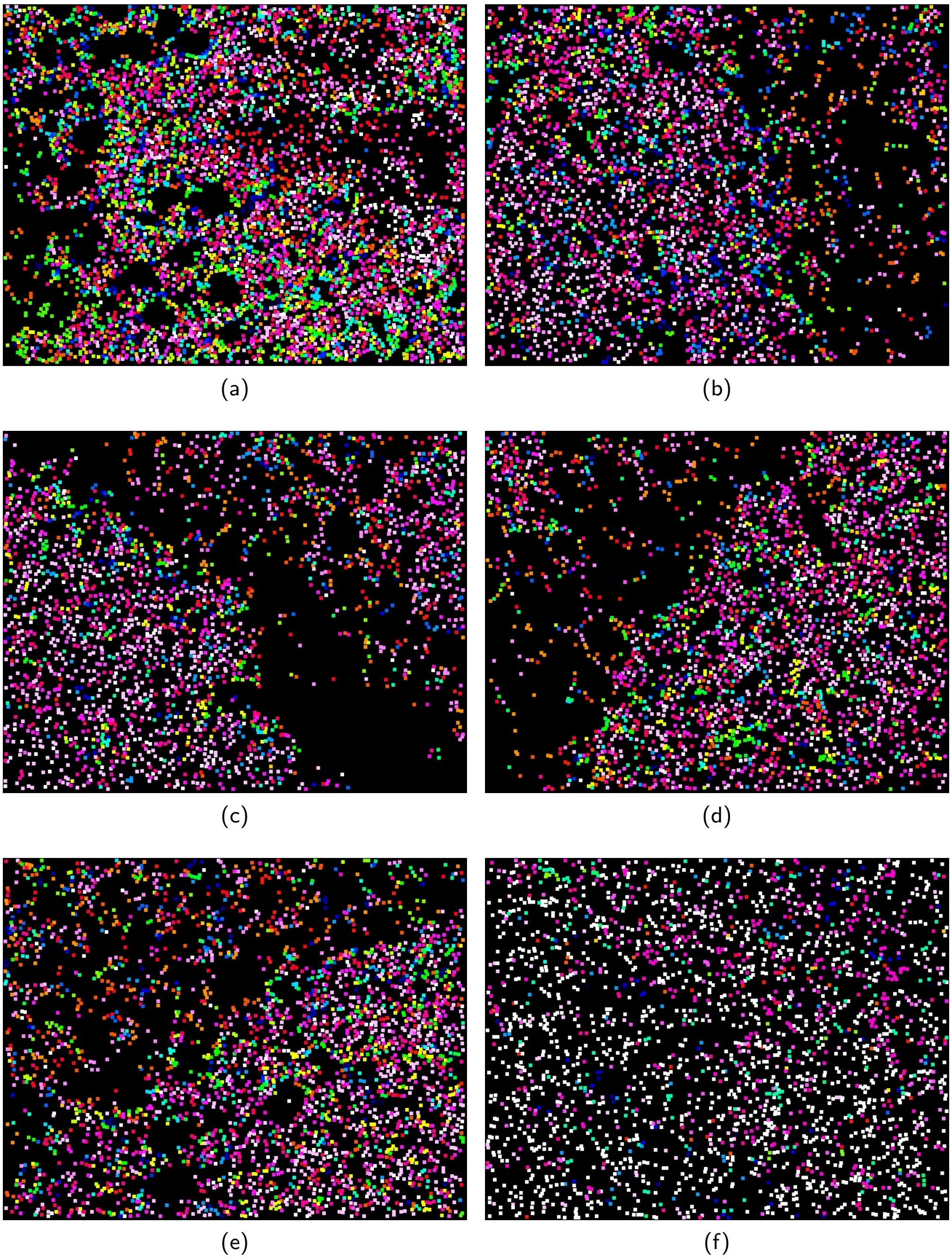


Figure A.6: Cluster maps generated for NG cluster results through PCA projection into the 3D RGB space. All 21 proteins were used in the data analysis.

A.4 Supplementary Tables

	PC1 Intel Penitum 4, 2.53 GHz, 2Gb Ram		PC1 Intel Core 2 Quad, 2.83 GHz, 4Gb Ram	
	Run 1	Run 2	Run 3	Run 4
Average consensus (AC)	100%	100%	100%	100%

Table A.2: To thoroughly test the reproducibility of one trained i3S in a maximum conservative way, it was checked, if the i3S can be slightly influenced by numerical operations at different time or on different PC hardware. To test the reproducibility in this regard, a i3S$_{s'}$ was trained using the reference set ER$_{s'}$ and applied to one other image and recorded the detected synapse positions in list L_1. Afterwards, the same i3S$_{s'}$ was applied again to the same input image and the result were recorded in list L_2. The consensus between both lists was defined as the percentage of those synapses listed in L_1 that are also listed in L_2 with identical coordinates. This was done for each i3S$_{s'}$ and I^s (with $s \neq s'$) so for each i3S$_{s'}$ the average consensus (AC$_s$) was computed for each s over s'. Afterwards, the average consensus (AC) was computed over all (AC$_{s'}$). The whole experiment was repeated in four runs. In the first two runs, the same PC (PC 1) was used for training the i3S$_{s'}$ and computing both lists L_1 and L_2. In run three and four, the i3S$_{s'}$ was trained on PC1 and L_1 was computed on PC1 but the second result L_2 was computed on another PC (PC 2, details given in the table).

	$\mathbf{I}^1$	$\mathbf{I}^2$	$\mathbf{I}^3$	$\mathbf{I}^4$	$\mathbf{I}^5$	$\mathbf{I}^6$
$\mathbf{I}^1$	1	0.087	0.054	0.084	0.089	0.6
$\mathbf{I}^2$	0.087	1	0.92	0.88	0.81	0.023
$\mathbf{I}^3$	0.054	0.92	1	0.89	0.85	0.011
$\mathbf{I}^4$	0.084	0.88	0.89	1	0.89	0.026
$\mathbf{I}^5$	0.089	0.81	0.85	0.89	1	0.012
$\mathbf{I}^6$	0.6	0.023	0.011	0.026	0.012	1

Table A.3: Pearson correlation r^2 between the six image sets based on the clustering result obtained for NG clustering. (see section 6.6.3).

	$\mathbf{I}^1$	$\mathbf{I}^2$	$\mathbf{I}^3$	$\mathbf{I}^4$	$\mathbf{I}^5$	$\mathbf{I}^6$
$\mathbf{I}^1$	1	0.6	0.52	0.53	0.47	0.2
$\mathbf{I}^2$	0.6	1	0.91	0.84	0.81	0.49
$\mathbf{I}^3$	0.52	0.91	1	0.88	0.9	0.5
$\mathbf{I}^4$	0.53	0.84	0.88	1	0.92	0.31
$\mathbf{I}^5$	0.47	0.81	0.9	0.92	1	0.37
$\mathbf{I}^6$	0.2	0.49	0.5	0.31	0.37	1

Table A.4: Spearman's rank correlation between the six image sets based on the clustering result obtained for NG clustering. (see section 6.6.3).

Bibliography

C. N. G. Anderson and S. G. N. Grant. High throughput protein expression screening in the nervous system - needs and limitations. *The Journal of Physiology*, 575:367–372, 2006.

C. Angeletti, N. R. Harvey, V. Khomitch, A. H. Fischer, R. M. Levenson, and D. L. Rimm. Detection of malignancy in cytology specimens using spectral-spatial analysis. *Laboratory Investigations*, 85:1555–1564, 2005.

I. Avril, B. Blondeau, B. Duchene, P. Czernichow, and B. Breant. Decreased beta-cell proliferation impairs the adaptation to pregnancy in rats malnourished during perinatal life. *Journal of Endocrinology*, 174:215–223, 2002.

S. Bandyopadhyay, U. Maulik, and A. Mukhopadhyay. Multiobjective genetic clustering for pixel classification in remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 45:1506–1511, 2007.

I. N. Bankman, editor. *Handbook of Medical Imaging Processing and Analysis*. Academic Press, 2000.

L. Barbe, E. Lundberg, P. Oksvold, A. Stenius, E. Lewin, E. Björling, A. Asplund, F. Ponten, H. Brismar, M. Uhlen, and H. Andersson-Svahn. Towards a confocal subcellular atlas of the human proteome. *Molecular & Cellular Proteomics*, 7:499–508, 2008.

P. Barber, B. Vojnovic, J. Kelly, C. Mayes, P. Boulton, M. Woodcock, and M. Joiner. Automated counting of mammalian cell colonies. *Physics in Medicine and Biology*, 46: 63–76, 2001.

R. A. Becker and W. S. Cleveland. Brushing scatterplots. *Technometrics*, 29:127–142, 1987.

J. A. Beliën, H. A. van Ginkel, P. Tekola, L. S. Ploeger, N. M. Poulin, J. P. Baak, and P. J. van Diest. Confocal dna cytometry: A contour-based segmentation algorithm for automated three-dimensional image segmentation. *Cytometry Part A*, 1:12–21, 2002.

C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., 2004.

M. Bode and A. Krusche. Toponome imaging system (TIS): imaging the proteome with functional resolution. *Nature Methods*, 4:541–47, 2007. Application note.

M. Bode, M. Irmler, M. Friedenberger, C. May, K. Jung, C. Stephan, H. Meyer, C. Lach, R. Hillert, A. Krusche, J. Beckers, K. Marcus, and W. Schubert. Interlocking transcriptomics, proteomics and toponomics technologies for brain tissue analysis in murine hippocampus. *Proteomics*, 8:1170–1178, 2008.

S. Bolte and F. P. Cordelières. A guided tour into subcellular colocalization analysis in light microscopy. *Journal of Microscopy*, 224:213–232, 2006.

L. Boucheron, Z. Bi, N. Harvey, B. S. Manjunath, and D. Rimm. Utility of multispectral imaging for nuclear classification of routine clinical histopathology imagery. *BMC Cell Biology*, 8, 2007.

L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–67, 1998.

M. E. Calhoun, M. Jucker, L. J. Martin, G. Thinakaran, D. L. Price, and P. R. Mouton. Comparative evaluation of synaptophysin-based methods for quantification of synapses. *Journal of Neurocytology*, 25:821–828, 1996.

T. Calinski and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics - Theory and Methods*, 3:1–27, 1974.

F. Camastra and A. Vinciarelli. *Machine learning for audio, image and video analysis*. Advanced Information and Knowledge Processing. Springer Verlag, theory and applications edition, 2008.

D. A. Cano, I. C. Rulifson, P. W. Heiser, L. B. Swigart, S. Pelengaris, M. German, G. I. Evan, J. A. Bluestone, and M. Hebrok. Regulated beta-cell regeneration in the adult mouse pancreas. *Diabetes*, 57:958–966, 2008.

A. E. Carpenter. Software opens the door to quantitative imaging. *Nature Methods*, 4: 120–121, 2007.

J. M. Chambers, W. S. Cleveleand, B. Kleiner, and P. A. Tukey. *Graphical Methods for Data Analysis*. Duxbury Press, 1983.

O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46:131–159, 2002.

S.-C. Chen, T. Zhao, G. J. Gordon, and R. F. Murphy. Automated image analysis of protein localization in budding yeast. In *ISMB/ECCB (Supplement of Bioinformatics)*, pages 66–71, 2007.

H. Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68:361–368, 1973.

C. Conrad, H. Erfle, P. Warnat, N. Daigle, T. Lörch, J. Ellenberg, R. Pepperkok, and R. Eils. Automatic identification of subcellular phenotypes on human cell arrays. *Genome Research*, 14:1130–1136, 2004.

N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kerne-based learning methods*. Camebridge University Press, 2003.

G. E. Crooks, G. Hon, J.-M. Chandonia, and S. E. Brenner. Weblogo: A sequence logo generator. *Genome Research*, 14:1188–1190, 2004.

P. Dalgaard. *Introductory statistics with R*. Springer Berlin, 2008.

D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1:224–227, 1979.

P. Diggle. *Statistical analysis of spatial point patterns*. Academic Press, 1983.

P. M. Dixon. *Ripley's K function*, pages 1796–1803. John Wiley & Sons, 2002.

K. Duan, S. S. Keerthi, and A. N. Poo. Evaluation of simple performance measures for tuning SVM hyperparameters. *Neurocomputing*, 51:41–59, 2003.

J. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3:32–57, 1973.

J. Egan. *Signal Detection Theory and ROC Analysis*. Series in Cognition and Perception. Academic Press, 1975.

I. El-Naqa, Y. Yang, M. Wernick, N. Galatsanos, and R. Nishikawa. A support vector machine approach for detection of microcalcifications. *IEEE Transactions on Medical Imaging*, 21: 1552–63, 2002.

W. Estelberger and G. Reibnegger. The rank correlation coefficient: an additional aid in the interpretation of laboratory data. *Clinica Chimica Acta*, 239:203–207, 1995.

T. Fawcett. Roc graphs: Notes and practical considerations for researchers, 2004.

F. Fleischer, M. Beil, M. Kazda, and V. Schmidt. *Case Studies in Spatial Point Process Modeling*, chapter Analysis of Spatial Point Patterns in Microscopic and Macroscopic Biological Image Data, pages 235–260. Springer New York, 2006.

E. Forgy. Cluster analysis of multivariate data: Efficiency versus interpretability of classification. *Biometrics*, 21:768–769, 1965.

M. Friedenberger, M. Bode, A. Krusche, and W. Schubert. Fluorescence detection of protein clusters in individual cells and tissue sections by using toponome imaging system: sample preparation and measuring procedures. *Nature Protocols*, 2:2285–94, 2007.

E. Galperin, V. V. Verkhusha, and A. Sorkin. Three-chromophore FRET microscopy to analyze multiprotein interactions in living cells. *Nature Methods*, 1:209–217, 2004.

Y. Geinsman, H. J. Gunderson, E. van der Zee, and M. J. West. Unbiased stereological estimation of the total number of synapses in a brain region. *Journal of Neurocytology*, 25:805–819, 1996.

T. Glasmachers and C. Igel. Uncertainty handling in model selection for support vector machines. In *Proceedings of the 10th international conference on Parallel Problem Solving from Nature*, pages 185–194, Berlin, Heidelberg, 2008. Springer-Verlag.

E. Glory and R. F. Murphy. Automated subcellular location determination and high-throughput microscopy. *Developmental Cell*, 12:7–16, 2007.

E. Glory, J. Newberg, and R. Murphy. Automated comparison of protein subcellular location patterns between images of normal and cancerous tissues. *Proceedings of the 2008 IEEE International Symposium on Biomedical Imaging (ISBI 2008)*, pages 304–307, 2008.

C. Gold, A. Holub, and P. Sollich. Bayesian approach to feature selection and parameter tuning for support vector machine classifiers. *Neural Networks*, 18:693–701, 2005.

R. Gonzalez and R. Woods. *Digital Image Processing (2nd Edition)*. Prentice Hall, New Jersey, 2002.

A. Gordon, A. Colman-Lerner, T. Chin, K. Benjamin, R. Yu, and R. Brendt. Single-cell quantification of molecules and rates using open-source microscope-based cytometry. *Nature Methods*, 4:175–81, 2007.

F. Goreaud and R. Pélissier. Avoiding misinterpretation of biotic interactions with the intertype k12-function: population independence vs. random labelling hypothesis. *Journal of Vegetation Science*, 14:681–692, 2003.

A. Gyenesei, R. Schlapbach, E. Stolte, and U. Wagner. Frequent pattern discovery without binarization: Mining attribute profiles. *Lecture Notes in Computer Science, Knowledge Discovery in Databases*, 4213:528–535, 2006.

M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17:107–145, 2001.

A. T. Harris. Spectral mapping tools from the earth sciences applied to spectral microscopy data. *Cytometry Part A*, 69A:872–879, 2006.

T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

G. Heidemann. Efficient vector quantization using the WTA-rule with activity equalization. *Neural Processing Letters*, 13:17–30, 2001.

T. Hermann, T. W. Nattkemper, W. Schubert, and H. J. Ritter. Sonification of multi-channel image data. In *Proc. of the Mathematical and Engineering Techniques in Medical and Biological Sciences (METMBS 2000)*, pages 745–750. CSREA Press, 2000.

J. Herold, S. Abouna, L. Zhou, S. Pelengaris, D. B. Epstein, M. Khan, and T. W. Nattkemper. A way towards analyzing high-content bioimage data by means of semantic annotation and visual data mining. In *SPIE Medical Imaging*, 2009.

J. Herold, W. Schubert, and T. W. Nattkemper. Automated detection and quantification of fluorescently labeled synapses in murine brain tissue sections for high throughput applications. *Journal of Biotechnology*, 2010.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

M. Hu, X. Ping, and Y. Ding. Automated cell nucleus segementation using improved snake. In *2004 International Conference on Image Processing*, volume 4, pages 2737–2740, 2004.

C. Huang, F. Snider, and J. C. Cross. Prolactin receptor is required for normal glucose homeostasis and modulation of β-cell mass during pregnancy. *Endocrinology*, 150:1618–1626, 2009.

K. Huang and R. Murphy. From quantitative microscopy to automated image understanding. *Journal of Biomedical Opttics*, 9:893–912, 2004.

W.-K. Huh, J. V. Falvo, L. C. Gerke, A. S. Carroll, R. W. Howson, J. S. Weissman, and E. K. O'Shea. Global analysis of protein localization in budding yeast. *Nature*, 425:686–691, 2003.

A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *VIS '90: Proceedings of the 1st conference on Visualization '90*, pages 361–378, Los Alamitos, CA, USA, 1990. IEEE Computer Society Press.

R. Jagoe, J. Steel, V. Vucicevic, N. Alexander, S. V. Noorden, R. Wootton, and J. Polak. Observer variation in quantification of immunocytochemistry by image analysis. *The Histochemical Journal*, 23:541–547, 1991.

K. Karhunen. Über Lineare Methoden in der Wahrscheinlichkeitsrechnungna. *Annales Academiae Scientiarum Fennicae*, 1:102–344, 1946.

D. A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8:1–8, 2002.

T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.

T. Kohonen. *Self-Organizing Maps*. Springer, 3 edition, 2001.

E. A. Krupinski. The importance of perception research in medical imaging. *Radiation Medicine*, 18:329–334, 2000.

W. J. Krzanowski, editor. *Principles of multivariate analysis: a user's perspective*. Oxford University Press, Inc., New York, NY, USA, 2000.

K. Kulasa and R. Henry. Pharmacotherapy of hyperglycemia. *Expert Opinion on Pharmacotherapy*, 10:2415–32, 2009.

G. P. Kuman, A. Sarkar, and N. C. Debnath. A new algorithm for frequent itemset generation in non-binary search space. In *Sixth International Conference on Information Technology:New Generations*, 2009.

J. LeBlanc, M. Ward, and N. Wittels. Exploring n-dimensional databases. In *Proceedings Visualization*, pages 230 – 237, 1990.

H. Levkovitz. Color icons: Merging color and texture perception for integrated visualization of multiple parameters. In *Proceedings Visualization*, pages 164–170, 1991.

H. Levkovitz and G. T. Herman. Color scales for image data. *IEEE Computer Graphics & Applications*, 12:72–80, 1992.

S. Lloyd. Least squares quantization in PCM. *IEEE Transaction on Information Theory*, 28: 129–137, 1982.

X. Long, W. Cleveland, and Y. Yao. Multiclass cell detection in bright field images of cell mixtures with ECOC probability estimation. *Image and Vision Computing*, 26:578–591, 2008.

J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proeedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

W. Magnus. *Noneuclidean Tesselations and Their Groups*. Academic Press, 1974.

N. Malpica, C. O. de Solorzano, J. J. Vaquero, A. Santos, I. Vallcorba, J. M. Garcia-Sagredo, and F. del Pozo. Applying watershed algorithms to the segmentation of clustered nuclei. *Cytometry*, 28:289–297, 1997.

E. Manders, J. Stap, G. Brakenhoff, R. van Driel, and J. Aten. Dynamics of three-dimensional replication patterns during the S-phase, analysed by double labelling of DNA and confocal microscopy. *Journal of Cell Science*, 103:857–862, 1992.

D. F. Marrone, J. C. LeBoutillier, and T. L. Petit. Complementary techniques for unbiased stereology of brain ultrastructure. *Journal of Electron Microscopy*, 52:425–428, 2003.

T. Martinetz and K. Schulten. A "neural-gas" network learns topologies. *Artificial Neural Networks*, I:397–402, 1991.

K. Matković, D. Gracanin, Z. Konyha, and H. Hauser. Color lines view: An approach to visualization of families of function graphs. In *Proceedings of 11th International Conference Information Visualization*, pages 59–64, 2007.

T. Mattfeldt, S. Eckel, F. Fleischer, and V. Schmidt. Statistical analysis of labelling patterns of mammary carcinoma cell nuclei on histological sections. *Journal of Microscopy*, 235: 106–118, 2009.

U. Maulik and S. Bandyopadhyay. Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24: 1650–1654, 2002.

L. A. McDonnell and R. M. A. Heeren. Imaging mass spectrometry. *Mass Spectrometry Reviews*, 26:606–643, 2007.

S. Megason and S. Fraser. Imaging in systems biology. *Cell*, 130:784–795, 2007.

F. Melgani and L. Bruzzone. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, 42: 1778– 1790, 2004.

C. E. Metz. Roc analysis in medical imaging: a tutorial review of the literature. *Radiological Physics and Technology*, 1:2–12, 2008.

K. D. Micheva and S. J. Smith. Array tomography: A new tool for imaging the molecular architecture and ultrastructure of neural circuits. *Neuron*, 55:25–36, 2007.

P. R. Mouton, D. L. Price, and L. C. Walker. Empirical assessment of synapse numbers in primate neocortex. *Journal of Neuroscience Methods*, 75:119–126, 1997.

R. F. Murphy. Putting proteins on the map. *Nature Biotechnology*, 24:1223–1224, 2006.

R. F. Murphy. Locations everyone: Lights, camera, action! *Journal of Proteome Research*, 8:1, 2009.

T. Nattkemper. Automatic segmentation of digital micrographs: A survey. In *Proceedings of 11th World Congress on Medical Informatics (MEDINFO)*, San Francisco, USA, 2004. AMIA/IMIA.

T. Nattkemper, H. Wersing, H. Ritter, and W. Schubert. A neural network architecture for automatic segmentation of fluorescence micrographs. *Neurocomputing*, 48:357–367, 2002.

T. Nattkemper, T. Twellmann, W. Schubert, and H. Ritter. Human vs. machine: Evaluation of fluorescence micrographs. *Computers in Biology and Medicine*, 33:31–43, 2003a.

T. W. Nattkemper. *A neural network-based system for high-throughput fluorescence micrograph evaluation*. PhD thesis, University of Bielefeld, Faculty of Technology, 2001.

T. W. Nattkemper, H. J. Ritter, and W. Schubert. Extracting patterns of lymphocyte fluorescence from digital microscope images. In *Intelligent Data Analysis in Medicine and Pharmacology (IDAMAP 99), Workshop Notes*, pages 79–88, 1999.

T. W. Nattkemper, H. Ritter, and W. Schubert. A neural classifier enabling high-throughput topological analysis of lymphocytes in tissue sections. *IEEE Transactions on Information Technology in Biomedicine*, 5:138–149, 2001.

T. W. Nattkemper, T. Hermann, W. Schubert, and H. Ritter. Look & listen: Sonification and visualization of multiparameter micrographs. In *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE*, pages 1311–1314, 2003b.

A. Nedzved, S. Ablameyko, and I. Pitas. Morphological segmentation of histology cell images. *ICPR*, 1:500–503, 2000.

J. Ontrup. *Semantic visualization with hyperbolic self-organizing maps - a novel approach for exploring structure in large data sets*. PhD thesis, Bielefeld University, 2008.

J. Ontrup and H. Ritter. Large-scale data exploration with the hierarchically growing hyperbolic SOM. *Neural Networks*, 19:751–761, 2006.

J. Ontrup, H. Ritter, S. W.Scholz, and R. Wagner. Detecting, assessing, and monitoring relevant topics in virtual information environments. *IEEE Transactions On Knowledge And Data Engineering*, 21:415–427, 2009.

N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on System Man and Cybernetic*, 9:62–66, 1979.

S. Park, B. D. Gallas, A. Badano, N. A. Petrick, and K. J. Myers. Efficiency of the human observer for detecting a gaussian signal at a known location in non-gaussian distributed lumpy backgrounds. *Journal of the Optical Society of America A*, 24:911–921, 2007.

S. Park, A. Badano, B. D. Gallas, and K. J. Myers. Incorporating human contrast sensitivity in model observers for detection tasks. *IEEE Transactions on Medical Imaging*, 28:339–347, 2009.

K. Pearson. On lines and planes of closest fit to a system of point in space. *Philosophical Magazin*, 2:559–572, 1901.

P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans Pattern Anal Mach Intell*, 12:629–639, 1990.

R. M. Pickett and G. G. Grinstein. Iconographic displays for visualizing multidimensional data. *Proceedings of the 1988 IEEE Conference on Systems, Man and Cybernetics*, 1: 514–519, 1988.

J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In A. Smola, P. Bartlett, B. Schoelkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74, 1999.

W. Pschyrembel. *Pschyrembel Klinisches Wörterbuch*. Gruyter, 2002.

B. Ripley. The second-order analysis of stationary point processes. *Journal of Applied Probability*, 13:255–266, 1976.

B. Ripley. *Spatial statistics*. Wiley, New York, 1981.

H. Ritter. Self-organizing maps on non-euclidean spaces. In *Kohonen Maps*, pages 97–108. Elsevier, 1999.

H. J. Ritter, T. M. Martinetz, and K. J. Schulten. *Neuronale Netze*. Addison-Wesley, München, Germany, 1991.

Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *Proceedings of the 1998 IEEE International Conference on Computer Vision*, 1998.

A. Saalbach. *Exploratory analysis of multivariate image data*. PhD thesis, Bielefeld University, Bielefeld, 2006.

A. Saalbach, J. Ontrup, H. Ritter, and T. W. Nattkemper. Image fusion based on topographic mappings using the hyperbolic space. *Information Visualization*, 4:266–275, 2005.

J. W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, 18:401–409, 1969.

T. D. Schneider and R. Stephens. Sequence logos: a new way to display consensus sequences. *Nucleic Acids Research*, 18:6097–6100, 1990.

A. J. Schölkopf, Bernhard; Smola. *Learning with Kernels*. Massachusetts Institute of Technology, 2002.

W. Schubert. *Advances in analytical cellular pathology*, chapter Multiple antigen-mapping microscopy of human tissue, pages 97–98. Excerpta Medica. Elsevier, 1990.

W. Schubert. *Topological Proteomics, Toponomics, MELK-Technology*, volume 83 of *Adv. in Biochem. Eng. Biotechnol.* Springer Heidelberg, 2003.

W. Schubert. Molecular Pattern Recognition Research Group, University of Magdeburg. personal communication, 2006.

W. Schubert. Breaking the biological code. *Cytometry Part A*, 71A:771–772, 2007.

W. Schubert, R. Prior, A. Weidemann, H. Dircksen, G. Multhaup, C. L. Masters, and K. Beyreuther. Localization of alzheimer beta A4 amyloid precursor protein at central and peripheral synaptic sites. *Brain Research*, 563:184–194, 1991.

W. Schubert, B. Bonnekoh, A. Pommer, L. Philipsen, R. Böckelmann, Y. Malykh, H. Gollnick, M. Friedenberger, M. Bode, and A. Dress. Analyzing proteome topology and function by automated multidimensional fluorescence microscopy. *Nature Biotechnology*, 24:1270–1278, 2006.

W. Schubert, M. Bode, R. Hillert, A. Krusche, and M. Friedenberger. Toponomics and neurotoponomics: a new way to medical systems biology. *Expert Review of Proteomics*, 5:361–369, 2008.

W. Schubert, A. Gieseler, A. Krusche, and R. Hillert. Toponome mapping in prostate cancer: detection of 2000 cell surface protein clusters in a single tissue section and cell type specific annotation by using a three symbol code. *Journal of Proteome Research*, 8:2696–2707, 2009.

P. Serocka. Visualization of high-dimensional biomedical image data. In *PCM*, pages 475–482, 2007.

B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *IEEE Symposium on Visual Languages*, pages 336–343, 1996.

B. Shneiderman. Science 2.0. *Science*, 391:1349–1350, 2008.

M. A. Silver and M. P. Stryker. A method for measuring colocalization of presynaptic markers with anatomically labeled axons using double label immunofluorescence and confocal microscopy. *Journal of Neuroscience Methods*, 94:205–215, 2000.

J. Simpson, R. Wellenreuther, A. Poustka, R. Pepperkok, and S. Wiemann. Systematic subcellular localization of novel proteins identified by large-scale cDNA sequencing. *EMBO reports*, 1:287–292, 2000.

P. J. Sjöström, B. R. Frydel, and L. U. Wahlberg. Artificial neural network-aided image analysis system for cell counting. *Cytometry*, 36:18–26, 1999.

A. J. Smola and B. Schölkopf. A tutorial on support vector regression. Technical report, Statistics and Computing, 2003.

C. O. D. Solorzano, R. Malladi, S. A. Lelièvre, and S. J. Lockett. Segmentation of nuclei and cells using membrane related protein markers. *Journal of Microscopy*, 201:404–415, 2000.

S. N. Somani A. K. Toponomics: visualizing cellular protein networks in health and disease-' a single picture is worth more than a thousand words!'. *Journal of Cutaneous Pathology*, 35:791–793, 2008.

M. Sonka and M. J. Fitzpatrick, editors. *Handbook of Medical Imaging Volume 2. Medical Image Processing and Analysis*. SPIE Press, Bellingham, Washington, 2000.

R. Spence. *Information Visualization: Design for Interactions: (2nd edition)*. Prentice Hall, 2007.

L. Squire, D. Berg, F. Bloom, S. D. Lac, A. Ghosh, and N. Spitzer, editors. *Fundamental Neuroscience*. Academic Press, 3. edition, 2008.

V. Starkuviene and R. Pepperkok. The potential of high-content high-throughput microscopy in drug discovery. *British Journal of Pharmacology*, 152:62–71, 2007.

D. Stoyan and A. Penttinen. Recent application of point process methods in forest statistics. *Statist. Science*, 15:61–78, 2000.

D. Stoyan and H. Stoyan. *Fractals, random shapes and point fields. Methods of geometrical statistics*. John Wiley & Sons, 1994.

F. Theis, Z. Kohl, H. Kuhn, H. Stockmeier, and E. Lang. Automated counting of labelled cells in rodent brain section images. In *IASTED International Conference of Biomedical Engineering*, pages 209–212, 2004.

M. E. Tipping. *Advances in Neural Information Processing Systems 12*, chapter The Relevance Vector Machine, pages 652–658. MIT Press, 2000.

C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *ICCV*, pages 839–46, 1998.

D. Tomaževič, B. Likar, and F. Pernuš. Comparative evaluation of retrospective shading correction methods. *Journal of Microscopy*, 208:212–223, 2002.

T. Twellmann, T. Nattkemper, and H. Ritter. Cell detection in micrographs of tissue sections using support vector machines. In *Proc. of ICANN: Workshop on Kernel & Subspace Methods for Computer Vision*, pages 79–88, 2001.

M. Uhlén, E. Björling, C. Agaton, C. A. Szigyarto, B. Amini, E. Andersen, A. C. Andersson, P. Angelidou, A. Asplund, C. Asplund, L. Berglund, K. Bergström, H. Brumer, D. Cerjan, M. Ekström, A. Elobeid, C. Eriksson, L. Fagerberg, R. Falk, J. Fall, M. Forsberg, M. G. Björklund, K. Gumbel, A. Halimi, I. Hallin, C. Hamsten, M. Hansson, M. Hedhammar, G. Hercules, C. Kampf, K. Larsson, M. Lindskog, W. Lodewyckx, J. Lund, J. Lundeberg, K. Magnusson, E. Malm, P. Nilsson, J. Odling, P. Oksvold, I. Olsson, E. Oster, J. Ottosson, L. Paavilainen, A. Persson, R. Rimini, J. Rockberg, M. Runeson, A. Sivertsson, A. Sköllermo, J. Steen, M. Stenvall, F. Sterky, S. Strömberg, M. Sundberg, H. Tegel, S. Tourle, E. Wahlund, A. Waldén, J. Wan, H. Wernérus, J. Westberg, K. Wester, U. Wrethagen, L. L. Xu, S. Hober, and F. Pontén. A human protein atlas for normal and cancer tissues based on antibody proteomics. *Molecular & Cellular Proteomics*, 4:1920–1932, 2005.

V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag New York, New York, 1996.

J. Venna and S. Kaski. Neighborhood preservation in nonlinear projection methods: An experimental study. In *Artificial Neural Networks-ICANN*, pages 485–491, 2001.

J. C. Venter, M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, G. G. Sutton, H. O. Smith, M. Yandell, C. A. Evans, and R. A. Holt. The sequence of the human genome. *Science*, 291:1304–1351, 2001.

T. Villmann, E. Mernyi, and B. Hammer. Neural maps in remote sensing image analysis. *Neural Networks*, 16:389–403, 2003.

C. Wählby, J. Lindblad, M. Vondrus, E. Bengtsson, and L. Björkesten. Algorithms for cytoplasm segmentation of fluorescence labelled cells. *Analytical Cellular Pathology*, 23: 101–111, 2002.

C. Wählby, I.-M. Sintorn, F. Erlandsson, C. Borgefors, and E. Bengtssson. Combining intensity, edge and shape information for 2D and 3D segmentation of cell nuclei in tissue sections. *Journal of Microscopy*, 215:67–76, 2004.

C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

D. N. Wei, J. You, K. Friehs, E. Flaschel, and T. W. Nattkemper. An in situ probe for on-line monitoring of cell density and viability on the basis of dark field micfroscopy in conjunction with image processing and supervised machine learning. *Biotechnology and Bioengineering*, 97:1489–500, 2007.

C. Weigle, W. Emigh, G. Liu, R. Taylor, J. Enns, and C. Healey. Oriented texture slivers: A technique for local value estimation of multiple scalar fileds. *Proceedings of Graphics Interface 2000*, pages 163–170, 2000.

T. Wiegand and K. A. Moloney. Rings, circles, and null-models for point pattern analysis in ecology. *OIKOS*, 104:209–229, 2004.

T. Wiegand, K. Moloney, J. Naves, and F. Knauer. Finding the missing link between landscape structure and population dynamics : A spatially explicit perspective. *American Naturialist*, 154:605–627, 1999.

P. C. Wong and Bergeron. 30 years of multidimensional multivariate visualization. *Scientific Visualization - Overviews, Methodologies and Techniques*, pages 3–33, 1997.

X. Wu, V. Kumar, Ross, J. Ghosh, Q. Yang, H. Motoda, G. Mclachlan, A. Ng, B. Liu, P. Yu, Z.-H. Zhou, M. Steinbach, D. Hand, and D. Steinberg. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14:1–37, 2008.

E. Zamir, B. Geiger, and Z. Kam. Quantitative multicolor compositional imaging resolves molecular domains in cell-matrix adhesions. *PLoS ONE*, 3:e1901, 2008.

L. Zhou and D. Epstein. Biomedical Research Institute and Mathematics Institute, University of Warwick. personal communication, 2009.