Inaugural-Dissertation

zur
Erlangung der Doktorwürde
der
Naturwissenschaftlich–Mathematischen Gesamtfakultät
der
Ruprecht–Karls–Universität
Heidelberg

vorgelegt von
M.Eng Xinghua Lou
aus Zhuji, Zhejiang, China

Tag der mündlichen Prüfung: 14. Juli 2011

Biomedical Data Analysis with Prior Knowledge: Modeling and Learning

Gutachter: Prof. Dr. Fred A. Hamprecht
PD Dr. Matthias P. Mayer

Abstract

Modern research in biology and medicine is experiencing a data explosion in quantity and particularly in complexity. Efficient and accurate processing of these datasets demands state-of-the-art computational methods such as probabilistic graphical models, graph-based image analysis and many inference/optimization algorithms. However, the underlying complexity of biomedical experiments rules out direct out-of-the-box applications of these methods and requires novel formulation and enhancement to make them amendable to specific problems. This thesis explores novel approaches for incorporating prior knowledge into the data analysis workflow that leads to quantitative and meaningful interpretation of the datasets and also allows for sufficient user involvement. As discussed in Chapter 1, depending on the complexity of the prior knowledge, these approaches can be categorized as **constrained modeling** and **learning**.

The first part of the thesis focuses on constrained modeling where the prior is normally explicitly represented as additional potential terms in the problem formulation. These terms prevent or discourage the downstream optimization of the formulation from yielding solutions that contradict the prior knowledge. In Chapter 2, we present a robust method for estimating and tracking the deuterium incorporation in the time-resolved hydrogen exchange (HX) mass spectrometry (MS) experiments with priors such as sparsity and sequential ordering. In Chapter 3, we introduce how to extend a classic Markov random field (MRF) model with a shape prior for cell nucleus segmentation.

The second part of the thesis explores learning which addresses problems where the prior varies between different datasets or is too difficult to express explicitly. In this case, the prior is first abstracted as a parametric model and then its optimum parametrization is estimated from a training set using machine learning techniques. In Chapter 4, we extend the popular Rand Index in a cost-sensitive fashion and the problem-specific costs can be learned from manual scorings. This set of approaches becomes more interesting when the input/output becomes structured such as matrices or graphs. In Chapter 5, we present structured learning for cell tracking, a novel approach that learns optimum parameters automatically from a training set and allows for the use of a richer set of features which in turn affords improved tracking performance.

Finally, conclusions and outlook are provided in Chapter 6.

Zusammenfassung

Die aktuelle Forschung in Biologie und Medizin erfährt derzeit einen rasanten Anstieg in der Datenmenge und insbesondere in der Datenkomplexität. Eine effiziente und präzise Verarbeitung solcher Datensätze verlangt nach neuesten rechnergestützen Methoden wie probibilistischen grafischen Modellen, graphbasierter Bildanalyse und modernen Inferenz- bzw. Optimierungsalgorithmen. Die Komplexität, die biomedizinischen Experimenten unterliegt, macht jedoch die direkte Anwendung dieser Methoden unmöglich und erfordert neue Formulierungen und Erweiterungen, die an spezifische Probleme anpassbar sind. Die vorliegende Arbeit erforscht neue Ansätze um Terme, die Vorwissen repräsentieren (sog. Prior Terme), in die Datenanalyse einzubinden. Diese lassen eine quantitative Interpretation der Datensätze zu und berücksichtigen eine explizite Nutzereinbindung. Wie in Kapitel 1 besprochen, können diese Ansätze — abhängig von der Komplexität des Vorwissens — als **Modellierung mit Zwangsbedingungen** oder **Lernen** kategorisiert werden.

Der erste Teil dieser Arbeit konzentriert sich auf die Modellierung mit Zwangsbedingungen, in der das Vorwissen gewöhnlich explizit in Form von zusätzlichen Potenzialtermen in der Problemformulierung repräsentiert wird. Diese Terme erschweren oder hindern die darauffolgende Optimierung daran, Ergebnisse zu liefern, die dem Vorwissen widersprechen. In Kapitel 2 präsentieren wir eine robuste Methode um die Deuterium Einbindung in zeitlich aufgelösten Wasserstoffaustausch-Massenspektrometrie-Experimenten ("hydrogen exchange mass spectrometry"; kurz HXMS) mit Vorwissen über die Daten wie Seltenheit ("Sparsity") und sequentielle Ordnung zu schätzen und nachzuverfolgen. In Kapitel 3 stellen wir vor wie man ein klassisches Markov Random Field (MRF) Modell mit Vorwissen über die äußere Form für Zellkern Segmentierung erweitern kann.

Der zweite Teil der Arbeit erforscht Lernverfahren, die Probleme behandeln, bei denen sich die Prior Terme abhängig vom Datensatz verändern oder sie zu schwierig sind, um sie explizit auszudrücken. In diesem Fall wird das Vorwissen zunächst in einem parametrischen Modell abstrahiert und dann die optimale Parametrisierung aus einem Trainingsdatensatz mit Hilfe von maschinellem Lernen geschätzt. In Kapitel 4 erweitern wir den weitverbreiteten Rand Index in Hinblick auf Kostensensitivität. Die problemspezifischen Kosten können aus manuellen Gewichtungen gelernt werden. Diese Ansätze werden besonders interessant wenn die Ein- und Ausgabe strukturiert ist, z.B. in Matrizen oder Graphen. In Kapitel 5 stellen wir strukturelles Lernen für das Tracking von Zellen vor; ein neuartiger Ansatz, der optimale Parameterwerte automatisch aus einem Trainingsdatensatz lernt und einen erweiterten Merkmalssatz verwendet, der wiederum zu einem verbessertem Tracking führt.

Schlussausführung und Ausblick schließlich sind Inhalt von Kapitel 6.

Acknowledgments

This thesis would not have seen the light of day without the significant amount of guidance, help and support that I received.

I want to especially thank my thesis advisor Prof. Fred Hamprecht for his inspiring guidance throughout my Ph.D. study. It was him who introduced me into the fascinating fields of machine learning, pattern recognition and image processing, and provided me with a stimulating environment for research. Not only have I acquired the scientific knowledge, I also developed the sense of conducting high-quality research as well as searching for unceasing excellence and tireless creativity.

I am greatly indebted to all members of the Multidimensional Image Processing group at the University of Heidelberg. They have offered me tremendous help and support to my research and my life. I would like to particularly thank Dr. Marc Kirchner and Dr. Bernhard Renard for their significant contribution to my first project, which guided me through a smooth transition from Physics to Computational Science. From them have I received many inspiring suggestions in science and beyond. It was also highly rewarding to have inspiring discussions on work and life in general with Dr. Ullrich Köthe, Dr. Michael Hanselmann, Björn Andres, Anna Kreshuk, Dr. Frederik Kaster, Bernhard Kausler, Martin Lindner, Thorben Kröger and Dr. Christoph Sommer.

My deep gratitude also goes to my collaborators Dr. Matthias Mayer, Prof. Jochen Wittbrodt, Dr. Yuki Oguchi as well as Dr. Christian Graf, Chung-Tien Lee and Burkhard Höckendorf. They provided me the unique opportunity to collaborate on highly interdisciplinary projects and granted me the insight to state-of-the-art biological research.

I am very grateful for the financial support by the Heinz Götze Memorial Fellowship and the Helmholtz-Alliance on Systems Biology. I would like to thank Prof. Dietrich Götze not only for financing my study but also for the yearly gathering with nice food, pleasant music and interesting discussions. I should also thank Dr. Dietlind Wünsche for taking good care of us international students and making us feel home in Heidelberg.

On a personal note, I am very thankful to many of my friends for kindly sharing their knowledge. This includes Dr. Shixia Liu, Bin Shen, Zhuang Lin, Dr. Jing Yuan and Dr. Hongwei Zheng. Most importantly, this thesis would have been impossible without the constant encouragement, support and patience from my parents and my wife. I can never repay what they have done and sacrificed for me and my gratitude to them is beyond what words can express.

To my family.

Contents

Chapter 1

Introduction

The notion “drowning in data” applies to many industries in modern society and is also particularly true for current scientific research in biology and medicine. The unceasing flood of data from large amount of experiments always baffles scientists with massive size and uninterpretable patterns. Fortunately, the collaboration between biomedical research and scientific computing has achieved notable progress during the past two decades and it catalyzed the success of many important scientific projects. This suggests a new era of data-intensive scientific discovery. While enjoying the success in the past, we shall also foresee the future of biomedical data analysis. In this chapter, we discuss the new challenges and opportunities in this field, which motivate several interesting and important directions including the focus of this thesis - prior knowledge.

1.1. Challenges and Opportunities in Biomedical Data Analysis

1.1.1. The Challenges

Among many successful stories of scientific computing helping biomedical discovery, the Basic Local Alignment Search Tool (BLAST) [5] is probably the most well-known algorithm which contributed significantly to the Human Genome Project [150, 38]. It is technically a one-dimensional sequence matching problem that is very common in computer science. However, the past decades have witnessed rapid advances in biomedical experimental techniques such as faster and higher-resolution microscopes, more accurate staining as well as more complete protocols. This drives scientists to study more fundamental questions such as “how do neurons in human brain connect and interact” or “how a single cell grows into life”. Particularly large amount of datasets with higher complexity are being produced. They encode the key information to the answers of those questions. The demand for advanced data analysis has reached a much higher level, so does the expectation. We first discuss the challenges in current biomedical data analysis.

High complexity is the first challenge that shall be considered. Unlike the aforementioned one-dimensional sequence matching problem, current datasets from biomedical experiments can be of much higher dimension (e.g. 5D of channel+xyz+time), many different modalities, and large variability in topology and other characteristics. The data analysis problems are therefore becoming much more sophisticated such as reconstructing the neuron connectivity from massive volumetric images [26] and tracking thousands of cells from 3D spatial temporal digital embryo datasets [76].

Accompanying the complexity is **large quantity**. This first attributes to the advances of instruments such as high-resolution microscopes (e.g. STED [62, 122], STORM [125], DSLM [76]) and mass spectrometer (e.g. Orbitrap [66]). Also, the trend towards studying the dynamics such as *in vivo* experiments [95] adds one new temporal dimension and increases the data size by a factor of hundreds or even thousands. Finally, experiments are normally repeated multiple times such as high-content screening [1] and high-throughput screening [63].

The challenges also stem from the **increasing expectation** on the analysis. Scientists now require more accurate data processing that delivers quantitative results [15]. Automation is a desired feature but researchers also like to have the flexibility of interacting with the algorithms [71]. Unlike early signal processing problems such as denoising, current analysis attempts to directly provide the researchers with important knowledge and patterns. This requires the employment of many machine learning techniques that can mimic human experts [11, 141].

1.1.2. The Opportunities

Advances in Computational Methods

We briefly introduce the advances in computational methods with respect to three methodological foundations that the work in this thesis is built on: machine learning, computer vision and image processing, and optimization. Yet, we restrict to several selected topics that are most relevant to this thesis.

In **machine learning**, the most notable advance probably attributes to the development of the theories, algorithms and applications of probabilistic graphical model (PGM) [115, 72]. PGM first provides means to a standard and intuitive mathematical formalism of biomedical applications with complex structure or dynamics [3]. In particular, PGM has the potential of unifying many previously proposed models from different applications and context. PGM is also accompanied with very deep theoretical analysis as well as a huge set of inference algorithms [78]. Another important advance is the development of classification algorithm for very high-dimensional and especially non-linear data. This mainly refers to kernel methods and ensemble learning. Kernel methods, with support vector machine (SVM) [27] being the representative, exploit the kernel trick to generalize the inner-product operator and implicitly project the raw data into a much higher dimension. This achieves a better separability but does not require explicit representation of the projection. It extends and improves the discriminative power of many existing classification, clustering and dimension reduction algorithms [129]. Ensemble learning, such as boosting [128, 52] and random forest (RF) [25], empowers the overall prediction model by combining the power of a collection of weak models. In particular, the RF algorithm exhibits superior performance when dealing with high-dimensional and non-linear data and is empirically the strongest classifier in the native form (without tuning) [31].

Yet another substantial advance in machine learning is the development of structured learning [10]. Unlikely conventional learning methods that work on flat data, structured learning accepts structured inputs (e.g. sequence, graph) and provides structured outputs (e.g. matrix, tree). Practically, structured data is more ubiquitous in real-world applications, which consolidates the significance of this method. Several algorithms have been proposed in the context of graphical models [142] as well as kernel methods [146]. In fact, structured learning aims at combining the ability of graphical models in capturing correlations in structured data and the ability of kernel methods in dealing with high-dimensional features (also fit the maximum-margin paradigm).

In **computer vision and image processing**, one important advance is the employment of graphs for modeling vision and image analysis problems (e.g. graph-based image segmentation [49], random walks [56] and graph cuts [20]). This emerges in conjunction with the development of graphical models as mentioned above and applies to both high- and low-level problems. One representative example is the development of modeling and inference algorithms for Markov random fields (MRFs) [140]. One particularly successful algorithm is graph cut (GC) which has been employed to solve a wide variety of problems from low-level denoising and segmentation to high-level stereo vision and texture synthesis [139]. Another important advance is the use and development of discriminative features for capturing large variability in the data (pixel-based features such as filter banks, Hessian matrix and structure tensor) as well as for detecting interesting points (e.g. SIFT features [99] and its variants). This is again boosted by the advances in machine learning which enable efficient use of the high-dimensional features as well as selection of them when necessary [59].

Since computer vision and image analysis problems can require very high computing power (e.g. feature extraction and model inference in large volumetric images), it is therefore worth mentioning the advance in high-performance computing for image processing. This includes technical improvements on GPUs (e.g. CUDA[1]) or parallelization (e.g. MPI[2]) and theoretical advances that provide means to decompose large problems into parallelized small ones (e.g. dual decomposition [81, 138]). Note that many other important but less relevant advances are not covered here such as scene understanding,

[1] http://www.nvidia.com/object/cuda_home_new.html

[2] http://www.mcs.anl.gov/mpi/

face recognition and robotic vision.

In **optimization**, one important advance was achieved in the context of energy minimization methods in computer vision and pattern recognition. In particular, one direction that has attracted a significant amount of research is discrete optimization for MRF models. For binary labeling problems, the *min-cut/max-flow* algorithm has the favorable property of global optimality [24]. For more sophisticated problems, convex relaxation (realized using linear programming) has been employed [82]. It is also worthy mentioning a particular focus on energy minimization with high-order potentials. This includes algorithms for specific high-order models such as Potts model [77] and clique reduction methods that transform high-order potentials to a combination of low-order ones [68]. Optimization for machine learning has also gained huge advance. This includes, for example, primal and dual optimization for efficient training of SVMs [129], cutting plane and bundle methods for structured learning [146, 144], and stochastic optimization for large-scale learning problems [136, 19]. Another important advance is on the methods and tools of basic mathematical programming. This particularly refers to the IBM ILOG CPLEX[3], which is generally considered the most efficient optimizer for linear programming (LP), integer linear programming (ILP) and quadratic programming (QP) problems. The corresponding software is also well packaged with standard interfaces to many popular programming languages such as C++, Python[4] and Matlab[5].

It is important to notice that all three major topics above are heavily intertwined. Computer vision and image processing frequently employs machine learning but it also motivates the future directions of learning techniques. They both rely on optimization as a fundamental building block but also develop optimization algorithms that are suitable for their respective applications.

1.1.3. Research Directions and Motivations

We now introduce two research directions in biomedical data analysis and motivate the topic of this thesis: prior knowledge.

Interactive and Generic Segmentation Tools

In biomedical image analysis, a high quality segmentation is an important prerequisite for any downstream analysis. This drives the development of several interactive segmentation tools or software frameworks that borrow techniques from image processing, pattern recognition and machine learning [131]. For example, CellProfiler has been widely used in quantitatively analysis of phenotypes [30] and V3D has contributed to the reconstruction of complex 3D neuronal structures from large brain images [118].

[3] http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/

[4] http://www.python.org/

[5] http://www.mathworks.com/

However, different biomedical image data exhibits a large variability of characteristics. It is hence interesting and important to develop a generic segmentation tool that can be adapted to various data by learning from the user interactions. One representative work in this direction is the ilastik software [135]. It integrates a very rich set of generic (non-linear) features and employs active learning to efficiently query user inputs. The underlying implementation has been optimized to ensure real-time interaction with the users. Several applications such as cell classification, neuron boundary prediction and synapsis detection [85] have demonstrated convincing applicability of ilastik.

Large-Scale Image Analysis in Life Sciences

High-resolution 3D microscopic imaging has advanced significantly during the last years. Vast amount of data with increased complexity are produced. In structural neurobiology, the neuron connectomics of the Inner Plexiform Layer (IPL) of the rabbit retina are encoded in volume images as large as $2,000^3$ voxels [26]. In developmental biology, monitoring the development of zebrafish embryos for the first 24 hours produces $1,000^3$ voxels/frame for totally 1200 frames [76].

The work in this direction aims at developing algorithms that allow accurate and efficient processing of such datasets and that automatically extract the encoded information which is practically inaccessible by any manual approach. For example of the connectomics study, researchers have developed a hierarchical segmentation approach that features a probabilistic factor graph model with a global optimization procedure to determine the optimal neuron connectivity configurations [7]. This work complements other related approaches based on convolutional neural network (CNN) [70] or axon tracking [100]. In developmental biology, several algorithms and software pipelines [102, 108, 96] are being developed for the goal of establishing a digital database of embryogenesis (also known as digital embryos[6]).

Biomedical Data Analysis with Prior Knowledge

Incorporating prior knowledge in biomedical data analysis was mainly motivated by two observations. First, each dataset from current biomedical experiments exhibit specific characteristics such as shape, topology and connectivity. These characteristics are crucial to the meaningful interpretation of the experiment. However, many computational methods in their native formulation can not capture such important information. This leads to suboptimal or even incorrect results and limits their applications. For example, directly applying the graph cut [24] algorithm to blood vessel segmentation may lead to unconnected vessel tracks as well as shrinking bias.

Second, many existing methods for biomedical data analysis can be abstracted as parametric models which require the users to directly interact with obscure parameters.

[6] http://www.embl.de/digitalembryo/

Tweaking such parameters is stressful and inefficient, and the results are suboptimal and inapplicable to another datasets with variations. Furthermore, the prior knowledge is only involved in the evaluation step in which users measure the quality of the intermediate analysis and enter another round of parameter tweaking, iteratively.

The focus of this thesis can well complement the two directions above. For example, priors can be integrated with the generic segmentation tool such that more complicated structured can be captured. Also, priors can be crucial for large-scale problems because it provides more accurate interpretation of the datasets and requires less user interventions.

1.2. Paradigms for Incorporating the Prior Knowledge

We developed two paradigms for incorporating the prior knowledge. They shall be appropriately employed for different problems, depending on the representativity and the complexity of the prior.

1.2.1. Constrained Modeling

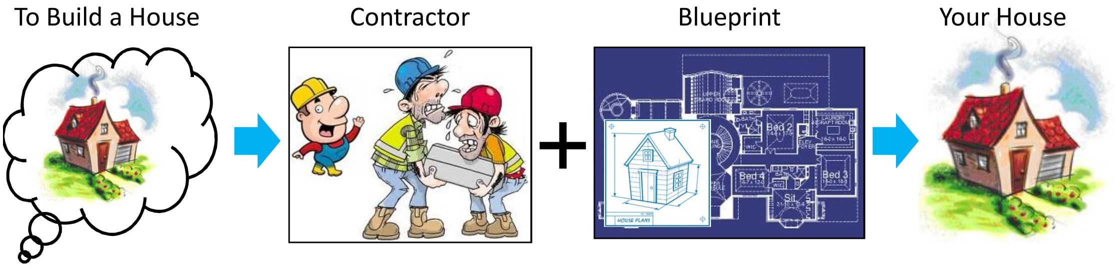


Figure 1.1.: A metaphorical representation of the constrained modeling paradigm.

Constrained modeling generally applies to problems in which the prior can be explicitly represented or quantified, such as “cells have convex shapes”, “human airways have tree structure” and “a parent cell divides into two daughter cells”. Such a prior is implemented by directly inserting additional potentials to the model of the problem. These potentials restrict the downstream optimization and prevent it from yielding solutions that contradict the prior. Metaphorically speaking, consider building a house as your problem and the hired contractor as your optimizer (Fig. 1.1). In this context, incorporating the prior is exactly analogous to letting the contractor follow a detailed construction blueprint. This is of course possible only if the blueprint is accessible, i.e. the prior is explicitly representable. Research in this direction generally responses to the first motivation of this thesis, such as connectivity prior [151, 107], topology prior [157] and shape prior [32, 98].

1.2.2. Learning

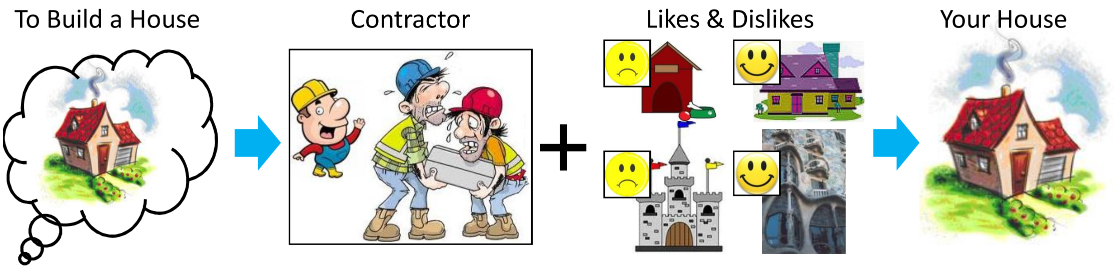


Figure 1.2.: A metaphorical representation of the learning paradigm.

Learning, on the other hand, deals with problems in which the prior cannot be explicitly represented or exactly quantified due to high complexity or incomplete knowledge. For example, in cell tracking we know cell movement is spatially restricted but cannot exactly quantify a proper displacement. Learning approaches can address this problem by estimating the functional dependency between some features as evidences of the inputs (i.e. training data) and manual annotated results as expected output (i.e. training labels). In the context of the same metaphor as above, the prior is now provided in disguise of examples of likes and dislikes (Fig. 1.2). These examples imply the expected results and require the contractor to design the blueprint, i.e. the learning. Obviously, this paradigm responses to the second motivation of this thesis.

1.3. Thesis Overview

The following chapters elaborate on these approaches in great detail with four applications (two for each paradigm):

- Chapter 2 introduces the use of sparsity prior and sequential ordering prior in the context of robust deuteration distribution estimation and tracking.
- Chapter 3 shows how to extend the MRF model with shape prior for cell nucleus segmentation.
- Chapter 4 introduces an extension to the popular Rand Index that learns to rank segmentations from manual scoring.
- Chapter 5 is about a novel cell tracking approach based on structured learning which operates on structured inputs (segmentation and raw image) as well as structured outputs (object association).

Finally, conclusions and outlook are provided in Chapter 6.

Chapter 2

Deuteration Distribution Estimation with Sparsity Prior

This chapter introduces an algorithmic workflow for robust deuteration distribution estimation and tracking for hydrogen exchange (HX) mass spectrometry (MS) experiments. We exploited two priors: first, a sparsity prior on the deuteration distribution helps to yield physically reasonable estimation; second, a sequential ordering prior on the LC retention time allows for jointly alignment and deuteration tracking.

Time-resolved hydrogen exchange (HX) followed by mass spectrometry (MS) is a key technology for studying protein structure, dynamics and interactions. HX experiments deliver a time-dependent distribution of deuteration levels of peptide sequences of the protein of interest. The robust and complete estimation of this distribution for as many peptide fragments as possible is instrumental to understanding dynamic protein-level HX behavior. Currently, this data interpretation step still is a bottleneck in the overall HX/MS workflow.

We propose *HeXicon*, a novel algorithmic workflow for automatic deuteration distribution estimation at increased sequence coverage. Based on an L_1-regularized feature extraction routine, HeXicon extracts the full deuteration distribution, which allows insight into possible bimodal exchange behavior of proteins, rather than just an average deuteration for each time point. Further, it is capable of addressing ill-posed estimation problems, yielding sparse and physically reasonable results. HeXicon makes use of existing peptide sequence information which is augmented by an inferred list of peptide candidates derived from a known protein sequence. In conjunction with a supervised classification procedure that balances sensitivity and specificity, HeXicon can deliver results with increased sequence coverage.

2.1. Introduction and Related Work

The determination of protein structure and dynamics is a key issue for the understanding of living systems [41]. By combining the information of the protein dynamics and other classical functional data, a more complete understanding of protein function can be obtained. In many cases, protein dynamics are directly related to specific protein functions such as conformational changes during enzyme activation and protein movements during binding [153]. Hydrogen exchange followed by mass spectrometry (HX/MS) has become a standard approach for interpreting HX experiments: the location and rate of deuteration are indicative of solvent accessibility and in particular hydrogen bonding and hence of conformation and dynamics [46]. They can be estimated by tracking the mass shift of peptide fragments in mass spectra over samples with different incubation times (Fig. 2.1) [45]. In comparison to Nuclear Magnetic Resonance (NMR) spectroscopy, mass spectrometry requires lower protein concentrations and amounts, provides higher measurement speed and better scalability in terms of protein size, and detects coexisting conformations [64]. Whereas manifold improvements in experimental methodology and instrumentation have been implemented for HX/MS experiments, data processing still remains a major difficulty in the overall experimental workflow [45]. First of all, the precise deuteration distribution is represented by complex peak patterns that are difficult to separate and quantitate even in 2D LC/MS (Liquid Chromatography/Mass Spectrometry) representation. Secondly, the peptide sequences of interest have to be pre-determined via MS/MS search report or selected empirically, yielding suboptimal sequence coverage of the protein of interest. Finally, manual analysis is time-consuming, error-prone as well as inaccurate in case of overlapping isotope clusters (Fig. 2.1).

Several methods and tools have been developed to facilitate the manual analysis. Palmblad and colleagues [112] modeled the deuterium incorporation as a binomial distribution and used χ^2-statistics to extract the optimal parameter. Weis and Engen [154] designed HX-Express as a semi-automatic data processing tool which measures the deuteration by the width of the given isotope pattern. TOF2H [106] is an integrated software framework designed specifically for semi-automatic LC-MALDI (Matrix-Assisted Laser Desorption/Ionization) data analysis.

Note that while the approaches mentioned above facilitate the analysis of HX/MS data, they do not yield the complete deuteration distribution, but only the average deuteration. The true deuteration distribution offers a more detailed characterization and more insightful description of the exchange process. In particular, it is suitable for discovering bimodal exchange behaviors of large protein oligomers, which are not detectable by average deuteration levels.

The algorithms developed for extracting deuteration distribution information mainly fall into two categories. The first set of methods fit a hypothetical deuterated isotope pattern to the observed spectrum by least-squares regression [2, 65, 134]. They exhibit the advantage of speed but have difficulties in handling ill-posed problems, which, as

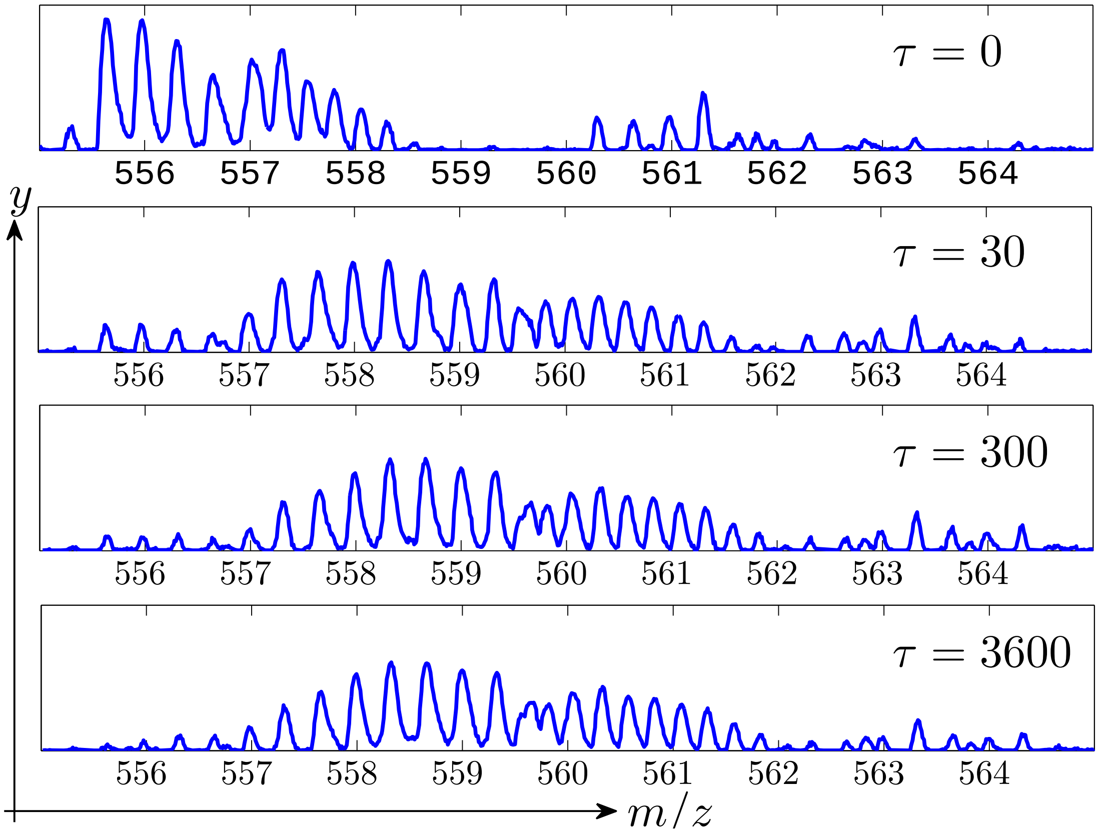


Figure 2.1.: Examples of HX/MS spectrum data from an incubation time series of 0, 30, 300 and 3600 seconds: the isotope envelope shifts to higher m/z values because of deuterium incorporation. The deuteration content is encoded in a complex mixture of isotope distributions. Due to the noise and overlapping isotope clusters, the separation of individual peptides is non-trivial. The abundance of the spectrum is labeled as y.

shown in the following, are common in large-scale HX/MS data analysis. It is possible to make use of padding methods to regularize the ill-posed regression problem. Given the optimal degree of padding, this approach can address data truncation problems and avoid over-fitting to noise [36]. The second set of methods is based on maximum entropy deconvolution [158, 2]. Those methods can handle ill-posed problems and yield non-negative outputs; however, they are computationally much more expensive [158]. One common limitation of these two categories is that they are designed for well-tuned and small-scale problems, i.e. the peptide sequence of interest is pre-selected in a well-separated spectrum, thus making them less applicable in practice, especially for large-scale HX/MS data processing. These methods have been implemented by several software tools such as Deuterator [114, 113] and Hydra [134]. Both frameworks focus on incorporating existing algorithms and providing user-friendly GUI and powerful visualization.

We propose a novel algorithmic approach named HeXicon to the deuteration distribution estimation problem for large-scale HX/MS experiments. HeXicon exploits information in the retention time and m/z domains for optimized separation of large HX/MS data and applies NITPICK [121] for LC/MS feature extraction, resulting in a robust and regularized estimation of the deuteration distribution. It integrates protein sequence and protein identification information in an attempt to increase the sequence coverage.

Section 2 of the manuscript elaborates the methodological development of our approach. Sections 3 describes the experimental setup and reports the results, focusing on the novelty of delivering a robust estimate of the deuteration distribution and the comparison to manual analysis. Discussion and conclusion are offered in sections 4 and 5, respectively.

2.2. Deuteration Distribution Estimation with Improved Sequence Coverage

As illustrated in Fig. 2.2, our approach consists of two major modules that jointly carry out our goals of robust deuteration distribution estimation and sequence coverage improvement. Given a hypothetical set of peptide sequences inferred in *Peptide Sequence Set Determination* (**A**), the *Deuteration Distribution Estimation* starts by constructing an over-complete set of basis functions (**B**) and then feeds them into the NITPICK algorithm to yield peak groups with features (**C**). Inter-experiment peak groups are then associated via correspondence estimation (**D**) and the deuteration distribution is derived for each association (**E**). The subsequent quality estimation of *Peptide Sequence Set Determination* retains the high-quality results and thus balances the sensitivity and specificity (**F, G**). Our approach makes extensive use of the NITPICK algorithm, a regularized, non-greedy, globally optimal linear mixture modeling algorithm for feature extraction from multicomponent mass spectra.

2.2.1. Deuteration Distribution Estimation

Definition Let p be a peptide sequence of interest. The deuteration level k is the number of deuterium exchanges at the back-bone hydrogens of p. The deuteration distribution $\rho(p, k, \tau)$ is the fraction of peptide with sequence p at deuteration level k for incubation time τ, where $k \in \{0, 1, \ldots, K(p)\}$ and $K(p)$ is the maximal possible deuteration level. The average deuteration $\eta(p, \tau)$ is the average deuteration level of all peptides with sequence p at incubation time τ.

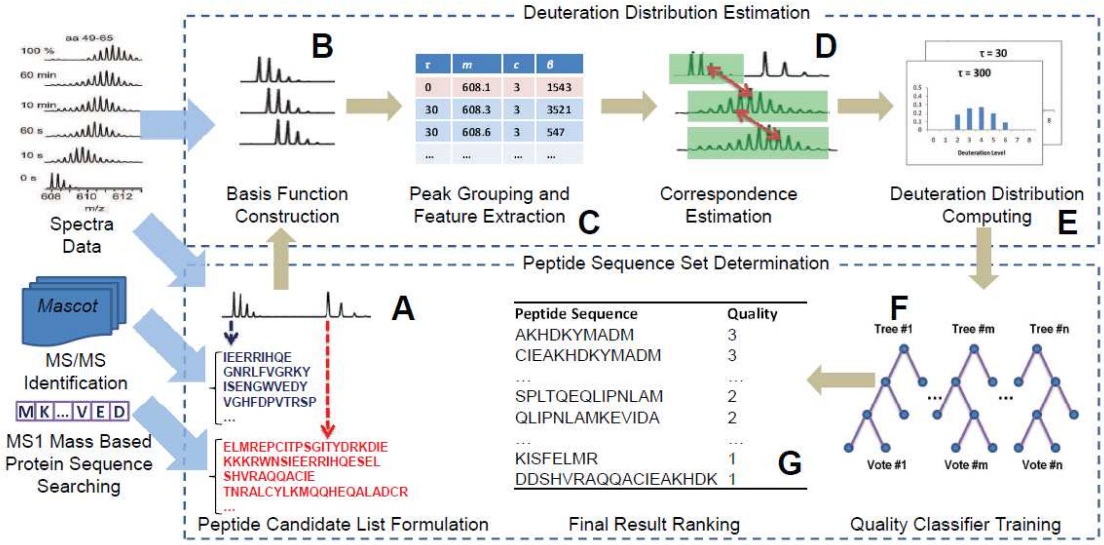


Figure 2.2.: Workflow of HeXicon. **A:** The list of peptide identification from MS/MS searches is automatically extended by matching theoretical peptides to observed masses to find peptide sequence candidates for previously unidentified peptides; **B:** A basis function set is created by modeling all possible deuteration levels for each peptide sequence; **C:** The spectra and basis function sets are inserted into the LC/MS segmentation and NITPICK routine and groups of peaks with features are extracted; **D:** The correspondence of inter-experiment peak groups are identified via a weighted Euclidean distance measure; **E:** The deuteration distribution is derived; **F:** A random forest classifier discriminates high-quality results from low-quality results; **G:** The final results are ranked by their quality score.

NITPICK Algorithm

We formulate the deuteration distribution estimation as a regression problem. That is, the observed spectrum $\boldsymbol{s}$ is explained as a linear combination of constituent basis spectra which represent a particular peptide. Each feasible basis spectrum is specified by one column of the regression matrix $\boldsymbol{\Phi}$, and the regression coefficients $\boldsymbol{\beta}$ determine the abundance of those constituents in the mixture. If the matrix $\boldsymbol{\Phi}$ contains more basis functions than are actually present in any given mixture $\boldsymbol{s}$, the regression problem is ill-posed and has to be constrained. [145] showed that the introduction of a $L1$-constraint leads to a sparse solution vector $\boldsymbol{\beta}$ which assigns non-zero abundance only to those basis functions that are contained in the mixture with high probability. The resulting regression problem is

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left\{ \left\{ ||\boldsymbol{s} - \boldsymbol{\Phi}\boldsymbol{\beta}||_2^2 + \lambda \, ||\boldsymbol{\beta}||_1 \right\} \right\} \text{ subject to } \boldsymbol{\beta} \geq 0, \tag{2.1}$$

which can be solved with the same computational efficiency as an ordinary least squares problem by the LARS algorithm [44]. The regularization parameter λ controls the model complexity based on the Bayesian Information Criterion (BIC) [130]. The NITPICK algorithm [121] determines its value automatically so that the number of degrees of freedom in the model is matched to the observed noise level of $\boldsymbol{s}$.

Basis Function Construction

Assuming that the peptide sequence set of interest P is known (see section 2.2.2), the solution to the regression problem must lie in a space spanned by all deuteration levels of all peptide sequences in the set (Fig. 2.2 **B**). Thus, we build the basis function set $\boldsymbol{\Phi}$ by combining the theoretical isotope distribution for every deuteration level of each peptide sequence in P:

$$\boldsymbol{\Phi} = \bigcup_{\forall p \in P, \forall k \in [0, K(p)]} \phi(p, k) \tag{2.2}$$

where $\phi(p, k)$ is the transformation function that computes the basis function for peptide sequence p at deuteration level k (i.e. its theoretical isotopic spectrum); $K(p)$ is the maximum number of exchangeable hydrogens [153]. To accommodate for the non-constant, m/z-dependent resolution [58, 35], we use a m/z-dependent peak shape function and learn its parametrization from the data (see Appendix).

Quantitative LC/MS Feature Extraction

This feature extraction procedure provides two key steps for the workflow: firstly, it selects a subset of basis functions $\hat{\boldsymbol{\Phi}} \subseteq \boldsymbol{\Phi}$ that optimally explain the observed spectrum and thus determines the peptide sequences of interest; secondly, it extracts features of selected basis functions for the following deuteration distribution computation and correspondence estimation.

We first apply segmentation techniques to achieve optimized separation of the LC/MS data, which yields better signal-noise-ratio (SNR) and groups signals that belong to the same peptide. Manual analysis and some existing methods normally use a heuristic window-based approach for separating the LC/MS data. The integration of LC/MS peaks along the entire retention time window yields suboptimal SNR and fails in case of overlapping peak clusters [6]. Therefore, we integrate over retention time only within segments and are thus able to benefit from better SNR. The exact retention time position of the peptide is then determined via a sparse elution profile estimation on the LC/MS data segment [18]. Thereafter, to determine the ratio of different deuteration levels of the peptide sequence of interest, the abundance of their corresponding basis function $\phi(p,k)$ is estimated using the NITPICK algorithm (Fig. 2.2 **C**). The regression problem (Eq. 2.1) is normally ill-posed because the basis function construction yields an over-complete set of explanatory variables. Also, NITPICK provides sparse solutions which represent a subset of the over-complete basis function set that is indeed necessary to explain the observed spectrum.

Eventually, for each incubation time τ, the feature extraction procedure outputs a list of peak groups $\boldsymbol{\mathcal{G}}^{\tau}$, where a group $\boldsymbol{g}^{\tau}$ corresponds to a certain segment and contains peaks with features:

$$\boldsymbol{\mathcal{G}}^{\tau} = \{\boldsymbol{g}^{\tau}\} = \left\{(m,\beta,z,t)_{\boldsymbol{g}^{\tau}}\right\}$$

where m is the monoisotopic m/z position, β is the abundance of the corresponding basis function, z is the charge and t is the estimated retention time.

Correspondence Estimation

This step determines the correspondences between the peak groups over incubation time points and the peptide sequences of interest (Fig. 2.2 **D**). Given a peptide sequence p of interest, its zero exchange peak group is first determined by matching a measured peak to its theoretical m/z value,

$$\hat{\boldsymbol{g}}^{0} = \underset{\boldsymbol{g}^{0}\in\boldsymbol{\mathcal{G}}^{0}}{\arg\min}\left\{|m_{\boldsymbol{g}^{0}} - f_{\text{theoretical}}(p,z,0)\,|\right\}, \tag{2.3}$$

where $f_{\text{theoretical}}(p,z,k)$ computes the theoretical m/z of p at charge z and deuteration level k. The corresponding peak group at every other incubation time is determined by minimal weighted Euclidean distance

$$\hat{\boldsymbol{g}}^{\tau} = \underset{\boldsymbol{g}^{\tau}\in\boldsymbol{\mathcal{G}}^{\tau}}{\arg\min}\left\{\sqrt{(\boldsymbol{g}^{\tau}-\boldsymbol{g}^{0})^{T}\boldsymbol{S}^{-1}(\boldsymbol{g}^{\tau}-\boldsymbol{g}^{0})}\right\}, \tag{2.4}$$

where $\boldsymbol{S}$ is a diagonal matrix which normalizes and weights the contributions of different features to the distance measure. The matrix $\boldsymbol{S}$ is designed to express the characteristics of signals belonging to the same peptide sequence over incubation time. To speed up the computation, we also applied a filtering procedure to eliminate unlikely candidates by charge consistency and thresholding via retention time window and m/z accuracy cutoff.

Deuteration Distribution Estimation

After determining the inter-experiment correspondence of peak groups with respect to a peptide sequence of interest, its deuteration distribution can be derived as (Fig. 2.2 **E**)

$$\rho(p, k, \tau) = \frac{\beta_{\hat{\boldsymbol{g}}^{\tau}}(k)}{\sum \beta_{\hat{\boldsymbol{g}}^{\tau}}} \tag{2.5}$$

where the $\beta_{\hat{\boldsymbol{g}}^{\tau}}(k)$ is the abundance of the basis function corresponding to deuteration level k. The average deuteration is merely the average of the deuteration distribution over all deuteration levels.

2.2.2. Peptide Sequence Set Determination

To perform complete deuteration distribution estimation for the entire protein, optimized protein sequence coverage is desirable. We achieve this goal by extending the peptide sequence set via sequence search and later using a supervised classification approach to discard incorrect or ambiguous peptide sequences.

Unsupervised Peptide Sequence Inference

We use a two-step procedure to infer possible peptide sequences directly from the observed spectrum and from prior knowledge (i. e. the protein sequence and the MS/MS report). We first perform peak picking on the observed spectrum using the NITPICK algorithm. In a second step, the picked monoisotopic masses, for which no MS/MS identifications are available, are matched to theoretical peptide sequences extracted from the known protein sequence. Eventually, a list of candidate peptide sequences is generated, which consists of peptide sequences from two sources: peptides that are identified by MS/MS data and peptides that are extracted by searching the protein sequence for subsequences with a mass proximate to the picked peaks.

Supervised Quality Estimation

The unsupervised peptide sequence inference procedure exploits information without sufficient concern for multiple assignments of peptide sequences to the same peak or peptide sequences hallucinated from noise peaks. Despite the fact that this apparently improves the system's sensitivity, the payoff is a reduced specificity, i.e. false positives are mixed into the peptide sequence set. Therefore, HeXicon implements a quality estimation procedure to recover reasonable specificity while maintaining high sensitivity. We tackle this problem using a supervised classification approach: given training data $\{\boldsymbol{x}, q\}$

Measure	CHIP	HtpG
Protein length	303	636
Protein weight (kDa)	34.8	72.8
Data size (MB)	ca. 121	ca. 671
Incubation time (minutes)	0, 0.5, 5, 60	0, 5, 10, 30
Manually selected peptide sequences	21	39
Manual analysis time	ca. 2 days	ca. 1 week

Table 2.1.: Summary of the CHIP and HtpG datasets.

where $\boldsymbol{x} \in \mathcal{X}$ is the quality feature vector and $q \in \mathcal{Q}$ is the quality label, train a classifier $h : \mathcal{X} \to \mathcal{Q}$ that maps $\boldsymbol{x}$ to its estimated quality q. In particular, we use the Random Forest classifier [25], a supervised, decision-tree based ensemble learning method with high prediction accuracy and little sensitivity to the hyper-parameter settings (Fig. 2.2 **F**).

A representative dataset was selected as training data and each reported peptide sequence was labeled with a quality score $q \in \{3, 2, 1\}$, in which 3 represents highly confident results, 2 indicates ambiguous results, i.e. unidentified peptide sequence resulting from multiple assignment to the same peak, and 1 contains all results containing no useful information. The quality features $\boldsymbol{x}$ are designed to characterize the quality of a peptide sequence from several different aspects. See [97] for a full list of quality features. Retraining is necessary for different instruments.

2.3. Experimental Results

HeXicon has been evaluated on two protein datasets of different complexity (Table 2.1): C terminus of Hsp70 Interacting Protein (CHIP) and High temperature protein G (HtpG). In each experiment, protein samples were first incubated in heavy water to induce a certain amount of exchange before being subjected to pepsin digestion. To identify peptic peptides from the investigated proteins we digested the undeuterated protein under the same conditions as later used for the exchange experiments. We then analyzed the peptic peptides by automated MS/MS using a 1 hour acetonitrile gradient either on a nanoLC-QSTAR MS system (CHIP, HtpG) and on a nanoLC-Orbitrap MS system (HtpG). Subsequently we determined, which of the identified peptides could be found consistently on the HPLC-QSTAR MS system using a 10 min acetonitrile gradient.

Both datasets have been processed manually, yielding average deuterations for selected peptide sequences that we use as ground truth. Segment retention time extensions are between 20s and 50s.

Dataset	Measure	Manual Analysis	HeXicon
CHIP	Number of extracted peptide sequences	21	31
	Sequence coverage	84.2%	90.4%
	Analysis time	2 days	1 hour
HtpG	Number of extracted peptide sequences	39	90
	Sequence coverage	78.5%	85.5%
	Analysis time	1 week	3 hours

Table 2.2.: Comparison to manual analysis: sequence coverage and analysis time.

2.3.1. Deuteration Distribution Estimation

For the CHIP spectra in Fig. 2.3 (first column), HeXicon provides a sparse and condensed estimation of the deuteration distribution which exhibits smoothness along the deuteration levels, as shown in Fig. 2.3 (second column). For comparison, we created a well-posed regression problem by constructing basis functions for the corresponding peptide CIEAKHDKYMADM and applied the non-negative least-squares regression based method described in [36]. We optimized the degree of padding by manually estimating the maximal deuteration level, yielding a solution very similar to the HeXicon's. Without optimizing the degree of padding, i.e. padding to the theoretically maximal possible deuteration level, Chik's approach selects several spurious basis functions due to overfitting (fourth column). Fig. 2.4 (top) shows a mixture of signals from two peptide sequences: AAERERELE and IAKKKRWNSIEER. HeXicon yields condensed deuteration distributions for both peptide sequences, as shown in Fig. 2.4 (bottom, first column). After padding optimization, Chik's approach gives a similar distribution for AAERERELE but the estimate for IAKKKRWNSIEER is questionable, see Fig. 2.4 (bottom, second column).

2.3.2. Sequence Coverage Enhancement

Combining MS/MS identifications and inferred peptide sequences, HeXicon yields an apparent improvement on the number of extracted peptide sequences with concomitant increases in sequence coverage when compared to the manual analysis (Table 2.3.2). For the manual analysis we only used those peptides identified that we could find consistently in the 10 min gradient runs on the QSTAR system.

2.3.3. Exchange Rate Inference

The deuteration distribution estimated by HeXicon can easily be transformed into an average deuteration estimate $\eta^{\mathrm{H}}(p, \tau)$ by computing the empirical mean. We validated HeXicon by comparing its average deuteration estimate to the manually obtained average deuteration estimate $\eta^{\mathrm{M}}(p, \tau)$. We applied two metrics to measure the accuracy: (i) the

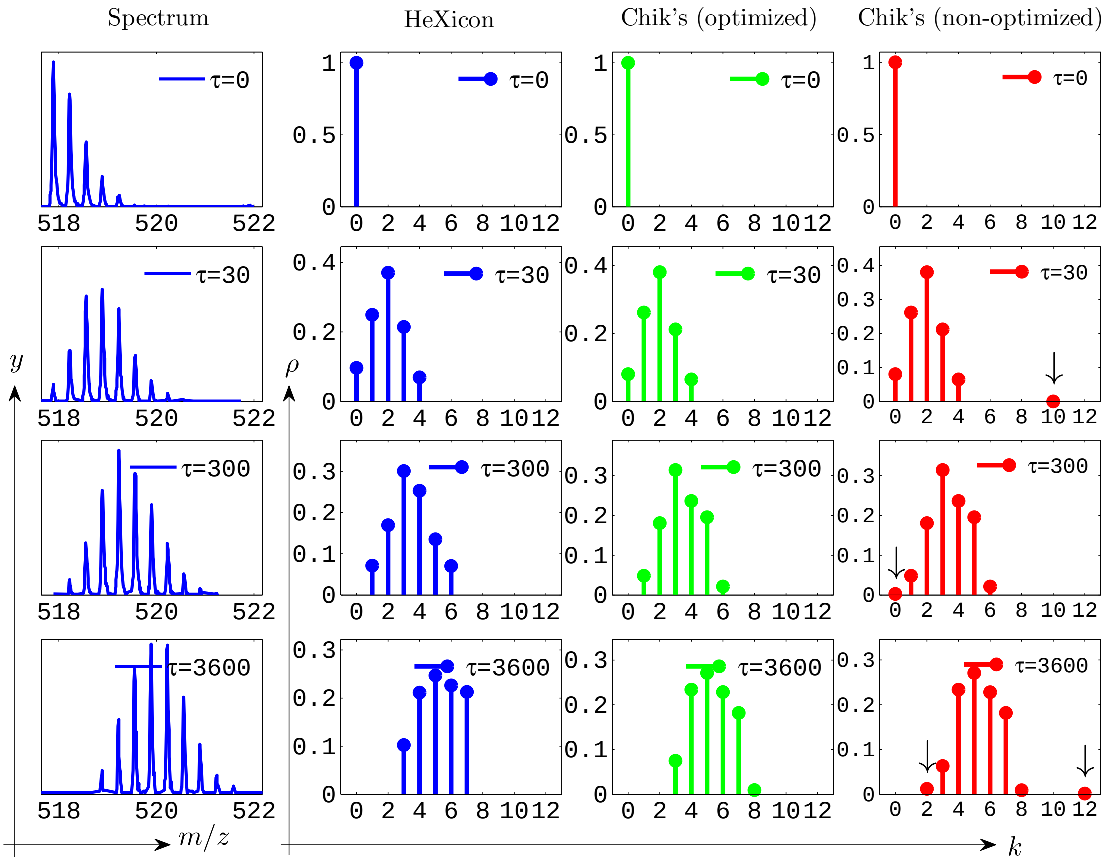


Figure 2.3.: Comparison of deuteration distribution estimation of CIEAKHDKYMADM from a time series of 0, 30, 300 and 3600 seconds (first column). HeXicon yields condensed solution and avoids overfitting (second column). With manually optimized degree of padding, Chik's approach results in similar estimates (third column). Without padding optimization, Chik's approach selects several spurious peaks (fourth column, marked by "↓") due to overfitting.

average m/z difference Δ_m is computed by $\Delta_m = \sum_\tau \left|\eta^{\mathrm{M}}(p,\tau) - \eta^{\mathrm{H}}(p,\tau)\right|/N_\tau$, where N_τ is the total number of incubation time points; (ii) the relative exchange rate difference Δ_κ is computed by $\Delta_\kappa = \left|\kappa^{\mathrm{M}} - \kappa^{\mathrm{H}}\right| / \max\left(\kappa^{\mathrm{M}}, \kappa^{\mathrm{H}}\right)$, where κ is the exchange rate inferred by fitting the average deuteration to the HX kinetic model function [83]. Since the fitting is non-linear and non-convex and since its first-order and second-order derivatives could be derived analytically, we applied a generalized Newton method to approximate the optimal solution (see Appendix).

For the CHIP dataset and 20 of 21 manually selected peptide sequences, the estimates

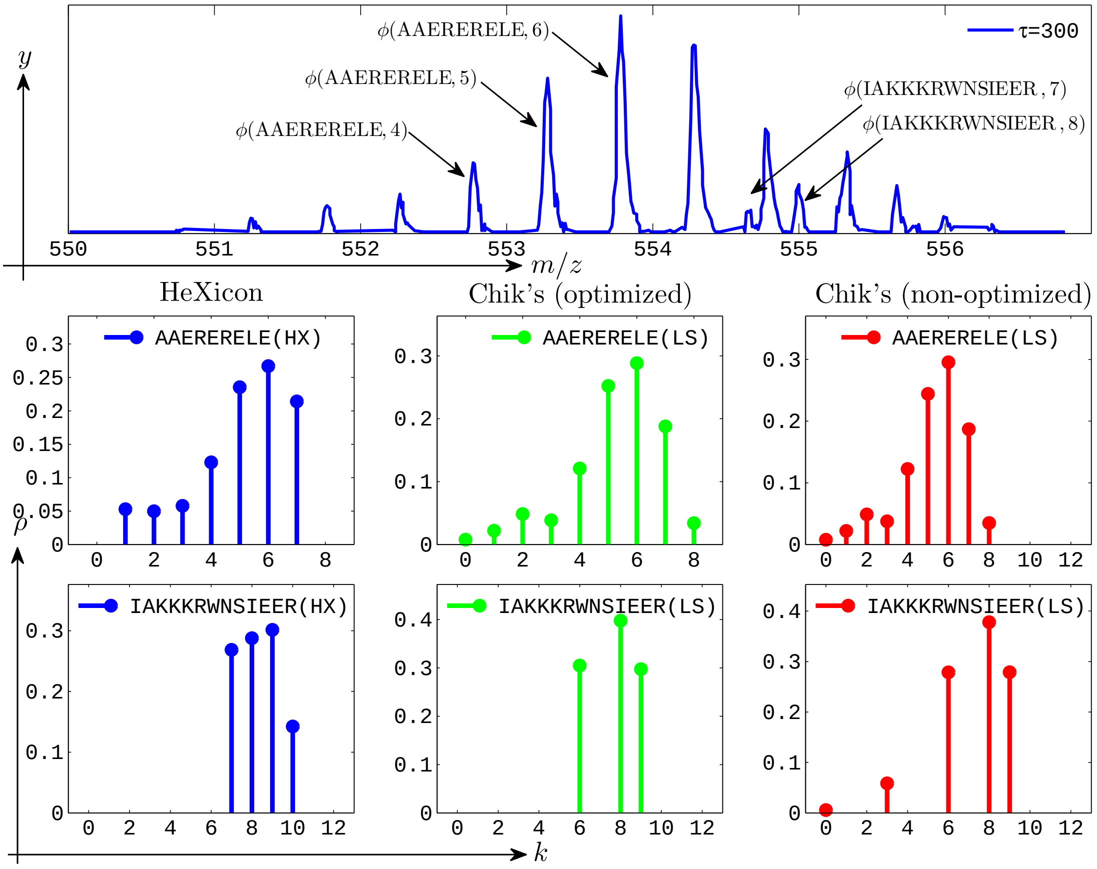


Figure 2.4.: Comparison of deuteration distribution estimation for overlapping patterns. Overlapping patterns consist of AAERERELE and IAKKKRWNSIEER (top). HeXicon yields condensed and smooth solutions for both peptide sequences (bottom, first column). Even with padding optimization, Chik's approach overfits the spectrum and yields an unrealistic deuteration distribution for IAKKKRWNSIEER (bottom, second column). Here $\phi(p, k)$ indicates the maximum peak position of the basis function of peptide p at deuteration level k.

by HeXicon coincide well with the manual analysis (see examples in Fig. 2.5, top-left and top-right), yielding an average m/z difference of 0.0688 ± 0.0307 Da (mean ± standard deviation) and a relative exchange rate difference of 0.0994 ± 0.0847. For the HtpG dataset, HeXicon correctly estimates the average deuteration for 32 of 39 manually selected peptide sequences and yields an average m/z difference of 0.0578±0.0339 Da and a relative exchange rate difference of 0.1205±0.0958 (e.g. Fig. 2.5, bottom-left). For the remaining seven manually selected peptides, the estimates are inaccurate (e.g. Fig. 2.5,

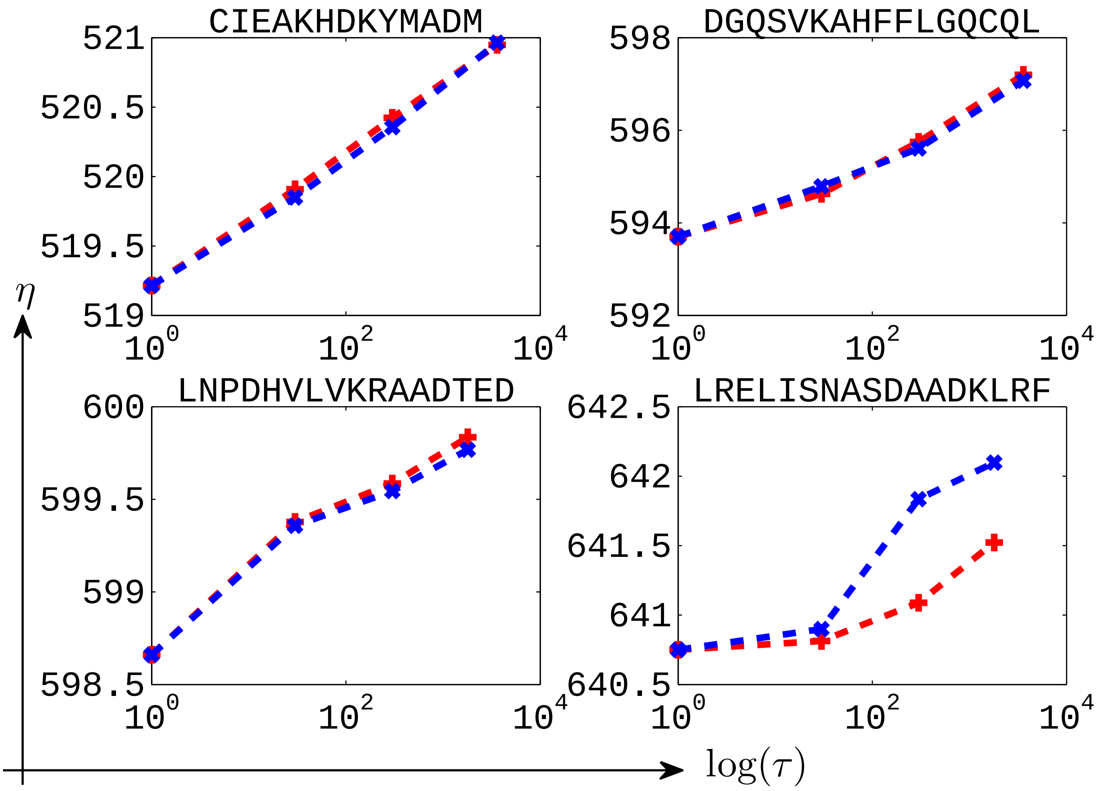


Figure 2.5.: Comparison of the exchange rate inference between manual analysis (red) and HeXicon (blue) for selected examples. While the estimate by HeXicon coincides well with the manual analysis for the peptides displayed on the top-left, top-right and bottom-left, the estimate for LRELISNASDAADKLRF (bottom-right) is incorrect due to under-segmentation of overlapping peptides in the LC/MS spectrum.

bottom-right). The complete list of peptide sequences and their average deuteration is given in the Appendix.

2.3.4. Quality Filtering Accuracy

The quality estimation step aims at identifying high-quality results and discarding the remaining results. We measure the cross validation performance of this step using common criteria from information theory: recall, precision and F-score. The results given in Table 2.3 indicate that the quality estimation step is accurate and generalizes well across data sets, providing an F-score over 90%.

Measure	Class 1	Class 2	Class 3
Recall	98.8	91.6	92.2
Precision	98.7	92.9	89.2
F-score	98.7	92.2	90.7

Table 2.3.: Cross validation performance: recall, precision and F-score (in %) are given for high quality (Class 3), medium quality (Class 2) and low quality (Class 1) results.

2.3.5. Software and Runtime

HeXicon has been implemented in C++ and the compiled software is available at http://hci.iwr.uni-heidelberg.de/software.php. As indicated in Table 2.3.2, HeXicon strongly reduces the analysis time compared to manual analysis. Since HeXicon is fully automated, it does not require any real-time user-interaction. Experiments were carried out without replicates. In order to perform replicate analysis, HeXicon results need to be obtained separately for each replicate and subsequently aggregated. The software package requires the spectrum data as mzXML files and other information (i.e. the protein sequence and the MS/MS search result) as plain text files. CSV files are the output.

2.4. Discussion

As shown in section 2.3.1, to avoid overfitting Chik's approach requires padding optimization by user-input or pre-processing. The reason is that the least-squares regression attempts to use each predictor without any restriction and thus overfits the data and causes several spurious basis functions to be selected, as shown in Fig. 2.3 (fourth column, marked by "$\downarrow$"). The proposed approach benefits from the sparsity of the $L1$-regularization and discards those spurious deuteration levels automatically, and thus requires no additional processing such as thresholding or any further user interaction. This overfitting problem becomes worse when overlapping patterns occur. As shown in Fig. 2.4 (bottom, right column), Chik's approach (with padding optimization) gives a reasonable distribution for AAERERELE, but yields an unrealistic estimate for IAKKKRWNSIEER, i.e. the large gaps between neighboring deuteration levels. HeXicon, on the other hand, keeps the intrinsic smoothness and sparsity of the deuteration levels. Although the estimate for the low-intensity IAKKKRWNSIEER is subject to low SNR, it is still represented by a compact deuteration distribution at the most relevant positions and appears to be physically reasonable. While maximum entropy deconvolution based methods [158] might theoretically be appealing, they are not applicable to the problem since they require a pre-defined noise level [2] which is usually not available to the users and may vary among different m/z regions or experiments. Further, these approaches are prone to overfitting and are computationally expensive [65].

The improved sequence coverage provided by HeXicon is particularly helpful to gain a more complete and detailed understanding of a dataset. Due to under-segmentation of crowded regions in the LC/MS data, HeXicon did not recover all manually selected peptide sequences from the HtpG dataset, but it still managed to yield a higher sequence coverage because other peptide sequences were selected to compensate for the missing ones. Further, as shown in Table 2.3.2, HeXicon finds more than twice the number of peptide sequences selected by human experts, which allows exchange behavior prediction in finer regions. For instance, the estimation of exchange rate at positions 279-284 can be inferred from both HLQRVGHFDPVTRSPLTQEQLIPNL (position 259-284) and HLQRVGHFDPVTRSPLTQE (position 259-278). Since we only considered those HeXicon results with the highest quality score for the computation of the sequence coverage, this number can be regarded as a conservative estimate. Additional lower quality results provided by HeXicon can guide users towards further targeted experiments. For instance, ambiguous results, when multiple peptide sequences could be assigned to the same spectrum, might motivate additional MS/MS run on specific peptide sequences of interest, and thereby allow further improvement on the sequence coverage.

2.5. Extensions

2.5.1. Detection of Bimodal Isotope Peak Distributions

Bimodal isotope distributions arise due to EX1 exchange mechanisms or different coexisting conformations in a protein. These distributions are of specific interest because they not only report on the kinetics of conformational changes but also provide an even greater challenge for automated data analysis. We extended the proposed HeXicon approach to search specifically for bimodal isotope distributions in large data sets [84]. We applied the modified program to a dataset from the E. coli Hsp90 homologue HtpG and compared the results with manual data analysis. All seven manually found bimodal cases were detected as bimodal by HeXicon. In addition, HeXicon also located nine previously unknown bimodal distributions, illustrating the benefit of automated data processing.

2.5.2. Gaussian Mixture Model for Asymmetric Spectrum Analysis

We also developed an extension of HeXicon to handle asymmetric spectrum using Gaussian mixture model (GMM). The asymmetric spectrum of interest, together with prior knowledge such as the mascot search report and the protein sequence, was first processed using HeXicon for deuteration distribution estimation. The resulting deuteration distribution was further processed to estimate the ratio of the conformation components that result in the observed asymmetric spectrum. In particular, the coexistence of different conformations is approximated by a Gaussian mixture model [16]. The ratio of the two conformations can be easily computed after inferring the GMM parameters using the Expectation-Maximization (EM) algorithm coupled with the Bayesian information criterion (BIC) for controlling the model complexity [16].

2.6. Conclusions and Outlook

In this article, we introduced HeXicon, a novel algorithmic workflow for the robust estimation of deuteration distributions with increased sequence coverage for HX/MS experiments. Comparisons to previous methods showed that the $L1$-regularization adopted in our method provides a sparse estimation of deuteration distributions and avoids overfitting. The overall sequence coverage is increased by inferring peptide sequences from prior knowledge, and the tradeoff between sensitivity and specificity is balanced using a supervised classification procedure. In comparison to manual analysis, we showed that HeXicon succeeds in accurately extracting the deuteration content while improving sequence coverage and reducing analysis time.

In the future, we plan to improve current HeXicon by replacing the local correspondence estimation with a global association method that incorporates a sequential ordering prior. Details are given as follows.

2.6.1. Jointly Alignment and Tracking with Sequential Ordering Prior

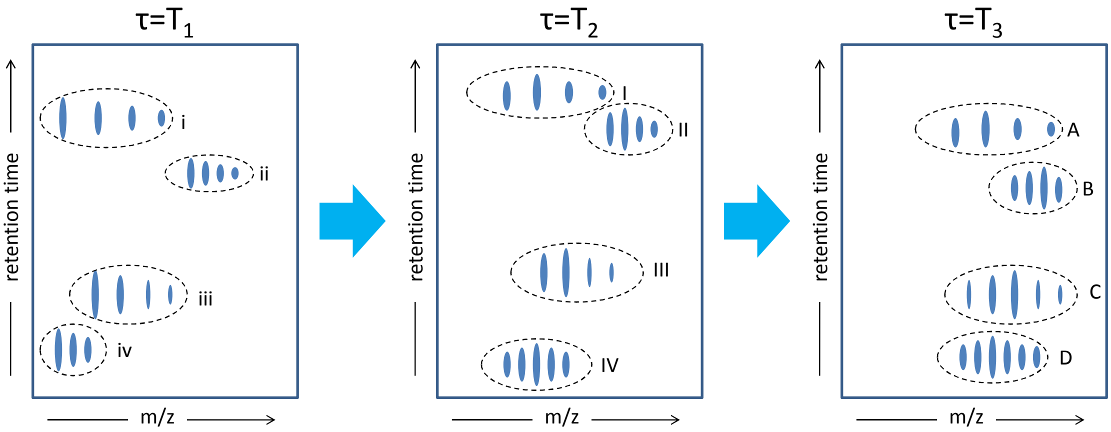


Figure 2.6.: The deuteration process of four peptide sequences. The peak group for each sequence shifts not only along the m/z axis but also along the retention time axis. But the sequential ordering between different peak groups remains consistent across different time steps.

One limitation of the proposed approach lies in the correspondence estimation (i.e. deuteration tracking), a local search method subject to the variation in LC retention time. An additional alignment step as pre-processing [85] can reduce the tracking errors but does not solve the substantial problem .

We made one observation that motivates an important improvement that jointly addresses the alignment and tracking problem. Fig. 2.6 shows the deuteration process of four peptide sequences in a two-dimensional LC/MS space. The key observation is that, no matter how non-linearly the retention time profile is, the sequence ordering of the peptide sequences in each respective time step remains consistent. That is, the ordering the four peptide sequences in time step T_1 ($i \to ii \to iii \to iv$) is consistent with their ordering in T_2 ($I \to II \to III \to IV$) and T_3 ($A \to B \to C \to D$), even though their exact retention time has a very non-linear correspondence.

Therefore, we can exploit this sequential ordering prior to jointly solve the alignment problem (i.e. variations in the retention time) and the deuteration tracking problem (i.e. shift in the m/z) using a single energy minimization formulation. Formally, we take a pair of time steps (e.g. T_1 and T_2 in Fig. 2.6) and let i, j and i', j' index the peak groups (cf. Section 2.2.1) from the first and the second time step, respectively. The energy minimization problem is as follows:

$$\begin{aligned} \hat{\boldsymbol{x}} \;\; = \;\; & \mathbf{arg\ min}_{\boldsymbol{x}} \quad \left\{ \sum_{i,i'} D_{ii'} \cdot x_{ii'} + \sum_{i,i'} \sum_{j,j'} R_{ii',jj'} \cdot x_{ii'} \cdot x_{jj'} \right\} \\ & \mathbf{subject\ to} \quad \forall i, \sum_{i'} x_{ii'} = 1 \text{ and } \forall i', \sum_{i} x_{ii'} = 1. \end{aligned} \tag{2.6}$$

Here, $x_{ii'}$ (and $x_{jj'}$) is a binary random variable which is true when peak group i from time step T_1 corresponds to peak group i' from time step T_2. The two constraints guarantee that one peak group can be only associated with a single peak group from the other time step. We elaborate the energy terms $D_{ii'}$ and $R_{ii',jj'}$ in the following.

First, the term $D_{ii'}$ represents the cost on associating peak group i with i':

$$D_{ii'} = \lambda_\beta \|\beta_i - \beta_{i'}\|^2 + \lambda_\rho \mathbf{dist}(\rho_i, \rho_{i'}, z_i). \tag{2.7}$$

Here, β represents the sum of abundance and $\|\beta_i - \beta_{i'}\|^2$ expresses the assumption that the abundance of peptide sequence is stable across different time steps. The function $\mathbf{dist}(\rho_i, \rho_{i'}, z_i)$ computes the Earth mover's distance (EMD) [124] constrained with charge state z_i, namely the difference between ρ_i and $\rho_{i'}$ must roughly divide z_i.

Second, the term $R_{ii',jj'} = \lambda_t \left(1 - \delta(\mathbf{sgn}(t_i - t_j), \mathbf{sgn}(t_{i'} - t_{j'}))\right)$ penalizes any pair of associations that contradicts the sequential ordering prior. Here, t is the retention time of the peptide sequence, $\mathbf{sgn}(a)$ returns the sign of a and $\delta(a, b)$ is the Kronecker delta function which returns one when a and b equal. Intuitively speaking, $R_{ii',jj'}$ is zero if two associations fulfil the sequential ordering prior, namely peak group i is behind (or before) j in T_1 indicates that peak group i' shall also be behind (or before) j' in T_2.

Appendix

2b. Unsupervised Peak Shape Function Learning

It is not feasible to directly compare the acquired real-world data and the theoretical isotopic distributions unless an instrument-specific peak shape is incorporated. From a signal processing point of view, this involves convolution of the theoretical isotopic distribution with an instrument-dependent peak shape function (PSF, also known as the *aperture* or *point spread function*). A Gaussian distribution is used to approximate the real-world spectrum s with a specific peak shape, as

$$s \sim \mathcal{N}(\mu, \sigma^2)$$

where the mean μ expresses the central m/z position of the ion signal and the variance σ^2 describes the deviation from the center. For a given peptide sequence p and charge state z, μ is merely the m/z position of the theoretical isotopic peak. But σ^2 has to be estimated appropriately because normally it has a non-linear correlation with the m/z position of the isotopic peak (denoted as m). For the TOF (time-of-flight) analyzer, the model that expresses the non-linearity is given in equation 2.8.

$$\sigma(m) = a\sqrt{m} + b \tag{2.8}$$

To learn the parametrization of a and b, we first sample the mean $\hat{\mu}$ and the variance $\hat{\sigma}^2$ from high quality spectra with high-abundance along the entire m/z dimension using maximum likelihood estimation:

$$\hat{\mu} = \frac{\sum_{i=1}^{N} m_i s_i}{\sum_{i=1}^{N} s_i}, \tag{2.9}$$

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{N} s_i - 1}{\sum_{i=1}^{N} s_i} \bar{\sigma}^2, \tag{2.10}$$

where the spectrum is denoted as $\{m_i, s_i\}_{i=1}^{N}$, and $\bar{\sigma}^2$ is the biased estimation

$$\bar{\sigma}^2 = \frac{\sum_{i=1}^{N} (m_i - \hat{\mu})^2 s_i}{\sum_{i=1}^{N} s_i}.$$

Because of overlapping isotope patterns and noise, the sample set normally contains samples from overlapping and distorted spectra, resulting in outliers (e.g. m/z range $[800, 1000]$ in Figure 2.7, left). We then use Random Sample Consensus (RANSAC) algorithm to detect those outliers and use least-squares regression to extract the optimal parametrization. Result of the above mentioned non-linear robust regression is shown in Figure 2.7 (right).

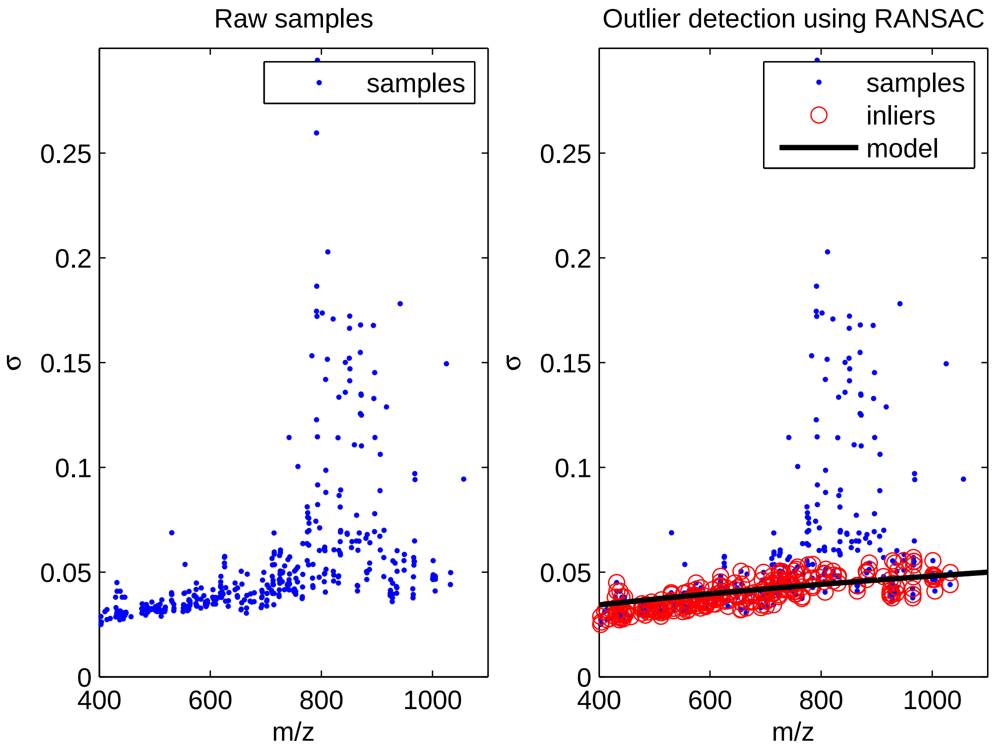


Figure 2.7.: PSF parameter extraction using RANSAC. The samples are shown on the left where outliers exist. The RANSAC algorithm provides a robust estimation where the outliers are successfully excluded from the regression.

2c. HX Kinetic Model Function Fitting: Convexity and Linearity

The fitting of estimated average deuteration $\{\tau_i, \eta_i\}_{i=1}^{T}$ to the HX kinetic model function $f_{\mathcal{A},\kappa}(\tau)$ is formulated as a non-linear regression problem:

$$\{\mathcal{A}, \kappa\} \quad = \quad \underset{\mathcal{A},\kappa}{\arg\min} \left\{ \textstyle\sum_{i=1}^{T} \left[\mathcal{A}(1 - e^{-\kappa\tau_i}) - \eta_i\right]^2 \right\} \tag{2.11}$$

where T is the total number of incubation time points, $\mathcal{A}$ is the total amount of exchangeable hydrogens, and κ is the fused exchange rate. Denoting the object function in Eq. 2.11 as $f(\mathcal{A}, \kappa)$, we now give the proof that $f(\mathcal{A}, \kappa)$ is non-linear and non-convex:

Proof.

First of all, the non-linearity of $f(\mathcal{A}, \kappa)$ is obvious: the parameter $\mathcal{A}$ and κ are not linearly combined.

Secondly, the second-order conditions theorem tells that the object function is convex if and only if **dom** $f(\mathcal{A},\kappa)$ (domain of the object function) is convex and its Hessian $\nabla^2 f(\mathcal{A},\kappa)$ is positive-semidefinite: $\forall \mathcal{A},\kappa \in \mathbf{dom}\ f(\mathcal{A},\kappa)$, $\nabla^2 f(\mathcal{A},\kappa) \succeq 0$. The domain of $f(\mathcal{A},\kappa)$ is convex, as

$$\mathbf{dom} f(\mathcal{A},\kappa) = \{\mathcal{A},\kappa | \mathcal{A} \geq 0, \kappa \geq 0, f(\mathcal{A},\kappa) < \infty\} \tag{2.12}$$

Computing the second derivatives of $f(\mathcal{A},\kappa)$, we have the Hessian matrix

$$H = \nabla^2 f(\mathcal{A},\kappa) = \left[\begin{array}{cc} \frac{\partial^2 f(\mathcal{A},\kappa)}{\partial \mathcal{A}^2} & \frac{\partial^2 f(\mathcal{A},\kappa)}{\partial \mathcal{A}\partial\kappa} \\ \frac{\partial^2 f(\mathcal{A},\kappa)}{\partial \mathcal{A}\partial\kappa} & \frac{\partial^2 f(\mathcal{A},\kappa)}{\partial \kappa^2} \end{array} \right] \tag{2.13}$$

where

$$\frac{\partial^2 f(\mathcal{A},\kappa)}{\partial \mathcal{A}^2} = \sum_{i=1}^{T} 2\beta_i^2$$

$$\frac{\partial^2 f(\mathcal{A},\kappa)}{\partial \mathcal{A}\partial\kappa} = \sum_{i=1}^{T} 2\tau_i\alpha_i(2\mathcal{A}\beta_i - \eta_i)$$

$$\frac{\partial^2 f(\mathcal{A},\kappa)}{\partial \kappa^2} = \sum_{i=1}^{T} 2\mathcal{A}\tau_i^2\alpha_i(\mathcal{A}\alpha_i + \eta_i - \mathcal{A}\beta_i)$$

and $\alpha_i = e^{-\kappa\tau_i}$ and $\beta_i = 1 - \alpha_i$ are introduced to simplify the notations. If $f(\mathcal{A},\kappa)$ is convex, the Hessian has to be positive-semidefinite, namely its diagonal entries must be real and non-negative. For $H_{1,1} = \sum_{i=1}^{T} 2\beta_i^2$, the condition is satisfied. For $H_{2,2} = \sum_{i=1}^{T} 2\mathcal{A}\tau_i^2\alpha_i(\mathcal{A}\alpha_i + \eta_i - \mathcal{A}\beta_i)$, the condition holds if and only if $\forall \mathcal{A},\kappa \in \mathbf{dom}\ f(\mathcal{A},\kappa)$

$$\sum_{i=1}^{T} 2\mathcal{A}\tau_i^2\alpha_i(\mathcal{A}\alpha_i + \eta_i - \mathcal{A}\beta_i) \geq 0 \tag{2.14}$$

Inserting the definition of α_i and β_i, Eq. 2.14 equals to

$$\sum_{i=1}^{T} (\mathcal{A}\alpha_i + \eta_i - \mathcal{A}\beta_i) \geq 0 \tag{2.15}$$

$$\sum_{i=1}^{T} \left(\mathcal{A}(2e^{-\kappa\tau_i} - 1) + \eta_i\right) \geq 0 \tag{2.16}$$

$$\sum_{i=1}^{T} \mathcal{A}\left(1 - 2e^{-\kappa\tau_i}\right) \leq \sum_{i=1}^{T} \eta_i \tag{2.17}$$

Taking the parameter $\hat{\mathcal{A}} > \mathbf{max}_{i\in\{1\ldots T\}}(\eta_i)$ and $\hat{\kappa} \leftarrow \infty$, we have $\hat{\mathcal{A}},\hat{\kappa} \in \mathbf{dom}\ f(\mathcal{A},\kappa)$

but

$$\sum_{i=1}^{T} \hat{\mathcal{A}}(1 - 2e^{-\hat{\kappa}\tau_i}) \approx \sum_{i=1}^{T} \hat{\mathcal{A}} > \sum_{i=1}^{T} \eta_i \qquad (2.18)$$

Therefore, Eq. 2.17 does not hold for certain parameter setting $\hat{\mathcal{A}}$ and $\hat{\kappa} \in \mathbf{dom}\, f(\mathcal{A}, \kappa)$, indicating that $H_{2,2} \not\geq 0$ and accordingly $\nabla^2 f(\mathcal{A}, \kappa) \not\succeq 0$. The object function $f(\mathcal{A}, \kappa)$ is not convex.

End of proof. □

2d. Pesudocode of the HeXicon Workflow

Algorithm No.	Description	In Fig. 2.2
1	LC/MS Segmentation and Peak Picking Routine	**B, C**
2	Peptide Sequence Candidate List Formulation	**A**
3	Correspondence Estimation Routine	**D**
4	Quality Estimation Routine	**F, G**

Table 2.4.: Pseudo code for the HeXicon algorithmic workflow

2e. HeXicon Vs. Manual Analysis

Table 2.5 and Fig. 2.8 show the comparison between HeXicon and manual analysis for the CHIP dataset. Table 2.6 and Fig. 2.9 show the comparison between HeXicon and manual analysis for the HtpG dataset.

Algorithm 1: LC/MS segmentation and peak picking routine.

```
Input: LC/MS spectrum data $\boldsymbol{D}$; List of peptides of interest $\boldsymbol{P}$.
Output: List of picked peaks $\boldsymbol{\mathcal{G}}$ with features.
// Data partition
$\boldsymbol{S} \leftarrow$ Watershed_Segmentation($\boldsymbol{D}$)
// Peak list initialization
$\boldsymbol{\mathcal{G}} \leftarrow \emptyset$
foreach $S \in \boldsymbol{S}$ do
    // Integration of LC/MS signals
    $\boldsymbol{s} \leftarrow$ LCMS_Data_Integration($S$)
    if $\boldsymbol{P} \equiv \varnothing$ then
        // Basis function construction using the Averagine model
        $\boldsymbol{\Phi} \leftarrow$ Averagine_Basis_Function_Construction($\boldsymbol{s}$)
    else
        // Basis function construction using the given peptide
           sequences
        $\boldsymbol{\Phi} \leftarrow$ Exact_Basis_Function_Construction($\boldsymbol{s}, \boldsymbol{P}$)
    end
    // Regression with $L1$-regularization
    $\hat{\boldsymbol{\beta}} \leftarrow \arg\min_{\boldsymbol{\beta}} \left\{ \left\{ ||\boldsymbol{s} - \boldsymbol{\Phi}\boldsymbol{\beta}||_2^2 + \lambda \, ||\boldsymbol{\beta}||_1 \right\} \right\}$, subject to $\boldsymbol{\beta} \geq 0$
    // Selecting basis functions with non-zero coefficients
    $\{\hat{\boldsymbol{m}}, \hat{\boldsymbol{z}}\} \leftarrow$ Basis_Function_Selection($\boldsymbol{\Phi}, \hat{\boldsymbol{\beta}} > 0$)
    // Sparse elution profile reconstruction
    $\hat{\boldsymbol{t}} \leftarrow$ Sparse_Elution_Profile_Reconstruction($S, \boldsymbol{\Phi}, \hat{\boldsymbol{\beta}} > 0$)
    // Saving picked peaks
    $\boldsymbol{\mathcal{G}} \leftarrow \boldsymbol{\mathcal{G}} \cup \left[ \hat{\boldsymbol{m}}, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{z}}, \hat{\boldsymbol{t}} \right]$
end
return $\boldsymbol{\mathcal{G}}$
```

Algorithm 2: Peptide sequence candidate list formulation.

```
Input: List of peaks $\boldsymbol{\mathcal{G}}$; list of peptides identified by MS/MS data $\boldsymbol{P}^{\mathrm{MS2}}$; protein
       sequence $\boldsymbol{l}$; mass difference threshold $\Delta$.
Output: List of peptide sequence candidates $\boldsymbol{P}$.
// Initializing the list of peptide sequence candidates
$\boldsymbol{P} \leftarrow \emptyset$
foreach $\boldsymbol{g} = \{m, \beta, z, t\} \in \boldsymbol{\mathcal{G}}$ do
    if $\exists p \in \boldsymbol{P}^{\mathrm{MS2}}, |\mathrm{Mass\_Of}(p) - \mathrm{Mass\_Of}(m, z)| < \Delta$ then
        // Adding the MS2 identified peptide
        $\boldsymbol{P} \leftarrow \boldsymbol{P} \cup p$
    else
        // Searching the protein sequence
        $p \leftarrow \mathrm{Protein\_Sequence\_Searching}(\boldsymbol{l}, m, z, \Delta)$
        $\boldsymbol{P} \leftarrow \boldsymbol{P} \cup p$
    end
end
return $\boldsymbol{P}$
```

Algorithm 3: Quality estimation routine.

```
Input: Peptide sequence candidate list $\boldsymbol{P}$; correspondence set $\boldsymbol{C}$; trained
       Random Forest quality classifier $\mathcal{F}$.
Output: Peptide sequence candidate list $\boldsymbol{P}$.
// Initializing the quality set
$\boldsymbol{q} \leftarrow \emptyset$
foreach $p \in \boldsymbol{P}$ do
    // Quality feature extraction
    $\boldsymbol{\mathcal{X}} \leftarrow \mathrm{Quality\_Feature\_Extraction}(p, \boldsymbol{C})$
    // Quality classification
    $q \leftarrow \mathrm{Random\_Forest\_Classification}(\mathcal{F}, \boldsymbol{\mathcal{X}})$
    // Saving the quality score
    $\boldsymbol{q} \leftarrow \boldsymbol{q} \cup q$
end
// Sorting the results
$\boldsymbol{P} \leftarrow \mathrm{Sorting\_By}(\boldsymbol{P}, \boldsymbol{q}, '\mathrm{descending}')$
return $\boldsymbol{P}$
```

Algorithm 4: Correspondence estimation routine.

```
Input: Peptide sequence candidate list $\boldsymbol{P}$; peak list series $\{\boldsymbol{\mathcal{G}}_0, \boldsymbol{\mathcal{G}}_1, \ldots, \boldsymbol{\mathcal{G}}_T\}$;
       Euclidean distance weighting matrix $\boldsymbol{S}$.
Output: Set of correspondence of peak groups.
// Initializing the correspondence set
$\boldsymbol{C} \leftarrow \emptyset$
foreach $p \in \boldsymbol{P}$ do
    // Determining the zero exchange peak group
    $\hat{\boldsymbol{g}}_0 \leftarrow \underset{\boldsymbol{g}=\{m,\beta,z,t\}\in\boldsymbol{\mathcal{G}}_0}{\arg\min} \{|\text{Mass\_Of}(p) - \text{Mass\_Of}(m, z)|\}$
    // Initializing the correspondence vector
    $\boldsymbol{c} \leftarrow [\hat{\boldsymbol{g}}_0]$
    // Iteratively determining the peak groups from other timepoints
    foreach $j \in \{1, 2, \ldots, T\}$ do
        // Weighted Euclidean distance measure
        $\hat{\boldsymbol{g}}_j \leftarrow \underset{\boldsymbol{g}=\{\boldsymbol{m},\boldsymbol{\beta},\boldsymbol{z},\boldsymbol{t}\}\in\boldsymbol{\mathcal{G}}_j}{\arg\min} \left\{\sqrt{(\boldsymbol{g} - \hat{\boldsymbol{g}}_0)'\boldsymbol{S}^{-1}(\boldsymbol{g} - \hat{\boldsymbol{g}}_0)}\right\}$
        if $\hat{\boldsymbol{g}}_j \equiv \varnothing$ then
            // Missing peptide at this timepoint
            break;
        else
            // Adding to the correspondence vector
            $\boldsymbol{c} \leftarrow [\boldsymbol{c}, \hat{\boldsymbol{g}}_j]$
        end
    end
    if length_of$(c) \equiv T + 1$ then
        // Adding to the correspondence set
        $\boldsymbol{C} \leftarrow \boldsymbol{C} \cup \boldsymbol{c}$
    end
end
return $\boldsymbol{C}$
```

Table 2.5.: Manual (top row) Vs. HeXicon(bottom row) on the CHIP dataset. The HeXicon results use monoisotope mass; the manual results use the average mass.

Peptide	z	0 s	30 s	300 s	3600 s	κ	Δ_m	Δ_κ
AKHDKYMADM	2	605.63	605.82	606.29	607.15	6.98	0.17	0.28
		605.28	605.43	605.80	606.96	5.01		
CIEAKHDKYMADM	3	519.21	519.91	520.42	520.95	19.88	0.11	0.20
		518.90	519.54	520.04	520.65	15.92		
DGQSVKAHF	2	494.97	496.35	497.26	497.28	109.32	0.10	0.14
		494.75	496.11	497.05	497.23	94.44		
DGQSVKAHFFLGQCQL	3	593.70	594.62	595.75	597.20	12.10	0.32	0.00
		593.30	594.38	595.20	596.67	12.14		
ELMREPCITPSGITYDRKDIEE	3	866.27	867.23	867.98	868.84	17.10	0.22	0.18
		865.77	866.64	867.32	868.26	13.95		
FSQVDEKRKKRDIPDYLCG	3	766.78	769.67	770.14	770.23	211.44	0.31	0.01
		766.40	769.20	769.55	769.74	208.93		
FSQVDEKRKKRDIPDYLCGKISFE	3	968.47	972.23	972.78	970.83	237.39	0.29	0.16
		967.83	971.40	972.19	970.05	199.88		
FVGRKYPEAAAC	2	656.62	659.91	660.14	659.98	290.29	0.11	0.06
		656.33	659.58	659.71	659.65	309.14		
HLQRVGHFDPVTRSPLTQE	3	740.11	741.24	742.56	743.47	20.05	0.14	0.06
		739.73	740.73	742.15	743.06	18.77		
HSYLSRL	2	438.43	438.53	438.71	438.99	8.98	0.10	0.22
		438.24	438.29	438.46	438.74	6.97		
IAAERERELEE	2	673.14	673.52	674.23	675.32	8.92	0.28	0.11
		672.85	673.07	673.77	674.79	7.94		
IAAERERELEECQRNHEGDEDDSHVRAQQA	4	881.39	883.93	884.48	884.97	145.72	0.15	0.01
		880.91	883.39	883.97	884.41	144.52		
ISENGWVEDY	2	606.61	608.69	609.34	609.21	170.42	0.08	0.06
		606.27	608.38	608.95	608.78	181.87		
KEVIDAF	2	411.46	411.45	411.44	411.67	1.00	0.17	0.00
		411.23	411.22	411.22	411.78	1.00		
MESYDEAIANL	2	628.60	629.74	630.40	632.05	10.03	0.06	0.02
		628.28	629.36	630.04	631.73	9.88		
MKGKEEKEGGARLGAGGGSPEKSPSAQE	3	925.36	929.15	929.51	929.75	225.54	0.11	0.04
		924.83	928.53	928.95	929.19	216.02		
MKGKEEKEGGARLGAGGGSPEKSPSAQELKEQGNRL	4	928.96	934.59	934.70	934.47	334.26	0.15	0.00
		928.49	934.06	934.16	933.99	335.29		
QRAYSLAKEQRLNFGDDIPSAL	3	831.91	834.73	835.91	836.41	116.73	0.22	0.12
		831.43	834.33	835.39	835.76	132.05		
RIAKKKRWNSIE	3	510.63	512.64	513.08	512.97	200.09	0.28	0.15
		510.31	512.52	512.85	512.73	234.72		
RRALELDGQSVKAHF	3	576.66	577.88	578.86	579.78	20.02	0.30	0.20
		576.31	577.32	578.37	579.38	15.93		
VYYTNRALC	2	552.10	552.96	554.80	555.04	37.63	0.14	0.11
		551.78	552.74	554.34	554.67	42.12		

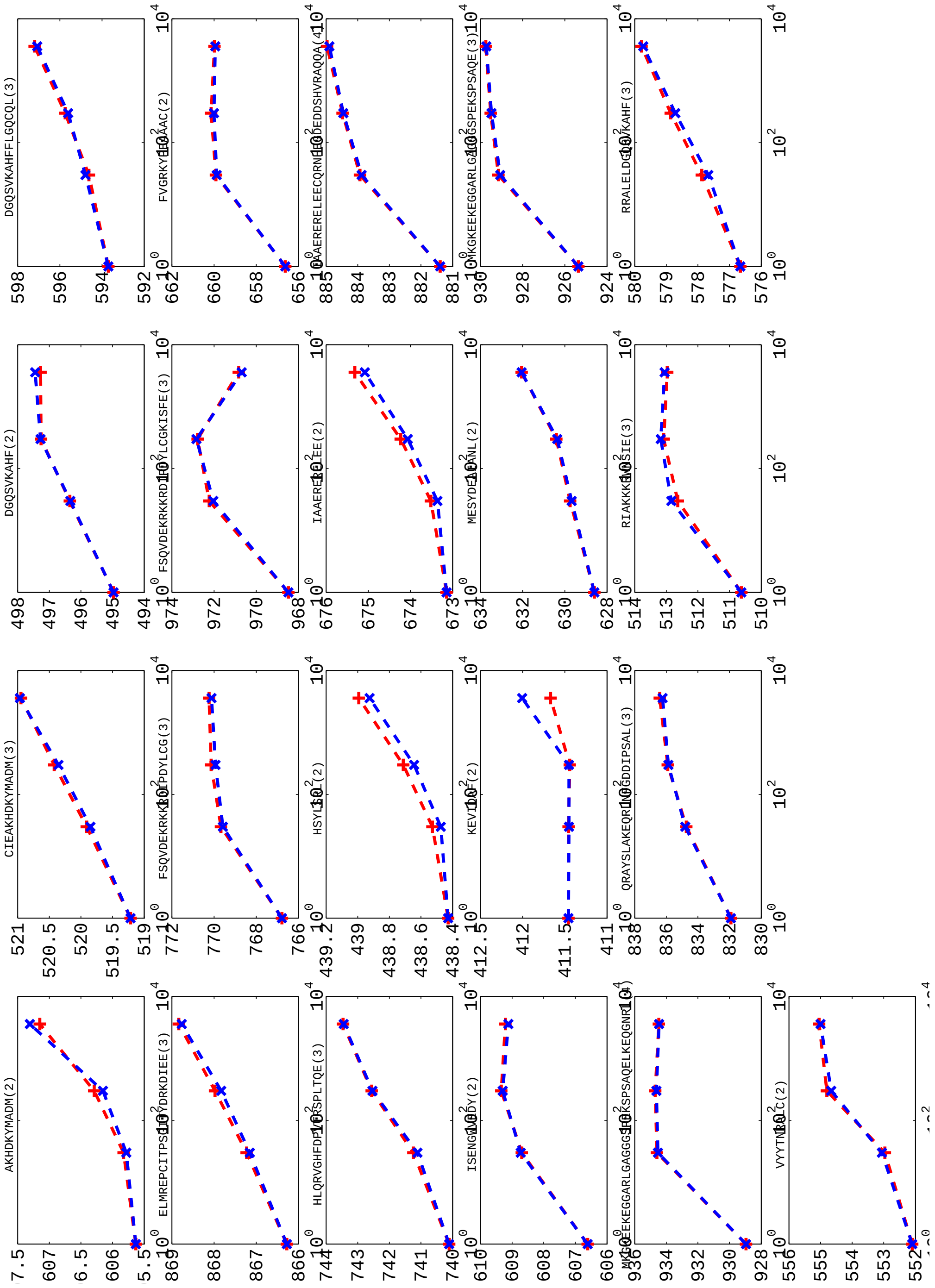


Figure 2.8.: Comparison of the average deuteration estimation for the CHIP dataset: Manual (red) Vs. HeXicon (blue).

Table 2.6.: Manual (top row) Vs. HeXicon(bottom row) on the HtpG dataset.

Peptide	z	0 s	30 s	300 s	3600 s	κ	Δ_m	Δ_κ
DAEQFMPNY	2	558.07	558.76	559.12	559.24	103.20	0.09	0.12
		557.73	558.50	558.81	558.95	116.93		
DITKEDRGTEITL	2	746.41	749.78	749.89	749.93	306.50	0.32	0.09
		745.84	749.36	749.55	749.62	280.13		
EDYVSRMKEGQEKIYYITA	3	775.56	776.21	777.36	777.60	38.02	0.11	0.08
		775.06	775.75	776.80	777.14	35.03		
EKLAKDDAEKYQTF	3	562.97	563.43	564.26	564.42	38.02	N/A	N/A
		N/A	N/A	N/A	N/A	N/A		
ESLGSDQAKDSQL	2	689.72	692.68	692.83	692.84	293.63	0.12	0.13
		689.33	692.12	692.38	692.43	254.90		
EVIDHLGTIA	2	534.83	536.30	536.48	536.60	203.46	0.10	0.04
		534.29	535.82	535.97	536.17	195.84		
FAAAGQKVPEVKY	3	470.24	472.51	472.85	472.83	234.41	0.23	0.02
		469.93	472.12	472.41	472.42	238.12		
HLREGEDEF	2	566.56	567.30	567.81	567.82	105.60	0.06	0.10
		566.26	567.02	567.46	567.47	116.97		
IFELNPDHVL	2	599.14	599.56	599.68	599.78	128.28	0.02	0.13
		598.82	599.25	599.37	599.42	148.11		
IGQFGVGF	1	824.80	827.88	828.69	828.62	181.90	0.79	0.40
		824.44	826.92	826.99	827.04	304.93		
IRRMNQL	2	466.05	466.23	466.48	466.93	8.94	0.06	0.22
		465.77	465.94	466.13	466.63	6.99		
IVADKVTVRTRAAGEKPENGVF	3	786.86	787.60	787.93	788.29	87.41	0.12	0.01
		786.44	787.14	787.51	787.77	86.69		
KEGPAEDFANQ	2	603.59	604.42	604.99	605.27	80.47	0.16	0.25
		603.28	604.25	604.55	604.91	106.72		
KSGTKSFL	2	434.65	435.23	436.04	436.11	60.28	0.22	0.29
		434.23	435.06	435.63	435.86	84.43		
LAERGTLEDPNL	2	664.69	665.75	666.90	667.18	64.91	0.19	0.25
		664.35	665.59	666.47	666.71	86.94		
LISNASDAADKLRFRALSNPDL	3	796.86	797.59	798.79	799.29	23.90	0.13	0.16
		796.44	797.13	798.35	798.97	19.96		
LLSDRIDE	2	481.05	481.69	481.95	482.09	111.36	0.02	0.06
		480.75	481.37	481.67	481.81	105.20		
LNPDHVLVKRAADTED	3	598.66	599.38	599.58	599.83	112.90	0.10	0.04
		598.32	599.02	599.21	599.43	117.79		
LRELISNASDAADKLRF	3	640.75	640.81	641.09	641.52	7.07	1.06	0.63
		640.36	640.51	641.44	641.71	18.95		
LRFASTHTDSSAQTVSL	3	608.00	609.01	609.69	609.87	7.07	N/A	N/A
		N/A	N/A	N/A	N/A	N/A		
LRKKGIEVL	2	528.66	528.84	529.55	529.82	18.12	0.17	0.01
		528.28	528.46	529.32	529.63	17.95		
LTEFDGKPFQSVSKVDESL	3	709.80	711.94	712.17	712.20	252.94	0.12	0.01
		709.37	711.48	711.64	711.72	255.62		
LYIPSQAPWDMWNRDHKHGLKL	3	903.00	904.91	905.54	905.66	150.54	0.32	0.15
		902.48	904.18	904.89	905.05	128.08		
MIHSLYSNKEIF	3	494.92	496.37	496.81	496.89	157.13	0.13	0.11
		494.59	495.92	496.46	496.53	139.55		
MKGQETRGFQSEVKQLLHL	3	744.15	747.92	748.27	748.25	264.85	0.05	0.03
		743.74	747.54	747.87	747.82	274.17		
PVEIEKREEKDGE	3	520.25	521.11	521.44	521.65	112.83	0.10	0.23
		519.93	520.91	521.13	521.32	146.21		
RFVRGLIDSSDLPLNV	3	601.36	602.40	602.82	602.82	146.98	0.64	0.26
		601.01	602.51	602.83	602.44	199.33		
RVKDVRLTHRLTDTPAIVST	3	760.66	761.96	762.56	762.86	146.98	N/A	N/A
		N/A	N/A	N/A	N/A	N/A		
RVRVSFDKDKRTLTISDNGVGMTRD	4	717.83	718.29	718.80	718.98	58.52	0.23	0.34
		717.38	717.94	718.30	718.45	89.08		
SREILQDSTVTRNLRNALTKRVLQM	3	982.24	985.28	986.32	986.47	150.05	0.10	0.03
		981.57	984.61	985.66	985.90	145.38		
TVISWEKINKAQAL	3	534.68	536.85	537.46	537.63	156.34	0.22	0.03
		534.32	536.59	537.19	537.35	161.59		
VRLTHRLTDTPAIVSTDADE	3	737.87	739.70	740.00	740.18	183.61	0.15	0.08
		737.39	739.10	739.50	739.63	168.61		
WESAGEGEYTVA	2	650.20	651.50	652.19	652.43	105.65	0.22	0.14
		649.79	651.30	651.88	652.15	122.99		
WQQFGLVL	2	496.04	496.10	496.41	496.43	26.19	0.17	0.68
		495.78	495.92	496.02	496.06	81.63		
WRVRSIISKYSDHIAL	3	649.07	650.71	651.86	651.96	100.09	0.24	0.02
		648.70	650.28	651.35	651.50	97.99		
YAAAKSSPHLEL	3	429.91	430.71	431.55	431.58	76.25	0.27	0.22
		429.56	430.55	431.27	431.32	98.10		
YKEFYKHIAHDFNDPLTWSHNRVEGKQEYTSL	4	989.60	991.08	991.83	992.07	109.49	0.15	0.11
		989.01	990.61	991.25	991.49	122.99		
YTVADITKEDRGTE	3	533.63	534.27	534.86	534.91	81.36	0.17	0.30
		533.27	534.07	534.53	534.56	115.99		
YVQRVF	2	406.48	406.53	407.23	407.51	14.87	0.05	0.06
		406.23	406.36	406.97	407.25	15.82		

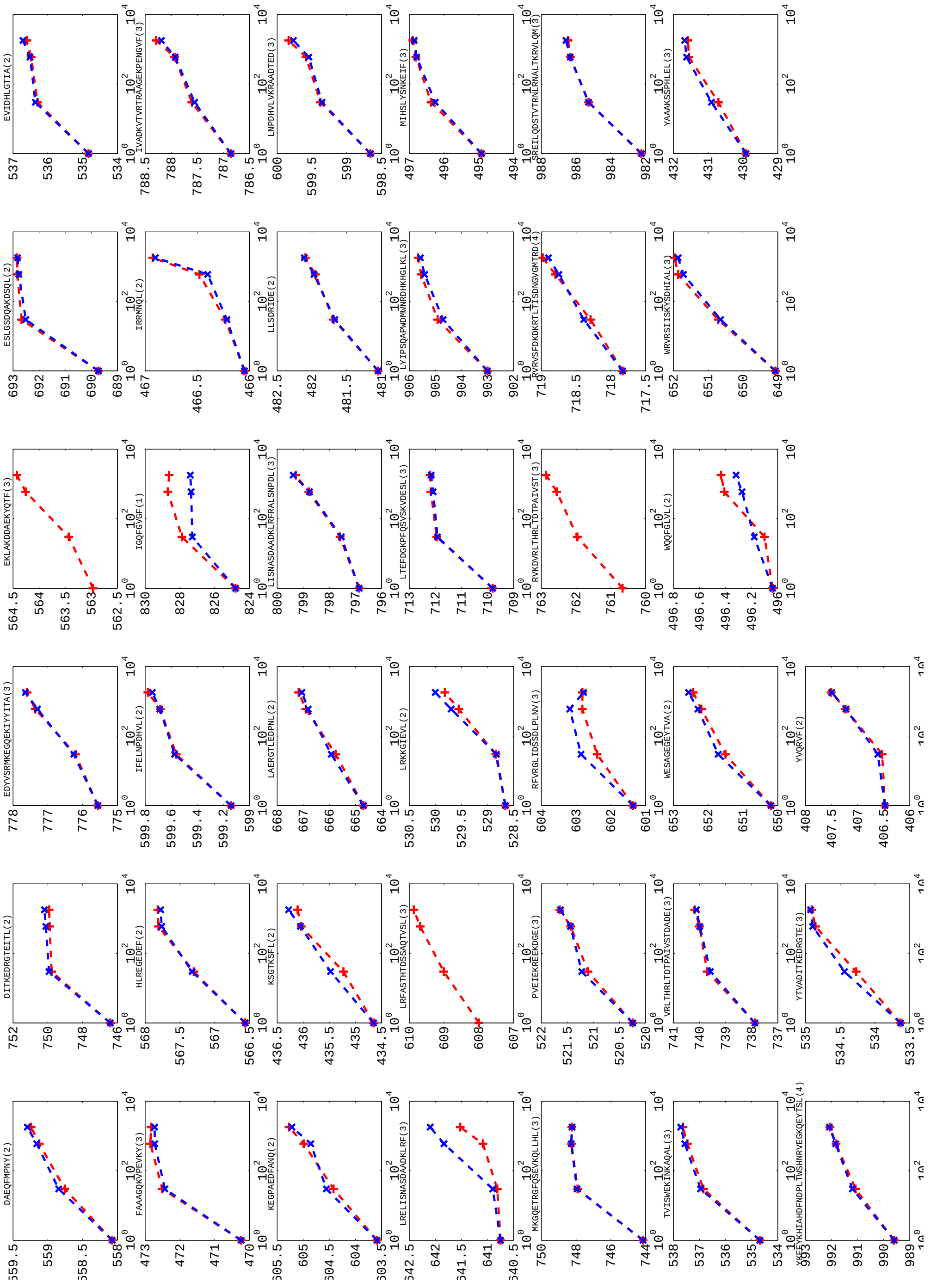


Figure 2.9.: Comparison of the average deuteration estimation for the HtpG dataset: Manual (red) Vs. HeXicon (blue).

Chapter 3

Markov Random Field with Shape Prior for Cell Nucleus Segmentation

This chapter studies the strategy of incorporating a shape prior to the cell nucleus segmentation problem. We extended the classic MRF energy formulation with two novel energy terms (thus, constrained modeling) that represent the shape and the length of segmented objects. This method has been evaluated on both 2D and 3D image data.

Accurate and automated segmentation of numerous cell nuclei from *in vivo* fluorescent microscope image data is critical for quantitative and high-throughput analysis of cell behaviors. This is a challenging problem due to uneven intensity distribution, large variability of appearance, and due to background disturbance as well as severe nuclei clutter. While manifold contributions have been made to this problem [110, 89, 88, 93, 34, 42, 156, 137], the underlying methods lack guarantee of optimality and the results have not been sufficiently evaluated with respect to the shape regularity. Poorly segmented cell nuclei not only hinder visual inspection and evaluation but also jeopardize downstream tasks such as growth phase classification and tracking. In this work, we propose *NuCut*, a novel cell *Nu*cleus segmentation method that captures a variety of important visual cues from texture to shape into a Markov Random Field (MRF) and performs the inference using Graph *Cut* to obtain global optimality. In particular, we extend the Graph Cut framework with a prior for simultaneously shape regularization of multiple cell nuclei. Unlike [156, 108], our method only requires a single nucleus image channel (such as GFP or Hoechst staining), allowing for better temporal resolution during image acquisition. Extensive experiments and evaluation on 2D and 3D data show qualitative and substantial quantitative improvement over a few recent segmentation methods. A few ideas in this article such as the object detection by multi-scale coherence and the adaptive multi-object shape prior may carry over to other applications.

3.1. Introduction

The rapid progress of microscopic imaging techniques creates compelling challenges for the life science image processing community [117]. The vast amount of data with increased complexity prohibits any manual analysis and raises a demand for incorporating state-of-the-art machine learning and optimization techniques into the analysis pipeline [155]. This is particularly true for many modern *in vivo* imaging experiments in developmental biology [103]. For example, the progress in [76, 108, 75] allows us to envision the automated extraction of full lineage trees for more complicated animals such as zebrafish and Drosophila. Such *digital embryo* databases have a far reaching impact on the field of developmental biology.

Sophisticated methods are being developed to process such complicated datasets. Segmentation, prior to tracking or growth phase classification, is a fundamental part of the overall processing pipeline. However, as shown in Fig. 3.1, this is a challenging task due to severe nuclei clutter and weak boundaries (A, B), limited image quality and spatial resolution (C), uneven intensity distribution (D, E), and due to large variability in brightness, size and texture (A, B). Many recent nucleus segmentation methods based on traditional image processing techniques lack sufficient guarantee of the optimality of the results and they usually do not consider the shape regularity of the nuclei. Segmented cell nuclei with misleading shape not only hinder visual inspection and evaluation but also imperil further tasks in the processing pipeline. In this work, we propose an automated method that captures several useful visual cues into a MRF and extends the Graph Cut framework with a multi-object shape prior. The main contributions of this work are:

- A cell nucleus detection algorithm based on multi-scale coherence of the eigensystem of the Hessian matrix.

- An automated label generation scheme for pixel classification to avoid the considerable amount of efforts on manual labeling.

- An extension of the Graph Cut framework that leverages a shape prior for multiple objects simultaneously by incorporating a gradient vector field (GVF). A function distance transform is introduced to eliminate the interference of gradient vectors from different sources by adaptively adjusting their range of influence.

- We conduct extensive experiments and evaluation on 2D and 3D datasets and thoroughly compare to other state-of-the-art methods.

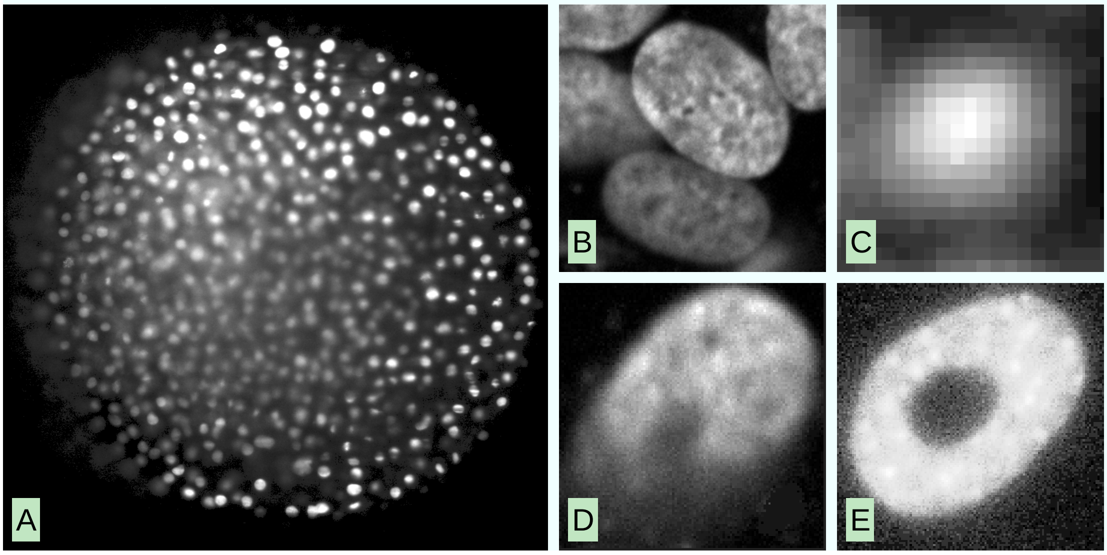


Figure 3.1.: Examples of cell nuclei images. Cell nuclei normally exhibit a variety of intensity, size and texture (A, B). The boundary may be weak due to cell nuclei clustering (B) or blurring (C). Also, uneven intensity distribution within a nucleus may cause unfavorable segmentation such as gulf (D) or hole (E). Image A and C are from the digital embryo dataset [76] and image B, D and E are from the hand-labeled benchmark [37]. See Section 3.4 for more details. The contrast of the images has been enhanced for visualization purpose.

The paper is organized as follows. In Section 4.2, we summarize related work on cell nucleus detection and segmentation, Markov Random Field, and the Graph Cut algorithm. Section 5.2 presents our method in detail. Section 3.4 describes the evaluation setup, including the datasets, the measures and the methods for comparison, followed by the results in Section 3.5. Finally, conclusions are offered in Section 3.6.

3.2. Related Work

3.2.1. Cell Nucleus Detection and Segmentation

Based on the underlying image processing concept they build on, we group the existing cell nucleus detection and segmentation methods into four categories as follows. The principal difference between detection and segmentation is that detection mostly locates the cell nuclei while segmentation also attempts to extract their boundaries as accurately as possible. In some approaches (including this work), detection is a coarse processing step before the segmentation.

Intensity thresholding based methods such as [110] consider supra-threshold contiguous regions as objects. They rarely consider spatial context and are sensitive to texture and noise. The authors in [76] used a local adaptive intensity thresholding method that handles under-segmentation well but only identifies local maxima as the segmented objects. This amounts to detection rather than segmentation and does not reveal the true extent of cell nuclei. Depending on the workflow, this may raise difficulties in further analysis.

The *Watershed* algorithm [123] has the advantage of speed but is known to yield severe over-segmentation unless an *ad hoc* merging operation is incorporated for post-processing [93, 34]. However, it is not easy to formulate the merging criteria that satisfy all variability in the data such as size and texture. Also, classic Watershed produces loose boundaries that cover irrelevant background regions in addition to the true nucleus body.

The *gradient flow tracking* method [89, 87] performs the segmentation with a gradient diffusion procedure followed by gradient flow tracking and local adaptive thresholding. The diffusion technique can regulate weak, noisy gradient vectors but not those due to intensity irregularity inside the nucleus body. Also, diffusion may further weaken vague boundaries and thus cause possible under-segmentation.

Some methods based on *contour evolution* (or surface evolution in 3D) have been successfully applied to segmentation [137] as well as tracking [90], sometimes jointly. The contour can be represented explicitly or implicitly. In the implicit representation, the contour is defined as the zero level set of a scalar function whose evolution is solved using a partial differential equation (PDE). Additional constraints have been introduced to improve the segmentation results. For example, the authors in [42] introduced coupling into multiple contour representations as a penalty to avoid under-segmentation. The authors in [156] used Subjective Surfaces [127] to handle gaps in boundaries. However, these methods, especially the ones using implicit contour representation (i.e. Level Sets), normally have high computational complexity. Also, solving PDEs requires special care to achieve convergence and numerical stability. Finally, there is no guarantee of obtaining the global optimum of the target energy functions.

3.2.2. Markov Random Field and Graph Cut

Markov Random Field has been used to model a wide variety of low-level computer vision and image processing problems [91]. As shown in Eq. 3.1, a classic first-order MRF for an image $\boldsymbol{I}$ can be formulated as a pixel-based energy function which consists of two components: a data term representing the cost of assigning label $l_p \in \mathcal{L}$ to a pixel $p \in \boldsymbol{I}$ and a smoothing term as the cost of assigning labels $\{l_p, l_q\}$ to a pixel pair $\{p, q\}$ that belongs to a neighborhood system $\boldsymbol{N}$. The data term can be the negative log-likelihood of pixel intensity (or color) being consistent with statistics over the region it belongs to. This region statistics can be mean gray value, intensity histogram [21], color Gaussian mixture model [17], or even be implicitly represented as a pixel classifier trained on pixel-based features [74]. The smoothing term, measuring the discontinuity between pixel pairs in the neighborhood system, can be a L_1 or L_2 distance (maybe truncated) or a Potts model. We refer the readers to [24] for more details. Given non-negative parameters λ_{data} and λ_{smooth} that weight the contributions of those two terms, the goal is to find the best labeling $\boldsymbol{l}$ that minimizes the energy function $E(\boldsymbol{l})$.

$$E(\boldsymbol{l}) = \lambda_{\text{data}} \sum_{p \in \boldsymbol{I}} E_{\text{data}}(l_p) + \lambda_{\text{smooth}} \sum_{\{p,q\} \in \boldsymbol{N}} E_{\text{smooth}}(l_p, l_q) \tag{3.1}$$

Early methods for this problem fail due to local optimality (e.g. Iterated Conditional Modes [14]) or slow speed (e.g. simulated annealing [54]). The Graph Cut algorithm, proposed in [57] and popularized in [21, 24, 17, 20], can effectively solve a large variety of such problems and has been successfully applied to many applications. In particular, given a binary MRF problem on an image $\boldsymbol{I}$, Graph Cut formulates it as a graph partition task of an undirected graph $\boldsymbol{G} = \langle \boldsymbol{V}, \boldsymbol{E} \rangle$ of vertices $\boldsymbol{V}$ and edges $\boldsymbol{E}$. The set of the vertices $\boldsymbol{V} = \{\boldsymbol{I}, s, t\}$ consists of all pixels of $\boldsymbol{I}$ and two special vertices (also known as terminals) identified as the source s and the sink t. Normally two types of edges are created: N-links connect pixel pairs in the neighboring system and T-links connect each pixel to both the source and sink. The costs in Eq. 3.1 are now associated with the edges: data term on the T-links and smoothing term on the N-links. Thereby, minimizing $E(\boldsymbol{l})$ becomes equivalent to finding the optimal cut that partition all vertices into two disjoint sets with the minimal sum of costs of the edges it severs. Graph Cut can be extended to handle multi-label problem by using α-expansion or $\alpha\beta$-swap [24].

Graph Cut has some favorable properties. In comparison to Level Sets, the *max-flow/min-cut* algorithm [80] for solving such MRF problem is guaranteed to obtain a global optimum given a submodular energy function. Also, the algorithm is numerically stable in practice [23]. Finally, it allows to integrate a wide range of visual cues and constraints, e.g. the shape regularization in this work. In fact, the authors in [4] have applied Graph Cut to cell nucleus segmentation. In particular, by considering every cell nucleus as an individual class, they use α-expansion [24] to propagate labels from identified seeds to the entire cell nuclei and adopt a graph coloring scheme for acceleration. However, formulating the problem as a multi-label segmentation task is an exaggeration since intrinsically only a binary segmentation is expected. Also, such a formulation loses the very important global optimality. The α-expansion solver requires multiple runs of the max-flow algorithm and thus increases the computational efforts.

3.2.3. Shape Prior for Graph Cut Segmentation

There are several methods that incorporate shape priors into the Graph Cut framework. The first set of methods incorporates the shape prior by penalizing deviations from a pre-defined mask [50, 152]. Normally such a mask is not shift or rotation invariant and thus requires additional alignment or registration steps to handle geometric transformations. Another set of methods attempts to loosen such strong prerequisites and is normally restricted to simple shapes. In [53], a blob energy term is proposed to favor a segmentation boundary that is perpendicular to rays from a pre-selected center. Similarly, an elliptical prior is used in [133]. The output quality of these methods is sensitive to the location of the center. These ideas are generalized in [149] by relaxing the segmentation boundary towards forming a star shape. In [40], the image is separated into four quadrants at a seed location and a set of prohibited assignments is defined to encourage Graph Cut to form a relatively simple and short boundary in each of these quadrants. This compact shape prior is not rotation invariant and is also sensitive to the seed location. All the work above addresses the problem of incorporating a shape prior for a single object. In this paper, we introduce a method to incorporate a shape prior for multiple objects with varying sizes.

3.3. Cell Nucleus Segmentation with Shape Regularization

sec:resultseq:shape-penaltyeq:shape-penaltyeq:shape-penalty

We formulate the cell nucleus segmentation problem as a MRF energy minimization problem and perform the inference using Graph Cut. Directly applying the classic MRF model in Eq. 3.1 is not sufficient for the following reasons:

- The images of cell nuclei, subject to their growth phase as well as the characteristics of the microscope, exhibit different intensities and textures (Fig. 3.1A, B). This makes the use of simple intensity-based modeling of the data term inapplicable.
- Multiple cell nuclei have to be extracted simultaneously and the boundary between proximate ones can be too weak to be captured by the data term (Fig. 3.1B).
- The formulation in Eq. 3.1 is subject to the shrinking bias, which tends to yield a shorter boundary in the presence of intensity gradient ramp (Fig. 3.1C).
- Cell nuclei exhibit a variety of appearance and uneven intensity distribution, which may consequently cause unfavorable segmentation results such as gulf-shaped boundary or holes within a segmented object (Fig. 3.1D, E).

To tackle those problems, we propose a novel energy function that integrates several important visual cues and enhances the segmentation by shape regularization (Eq. 3.2). The overall computational workflow is shown in Fig. 3.2. The method first performs a cell nucleus detection that roughly locates cell nuclei but does not provide sufficient characterization of their boundary. The resulting connected components are referred to as *seeds*. Using the seeds as positive samples and the watersheds from a seeded Watershed as negative ones, a binary Random Forest (RF) classifier is trained on local features.

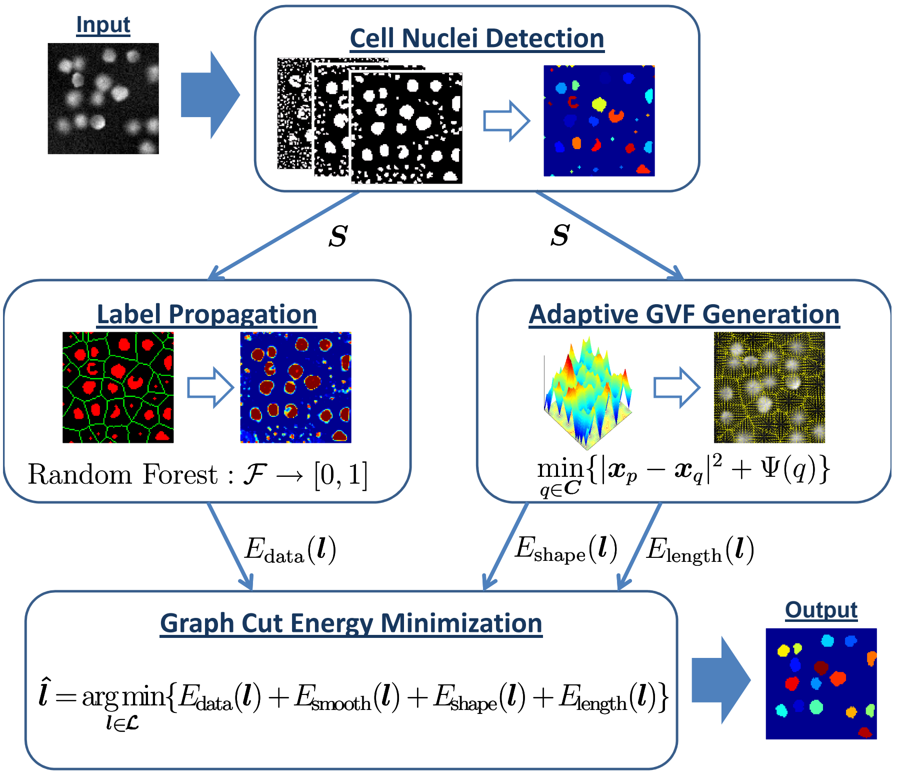

Input
Cell Nuclei Detection
$\boldsymbol{S}$
$\boldsymbol{S}$
Label Propagation
Random Forest : $\mathcal{F} \longrightarrow [0,1]$
Adaptive GVF Generation
$\min_{q \in \boldsymbol{C}} \{|\boldsymbol{x}_p - \boldsymbol{x}_q|^2 + \Psi(q)\}$
$E_{\text{data}}(\boldsymbol{l})$
$E_{\text{shape}}(\boldsymbol{l})$
$E_{\text{length}}(\boldsymbol{l})$
Graph Cut Energy Minimization
$\hat{\boldsymbol{l}} = \arg\min_{\boldsymbol{l} \in \boldsymbol{\mathcal{L}}} \{E_{\text{data}}(\boldsymbol{l}) + E_{\text{smooth}}(\boldsymbol{l}) + E_{\text{shape}}(\boldsymbol{l}) + E_{\text{length}}(\boldsymbol{l})\}$
Output

Figure 3.2.: Algorithmic workflow of our method. The seeds from a cell nucleus detection module are used as labels for training a pixel classifier based on local features, yielding the data term E_{data}. Also, the seeds generate an adaptive GVF which serves as a base of computing the shape term E_{shape} and the length term E_{length}. Ultimately, the energy function is minimized using the max-flow/min-cut algorithm.

This RF computes the probability of a pixel belonging to the fore- or background over the entire image, yielding $E_{\text{data}}(l_p)$ in Eq. 3.2. Furthermore, the seeds parameterize a function distance transform from which a GVF adaptive to the characteristic of each individual cell nucleus is generated. This GVF is at the base of $E_{\text{shape}}(l_p, l_q)$ for regularizing the shape and $E_{\text{length}}(l_p)$ for eliminating the shrinking bias. Finally, the energy function in Eq. 3.2 is minimized using the max-flow/min-cut algorithm with global optimality (proof available in Section 3.3.5). For the smoothing term $E_{\text{smooth}}(l_p, l_q)$, we use the same Potts model from [20].

$$\begin{aligned} E(\boldsymbol{l}) \quad = \quad & \lambda_{\text{data}} \sum_{p \in \boldsymbol{I}} E_{\text{data}}(l_p) + \lambda_{\text{smooth}} \sum_{\{p,q\} \in \boldsymbol{N}} E_{\text{smooth}}(l_p, l_q) + \\ & \lambda_{\text{shape}} \sum_{\{p,q\} \in \boldsymbol{N}} E_{\text{shape}}(l_p, l_q) + \lambda_{\text{length}} \sum_{p \in \boldsymbol{I}} E_{\text{length}}(l_p) \end{aligned} \tag{3.2}$$

3.3.1. Cell Nucleus Detection via Multi-scale Coherence

Our cell nucleus detection algorithm exploits the coherence of local maxima in scale-space [94] by measuring the eigenvalues of the Hessian matrix. Intuitively speaking, we subsequently use the fact that a *true* local maximum extends across several scales [109] while a *false* one induced by noise or from over-smoothed edges does not exhibit such coherence. Therefore, pixels which have all negative intensity curvature (i.e., both eigenvalues of the Hessian matrix are negative) across a pre-defined set of scales must be part of a nucleus. These pixels are grouped by a morphological closing operation to finally yield a labeled image $\boldsymbol{S}$ via connected component analysis [126]. These components, referred to as *seeds*, provide the starting point for the following regularized segmentation. Pseudo-code of the algorithm (for 2D images) is shown in Algorithm 5. In practice, the eigenvalue threshold η_0 can be set lower than zero to improve the robustness of the algorithm against noise or false background structure.

Algorithm 5: 2D multi-scale cell nucleus detection algorithm (analogously in 3D).

```
Input: 2D image $\boldsymbol{I}$; set of scales $\{\sigma\}$; template $\boldsymbol{T}$
Output: Labeled image $\boldsymbol{S}$ representing cell nuclei.
$\boldsymbol{M} \leftarrow \boldsymbol{true}$ // Initialize the indicator matrix
foreach $\sigma \in \{\sigma\}$ do
    $\boldsymbol{I}_\sigma \leftarrow \boldsymbol{I} \star \boldsymbol{G}_\sigma$ // Smooth the raw image at scale $\sigma$
    $[\boldsymbol{\eta}_1, \boldsymbol{\eta}_2] \leftarrow$ compute_eigenvalues_of_hessian($\boldsymbol{I}_\sigma$);
    $\boldsymbol{M} \leftarrow \boldsymbol{M} \wedge [\boldsymbol{\eta}_1 < 0] \wedge [\boldsymbol{\eta}_2 < 0]$ // Update the indicator matrix
end
$\boldsymbol{M} \leftarrow$ morphological_closing($\boldsymbol{M}, \boldsymbol{T}$) // Run the closing operation
$\boldsymbol{S} \leftarrow$ connected_component_analysis($\boldsymbol{M}$) // Generate seeds
return $\boldsymbol{S}$
```

3.3.2. The Data Term: Unbiased Label Propagation

Cell nuclei exhibit a variety of intensities and textures, which makes them recognizable only by considering their local context. Consequently, a simple global intensity model is insufficient to express the likelihood of pixels belonging to the fore- or background [21, 17]. Therefore, we use the probability map predicted by a pixel classification procedure using local features [7, 74]. Normally, the training data is supplied by human labelers. Such an interactive approach requires a sufficient amount of experts' input which may be expensive and laborious to obtain. Also, for this particular segmentation problem, we observe that manual labeling tends to be biased in favor of the nuclei and against the background. This can be seen in Fig. 3.3: focusing on the objects of interest, manual labelers tend to annotate *nucleus* at the red lines in A and C, which consequently

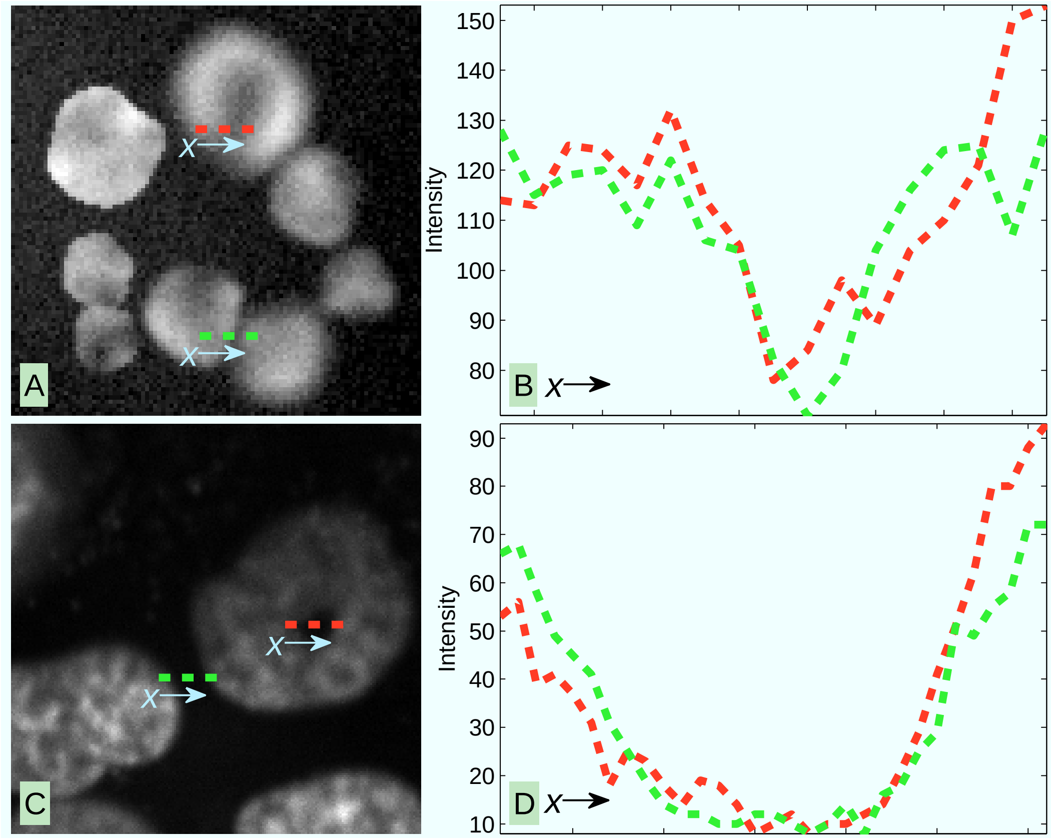


Figure 3.3.: Cell nuclei often exhibit uneven intensity distribution with the appearance of gulf (A) or hole (C). The local features of pixels at the gulf (A, red line) may be similar to a true boundary (A, green line). This can be seen from their corresponding intensity profiles in B (red: nucleus; green: boundary). Similarly, image C and plot D show another example as a "hole" inside the nucleus body. Example A and C are from [86] and [37], respectively.

may lead to possible under-segmentation at the respective green line since they have similar intensity profiles (see B and D, respectively). This bias can be compensated if an interactive realtime feedback system is incorporated (e.g. immediate update of the segmentation results), which suggests the labelers put more *background* labels at the green lines. But this requires considerable labeling efforts which is tedious to carry out, especially in 3D.

To reduce this bias and save manual labeling efforts, we propose an automated labeling scheme that produces an unbiased set of labels. In particular, we create a full set of labels by using the seeds as positive labels and the watersheds from a seeded Watershed on

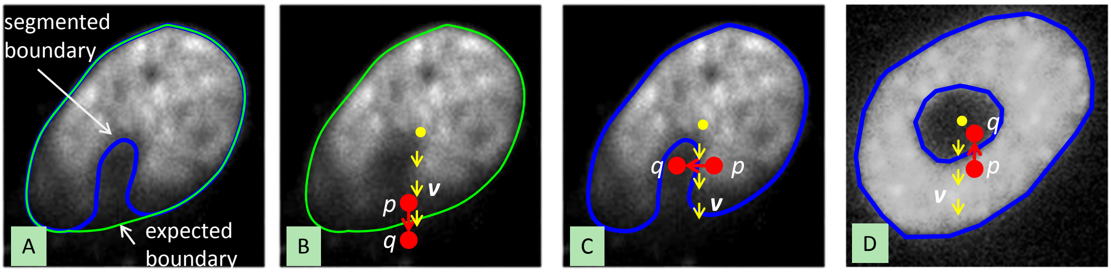


Figure 3.4.: Illustration of the GVF based shape prior. Without any shape regularization, the segmentation is subject to the uneven distribution of intensity, yielding unfavorable boundary (A). The shape prior encourages the Graph Cut to saw edges aligned with the GVF (B) rather than perpendicular (C) or opposite (D) to it. (best viewed in color)

the raw image as negative ones. These labels and the corresponding local features $\boldsymbol{\mathcal{F}}$ are used to train a Random Forest classifier [25] that produces a probability map of nucleus/background assignment throughout the entire image: $\boldsymbol{\mathcal{F}} \rightarrow [0,1]$. More study on the differences of manual labeling and the proposed labeling scheme can be found in Section 3.5.5. In terms of the energy function, this probability map is the data term $E_{\text{data}}(l_p)$ for pixel p with label l_p,

$$E_{\text{data}}(l_p) = 1 - \Pr(l_p|\boldsymbol{\mathcal{F}}_p). \tag{3.3}$$

3.3.3. The Regularization Terms: Multi-object Shape Prior

Cell nuclei usually exhibit an ellipsoid shape, which suggests incorporating a shape prior. Taking the cell nucleus in Fig. 3.4A as an example, the segmentation by Level Sets suffers from the uneven distribution of intensity and thus yields a gulf-shaped boundary (blue line). Now consider a unit gradient vector field $\boldsymbol{v}$ originated from the nucleus center (yellow spot). The true boundary is expected to be mostly perpendicular to $\boldsymbol{v}$ (Fig. 3.4B) while the initial boundary estimate is partly parallel to $\boldsymbol{v}$ (Fig. 3.4C). In terms of the graph representation, this is equivalent to the fact that the favorable edges severed by Graph Cut (3.4B, pixel p(foreground) $\rightarrow$ q(background)) are largely aligned with $\boldsymbol{v}$ and the unfavorable ones are perpendicular (Fig. 3.4C) or even opposite (Fig. 3.4D) to $\boldsymbol{v}$. Therefore, we formulate the shape term as a pair-potential based on a pre-defined GVF $\boldsymbol{v}$. Formally, given a pair of neighboring pixels p and q (labeled as l_p and l_q), the shape term is defined as

$$E_{\text{shape}}(l_p, l_q) = \exp\left(-\langle \boldsymbol{v}_p, \boldsymbol{u}_{p\rightarrow q}\rangle\right) \cdot \mathcal{I}\left(\langle \boldsymbol{v}_p, \boldsymbol{u}_{p\rightarrow q}\rangle \leq 0\right) \cdot \mathcal{I}\left(l_p \neq l_q\right), \tag{3.4}$$

where $\boldsymbol{v}_p$ is the gradient vector at pixel p, $\boldsymbol{u}_{p \to q}$ is a unit vector oriented from pixel p to pixel q, and $\mathcal{I}\left(c\right)$ is an indicator function that returns 1 when the condition c is satisfied and 0 otherwise.

It is known that surface regularization techniques such as Graph Cut are subject to a shrinking bias in the presence of an intensity gradient ramp. In particular, the optimization algorithm normally attempts to minimize the sum of the costs and consequently favors a shorter boundary to reduce the total cost. Generally speaking, this is the consequence of inaccuracy in the data term, which in our case is the high uncertainty of the Random Forest classifier at weak boundaries. Obviously, this bias is further exacerbated in the proposed energy function because the shape regularization term introduces extra costs. This makes a length regularization of the segmentation boundary indispensable. We use the flux maximizing method described in [148] as the boundary length regularization term. Using again the unit GVF $\boldsymbol{v}$ from Eq. 3.4, the purpose of flux maximizing is to maximize the flow of the GVF through the segmentation boundary. Formally, the flux that passes through p is given as

$$w(p) = \sum_{q \in \boldsymbol{N}(p)} \langle \boldsymbol{v}_q, \boldsymbol{u}_{p \to q} \rangle, \tag{3.5}$$

where $\boldsymbol{u}_{p \to q}$ is a unit vector oriented from pixel p to q and $\boldsymbol{N}(p)$ represents the set of neighbors of p. The corresponding energy term (see Eq. 3.2) is formulated as [79]

$$E_{\text{length}}(l_p) = \begin{cases} -w(p) & \text{if } w(p) \leq 0 \text{ and } l_p = 1 \\ 0 & \text{if } w(p) \leq 0 \text{ and } l_p = 0 \\ 0 & \text{if } w(p) > 0 \text{ and } l_p = 1 \\ w(p) & \text{if } w(p) > 0 \text{ and } l_p = 0 \end{cases} \tag{3.6}$$

Finally, we choose the spatial smoothing term $E_{\text{smooth}}(l_p, l_q)$ the same as [20]. Intuitively, this function gives a small penalty if $||I_p - I_q||$ is large, i.e. pixel p and pixel q are more likely to be on an edge, as

$$E_{\text{smooth}}(l_p, l_q) = \exp\left(-\frac{||I_p - I_q||^2}{2\sigma_Z^2}\right) \cdot \frac{1}{\text{dist}(p, q)} \cdot \mathcal{I}\left(l_p \neq l_q\right), \tag{3.7}$$

where σ_Z is a user-adjustable parameter, formally the standard deviation of noise. Note that I_p and I_q are vectors in the case of multivariate images.

To simultaneously segment multiple cell nuclei, the regularization terms (Eq. 3.4 and 3.6) have to be extended by generating a proper GVF for each cell nucleus. The normalized intensity gradient is not suitable due to the intensity inhomogeneity and interior texture (Fig. 3.5A). The same applies to the normalized gradient of a distance map computed from the seeds (Fig. 3.5B). To eliminate the irregularity introduced by intensity unevenness, texture and noise, one can use the normalized gradient of a distance

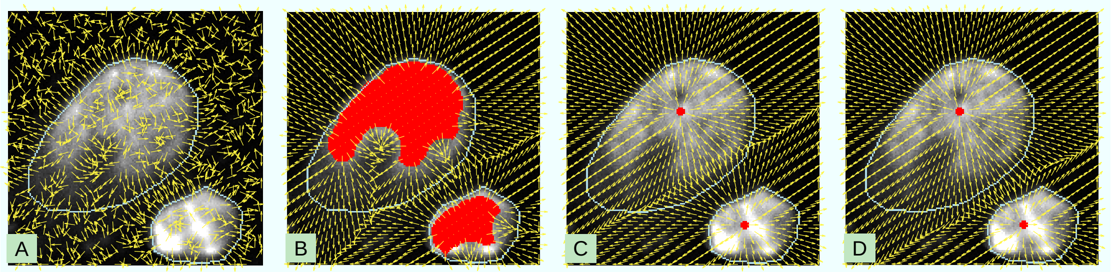


Figure 3.5.: Comparison of possible GVFs as multi-object shape prior. Both the intensity gradient of the raw image (A) and the gradient of the distance transform of the seeds (B) are subject to the intensity unevenness. The gradient of a binary distance transform of the seed centers introduces interference (C) from proximate cell nuclei or a false detection. Our method yields a GVF that is adaptive to the characteristics of its sources and eliminates the interference (D). White lines are the expected segmentation boundaries. Red area represents the seeds or their centers. Yellow arrows represent the GVF. (best viewed in color)

transform of the seed centers. However, a simple binary Euclidean distance transform (bEDT, see [47]) from the seed centers discards all information of the raw seeds but their coordinates, introducing possible interference between proximate cell nuclei or from false detections. As shown in Fig. 3.5C, the GVF from the lower-right cell nucleus extends into the interior of the other one, preventing Graph Cut from yielding the expected boundary (white lines, Fig. 3.5C). As an alternative interpretation, bEDT creates a Voronoi diagram [9] which possibly splits a single cell nucleus. To overcome the limitations of the bEDT, we now present a novel method for efficiently generating a GVF that adaptively adjusts the range of influence with respect to individual seeds (Fig. 3.5D) as follows.

3.3.4. Adaptive GVF by Function Distance Transform

The Euclidean distance from a seed center can be visualized as a 2D parabola rooted at it. Accordingly, computing the bEDT from multiple seed centers is equivalent to determining the lower envelope of a set of parabolas rooted at their respective seed centers [47]. Formally, given a set of pixels $\boldsymbol{C}$ representing the geometric centers of all connected components of the seeds, the bEDT is computed by Eq. 3.8 where $\boldsymbol{\Omega}^2 = \{1, \ldots, n\} \times \{1, \ldots, m\}$ is the image domain and $\boldsymbol{x}_p$ represents the coordinates of pixel p. The range of influence of the GVF from each seed extends to the border of the Voronoi region, i.e. to the intersections of its parabola with others (Fig. 3.6A). This is the idea behind a few algorithms which have time complexity $\mathcal{O}(nm)$ and outperform

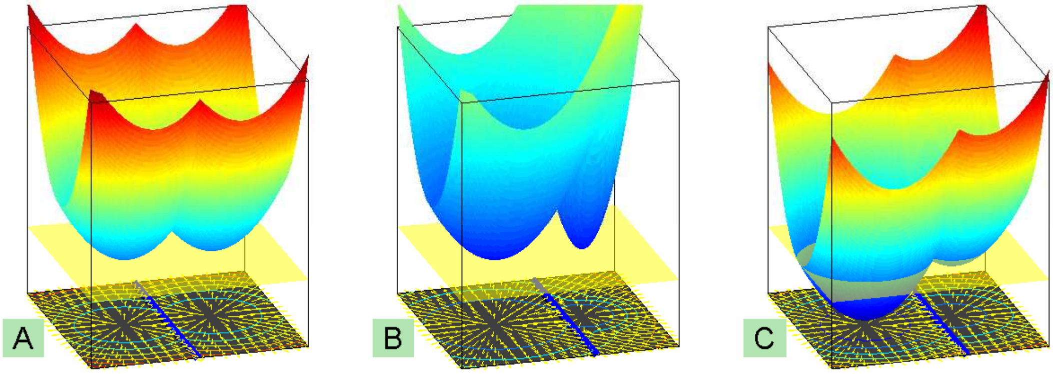


Figure 3.6.: Examples of the parabola metaphor for 2D distance transforms: the bEDT (A), the abEDT(B) that adjusts the curvature of the parabolas, and fEDT(C) that pulls down the parabolas to adjust the influence of individual sources.

other relevant algorithms [47].

$$\forall p \in \boldsymbol{\Omega}^2, f_{\text{bEDT}}(p) = \min_{q \in \boldsymbol{C}} \left\{ \{ |\boldsymbol{x}_p - \boldsymbol{x}_q|^2 \} \right\} . \tag{3.8}$$

One way to adjust the range of influence from an individual seed $q \in \boldsymbol{C}$ is to encode the characteristics (size, intensity, etc.) of the seed into the curvature of the corresponding parabola (Fig. 3.6B). This is equivalent to solving the following minimization problem,

$$\forall p \in \boldsymbol{\Omega}^2, f_{\text{abEDT}}(p) = \min_{q \in \boldsymbol{C}} \left\{ \{ \Phi(q) |\boldsymbol{x}_p - \boldsymbol{x}_q|^2 \} \right\} , \tag{3.9}$$

where "abEDT" stands for adaptive binary Euclidean distance transform and the curvature $\Phi(q)$ characterizes the seeds. Reducing $\Phi(q)$ increases the range of influence from q and vice versa. In fact, $\Phi(q)$ is constantly 1 in bEDT. However, it is difficult to extend the classic bEDT algorithms for the minimization of Eq. 3.9 with the same $\mathcal{O}(nm)$ complexity. This is due to the fact that, since each parabola has its own curvature, the lower envelope of the parabolas at pixel p could be determined by a parabola rooted far away from rather than the one being closest to p. Therefore, the base time complexity is $\mathcal{O}(|C|nm)$ and $|C|$ represents the number of seed centers.

Considering the high time complexity introduced by adjusting the curvature of the parabolas, we propose an alternative approach for generating the expected GVF with time complexity $\mathcal{O}(nm)$. Note that, similar to the curvature idea, the range of influence of the seeds can also be controlled by pulling down the parabolas differently, i.e. imposing different offset to the parabolas (Fig. 3.6C). This corresponds to the following minimization problem,

$$\forall p \in \boldsymbol{\Omega}^2, f_{\mathrm{fEDT}}(p) = \min_{q \in \boldsymbol{C}} \left\{ \{ |\boldsymbol{x}_p - \boldsymbol{x}_q|^2 + \Psi(q) \} \right\}, \tag{3.10}$$

where Ψ is a function defined on the image domain $\boldsymbol{\Omega}^2$. In particular, Ψ encodes the seed-dependent offset at the seed centers $\boldsymbol{C}$ and is infinite elsewhere, as

$$\forall p \in \boldsymbol{\Omega}^2, \Psi(p) = \begin{cases} -\Phi(p) & \text{if } p \in \boldsymbol{C} \\ \infty & \text{else} \end{cases} \tag{3.11}$$

The authors in [48] have proposed a method for efficiently solving Eq. 3.11 and achieved a time complexity of $\mathcal{O}(nm)$ for 2D images. This method can also be easily extended to arbitrary dimensions. For example of 3D images, let $\boldsymbol{\Omega}^3$ be the 3D image domain and $\Psi : \boldsymbol{\Omega}^3 \to \mathbb{R}$ a function defined on it. Let $\boldsymbol{x} = [x, y, z]$ be the spatial coordinates. We have

$$\begin{aligned} f_{\mathrm{fEDT}}(p) &= \min_{q} \left\{ \{ |\boldsymbol{x}_p - \boldsymbol{x}_q|^2 + \Psi(q) \} \right\} \\ &= \min_{x_q, y_q, z_q} \left\{ \{ (x_p - x_q)^2 + (y_p - y_q)^2 + (z_p - z_q)^2 + \Psi(q) \} \right\} \\ &= \min_{z_q} \left\{ \left\{ (z_p - z_q)^2 + \min_{x_q, y_q | z_q} \left\{ \{ (x_p - x_q)^2 + (y_p - y_q)^2 + \Psi(q) \} \right\} \right\} \right\} \\ &= \min_{z_q} \left\{ \{ (z_p - z_q)^2 + f_{\mathrm{fEDT}}(p|z_q) \} \right\}. \end{aligned} \tag{3.12}$$

This indicates that the fEDT of 3D images is performed by first computing the 2D transform on each slice along the z-axis (i.e. computing $f_{\mathrm{fEDT}}(p|z_q)$) and then computing the one dimensional transforms on the z-axis of the result over the entire $x - y$ plane. Similarly, the 2D $f_{\mathrm{fEDT}}(p|z_q)$ is computed by first performing one dimensional transforms along each column of the grid, and then performing one dimensional transforms along each row of the result [48]. Notice that this fEDT does not provide a conceptually valid image distance transform as the bEDT. However, the gradient of the transformed result ∇_{fEDT} is independent of the local offset, allowing us to directly obtain a GVF with the required property at a low computational cost. In practice, we design the function Ψ as follows:

- For each connected component from the cell nucleus detection, we compute its geometric center c and the minimum bounding box R_c. All the centers form a set of pixels $\boldsymbol{C}$.
- For each center (also pixel) $c \in \boldsymbol{C}$, Ψ is set to be $-\left[\zeta(R_c)/2\right]^2$, where $\zeta(R_c)$ is the diagonal length of the bounding box. For the remaining pixels, Ψ is set infinite.

We notice that an alternative interpretation of Eq. 3.11 is the erosion operation on the gray-value image defined by Ψ with a parabolic structuring element [55].

3.3.5. Submodularity and Global Optimality

We here prove that the energy function in Eq. 3.2 is submodular so that Graph Cut can obtain the global optimum.

Proof. All functions of one variable are submodular. Thus, $E_{\text{data}}(l_p)$ and $E_{\text{length}}(l_p)$ are submodular. A second-order binary energy function $f(l_p, l_q)$ is submodular if it satisfies the following inequality

$$f(0,0) + f(1,1) \leq f(1,0) + f(0,1). \tag{3.13}$$

Obviously, the second-order terms in Eq. 3.2 that are defined on a binary Potts model fulfils this condition. Thus, both $E_{\text{smooth}}(l_p, l_q)$ and $E_{\text{shape}}(l_p, l_q)$ are submodular. Since the sum of submodular functions is submodular again, we see that energy function Eq. 3.2 is submodular. □

3.4. Experiments

Our method has been evaluated on several datasets with different complexity and characteristics, including the SIMCEP benchmark [86], the hand-labeled 2D nucleus segmentation benchmark [37], and the digital embryo dataset [76]. Furthermore, we also conduct extensive comparison between our method and a few methods listed in Section 4.2.

3.4.1. Evaluation Datasets

The SIMCEP benchmark [86] models a complete generation of typical 2D fluorescent microscope images, including local properties such as shape and texture, and global properties such as population, point spread function (PSF), autofluorescence background, uneven illumination and noise. We generated 20 instances of noise-free images (with ground truth) and corrupted them with different levels of noise (Fig. 3.7). Each image has size 501×501 and contains a total of 160 cell nuclei. The SNR varies between 45 and 15 dB. Here, SNR is defined as $20 \log_{10} [(I_{\max} - I_{\min})/\sigma_Z]$, where $I_{\max} - I_{\min}$ controls the bounds of the intensity and σ_Z controls the noise level.

The hand-labeled 2D nucleus segmentation benchmark was presented in [37] to tackle the problem of subjective and insufficient evaluation of individual nucleus segmentation methods. The dataset consists of two different collections (*U2OS* cells and *NIH3T3* cells) of 98 fluorescence microscopy images (a total of 4009 cell nuclei) that exhibit different characteristics such as inhomogeneity of size/shape/intensity as well as clustering and debris (Fig. 3.8).

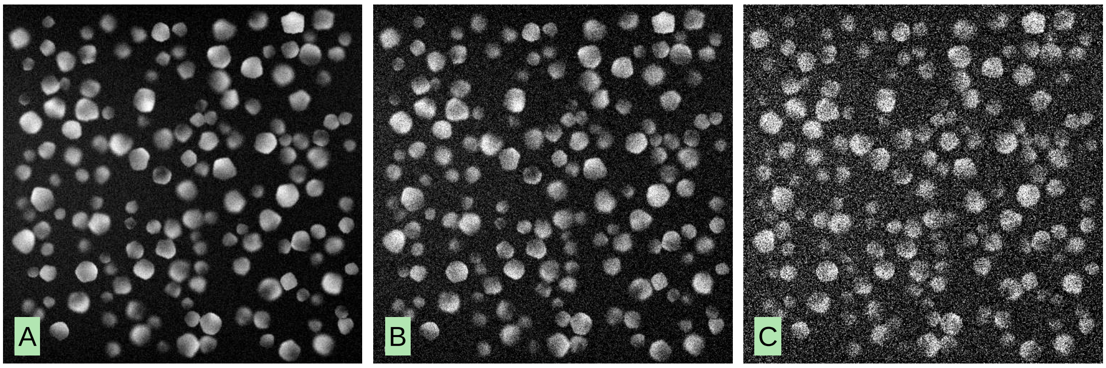


Figure 3.7.: Examples of simulated cell nuclei image using SIMCEP [86]. From left to right, the SNR is 30.1, 22.6, and 15.2. See Section 3.4.1 for detailed description.

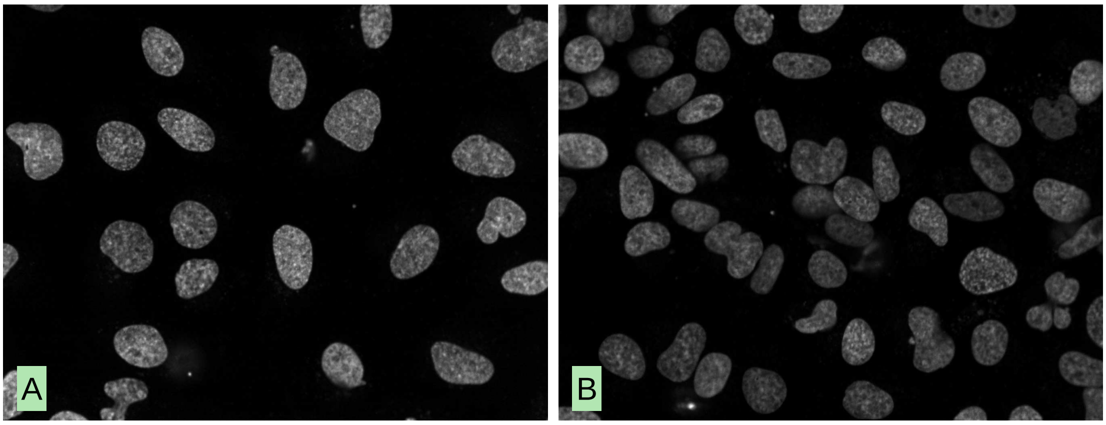


Figure 3.8.: Examples of images from the hand-labeled benchmark [37]. Most images in the dataset lie in between these two examples.

The digital embryo dataset was acquired using digital scanned laser light sheet fluorescence microscopy (DSLM) [76]. DSLM illuminates the specimen from the side with a "light sheet" and rapidly moves this plane vertically (or horizontally) through the specimen to generate volumetric fluorescence images. The data record the location of stained cell nuclei in an entire zebrafish embryo over the first 24 hours of development with a temporal resolution of 90 s. As shown in Fig. 3.9 and Fig. 3.10, segmenting such dataset has the following challenges: relatively low SNR w.r.t 2D images, high variability of brightness (16-bit, varying from ca. 300 to ca. 10000), strong background structure at early time steps, and severe cell nuclei clutter at late time steps.

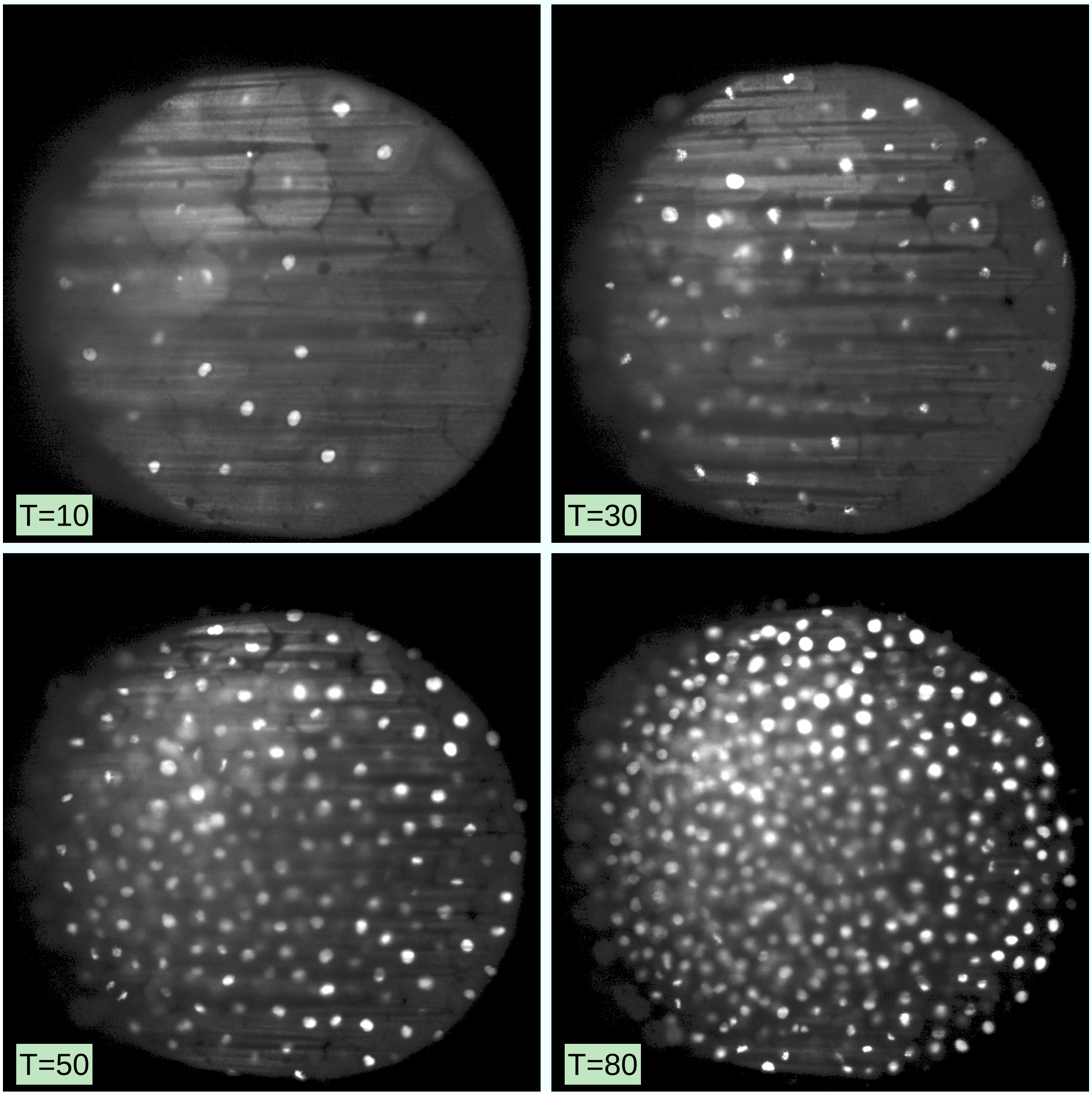


Figure 3.9.: Slice view (x-y) of a single slice from selected time steps of the digital embryo dataset [76]. Segmenting such dataset has the following challenges: relatively low SNR w.r.t 2D images, high variability of brightness, strong background structure at early time steps, and severe cell nuclei clutter at late time steps.

3.4.2. Implementation

Our method has been implemented in C++ and the software will be available to the public. To reduce the computational and memory load especially for large datasets,

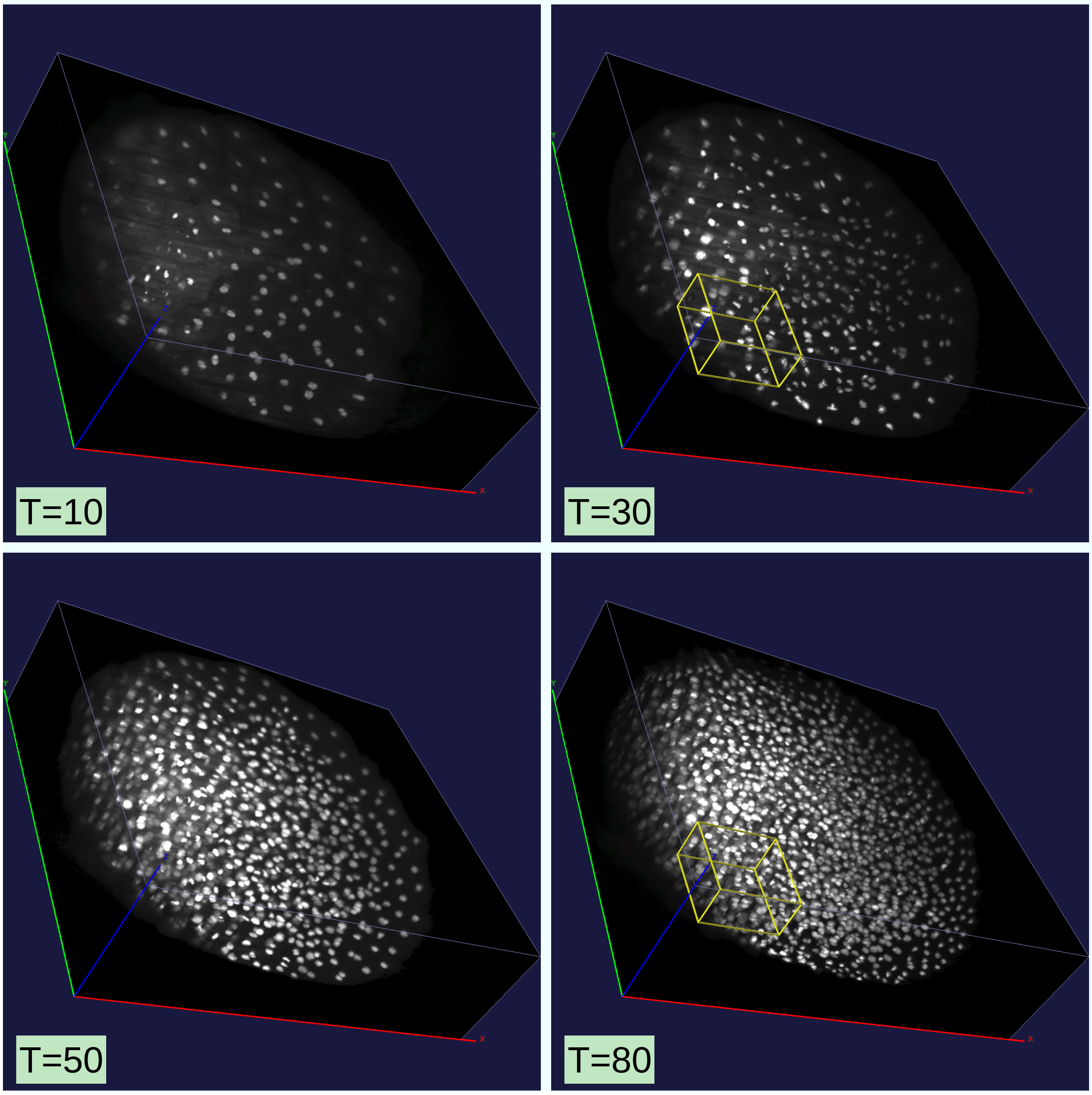


Figure 3.10.: Volume visualization [118] of selected time steps of the digital embryo dataset [76]. A subvolume (yellow box) in time step 30 and 80 is extracted for visualizing the segmented surface in detail, see Fig. 3.18 and Fig. 3.19.

we restrict the max-flow optimization to the region above certain intensity threshold. The corresponding sparse graph construction is shown in Fig. 3.11. In particular, all nodes below the intensity threshold are fixed to the sink (background) and the normal max-flow optimization occurs only within the intensity iso-contour (yellow dash line).

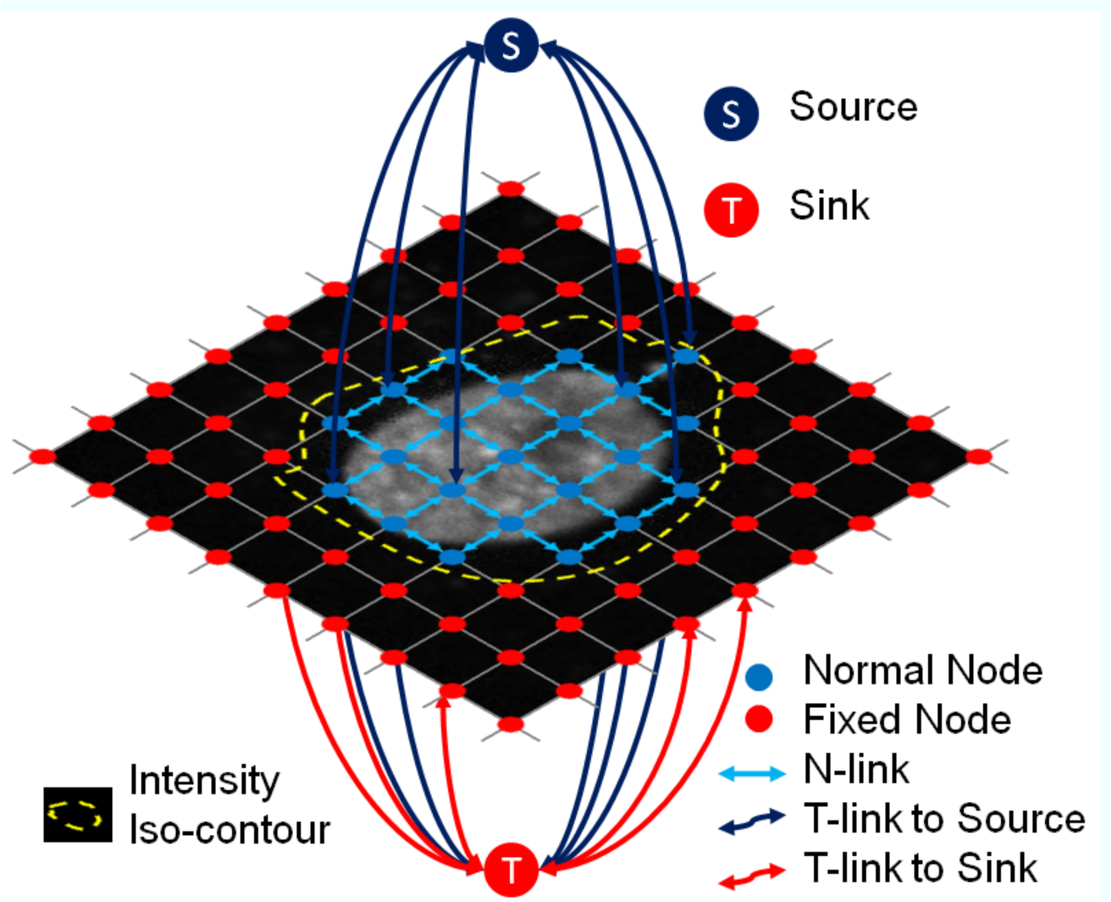


Figure 3.11.: Sparse graph construction for background removal. All pixels outside the yellow intensity iso-contour are fixed to the sink (background) and no N-links are created between them. Note that not all T-links are draw in the figure to avoid visual clutter. (best viewed in color)

The implementation also features multi-threading for parallelization on shared memory systems and block-processing for large volumetric data.

3.4.3. Relevant Methods for Comparison

The methods used for comparison are:

- The Level Sets (**LS**) method using the Chan-Vese model [33] which is the basic building block for a few segmentation or tracking methods;
- The method from [89] based on gradient flow tracking (**GFT**);

- The method described in [4] (**FS**) which uses a cascade of two Graph Cut runs for image binarization and segmentation pruning;
- The local adaptive thresholding (**LAT**) used in [76];
- The classic Graph Cut (**GC**) with a data term and a smoothing term only.
- The method in this paper is referred as **Ours**. We also report the intermediate cell nucleus detection results from the multi-scale analysis (**MSA**).

The parameters for all algorithms above are tuned for each dataset individually. For the datasets with ground truth [86, 37], one representative image is selected and the optimal parameters are chosen via a grid search in the parameter space. For the digital embryo dataset with no ground truth, we determine the optimal parameters by visually examining the results w.r.t. the raw data. Note that the tuning aims at a fair comparison in terms of segmentation quality, disregarding the runtime. For our methods, the key parameters are the scales for cell nucleus detection (Algorithm 5) and the parameters that weight the contributions of energy terms in Eq. 3.2. The scales are chosen in a similar way as in [13]. We first roughly determine the diameter range $[d_1, d_2]$ of the nuclei and then compute the corresponding scale range $[\sigma(d_1), \sigma(d_2)]$ by $\sigma(d) = 1.2 \times \frac{d}{9}$. Then we evenly sample 3 or 5 scales from this range. For the parameters in Eq. 3.2, we adjust them empirically and a typical setting is $\lambda_{\text{data}} = \lambda_{\text{smooth}} = 0.30$ and $\lambda_{\text{shape}} = \lambda_{\text{length}} = 0.20$.

3.5. Results

To deliver a qualitative and quantitative evaluation of our method, we use several metrics to characterize the segmentation results. As described in [37], a matching procedure is performed to assign objects (i.e. connected components) between the segmentation and the ground truth. In detail, we first assign each object in the segmentation to the object in the ground truth with which it shares the most pixels and then repeat this procedure after switching the segmentation and the ground truth. Four types of errors are defined based on these assignments: **False Split**, two objects in the segmentation are assigned to the same object in the ground truth; **False Merge**, two objects in the ground truth are assigned to the same object in the segmentation; **Spurious**, one object in the segmentation is assigned to the background in the ground truth; and **Missing**, one object in the ground truth is assigned to the background in the segmentation. Furthermore, two additional measures are defined to characterize the overall qualitative performance: **Sensitivity** measures the proportion of correctly segmented cell nuclei w.r.t. the ground truth and **Specificity** measures the proportion of correctly segmented cell nuclei w.r.t. the total number of segmented objects. Eq. 3.14 and Eq. 3.15 define these two measures,

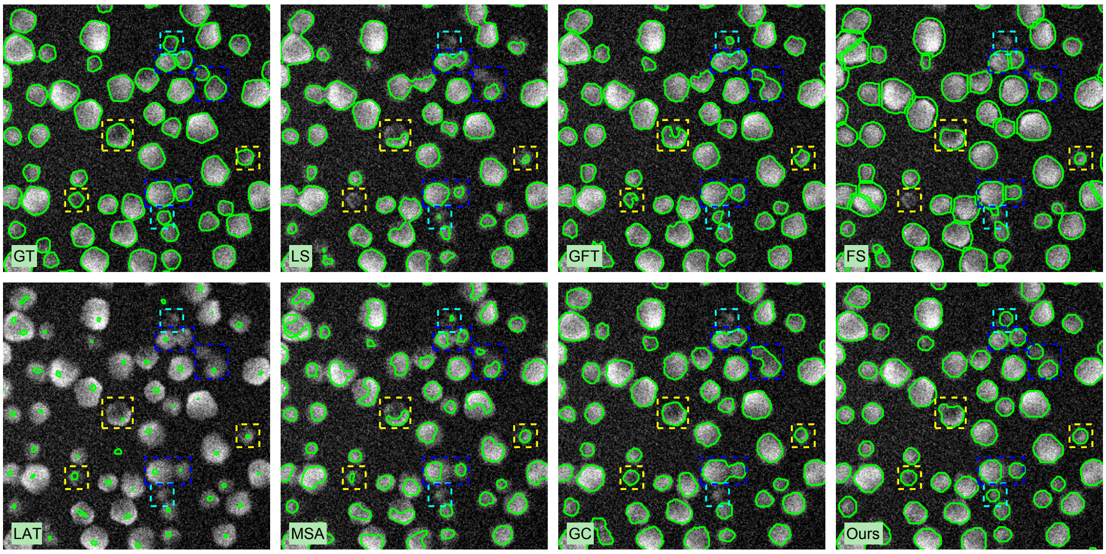


Figure 3.12.: Segmentation by the methods in comparison on one SIMCEP image. See Section 3.5.1 for detailed description. (best viewed in color)

where N_{Nuclei} is the true number of cell nuclei.

$$\text{Sensitivity} = \frac{N_{\text{Nuclei}} - N_{\text{Missing}} - N_{\text{FalseSplit}} - N_{\text{FalseMerge}}}{N_{\text{Nuclei}}} \tag{3.14}$$

$$\text{Specificity} = \frac{N_{\text{Nuclei}} - N_{\text{Missing}} - N_{\text{FalseSplit}} - N_{\text{FalseMerge}}}{N_{\text{Spurious}} + N_{\text{Nuclei}}} \tag{3.15}$$

All measures introduced above can be regarded as qualitative metrics. For quantitative evaluation, we choose the Rand Index and the Hausdorff metric as in [37]. The **Rand Index** measures the ratio of pixel pairs that are labeled consistently or not between the ground truth and the segmentation. The **Hausdorff** metric is defined as the largest distance of the segmentation boundary to the reference border. Note that the Hausdorff metric is computed only between pairs of matched cell nuclei.

3.5.1. Evaluation on the SIMCEP Benchmark

Fig. 3.12 shows the segmentations on one example image of the SIMCEP benchmark. Compared with the ground truth (GT), LS mainly suffers from under-segmentation, which is apparently improved by GFT and FS. However, GFT is subject to local intensity unevenness and FS is prone to over-segmentation. Despite several false positives, LAT correctly locates most of the cell nuclei but barely gives any information on their

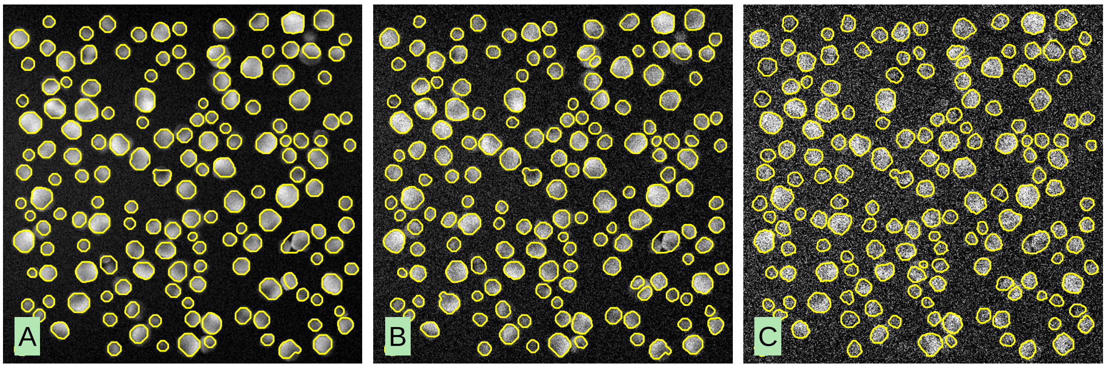


Figure 3.13.: Segmentation contour from our method w.r.t. different noise levels. Inspite of the obvious difference in image quality, our method yields stable results.

shape/size. For the purpose of cell nucleus detection, MSA improves over LAT by offering better detection accuracy and also providing coarse information on their shape/size. However, all these methods are subject to the intensity unevenness and yield unfavorable segmentations such as gulf (yellow boxes). GC based on manual labeling solves this problem at a cost of imposing a bias towards the nucleus class, which leads to a number of under-segmentations (blue boxes). Also, due to its limitation in describing the true distribution of cell nuclei in feature space (see Section 3.5.5), miss detections are also observed (cyan boxes). Our method segments regularly shaped cell nuclei and successfully avoids under-segmentation.

Fig. 3.13 shows the robustness of our method towards noise (raw images shown in Fig. 3.7). The segmentations remain stable even when the image is severely corrupted (Fig. 3.7C). Note that the only parameter adjusted is the standard deviation of noise σ_Z in the smoothing term in Eq. 3.2.

Fig. 3.14 shows the averages of selected measures over all test images at different noise levels. In general, the proposed cell nucleus detection method (MSA) delivers superior qualitative result by measure of Sensitivity/Specificity and also provides sufficient robustness against noise. This advantage carries over to the regularized segmentation which yields obvious improvement of Rand Index. Note that GC gives comparable Rand Index but obviously lower Sensitivity/Specificity. This corresponds to the fact that GC correctly detects the foreground but is subject to under-segmentation. Runtime was measured on a computer with Intel Core 2 Duo Processor T5470 2.0GHz (only one core is used for the computation).

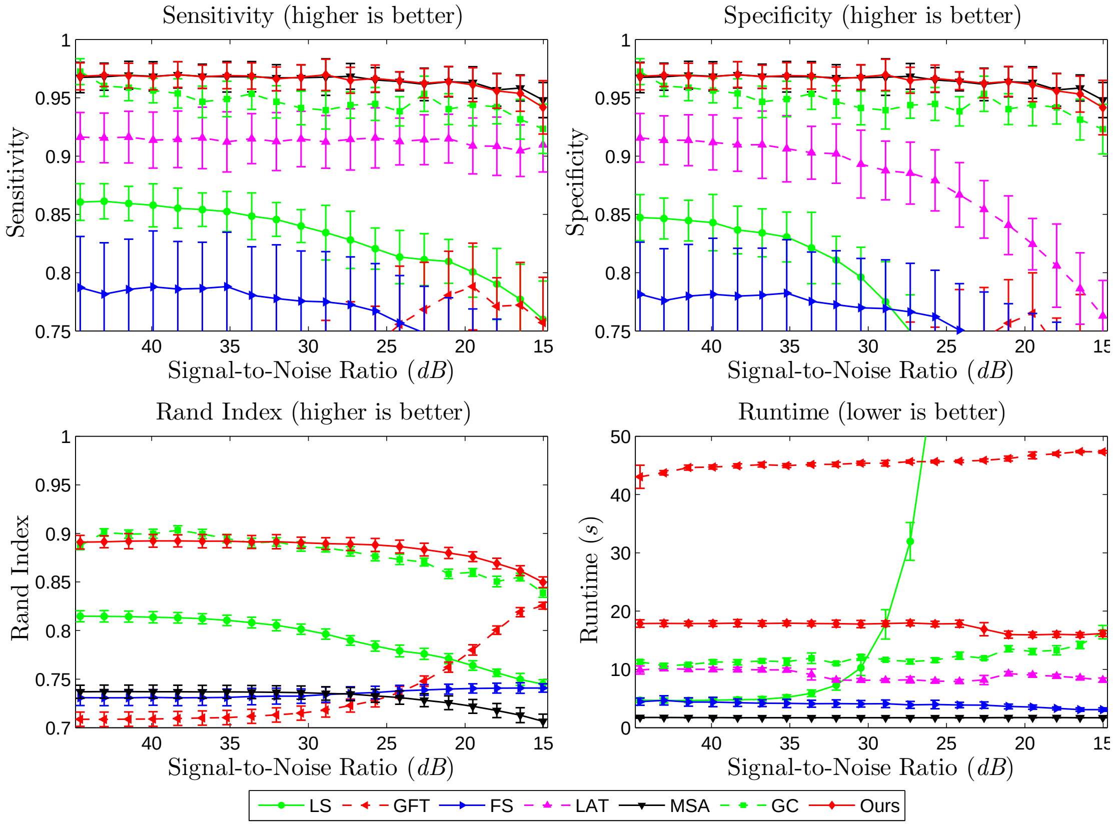


Figure 3.14.: Comparison of segmentation methods on the SIMCEP benchmark with varying noise level. The error bars indicate the standard deviations of the respective measure. Our method delivers convincing results both qualitatively and quantitatively. (best viewed in color)

3.5.2. Evaluation on the Hand-labeled Benchmark

We followed the evaluation procedure described in [37] on their hand-labeled cell nucleus segmentation benchmark. The comparison of segmentation methods is shown in Table 3.1 for the U2OS collection and Table 3.2 for the NIH3T3 collection. The Merging Algorithm (**MA**) [93], reported as the best scoring method in [37], is also included. Note that the numbers are the average over all images in the respective collection.

Qualitatively, our method yields high sensitivity ($\geq 94.6\%$) while keeping reasonable specificity ($\geq 85.2\%$), which is or is very close to the best among all methods in comparison. The other methods are either subject to severe False Merge/Split (LS, GFT and FS) or missing detections (LAT). Surprisingly, GC performs quite poorly on this dataset. This is probably due to the fact that any supervised segmentation procedure relies on the quality of the training dataset and thus is sensitive to variations between images. Quantitatively, the advantage of our method is further emphasized: our method yields the best Rand Index (R. I.) and Hausdorff distance among all methods in comparison. In particular, the improvement is substantial on the more difficult NIH3T3 collection (Table 3.2), i.e. 6.6% increase in Rand Index and 5.6 decrease in Hausdorff distance over the respective second best method.

Method	R. I.	Hausdorff	$N_{\text{F.Split}}$	$N_{\text{F.Merge}}$	N_{Spurious}	N_{Missing}	Sensi.	Speci.
LS	95.3%	10.5	0.7	2.4	4.1	0.1	93.1%	85.2%
GFT	94.2%	20.4	0.6	2.1	8.4	**0.0**	94.0%	79.6%
FS	95.7%	6.0	4.4	3.2	5.3	**0.0**	83.5%	74.6%
LAT	67.1%	33.9	**0.2**	**0.2**	2.4	2.1	94.5%	**88.9%**
MSA	85.7%	18.3	1.1	0.5	4.7	0.3	95.9%	86.6%
GC	84.8%	21.7	1.5	1.0	2.5	5.7	81.7%	77.4%
Ours	**96.2%**	**6.0**	0.6	1.2	4.0	**0.0**	**96.0%**	87.8%
MA	96.0%	12.9	1.8	2.1	**1.0**	3.3	N/A	N/A

Table 3.1.: Comparison of segmentation methods on the hand-labeled benchmark [37]: U2OS collection. See Section 3.5.2 for detailed description.

Method	R. I.	Hausdorff	$N_{\text{F.Split}}$	$N_{\text{F.Merge}}$	N_{Spurious}	N_{Missing}	Sensi.	Speci.
LS	82.6%	14.6	1.8	3.9	**3.9**	2.9	83.6%	78.7%
GFT	80.8%	20.4	2.7	4.0	11.1	1.1	84.8%	72.9%
FS	83.3%	15.2	4.0	4.7	5.7	1.7	81.5%	74.9%
LAT	72.8%	28.3	**0.1**	**0.7**	4.0	4.4	90.8%	84.9%
MSA	85.2%	15.8	1.0	1.0	11.2	0.3	**96.0%**	81.4%
GC	80.4%	27.7	3.1	2.9	19.8	0.8	86.9%	71.8%
Ours	**91.8%**	**9.0**	1.4	1.6	6.6	**0.0**	94.6%	**85.2%**
MA	83.0%	15.9	1.6	3.0	6.8	5.9	N/A	N/A

Table 3.2.: Comparison of segmentation methods on the hand-labeled benchmark [37]: NIH3T3 collection. See Section 3.5.2 for detailed description.

Fig. 3.15 shows the segmentation on the difficult example in Fig. 3.8. GFT, GC and our method are chosen for comparison with the ground truth (GT). GFT successfully segments the isolated cell nuclei but is subject to obvious under-segmentation for those in close proximity. Also, it overfits to the intensity inhomogeneity, producing holes in the segmented objects. These two issues are well addressed by our method. The under-segmentation problem is avoided by using the proposed unbiased labeling scheme. The problem of overfitting to the intensity inhomogeneity is avoided by shape regularization since a hole inside would have generated a high cost by the shape term. GC, in this case,

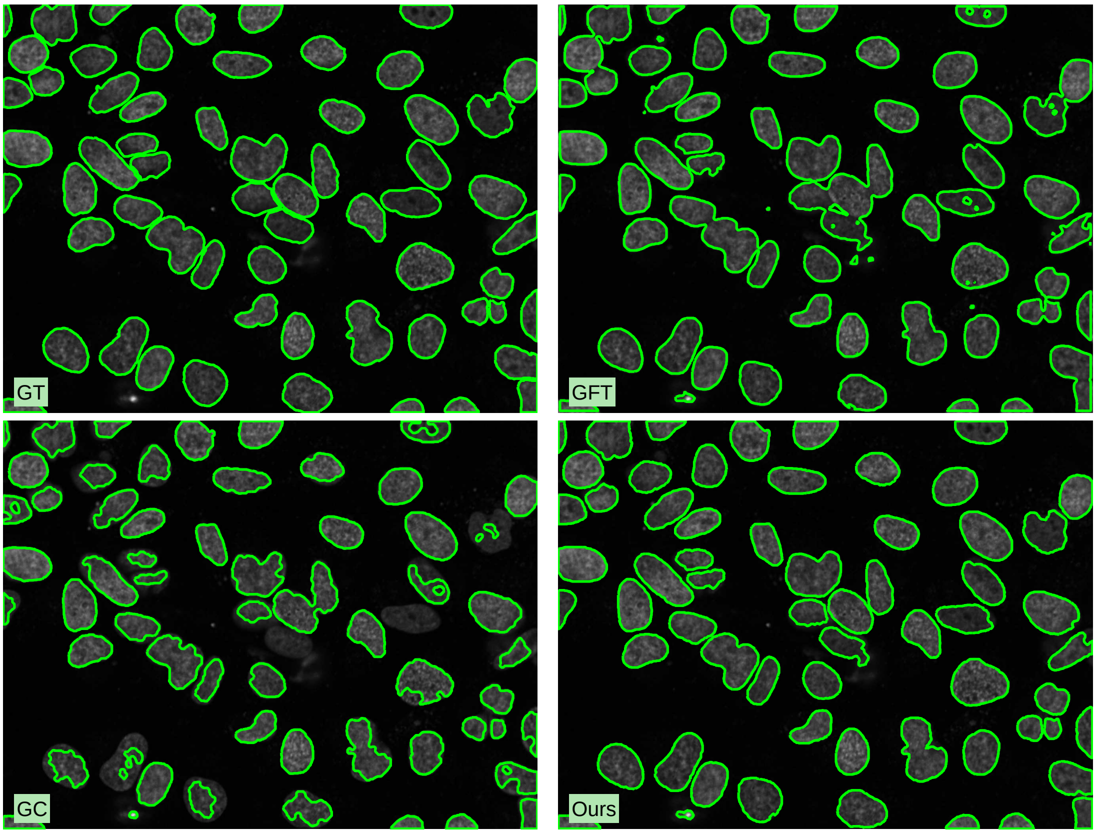


Figure 3.15.: Comparison of the segmentation between LS, GFT and our method. Our method gives superior performance by yielding fewer under-segmentations and by avoiding overfitting to the local intensity unevenness.

yields the worst segmentation. Furthermore, Fig. 3.16 illustrates two typical examples of gulf and hole free segmentations for the images in Fig. 3.1D and E.

We also observe that all methods are subject to relatively high Spurious error. We randomly selected 12 "spurious" cell nuclei for each method from the U2OS collection (Fig. 3.17). Many of them are in fact debris with nucleus-like appearance but normally much smaller.

3.5.3. Evaluation on the 3D Digital Embryo Dataset

The digital embryo dataset contains a sequence of volumetric images acquired within the first 24 hours of the development of a zebrafish embryo. We evaluated our method on selected volumes from time step 0 to 100 (2 to 4 hours of development). Qualitatively,

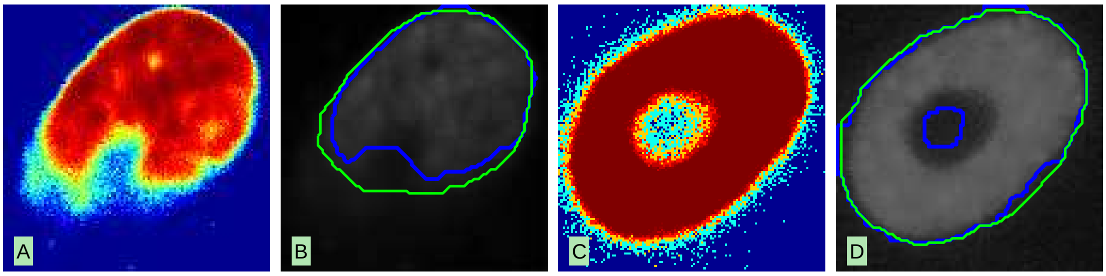


Figure 3.16.: Examples of gulf and hole free segmentation. Given inaccurate data terms (A and C), the classic Graph Cut yields gulf-like segmentation (blue line, B) or hole (blue line, D). Such limitations are successfully overcome by our method (green line, B and D).

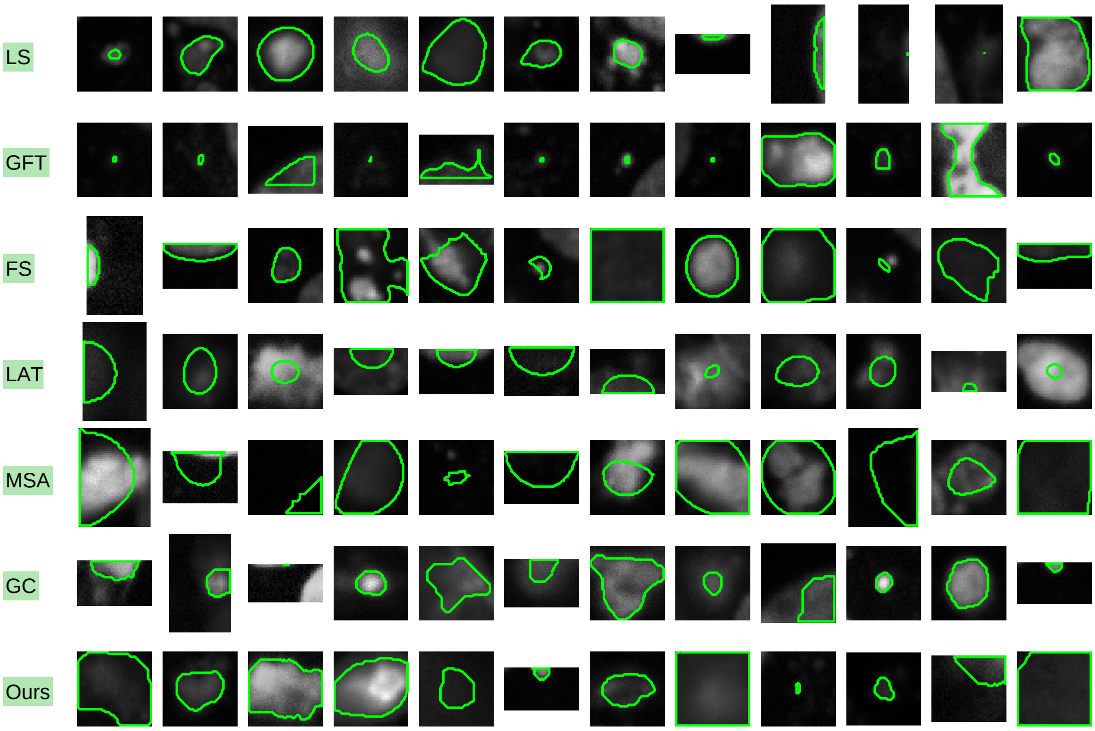


Figure 3.17.: Randomly selected examples of spurious cell nuclei.

the results are compared against LAT [76] which was first used to segment such dataset. As shown in Table 3.3, our method is superior to LAT at early time steps when the background structure is strong and at late time steps when cell nuclei clutter severely.

We also observe that the overall accuracy is increasing, which is mainly because the background structure is fading away (Fig. 3.10). This indicates that both methods handle cell nuclei clutter quite well but LAT tends to pick more false nuclei hallucinated from noisy background structures. This reflects one major weakness of the LAT method: it relies on a smoothing step at a single scale which is difficult to determine optimally and is insufficient for the entire volume.

Time Step		0	10	30	50	70	80
	N_{Nuclei}	77	354	748	922	1956	2393
LAT	Specificity	93.5%	92.9%	95.5%	95.7%	96.4%	97.9%
	Sensitivity	91.1%	93.5%	93.6%	92.7%	97.9%	98.4%
	F-measure	92.3%	93.2%	94.5%	94.2%	97.1%	97.6%
Ours	Specificity	96.1%	97.1%	98.6%	98.3%	98.2%	98.5%
	Sensitivity	96.1%	95.0%	96.6%	95.4%	98.9%	99.1%
	F-measure	96.1%	96.0%	97.6%	96.9%	98.5%	98.8%

Table 3.3.: Qualitative comparison between LAT and our method for selected time steps. Our method is constantly superior to LAT at early time steps when the background structure is strong as well as late ones when cell nuclei clutter severely. We also observe that the overall accuracy is increasing, which is mainly because the background structure is fading away (Fig. 3.10). This indicates that both methods handle cell nuclei clutter quite well but LAT appears to be more sensitive to the background disturbances.

We now compare our method to LAT, FS and GFT visually using segmentation surface rendering [118]. Our segmentation of the entire volume at time step 30 (raw data shown in Fig. 3.10) is shown in Fig. 3.18A with each segmented object colored differently. The segmentations on an example subvolume (B) is shown in C-F. Our method (C) correctly detects all the cell nuclei and yields smooth and regular shape. In comparison, LAT (D), though accurate in cell nucleus detection, hardly gives any information on their size and shape. FS (E) is subject to a few miss detections and "blockness" artifacts due to metrication artifacts [22]. GFT (F) is more sensitive to the background structure than the others. The advantage of our method becomes more apparent at late time steps when cell nuclei clutter severely. As shown in Fig. 3.19, our method (C) maintains a similar segmentation quality. FS (E) basically fails in such case because irrelevant background regions are produced. GFT (F) is obviously subject to severe under-segmentation because the boundary between cell nuclei is not always sharp.

3.5.4. Multi-object Shape Prior: fEDT Vs. bEDT

As discussed in Section 5.2, generating the GVF using a binary Euclidean distance transform (bEDT) causes interference among proximate seeds. Fig. 3.20 represents instances from the SIMCEP benchmark [86] where a large cell nucleus is close to a few small ones (A) or a false detection (D). Their respective GVF based on bEDT is shown in B and E, respectively. Apparently the large cell nucleus receives strong interference of the vector field from other seeds and the segmentation is subject to obvious shrinkage (blue dashed line in A and D). The proposed fEDT approach overcomes this limitation by adaptively adjusting the range of influence (C and F) and yields better segmentation results (green solid line in A and D).

3.5.5. Label Propagation Vs. Manual Labeling

To further study the bias of manual labeling, we compare the distribution of labeled pixels in the feature space selected by manual labeling and by the proposed labeling scheme on one SIMCEP image [86]. As shown in Fig. 3.21. The high-dimensional feature space is projected onto a 2D space by Multidimensional Scaling (MDS) [39]. Knowing the ground truth, each pixel is represented as either a blue (nucleus) or red (background) dot in the projected feature space. The pixels selected as labels are highlighted with a square (nucleus) or a circle (background). The distribution of true positives and true negatives indicates a certain degree of overlapping in the feature space, especially around coordinate $[0.6, 0.2]$. The proposed labeling scheme (top) follows an unbiased policy of labeling the positive class and thus generates very few positive labels at the overlap. The manual labeling (bottom) selects more positive labels around $[0.6, 0.2]$ and thus encourages the classifier to favor the nucleus class. To further illustrate this, we estimate the density of positive labels using Kernel Density Estimation [60] and visualize the contours of density isolines. The density of negative labels are not shown to avoid visual clutter. Obviously, manual labeling yields a higher degree of overlapping. Another observation is that manual labeling does not describe the true class distribution as completely as the proposed labeling scheme, especially for the positive class.

3.6. Discussion and Conclusions

In this paper we present a novel method that incorporates shape regularization for simultaneous segmentation of all cell nuclei in fluorescent microscope image data. Based on the Graph Cut algorithm, our method extracts smooth and regularly shaped cell nuclei and discourages unfavorable segmentations such as hole or gulf. Such improved shape estimation allows for more accurate feature extraction on the resulting cell nuclei, which is valuable for downstream tasks such as mitosis detection or cell tracking. Evaluation on three datasets with different complexity and comparison to several relevant methods show promising results by our method. Particularly, our method successfully segments the zebrafish digital embryo volumes in the presence of strong background disturbance and severe cell nuclei clutter. Furthermore, our automated method does not requires any manual labeling efforts and handles large volumes well by sparse graph construction and block processing.

We also notice a few limitations of the method and are investigating for improvements. First, the Graph Cut algorithm is subject to notable metrication error, i.e. the “blockness”. We use the 26-neighborhood system to reduce this error but this introduces more memory consumption because the graph has to be explicitly represented. Though the sparse graph construction helps to handle large volumes, the problem is not solved fundamentally. Formulating the max-flow problem in continuous space can be an alternative [8]. Second, considering the memory limitation of normal desktop computers, our implementation features block processing (with overlaps) for handling very large volumes. However, technically it does not obtain the global optimality on the entire volume since the segmentations on each block are simply merged. Methods such as dual decomposition [138] can address this problem but introduce much more computational load. In practice, we observe some block processing artifacts such as non-smooth boundary at the block bounds.

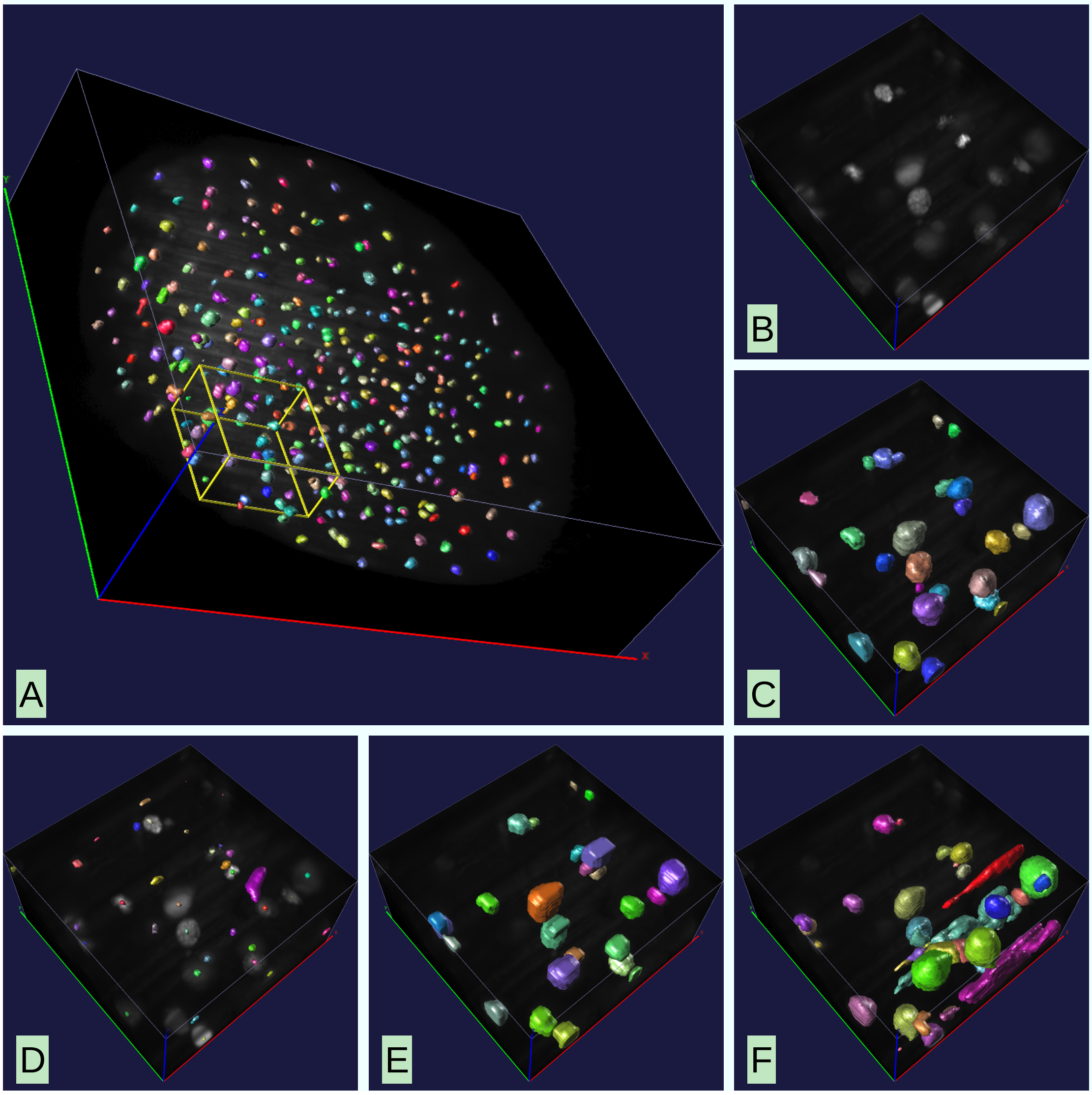


Figure 3.18.: Volume visualization (A) of the segmentation by our method overlayed on the raw data (T=30). For an example subvolume (B), our method segments smooth and regularly shaped cell nuclei (C). LAT hardly provides any information on size and shape (D). FS shows apparent blockness artifacts (E). GFT incorrectly picks several regions that belong to the background structure (F). (best viewed in color)

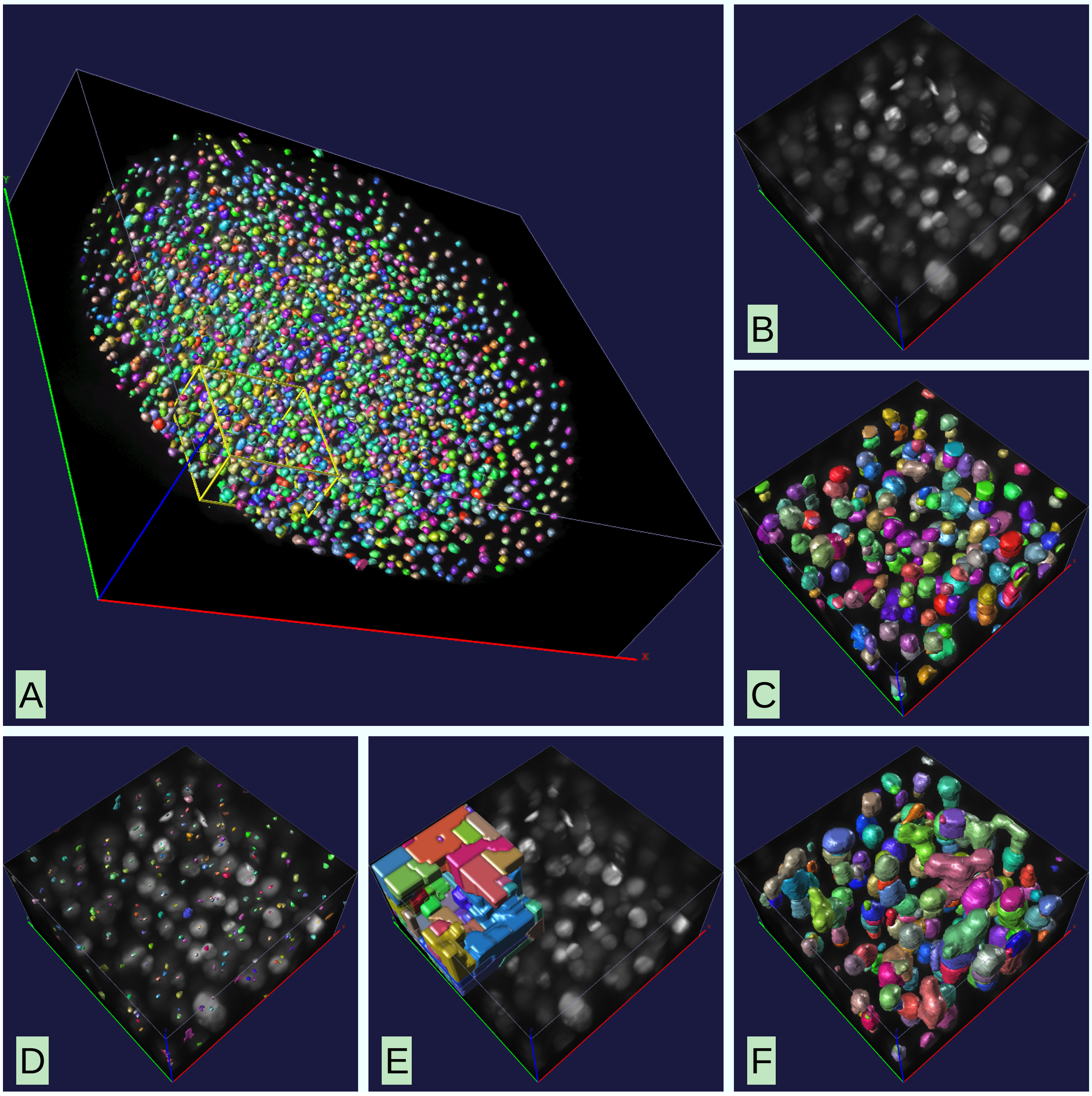


Figure 3.19.: Volume visualization (A) of the segmentation by our method overlayed on the raw data (T=80). For an example subvolume (B), our method segments smooth and regularly shaped cell nuclei (C). LAT hardly extracts any information on size and shape (D). FS basically fails for producing irrelevant background regions (E). GFT is subject to obvious under-segmentation since the boundary is not crisp (F). (best viewed in color)

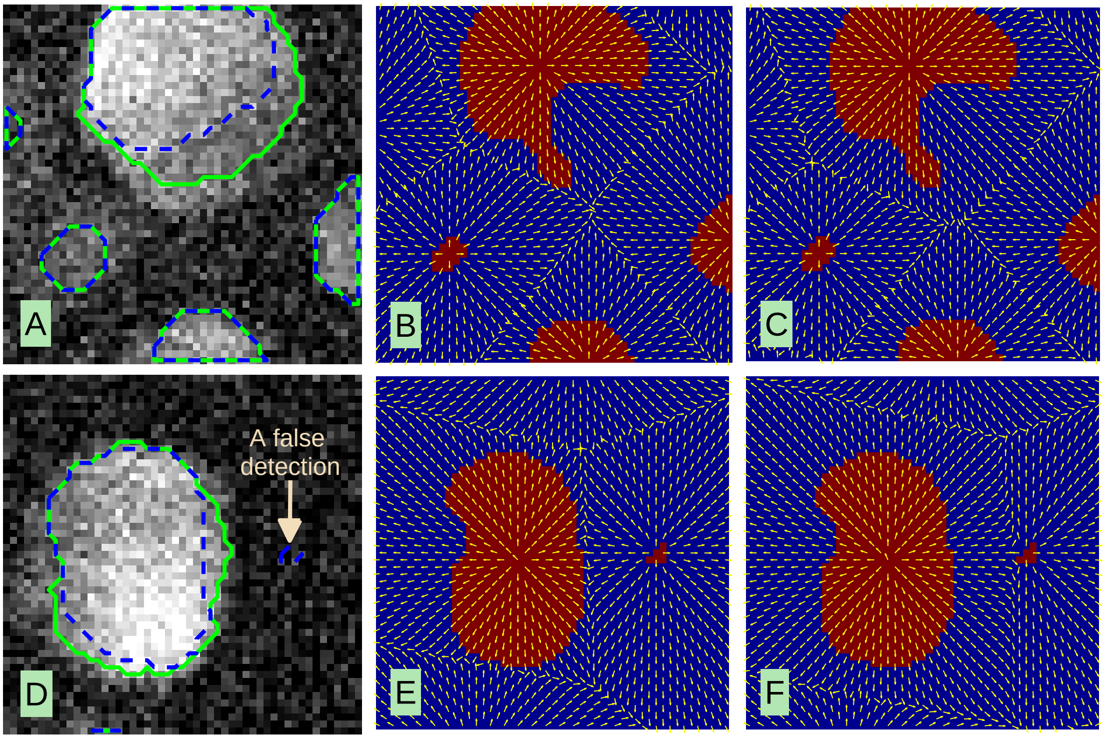


Figure 3.20.: Comparison of the segmentation result using multi-object shape prior based on bEDT (blue) and fEDT (green). See Section 3.5.4 for detailed description. (best viewed in color)

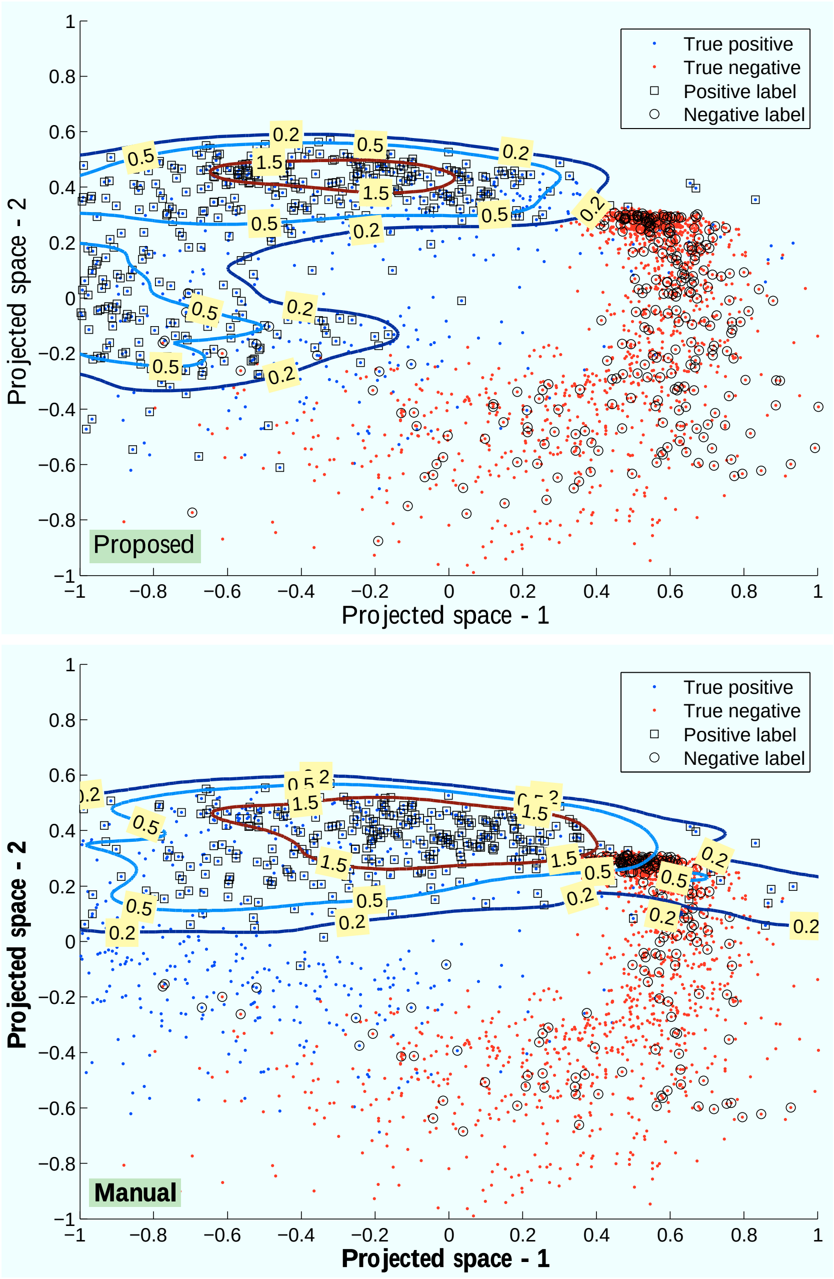


Figure 3.21.: Comparison of label distribution in feature space by manual and the proposed labeling. All pixels are projected from the original high-dimensional feature space to a 2D space by multidimensional scaling and presented as a dot. True positives (nucleus) and true negatives (background) are colored in blue and red, respectively. The pixels selected as labels for the classification are marked by a square (positive) or a circle (negative). (best viewed in color)

Chapter 4

Ranking Segmentation by Learning

Rand Index is a popular measure of segmentation quality. However, in the default shape it does not capture the varying requirement in different applications. This chapter presents a extension to Rand Index that learns the user's preference of segmentation such that the resulting new measure becomes applicable for the specific application.

Appropriate and sufficient evaluation of segmentation is a significant step in most biomedical image analysis pipelines. It is required for the comparison of segmentation algorithms as well as for the optimization of parameter settings. Existing indices or distances as measures of the segmentation focus either on the object level or on the pixel level. These measures may yield quite distinct results on the same segmentation, forcing users to consider multiple measures simultaneously or even to incorporate tremendous manual inspection. This paper proposes Cost-Sensitive Rand Index (CSRI), a measure that fuses the object- and pixel-level information and is particularly suitable for biomedical image analysis. This measure is intuitive, easy to compute, and effective in characterizing the overall segmentation quality.

4.1. Introduction

Several measures for comparing segmentations have been well studied in natural image processing [101, 147] but not clearly adopted into the field of biomedical image analysis. The fundamental differences between these fields forbid a direct import of those concepts into biomedical applications. First, humans differ in the level of detail when perceiving natural images, indicating the availability of multiple subjective references. Thus, for example, the work in [147] focuses on fusing those subjective segmentations. On the contrary, biomedical image analysis is largely objective with clear definitions of the regions of interest. Second, due to the huge variation of objects of interest in shape, texture, illumination, etc., natural image analysis ultimately delivers qualitative and semantic results (e.g. content-based image retrieval). However, biomedical images are normally acquired under restricted experiment conditions and the purpose is to enhance our understanding of the biomedical entities accurately and, most of the time, quantitatively.

Among those measures, Rand Index (RI) is becoming a popular one due to its computational efficiency and allowance for label refinement [120, 147]. Intuitively, RI can be interpreted as a ratio of the amount of pixel pairs that are labeled consistently (agreement) or not (disagreement) between the reference and the test segmentation. Formally, consider two valid segmentations $\boldsymbol{S}$ and $\boldsymbol{S}'$ of the image $\boldsymbol{I}$ that assign label $l_p \in \{0, 1, \ldots, N\}$ and $l'_p \in \{0, 1, \ldots, N'\}$ to pixel $p \in \boldsymbol{I}$ (N and N' not necessarily equal). RI is defined as

$$\lambda(\boldsymbol{S}, \boldsymbol{S}') = \frac{a+b}{a+b+c+d}, \tag{4.1}$$

where $a + b$ describe the number of agreements between $\boldsymbol{S}$ and $\boldsymbol{S}'$ while $c + d$ describe the number of disagreements between them. Formally,

- $a = \sum_{p \neq q \in \boldsymbol{I}} \mathcal{I}(l_p = l_q \wedge l'_p = l'_q)$ is the number of pixel pairs that are labeled identically in $\boldsymbol{S}$ and $\boldsymbol{S}'$;

- $b = \sum_{p \neq q \in \boldsymbol{I}} \mathcal{I}(l_p \neq l_q \wedge l'_p \neq l'_q)$ is the number of pixel pairs that are labeled differently in $\boldsymbol{S}$ and $\boldsymbol{S}'$;

- $c = \sum_{p \neq q \in \boldsymbol{I}} \mathcal{I}(l_p \neq l_q \wedge l'_p = l'_q)$ is the number of pixel pairs that are labeled differently in $\boldsymbol{S}$ but identically in $\boldsymbol{S}'$;

- $d = \sum_{p \neq q \in \boldsymbol{I}} \mathcal{I}(l_p = l_q \wedge l'_p \neq l'_q)$ is the number of pixel pairs that are labeled identically in $\boldsymbol{S}$ but differently in $\boldsymbol{S}'$.

Here, $\mathcal{I}(v)$ is an indicator function which returns 1 when the condition v is satisfied and 0 otherwise.

In biomedical image analysis, the authors in [37] used RI to quantify binary segmentations of cell nuclei images, i.e. $l_p \in \{0, 1\}$ and $l'_p \in \{0, 1\}$. However, such a binary RI measure (λ^{bin}) is simply equivalent to a weighted version of the pixel error rate[1] (ρ) and they both are insensitive to topological differences. For example, Fig. 4.1B-F illustrate several types of segmentation errors. Inspite of their apparently distinct interpretations, they have identical pixel error rate ($1 - \rho = 99.5\%$) as well as binary RI ($\lambda^{\text{bin}} = 99.0\%$), as shown in Table 4.1. In terms of segmentation quality, for example, a little Shrink error at the boundary (Fig. 4.1D) is definitely more acceptable than Merge (under-segmentation) (Fig. 4.1B). Assistance form additional measures such as error counts is demanded to finally determine the segmentation quality.

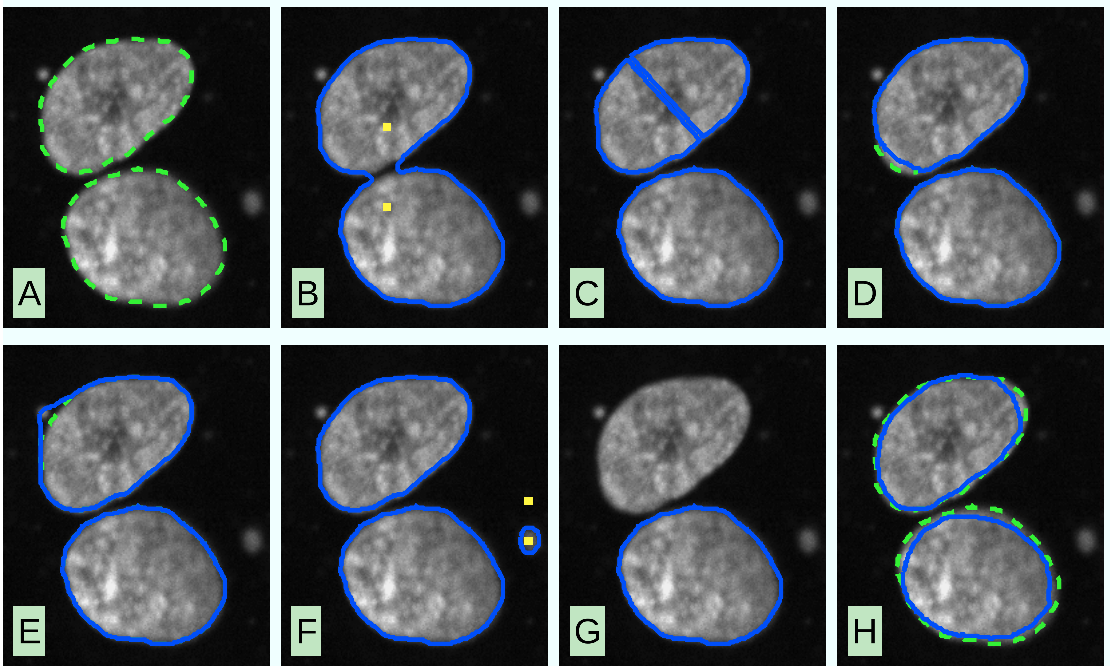


Figure 4.1.: Example of typical segmentation errors. With respect to the reference in A, B to G represent *Merge*, *Split*, *Shrink*, *Stretch*, *Spurious*, and *Missing*. H shows one example of more severe but still friendly Shrink error. Please refer to the text and Table 4.1 for more detailed description.

Fig.4.1-	B	C	D	E	F	G	H
$1-\rho$	99.5	99.5	99.5	99.5	99.5	82.7	95.3
λ^{bin}	99.0	99.0	99.0	99.0	99.0	71.4	91.0
λ^{obj}	92.1	97.6	99.1	99.2	99.3	76.7	92.0

Table 4.1.: Comparison of segmentation errors in Fig. 4.1 by pixel error rate ρ, binary RI (λ^{bin}) and object RI (λ^{obj}). The unit is %.

An improvement is to treat each object as an individual class, namely we define $l_p \in \{0, 1, \ldots, N\}$ and $l'_p \in \{0, 1, \ldots, N'\}$ where 0 represents the background the each other number represents an object (i.e. connected component) respectively for the reference $\boldsymbol{S}$ and the test segmentation $\boldsymbol{S}'$. However, as shown in Table 4.1, this object based RI (λ^{obj}) does not solve the fundamental problem: the object RI of the Merge example in Fig.4.1B is reduced to 92.1%, which is still almost identical to the friendly Shrink error in Fig.4.1H ($\lambda^{\text{obj}} = 92.0\%$). Also, the Spurious error in Fig.4.1F obtains higher value

[1] Pixel error rate refers to the percentage of pixels that are mis-segmented.

than the Shrink and Stretch in Fig.4.1D and E, which are nearly negligible in practice.

To overcome the limitations of those approaches, we propose Cost-Sensitive Rand Index (CSRI, λ^{cs}). In short, CSRI seamlessly fuses the object-level correlation and pixel-level granularity of label assignments into a single but effective measure for characterizing the overall segmentation quality. The key novelties arise from the exploration of the cross-level correspondences of labeling as well as the association of costs with them. CSRI is also computationally efficient and suitable for biomedical image analysis.

4.2. Related Work

We review related measures for segmentation evaluation mainly by their similarity in definition.

Several measures work by computing the degree of overlap of segmented regions between the segmentation and the reference. A very basic set of such measures are the error counts such as Split, Merge, Missing and Spurious [37]. However, those measures do not take into account varying degrees of label refinement. The authors in [101] proposed Local Consistency Error (LCE) and Global Consistency Error (GCE) that compute the overall error by summing up the local inconsistency at each pixels. Both LCE and GCE penalize inconsistency uniformly through the image, making them subject to the same problem as Rand Index (Fig. 4.1). The work [119] improves this by incorporating Jaccard index into the error formulation to penalize both over- and under-segmentation. Similarly, the work [29] proposed four measures based on partition distance for different cases such as over- and under-segmentation, respectively.

Another set of measures depend on a designated matching procedure of objects between the segmentation and the reference. The matching can be performed on the segmented boundaries [51] or regions [12, 132]. Such a matching strategy has a few drawbacks. First, the final output relies on the quality of the matching. Second, it introduces additional runtime which can be expensive for large images. Finally, it is unclear how to deal with unmatched objects since simplify discarding them causes loss of information.

The author in [104] proposed variation of information (VI), a measure based on the conditional entropies between the distribution of labels among clusters. As a metric, VI shows several promising properties and has been widely used for comparing data clustering results. However, its application in the image segmentation domain, where the spatial correlation matters more, is not clearly studied.

Recently, a novel *warpping error* was proposed in [69]. It is defined as the best Hamming distance between the test segmentation and the warpings of the reference. Here, warping is a sequence of pixel flips that preserve some desired topological properties and occur only within a mask. This measure, though expressing very intriguing theoretical interpretation of topological errors, is practically expensive to compute: a minimization problem has to be solved and only local minima is guaranteed. Also, the warping relies on a set of topological constraints and a mask, which makes it more complicated and less general.

Our method is based on Rand Index from the statistics community. We refer the readers to [67] for more details.

4.3. Cost-Sensitive Rand Index

4.3.1. Object-Level Correlation & Pixel-Level Granularity

The aforementioned measures are mostly defined either on object-level correlation or on pixel-level granularity. Neither of them alone is complete in revealing the true segmentation quality. In attempt to fuse such cross-level information, we observe that object-level correlation can be represented by the labeling of pixel pairs in granularity. For example, a spurious object results in disagreement of labeling between one pixel inside the spurious object and another background pixel outside (Fig. 4.1F, pixels in yellow). A merge of two objects indicates identical labeling of a pair of pixels which are labeled differently according to their respective object in the reference (Fig. 4.1B, pixels in yellow).

Based on this observation, we define six object-level correlations that describe most segmentation errors in practice: *Merge*, *Split*, *Shrink*, *Stretch*, *Spurious* and *Missing* [37]. Examples of these errors are shown in Fig. 4.1.

4.3.2. Definition of CSRI

Consider that Rand Index (RI) intrinsically measures the agreement/disagreement of pixel pair labeling and each disagreement corresponds to one object-level segmentation error defined above. Also, consider that errors may differ in degrading the ultimate segmentation quality. We derive Cost-Sensitive Rand Index (CSRI) by encoding costs into the disagreement of pixel pair labeling according the segmentation error it represents. Formally, denote $\boldsymbol{n}$ as a vector of counts of pixel pairs representing those segmentation errors and $\boldsymbol{w}$ as the additional costs associated with them. CSRI is defined as

$$\lambda^{\text{cs}}(\boldsymbol{S}, \boldsymbol{S}'; \boldsymbol{w}) \quad = \quad \frac{a+b}{a+b+c+d+\langle \boldsymbol{n}, \boldsymbol{w} \rangle}, \tag{4.2}$$

where a, b, c, d are defined in the same way as in Eq. 4.1 and
$\boldsymbol{n} = [n_{\text{Merge}}, n_{\text{Split}}, n_{\text{Shrink}}, n_{\text{Stretch}}, n_{\text{Spurious}}, w_{\text{Missing}}]'$,
$\boldsymbol{w} = [w_{\text{Merge}}, w_{\text{Split}}, w_{\text{Shrink}}, w_{\text{Stretch}}, w_{\text{Spurious}}, w_{\text{Missing}}]'$.

In addition, $\boldsymbol{w}$ satisfies $\forall w \in \boldsymbol{w}, w > -1$ so $\lambda^{\text{cs}} \in [0, 1]$. Obviously, RI has a uniform cost of 1 (i.e., $\boldsymbol{w} \equiv \boldsymbol{0}$) while CSRI adjusts the costs for particular segmentation errors to $\boldsymbol{1} + \boldsymbol{w}$. Note that the error counts in $\boldsymbol{n}$ have already been expressed in c and d (thus $\boldsymbol{w}$

is additive). We keep parts of the RI definition (Eq. 4.1) so that the connection between CSRI and RI can be immediately clear.

4.3.3. Learning the Costs using Generalized Linear Model

The costs $\boldsymbol{w}$ can be manually parameterized using heuristics or obtained by learning from training examples. To learn $\boldsymbol{w}$, we first simplify Eq. 4.2 as

$$\begin{aligned}\lambda^{\text{cs}}(\boldsymbol{S},\boldsymbol{S}';\boldsymbol{w}) &= \left[(1+\frac{c+d}{a+b})+\langle\frac{\boldsymbol{n}}{a+b},\boldsymbol{w}\rangle\right]^{-1} \\ &= \langle\boldsymbol{x},\boldsymbol{\beta}\rangle^{-1} \qquad (4.3)\end{aligned}$$

where $\boldsymbol{x}=\left[1+\frac{c+d}{a+b},\frac{\boldsymbol{n}}{a+b}\right]$ and $\boldsymbol{\beta}=[1,\boldsymbol{w}]$ are extended vectors of normalized counts and costs, respectively.

Eq. 4.3 shows that CSRI is defined based on a Generalized Linear Model (GLM) whose error term follows a Gamma distribution [105]. The maximum-likelihood estimation of such model can be obtained using Iteratively Reweighted Least Squares (IRS). In particular, given K training segmentation pairs $\{\boldsymbol{S}_k,\boldsymbol{S}'_k\}$ and their desired index value $\hat{\lambda}^{\text{cs}}_k$ assigned by the experts, $k\in K$, the learning problem is formulated as

$$\begin{aligned}&\underset{\boldsymbol{\beta}}{\text{minimize}} \quad \left\{\sum_{k=1}^{K}\left(\hat{\lambda}^{\text{cs}}_k-\langle\boldsymbol{x},\boldsymbol{\beta}\rangle^{-1}\right)^2\right\} \\ &\text{subject to} \quad \beta_1=1 \text{ and } \forall\beta\in\boldsymbol{\beta}/\{\beta_1\},\beta>-1.\end{aligned} \qquad (4.4)$$

4.3.4. Efficient Computation with Contingency Table

The efficient computation of RI has been exploited in [67]. Similarly, CSRI can also be analytical formulated by representing the labeling consistency between $\boldsymbol{S}$ and $\boldsymbol{S}'$ with a contingency table, as shown in Table 4.2. The row and column represent labels from those two segmentations, respectively. The element n_{ij} represents the amount of elements that are labeled as i in $\boldsymbol{S}$ and as j in $\boldsymbol{S}'$.

Let $\boldsymbol{S}$ and $\boldsymbol{S}'$ represent the segmentation and reference, respectively, and let label 0 be the background. All the counts in Eq. 4.2 can be analytically represented by combinations of elements in Table 4.2. For example, a pixel pair $\{p,q\}$ that is labeled as $\{l_p=1,l_q=2\}$ in $\boldsymbol{S}$ but as $\{l'_p=1,l'_q=1\}$ in $\boldsymbol{S}'$ (i.e. disagreement due to Merge) corresponds to one instance in n_{11} and one instance in n_{21}. Thus, the count of all such pixel pairs is $n_{11}\cdot n_{21}$ and the count of all pixel pairs of Merge is $\sum_{i=1}^{N}\sum_{j=1}^{N'}\sum_{u=i+1}^{N}n_{ij}\cdot n_{uj}$. All the rest error counts can be represented similarly (Table 4.3). The formulas for computing a, b, c and d are given in [67]. The computational complexity has an upper bound of $\mathcal{O}(n+(NN')^2)$, where n is the total number of pixels

l' l	0	1	2	$\dots$	N'
0	n_{00}	n_{01}	n_{02}	$\dots$	$n_{0N'}$
1	n_{10}	n_{11}	n_{12}	$\dots$	$n_{1N'}$
2	n_{20}	n_{21}	n_{22}	$\dots$	$n_{2N'}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
N	n_{N1}	n_{N1}	n_{N2}	$\dots$	$n_{NN'}$

Table 4.2.: An example of contingency table showing the consistency of labeling between two segmentations.

Error	$\{l_p, l_q\} \to \{l'_p, l'_q\}$	Count Formula n_{Error}
Merge	$\{1, 2\} \to \{1, 1\}$	$\sum_{i=1}^{N} \sum_{j=1}^{N'} \sum_{u=i+1}^{N} n_{ij} \cdot n_{uj}$
Split	$\{1, 1\} \to \{1, 2\}$	$\sum_{i=1}^{N} \sum_{j=1}^{N'} \sum_{v=j+1}^{N'} n_{ij} \cdot n_{iv}$
Shrink	$\{1, 1\} \to \{1, 0\}$	$\sum_{i=1}^{N} \sum_{v=1}^{N'} n_{i0} \cdot n_{iv}$
Stretch	$\{1, 0\} \to \{1, 1\}$	$\sum_{u=1}^{N} \sum_{j=1}^{N'} n_{0j} \cdot n_{uj}$
Spurious	$\{0, 0\} \to \{0, 1\}$	$\sum_{j=0}^{N'} \sum_{v=j+1}^{N'} n_{0j} \cdot n_{0v}$
Missing	$\{0, 1\} \to \{0, 0\}$	$\sum_{i=0}^{N} \sum_{u=i+1}^{N} n_{i0} \cdot n_{u0}$

Table 4.3.: Formulas for the counts of pixel pairs that represent segmentation errors. The middle column shows examples of the corresponding disagreement on pixel pair labeling.

4.4. Experiment and Results

We experimented CSRI for the comparison of four segmentation methods (named as Method 1-4) on the hand-labeled U2OS cell nuclei images from [37]. We address two important issues as follows. First, evaluation on the full image fails to reflect the true performance of those methods due to the sparse distribution of nuclei. Therefore, we partition the images and evaluate on the resulting local patches which contain one or a few nuclei. Second, to allow direct comparison of CSRI (or other variants) between patches with varying background, we fix the size of the background class as the average size of foreground objects in the reference.

We labeled 100 patches of representative segmentation quality for the cost learning, following these guidelines: (I) Merge is the most unfavored, followed by Split; (II) Spurious is more acceptable than Missing; (III) We tolerate friendly Shrink unless too many pixels are lost (the same applies to Stretch). Accordingly, the patches in Fig. 4.1 were assigned with desired index value of 30.0%(B), 60.0%(C), 99.0%(D), 99.0%(E), 85.0%(F), 50.0%(G), and 95.0%(H). Table 4.4 shows the learned costs, which suggests: (i) Shrink and Stretch errors are basically canceled out during the computation; (ii) w_{Spurious} is greater than w_{Missing} because Missing occurs when all pixels belonging to one nucleus are lost while Spurious is counted even for a small false object.

w_{Merge}	w_{Split}	w_{Shrink}	w_{Stretch}	w_{Spurious}	w_{Missing}
8.08	3.98	-1.00	-1.00	0.24	-0.36

Table 4.4.: Learned costs from manually labeled patches.

We then computed the CSRI (λ^{cs}) for 500 patches (out of 1549) and plotted them with their associated object RI (λ^{obj}) values for comparison. Interesting patterns emerge, as shown in Fig. 4.2. First, for the points close to the diagonal line ($\lambda^{\text{cs}} \equiv \lambda^{\text{obj}}$) the CSRI and RI form a correlation close to linear. Those are mostly the patches with only a single nucleus. Second, Method 1 and Method 2 experience huge drops in CSRI for many patches. Random samples from their respective region of drop (Fig. 4.2, black/red polygon) reveal their susceptibleness to Merge or Split error (Fig. 4.3, 1st/2nd row). Such truth is not well expressed by λ^{obj} due to its uniform penalty. Third, Method 3 obtains low scores in both measures mainly due to severe loss of foreground pixels (see sampled patches in Fig. 4.3, 3rd row). Finally, Method 4 largely gains score in CSRI mainly because it is more robust against Merge/Split and CSRI is tolerant of Shrink (see sampled patches in Fig. 4.3, 4th row).

Table 4.5 shows the standard error of CSRI and object RI yielded by those methods. If we had used object RI as the quality measure, we may have drawn a false conclusion that Method 1 and Method 2 are more favorable than Method 4. This is avoided when using CSRI, which clearly separates Merge/Split errors from friendly Shrink error with concomitant suggestion that Method 1 and 2 are not robust and Method 4 is overall superior than the rest.

Meas.	Method 1	Method 2	Method 3	Method 4
λ^{obj}	88.7 ± 9.5	88.6 ± 7.0	48.2 ± 6.7	82.2 ± 9.5
λ^{cs}	86.0 ± 20.6	80.5 ± 19.9	46.5 ± 6.0	85.6 ± 10.7

Table 4.5.: Comparison of standard errors by object RI and CSRI (unit: %).

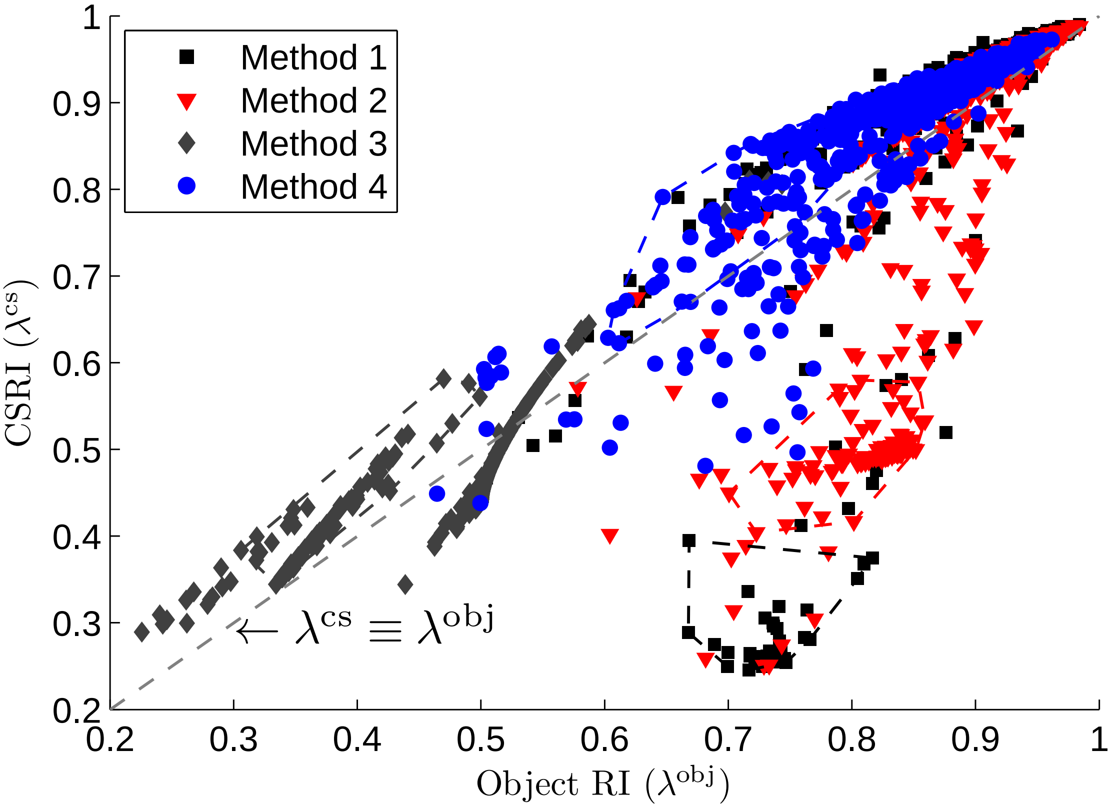


Figure 4.2.: CSRI Vs. object RI for 500 randomly sampled patches. Each point represents the CSRI (y-axis) and RI (x-axis) of one image patch segmented by a specific method. The symbols represent different segmentation methods.

4.5. Discussion and Conclusions

We represented a novel Cost-Sensitive Rand Index for the evaluation of segmentation and demonstrated its applicability in cell nuclei segmentation. Our measure encodes object-level meaningful costs into pixel-level granularity and incorporates a learning procedure to suit various biomedical applications. For example, segmenting cell nuclei images particularly disfavors Merge/Split but tolerates friendly Shrink/Stretch at the boundary. However, boundary accuracy becomes very critical in computer assisted medical treatment. Thus, the costs can be re-learned to cope with such critical demand.

We also notice the limitations of the proposed approach. First, it requires additional efforts on the learning of costs when applied to a new application. Second, unlike the warping error [69], it is insensitive to complicated topological errors such as holes or handles.

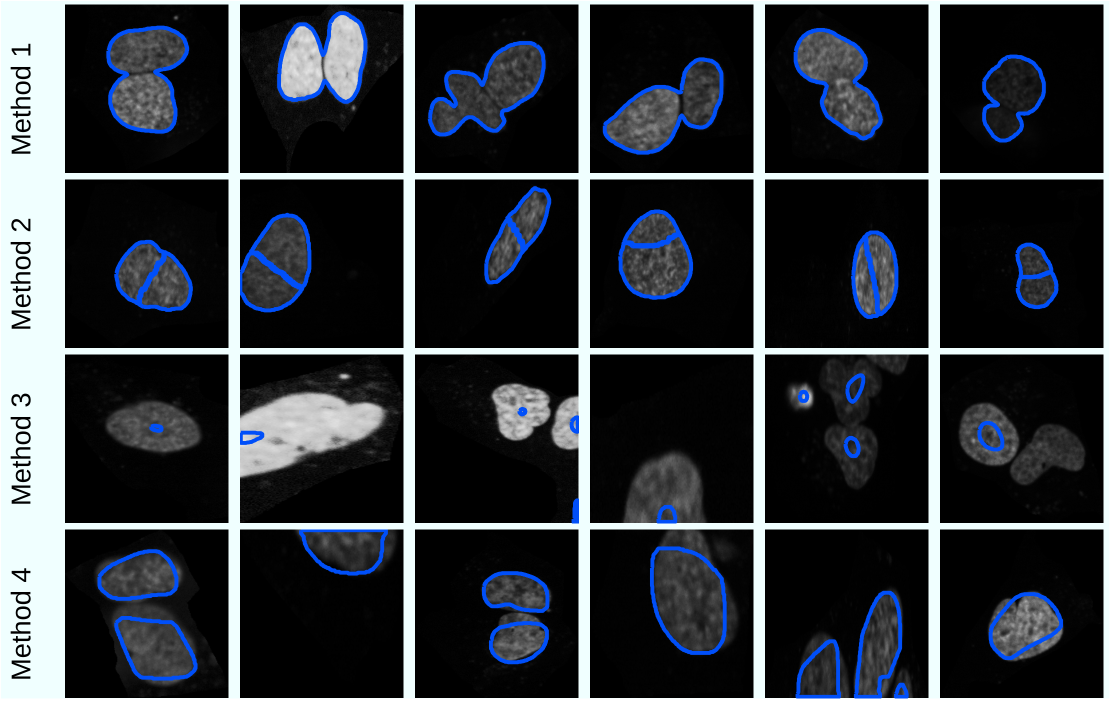


Figure 4.3.: Each method produces typical segmentation errors, as evidenced by the non-overlapping distributions in Fig. 4.2 and shown by exemplary cases here.

Chapter 5

Structured Learning for Cell Tracking

The prior learning problem becomes more challenging when the input and output have structured data. This chapter presents a structured learning based cell tracking algorithm that learns from manually annotated tracks to optimize the parameters and brings apparent improvement over state-of-the-art methods for the same problem.

Reliable cell tracking in time-lapse microscopic image sequences is important for modern biomedical research. Existing cell tracking methods are usually kept simple and use only a small number of features to allow for manual parameter tweaking or grid search. We propose a structured learning approach that allows to learn optimum parameters automatically from a training set. This allows for the use of a richer set of features which in turn affords improved tracking compared to recently reported methods on a benchmark sequence. Matlab source code is made available at http://hci.iwr.uni-heidelberg.de/MIP/Software/.

5.1. Introduction and Related Work

One distinguishing property of life is its temporal dynamics, and it is hence only natural that time lapse experiments play a crucial role in current research on signaling pathways, drug discovery and developmental biology [90]. Such experiments yield a very large number of images, and reliable automated cell tracking emerges naturally as a prerequisite for further quantitative analysis.

Even today, cell tracking remains a challenging problem in dense populations, in the presence of complex behavior or when image quality is poor. Existing cell tracking methods can broadly be categorized as deformable models, stochastic filtering and object association. Deformable models combine detection, segmentation and tracking by initializing a set of models (e.g. active contours) in the first frame and updating them in subsequent frames (e.g. [90, 43]). Large displacements are difficult to capture with this class of techniques and are better handled by state space models, e.g. in the guise of stochastic filtering. The latter also allows for sophisticated observation models (e.g. [103]). Stochastic filtering builds on a solid statistical foundation, but is often limited in practice due to its high computational demands. Object association methods approximate and simplify the problem by separating the object detection and assignment steps: once object candidates have been detected and characterized, a second step suggests assignments between object candidates at different frames. This class of methods scales well [111, 87, 73] and allows the tracking of thousands of cells in 3D [96].

All of the above approaches contain parameters which may be tedious or difficult to adjust. Our work is in the spirit of [92, 116] in that we undertake to learn the parameters empirically from a training set.

We first present an extended formulation of the object association models proposed in the literature. This generalization improves the expressiveness of the model, but also increases the number of parameters. We hence, secondly, propose to use structured learning to automatically learn optimum parameters from a training set, and hence profit fully from this richer description. An evaluation shows that this framework inherits the runtime advantage of object association while addressing many of its limitations.

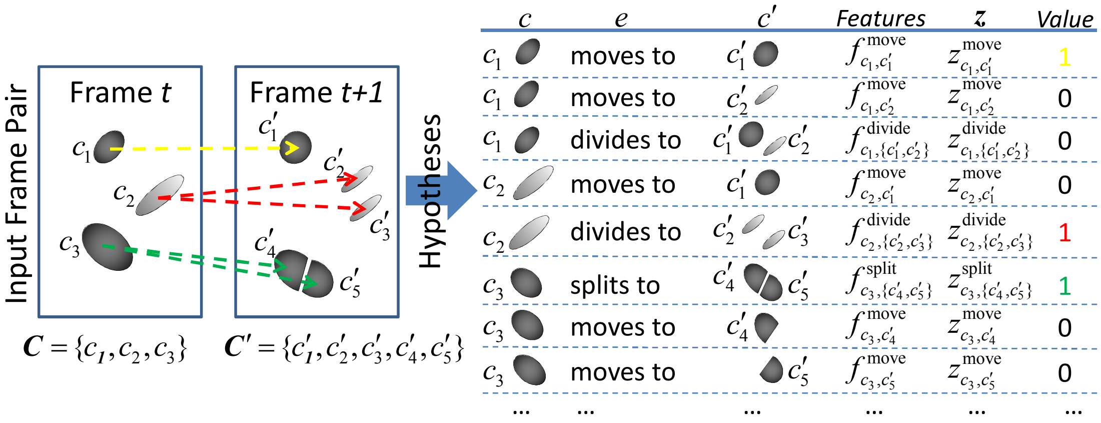
5.2. Structured Learning for Cell Tracking

5.2.1. Assignment Hypotheses and Scoring

We assume that a previous detection and segmentation step has identified object candidates in all frames, see Fig. 5.1. We set out to find that set of object associations that best explains these observations. To this end, we admit the following set $\boldsymbol{E}$ of standard events [111, 73]: a cell can *move* or *divide* and it can *appear* or *disappear*. In addition, we allow two cells to (seemingly) *merge*, to account for occlusion or undersegmentation; and a cell can (seemingly) *split*, to allow for the lifting of occlusion or oversegmentation. These additional hypotheses are useful to account for the errors that typically occur in the detection and segmentation step in crowded or noisy data. The distinction between division and split is reasonable given that typical fluorescence stains endow the anaphase with a distinctive appearance.

Given a pair of object candidate lists $\boldsymbol{x} = \{\boldsymbol{C}, \boldsymbol{C}'\}$ in two neighboring frames, there is a multitude of possible assignment hypotheses, see Fig. 5.1. We have two tasks: firstly, to allow only consistent assignments (e.g. making sure that each cell in the second frame is accounted for only once); and secondly to identify, among the multitude of consistent hypotheses, the one that is most compatible with the observations, and with what we have learned from the training data.

We express this compatibility of the association between $c \in \mathcal{P}(\boldsymbol{C})$ and $c' \in \mathcal{P}(\boldsymbol{C}')$



c	e	c'	Features	z	Value
c_1	moves to	c'_1	$f^{\text{move}}_{c_1,c'_1}$	$z^{\text{move}}_{c_1,c'_1}$	1
c_1	moves to	c'_2	$f^{\text{move}}_{c_1,c'_2}$	$z^{\text{move}}_{c_1,c'_2}$	0
c_1	divides to	c'_1 c'_2	$f^{\text{divide}}_{c_1,\{c'_1,c'_2\}}$	$z^{\text{divide}}_{c_1,\{c'_1,c'_2\}}$	0
c_2	moves to	c'_1	$f^{\text{move}}_{c_2,c'_1}$	$z^{\text{move}}_{c_2,c'_1}$	0
c_2	divides to	c'_2 c'_3	$f^{\text{divide}}_{c_2,\{c'_2,c'_3\}}$	$z^{\text{divide}}_{c_2,\{c'_2,c'_3\}}$	1
c_3	splits to	c'_4 c'_5	$f^{\text{split}}_{c_3,\{c'_4,c'_5\}}$	$z^{\text{split}}_{c_3,\{c'_4,c'_5\}}$	1
c_3	moves to	c'_4	$f^{\text{move}}_{c_3,c'_4}$	$z^{\text{move}}_{c_3,c'_4}$	0
c_3	moves to	c'_5	$f^{\text{move}}_{c_3,c'_5}$	$z^{\text{move}}_{c_3,c'_5}$	0
...	...	...	...	...	...

Figure 5.1.: Toy example: two sets of object candidates, and a small subset of the possible assignment hypotheses. One particular interpretation of the scene is indicated by colored arrows (left) or equivalently by a configuration of binary indicator variables $\boldsymbol{z}$ (rightmost column in table).

by event $e \in \boldsymbol{E}$ as an inner product $\left\langle \boldsymbol{f}^e_{c,c'} \boldsymbol{w}^e \right\rangle$. Here, $\boldsymbol{f}^e_{c,c'}$ is a feature vector that characterizes the discrepancy (if any) between object candidates c and c'; and $\boldsymbol{w}^e$ is a parameter vector that encodes everything we have learned from the training data. Summing over all object candidates in either of the frames and over all types of events gives the following compatibility function:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{w}) = \sum_{e \in \boldsymbol{E}} \sum_{c \in \mathcal{P}(\boldsymbol{C})} \sum_{c' \in \mathcal{P}(\boldsymbol{C}')} \langle \boldsymbol{f}^e_{c,c'}, \boldsymbol{w}^e \rangle z^e_{c,c'} \tag{5.1}$$

$$\text{s. t.} \sum_{e \in \boldsymbol{E}} \sum_{c \in \mathcal{P}(\boldsymbol{C})} z^e_{c,c'} = 1 \text{ and } \sum_{e \in \boldsymbol{E}} \sum_{c' \in \mathcal{P}(\boldsymbol{C}')} z^e_{c,c'} = 1 \text{ with } z^e_{c,c'} \in \{0,1\} \tag{5.2}$$

The constraints in the last line involve binary indicator variables $\boldsymbol{z}$ that reflect the consistency requirements: each candidate in the first frame must have a single fate, and each candidate from the second frame a unique history. As an important technical detail, note that $\mathcal{P}(\boldsymbol{C}) := \boldsymbol{C} \cup (\boldsymbol{C} \otimes \boldsymbol{C})$ is a set comprising each object candidate, as well as all ordered pairs of object candidates from a frame[1]. This allows us to conveniently subsume cell divisions, splits and mergers in the above equation. Overall, the compatibility function $\mathcal{L}(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{w})$ states how well a set of assignments $\boldsymbol{z}$ matches the observations $\boldsymbol{f}(\boldsymbol{x})$ computed from the raw data $\boldsymbol{x}$, given the knowledge $\boldsymbol{w}$ from the training set.

The remaining tasks, discussed next, are how to learn the parameters $\boldsymbol{w}$ from the training data (section 5.2.2); given these, how to find the best possible assignments $\boldsymbol{z}$

[1] For the example in Fig. 5.1, $\mathcal{P}(\boldsymbol{C}) = \{c_1, c_2, c_3, \{c_1, c_2\}, \{c_1, c_3\}, \{c_2, c_3\}\}$.

(section 5.2.3); and finding useful features (section 5.2.6).

5.2.2. Structured Max-Margin Parameter Learning

In learning the parameters automatically from a training set, we pursue two goals: first, to go beyond manual parameter tweaking in obtaining the best possible performance; and second, to make the process as facile as possible for the user. This is under the assumption that most experimentalists find it easier to specify what a correct tracking should look like, rather than what value a more-or-less obscure parameter should have.

Given N training frame pairs $\boldsymbol{X} = \{\boldsymbol{x}_n\}$ and their correct assignments $\boldsymbol{Z}^* = \{\boldsymbol{z}_n^*\}$, $n = 1, \ldots, N$, the best set of parameters is the optimizer of

$$\arg\min_{\boldsymbol{w}} \mathcal{R}(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{Z}^*) + \lambda\Omega(\boldsymbol{w}) \tag{5.3}$$

Here, $\mathcal{R}(\boldsymbol{w; X, Z}^*)$ measures the empirical loss of the current parametrization $\boldsymbol{w}$ given the training data $\boldsymbol{X}, \boldsymbol{Z}^*$. To prevent overfitting to the training data, this is complemented by the regularizer $\Omega(\boldsymbol{w})$ that favors parsimonious models. We use L_1 or L_2 regularization ($\Omega(\boldsymbol{w}) = ||\boldsymbol{w}||_p^p/p$, $p = \{1, 2\}$), i.e. a measure of the length of the parameter vector $\boldsymbol{w}$. The latter is often used for its numerical efficiency, while the former is popular thanks to its potential to induce sparse solutions (i.e., some parameters can become zero). The empirical loss is given by $\mathcal{R}(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{Z}^*) = \frac{1}{N}\sum_{i=1}^{N} \Delta(\boldsymbol{z}_n^*, \hat{\boldsymbol{z}}_n(\boldsymbol{w}; \boldsymbol{x}_n))$. Here $\Delta(\boldsymbol{z}^*, \hat{\boldsymbol{z}})$ is a loss function that measures the discrepancy between a true assignment $\boldsymbol{z}^*$ and a prediction by specifying the fraction of missed events w.r.t. the ground truth:

$$\Delta(\boldsymbol{z}^*, \hat{\boldsymbol{z}}) = \frac{1}{|\boldsymbol{z}^*|} \sum_{e \in \boldsymbol{E}} \sum_{c \in \mathcal{P}(\boldsymbol{C})} \sum_{c' \in \mathcal{P}(\boldsymbol{C}')} z_{c,c'}^{*e}(1 - \hat{z}_{c,c'}^{e}). \tag{5.4}$$

This decomposable function allows for exact inference when solving Eq. 5.5 [28].

Importantly, both the input (objects from a frame pair) and output (associations between objects) in this learning problem are *structured*. We hence resort to max-margin structured learning [146]. In particular, we attempt to find the decision boundary that maximizes the margin between the correct assignment $\boldsymbol{z}_n^*$ and the closest runner-up solution. An equivalent formulation is the condition that the score of $\boldsymbol{z}_n^*$ be greater than that of any other solution. To allow for regularization, one can relax this constraint by introducing slack variables ξ_n, which finally yields the following objective function for the max-margin structured learning problem from Eq. 5.3:

$$\begin{array}{ll} \arg\min\limits_{\boldsymbol{w}, \boldsymbol{\xi} \geq \boldsymbol{0}} & \frac{1}{N}\sum_{n=1}^{N} \xi_n + \lambda\Omega(\boldsymbol{w}) \\ \text{s. t.} & \forall n, \forall \hat{\boldsymbol{z}}_n \in \mathcal{Z}_n : \mathcal{L}(\boldsymbol{x}_n, \boldsymbol{z}_n^*; \boldsymbol{w}) - \mathcal{L}(\boldsymbol{x}_n, \hat{\boldsymbol{z}}_n; \boldsymbol{w}) \geq \Delta(\boldsymbol{z}_n^*, \hat{\boldsymbol{z}}_n) - \xi_n, \end{array} \tag{5.5}$$

where $\mathcal{Z}_n$ is the set of possible consistent assignments and $\Delta(\boldsymbol{z}_n^*, \hat{\boldsymbol{z}}_n) - \xi_n$ is known as "margin-rescaling" [146]. Intuitively, it pushes the decision boundary further away from the "bad" solutions with high losses.

5.2.3. Inference with Bundle Method

Since Eq. 5.5 involves an exponential number of constraints, the learning problem cannot be represented explicitly, let alone solved directly. We thus resort to the *bundle method* [144] which, in turn, is based on the *cutting-planes* approach [146].

The basic idea is as follows: Start with some parametrization $\boldsymbol{w}$ and no constraints. Iteratively find, first, the optimum assignments for the current $\boldsymbol{w}$ by solving, for all n, $\hat{\boldsymbol{z}}_n = \arg\max_{\boldsymbol{z}}\{\mathcal{L}(\boldsymbol{x}_n, \boldsymbol{z}; \boldsymbol{w}) + \Delta(\boldsymbol{z}_n^*, \boldsymbol{z})\}$. Use all these $\hat{\boldsymbol{z}}_n$ to identify the most violated constraint, and add it to Eq. 5.5. Update $\boldsymbol{w}$ by solving Eq. 5.5 (with added constraints), then find new best assignments, etc. pp. For a given parametrization, the optimum assignments can be found by integer linear programming (ILP) [87, 111, 73].

Considering a set of events $\boldsymbol{\mathcal{E}} = \{\mathcal{E}_1, \mathcal{E}_2, \ldots\}$, the overall likelihood can be reformulated as follows:

$$\begin{aligned} \mathcal{L}(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{w}) &= \textstyle\sum_{\mathcal{E}} \sum_{\boldsymbol{s},\boldsymbol{s}'} \langle \boldsymbol{f}^{\mathcal{E}}_{\boldsymbol{s},\boldsymbol{s}'}, \boldsymbol{w}^{\mathcal{E}} \rangle z^{\mathcal{E}}_{\boldsymbol{s},\boldsymbol{s}'} \\ &= \textstyle\sum_{\mathcal{E}} \left\langle \sum_{\boldsymbol{s},\boldsymbol{s}'} z^{\mathcal{E}}_{\boldsymbol{s},\boldsymbol{s}'} \boldsymbol{f}^{\mathcal{E}}_{\boldsymbol{s},\boldsymbol{s}'}, \boldsymbol{w}^{\mathcal{E}} \right\rangle \\ &= \langle \boldsymbol{\Phi}(\boldsymbol{x}, \boldsymbol{z}), \boldsymbol{w} \rangle, \end{aligned} \tag{5.6}$$

where $\boldsymbol{\Phi}(\boldsymbol{z})$ is the so-called joint feature vector [146] and $\boldsymbol{w}$ is the joint set of weights:

$$\boldsymbol{\Phi}(\boldsymbol{x}, \boldsymbol{z}) = \begin{bmatrix} \sum\limits_{\boldsymbol{s},\boldsymbol{s}'} z^{\mathcal{E}_1}_{\boldsymbol{s},\boldsymbol{s}'} \boldsymbol{f}^{\mathcal{E}_1}_{\boldsymbol{s},\boldsymbol{s}'} \\ \sum\limits_{\boldsymbol{s},\boldsymbol{s}'} z^{\mathcal{E}_2}_{\boldsymbol{s},\boldsymbol{s}'} \boldsymbol{f}^{\mathcal{E}_2}_{\boldsymbol{s},\boldsymbol{s}'} \\ \ldots \end{bmatrix} \text{ and } \boldsymbol{w} = \begin{bmatrix} \boldsymbol{w}^{\mathcal{E}_1} \\ \boldsymbol{w}^{\mathcal{E}_2} \\ \ldots \end{bmatrix}. \tag{5.7}$$

Note that here all the features $\boldsymbol{f}^{\mathcal{E}}$ and weights $\boldsymbol{w}^{\mathcal{E}}$ are column vectors.

The pseudocode for the bundle method is shown in Alg. 6 [28, 144]. In terms of the objective function $J(\boldsymbol{w})$ that the learning attempts to minimize, it is equivalent to approximating the empirical loss term $\frac{1}{N}\sum_{n=1}^{N}\{\Delta(\boldsymbol{z}_n^*, \hat{\boldsymbol{z}}_n; \boldsymbol{w})\}$ with a piecewise linear lower bound $\mathbf{max}\left(0, \max\limits_{j\leq i+1}\{\langle \boldsymbol{w}, \boldsymbol{a}_j\rangle + b_j\}\right)$ [144]. The approximation gap, i.e. the smallest difference between the $J(\boldsymbol{w})$ and the approximation $\hat{J}(\boldsymbol{w})$, is used as the stopping criterion. Details on a few key optimization steps are provided in the following sections.

Algorithm 6: Bundle method for learning the weight vector $\boldsymbol{w}$.

```
Input: Training examples $\{\boldsymbol{x}, \boldsymbol{z}^*\}_{n=1,\dots,N}$, initial weight vector $\boldsymbol{w}_1$,
       regularization strength $\lambda$, approximation gap tolerance $\epsilon$.
Output: Weight vector $\boldsymbol{w}$.
Definition:
$\boldsymbol{\Psi}(\boldsymbol{x}_n, \boldsymbol{z}_n^*, \hat{\boldsymbol{z}}_n) := \boldsymbol{\Phi}(\boldsymbol{x}_n, \boldsymbol{z}_n^*) - \boldsymbol{\Phi}(\boldsymbol{x}_n, \hat{\boldsymbol{z}}_n)$ // Joint feature difference.
$J(\boldsymbol{w}) = \lambda\Omega(\boldsymbol{w}) + \frac{1}{N}\sum_{n=1}^{N}\{\Delta(\boldsymbol{z}_n^*, \hat{\boldsymbol{z}}_n; \boldsymbol{w})\}$ // Objective function.
$H(\boldsymbol{x}_n, \boldsymbol{z}_n^*, \hat{\boldsymbol{z}}_n; \boldsymbol{w}) = \langle \boldsymbol{\Phi}(\boldsymbol{x}_n, \hat{\boldsymbol{z}}_n), \boldsymbol{w} \rangle + \Delta(\boldsymbol{z}_n^*, \hat{\boldsymbol{z}}_n; \boldsymbol{w})$ // Likelihood+loss.
$\hat{J}(\boldsymbol{w}) = \lambda\Omega(\boldsymbol{w}) + \mathbf{max}\left(0, \max_{j \leq i+1}\{\langle \boldsymbol{w}, \boldsymbol{a}_j \rangle + b_j\}\right)$ // Approximation.
Initialization:
$i = 1$ // Iteration index.
$\boldsymbol{W} = \emptyset$ // Matrix of intermediate solutions.
$\boldsymbol{A} = \emptyset$ // Matrix of gradients.
$\boldsymbol{b} = \emptyset$ // Matrix of offsets.
repeat
    // Determine the most violated constraint.
    for $n \in \{1, \dots, N\}$ do $\hat{\boldsymbol{z}}_n = \arg\max_{\boldsymbol{z}}\{H(\boldsymbol{x}_n, \boldsymbol{z}_n^*, \boldsymbol{z}; \boldsymbol{w}_i)\}$
    // Compute the gradient and offset.
    $\boldsymbol{a}_i = -\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\Psi}(\boldsymbol{x}_n, \boldsymbol{z}_n^*, \hat{\boldsymbol{z}}_n)$
    $b_i = \frac{1}{N}\sum_{n=1}^{N}\{\Delta(\boldsymbol{z}_n^*, \hat{\boldsymbol{z}}_n; \boldsymbol{w}_i) - \langle \boldsymbol{\Psi}(\boldsymbol{x}_n, \boldsymbol{z}_n^*, \hat{\boldsymbol{z}}_n), \boldsymbol{w}_i \rangle\}$
    // Update the matrix of gradients and offsets.
    $\boldsymbol{A} = [\boldsymbol{A}, \boldsymbol{a}_i]$
    $\boldsymbol{b} = \left[\boldsymbol{b}^T, b_i\right]^T$
    // Update the model parameters.
    $\boldsymbol{w}_{i+1} = \arg\min_{\boldsymbol{w}}\left\{\lambda\Omega(\boldsymbol{w}) + \mathbf{max}\left(0, \max_{j \leq i}\{\langle \boldsymbol{w}, \boldsymbol{a}_j \rangle + b_j\}\right)\right\}$
    // Save the intermediate solution.
    $\boldsymbol{W} = [\boldsymbol{W}, \boldsymbol{w}_{i+1}]$
    // Compute the approximation gap.
    $\epsilon_i = \min_{j \leq i+1}\left\{J(\boldsymbol{w}_j) - \hat{J}(\boldsymbol{w}_{i+1})\right\}$
    // Enter the next iteration.
    $i = i + 1$
until $\epsilon_i < \epsilon$
return $\boldsymbol{w}$
```

5.2.4. Implementation Issues and Software

Our framework has been implemented mainly in Matlab, including a labeling GUI for the generation of training set assignments, feature extraction, model inference and the bundle method. To reduce the search space and eliminate hypotheses with no prospect of being realized, we constrain the hypotheses to a k-nearest neighborhood with distance thresholding. We use IBM ILOG CPLEX[2] as the underlying optimization platform for the ILP, quadratic programming and linear programming as needed for solving Eq. 5.5 [144].

5.2.5. Optimization

Solving the Augmented Optimization Problem

Here we show how to solve the augmented optimization problem. Let $\boldsymbol{z}$ be the ground truth and $\hat{\boldsymbol{z}}$ the prediction, the augmented optimization is:

$$\begin{aligned} \hat{\boldsymbol{z}} &= \arg\max_{\boldsymbol{z}} \{H(\boldsymbol{x}, \boldsymbol{z}^*, \boldsymbol{z}; \boldsymbol{w})\} \\ &= \arg\max_{\boldsymbol{z}} \{\langle \boldsymbol{\Phi}(\boldsymbol{x}_n, \boldsymbol{z}), \boldsymbol{w}\rangle + \Delta(\boldsymbol{z}^*, \boldsymbol{z}; \boldsymbol{w})\}\,. \end{aligned} \tag{5.8}$$

where the loss function takes the form $\Delta(\boldsymbol{z}^*, \boldsymbol{z}) = \frac{1}{|\boldsymbol{z}|}\sum_{\mathcal{E}}\sum_{\boldsymbol{s},\boldsymbol{s}'} z^{*\mathcal{E}}_{\boldsymbol{s},\boldsymbol{s}'}(1 - z^{\mathcal{E}}_{\boldsymbol{s},\boldsymbol{s}'})$.

This function has the advantage of being decomposable [28]. Therefore, the augmented optimization problem in Alg. 6 can be expressed as

$$\begin{aligned} H(\boldsymbol{x}, \boldsymbol{z}^*, \boldsymbol{z}; \boldsymbol{w}) &= \langle \boldsymbol{\Phi}(\boldsymbol{x}, \boldsymbol{z}), \boldsymbol{w}\rangle + \Delta(\boldsymbol{z}^*, \boldsymbol{z}; \boldsymbol{w}) \\ &= \sum_{\mathcal{E}}\sum_{\boldsymbol{s},\boldsymbol{s}'}\langle \boldsymbol{f}^{\mathcal{E}}_{\boldsymbol{s},\boldsymbol{s}'}, \boldsymbol{w}^{\mathcal{E}}\rangle z^{\mathcal{E}}_{\boldsymbol{s},\boldsymbol{s}'} + \frac{1}{|\boldsymbol{z}|}\sum_{\mathcal{E}}\sum_{\boldsymbol{s},\boldsymbol{s}'} z^{*\mathcal{E}}_{\boldsymbol{s},\boldsymbol{s}'}(1 - z^{\mathcal{E}}_{\boldsymbol{s},\boldsymbol{s}'}) \\ &= \sum_{\mathcal{E}}\sum_{\boldsymbol{s},\boldsymbol{s}'}\left(\langle \boldsymbol{f}^{\mathcal{E}}_{\boldsymbol{s},\boldsymbol{s}'}, \boldsymbol{w}^{\mathcal{E}}\rangle - \frac{1}{|\boldsymbol{z}|} z^{*\mathcal{E}}_{\boldsymbol{s},\boldsymbol{s}'}\right) z^{\mathcal{E}}_{\boldsymbol{s},\boldsymbol{s}'} + Const. \end{aligned} \tag{5.9}$$

We see that the formulation of $H(\boldsymbol{x}, \boldsymbol{z}, \hat{\boldsymbol{z}}; \boldsymbol{w})$ is similar to the formulation of the likelihood. As a result, the augmented optimization problem of maximizing $H(\boldsymbol{x}, \boldsymbol{z}^*, \boldsymbol{z}; \boldsymbol{w})$ can be solved by the same integer linear programming (ILP) method which offers efficient exact inference.

Optimizing Model Update with L_2/L_1 Regularization

In this section, we introduce how to solve the model update problem with different regularization terms. The model update problem refers to solving the approximated objective function:

$$\boldsymbol{w}_{i+1} = \arg\min_{\boldsymbol{w}} \left\{ \lambda\Omega(\boldsymbol{w}) + \mathbf{max}\left(0, \max_{j\leq i}\{\langle \boldsymbol{w}, \boldsymbol{a}_j\rangle + b_j\}\right)\right\}. \tag{5.10}$$

When using L_2 Regularization ($\Omega(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|_2^2$), the problem of model update can be

[2] http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/

rewritten as a constrained quadratic programming (QP) problem [144]:

$$\begin{aligned} \boldsymbol{w}_{i+1} &= \arg\min_{\boldsymbol{w}} \left\{ \tfrac{\lambda}{2}\|\boldsymbol{w}\|_2^2 + \xi \right\} \\ & \quad \text{subject to } \xi \geq 0 \text{ and } \forall j \leq i, \xi \geq \langle \boldsymbol{w}, \boldsymbol{a}_j \rangle + b_j. \end{aligned} \tag{5.11}$$

This problem can be solved in its current prime form or in a dual form which brings better computational efficiency [143]. Let $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_i]$ and $\boldsymbol{b} = [b_1, b_2, \ldots, b_i]^T$. The dual form of this problem is

$$\begin{aligned} \boldsymbol{\alpha}_{i+1} &= \arg\max_{\boldsymbol{\alpha}} \left\{ -\tfrac{1}{2\lambda}\boldsymbol{\alpha}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{\alpha} + \boldsymbol{\alpha}^T\boldsymbol{b} \right\} \\ & \quad \text{subject to } \boldsymbol{\alpha} \geq \mathbf{0} \text{ and } \|\boldsymbol{\alpha}\|_1 \leq 1. \end{aligned} \tag{5.12}$$

For the case of L_1 Regularization ($\Omega(\boldsymbol{w}) = \|\boldsymbol{w}\|_1$), define $\boldsymbol{w} = \boldsymbol{u} - \boldsymbol{v}$ and $\boldsymbol{u} \geq \mathbf{0}, \boldsymbol{v} \geq \mathbf{0}$. This problem with L1 regularization can be represented as a constrained linear programming (LP) problem [143]:

$$\begin{aligned} \boldsymbol{w}_{i+1} &= \arg\min_{\boldsymbol{w}} \left\{ \lambda\mathbf{1}^T\boldsymbol{u} + \lambda\mathbf{1}^T\boldsymbol{v} + \xi \right\} \\ & \quad \text{subject to } \xi \geq 0 \text{ and } \forall j \leq i, \xi \geq \langle \boldsymbol{u}, \boldsymbol{a}_j \rangle - \langle \boldsymbol{v}, \boldsymbol{a}_j \rangle + b_j, \end{aligned} \tag{5.13}$$

where $\boldsymbol{u} = (|\boldsymbol{w}| + \boldsymbol{w})/2$ and $\boldsymbol{v} = (|\boldsymbol{w}| - \boldsymbol{w})/2$ and thus this formulation is identical to Eq. 5.10 with $\Omega(\boldsymbol{w}) = \|\boldsymbol{w}\|_1$.

In practice, the iteration index i is usually in the order of 10s to 100s. Thus, both optimization problem can be efficiently solved using off-the-shelf solves (IBM ILOG CPLEX in our implementation).

5.2.6. Features

To differentiate similar events (e.g. division and split) and resolve ambiguity in model inference, we need rich features to characterize different events. In additional to basic features such as size/position [111] and intensity histogram [87], we also designed new features such as "shape compactness" for oversegmentation and "angle pattern" for division. Shape compactness relates the summed areas of two object candidates to the area of their union's convex hull. Angle pattern describes the constellation of two daughter cells relative to their mother.

To make the features comparable in scale and avoid numerical instability, we precondition each feature in two steps. Firstly, the features are centralized and linearly rescaled to approximately $[-1, 1]$. To ensure the robustness against noisy observations, the rescaling is based on the 5 percentile and the 95 percentile of the respective features. Secondly, the resulting features are transformed to $[0, 1]$ by a logistic function $F(x) = 1/(1 + e^{-x})$.

5.3. Results

We first evaluated the proposed method on the publicly available[3] image sequence provided in conjunction with the DCellIQ project [87]. It comprises 100 frames with ca. 10'000 cells in total (summed over all frames). The GFP stained cell nuclei were segmented using the method in [96]. The resulting segmentation has an F-measure of 99.5%. Ground truth associations for the training set were generated with a Matlab GUI tool at a rate of approximately 30 frames/hour. We also evaluated the model's generalization capabilities on another sequence.

5.3.1. Comparison to Related Methods

For a fair comparison, we extended Padfield's method [111] to account for the six events described in section 5.2.1 and used the same features (viz., size and position) and weights as in [111]. Hand-tuning of the parameters results in a high accuracy of 98.4% (i.e. 1 - total loss) as shown in Table 5.1 (2nd row). A detailed analysis of the error counts for specific events shows that the method accounts well for moves, but has difficulty with disappearance and split events. This is mainly due to the limited descriptive power of the simple features used. To study the difference between manual tweaking and learning of the parameters, we used the learning framework presented here to optimize the model and obtained a reduction of the total loss from 1.64% to 0.65% (3rd row). This can be considered as the limit of this model. Note that the learned parametrization actually deteriorates the detection of divisions because the learning aims at minimizing the overall loss across all events. In obtaining these results, every third pair of frames was used for training, just as in all subsequent comparisons.

Table 5.1.: Performance comparison. The header row shows the number of events occurring for moves, divisions, appearance, disappearance, splits and mergers. The remaining entries give the error counts for each event, summed over the entire sequence.

	mov	div	app	dis	spl	mer	total loss
	10156	104	78	76	54	55	-
Padfield *et al.* [111]	71	18	16	26	30	12	1.64%
Padfield *et al.* w/ learning	21	25	5	5	6	10	0.65%
Ours w/ manual tweaking	56	24	16	19	2	**5**	1.12%
Ours w/ learning (L_2 regula.)	**15**	**6**	**4**	**1**	**2**	6	**0.30%**
Ours w/ learning (L_1 regula.)	22	6	9	3	4	9	0.45%
Li *et al.* [87]	-	-	-	-	-	-	6.18%[a]

[a] Here we use the best reported error matching rate in [87] (similar to our loss).

[3] http://www.cbi-tmhs.org/Dcelliq/files/051606_HeLaMCF10A_DMSO_1.rar

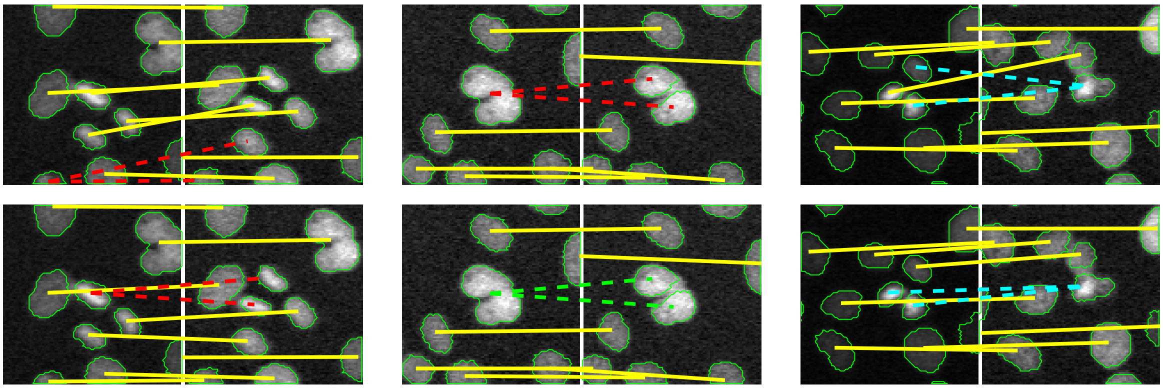


Figure 5.2.: Some diverging associations by [111] (top) and structured learning (bottom). Color code: yellow – move; red – division; green – split; cyan – merger.

For our model with 37 features in total, we first tried manual tweaking with a time limit of 1 hour (the same amount of time as required for generating the training samples). In comparison to [111], this model improves the overall loss but still makes many errors, especially for division (Table 5.1, 4th row). The limitations of manual tweaking as regards performance and user friendliness are successfully addressed by structured learning. The optimized model (with L_2 regularization) yields a total loss of only 0.30% (5th row) which is a significant improvement over [111, 87] (7th row). Some example assignments are shown in Fig. 5.2.

The learned parameters are summarized in Fig. 5.3. They afford the following observations: Firstly, cell size and shape are less useful than the intensity integral for move events because sudden deformations of cells occur before mitosis. Secondly, shape compactness is positively correlated with split but negatively with division. This is in line with the intuition that an oversegmentation conserves compact shape, while a true division seemingly pushes the daughters far away from each other (in the present kind of data, where only DNA is labeled). Finally, in spite of the regularization, many features are associated with large parameter values, which is key to the improved expressive power.

5.3.2. Sensitivity to Training Set

The success of supervised learning depends on the representativeness (and hence also size) of the training set. To test the sensitivity of the results to the training data used, we drew different numbers of training image pairs randomly from the entire sequence and used the remaining pairs for testing. For each training set size, this experiment is repeated 10 times. The mean and deviation of the losses on the respective test sets is shown in Fig. 5.5. According to the one-standard-error-rule, assignments between at

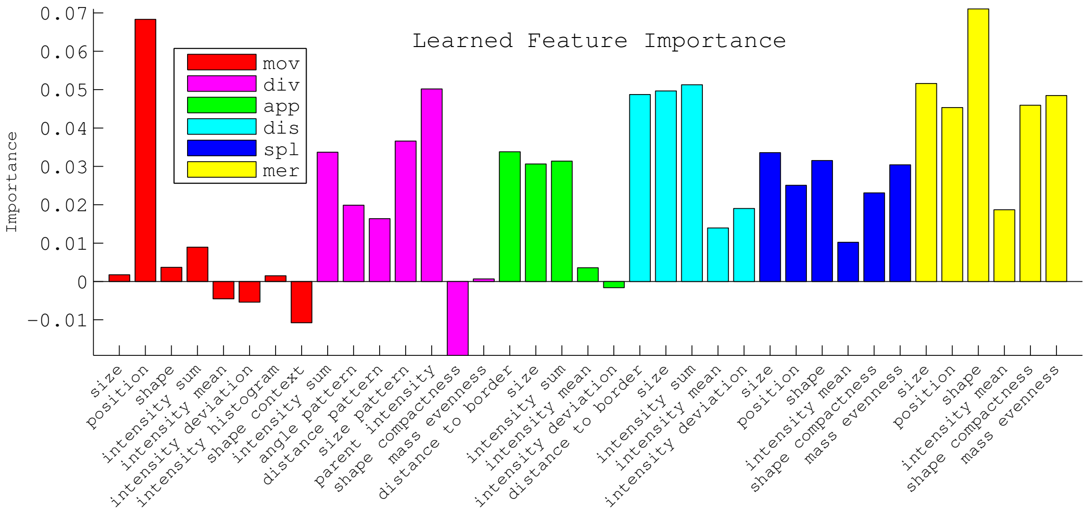


Figure 5.3.: Parameters $\boldsymbol{w}$ learned from the training data ($\Omega(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|_2^2$). Parameters weighing the features for different events are colored differently. The vector is normalized to unit 1-norm, i.e. $\|\boldsymbol{w}\|_1 = 1$.

least 15 or 20 image pairs are desirable, which can be accomplished in well below an hour of annotation work.

5.3.3. L_1 vs. L_2 Regularization

The results of L_1 vs. L_2 regularization are comparable (Table 5.1, 6th row). While L_1 regularization yields sparser vector, it has a much slower convergence rate (Fig. 5.6). The staircase structure shows that, due to sparse feature selection, the bundle method has to find more constraints to escape from a local minimum. The learned weight vector with L1 Regularization is shown in Fig. 5.4.

5.3.4. Distribution of Mitosis Events and Detection Errors

Fig. 5.7 shows the spatial distribution of all mitosis events detected by our method through the entire sequence. Our prediction (red squares) coincides well with the ground truth (green circles). The background image is the maximum-intensity projection (MIP) of 2D images along the entire temporal dimension.

Fig. 5.8 shows the spatial distribution of all tracking errors by our method through the entire sequence. Many of them occurred at the border of the view because the cell nuclei are partially visible and thus the features from them are particularly noisy. The background image is the same MIP of 2D images along the temporal dimension.

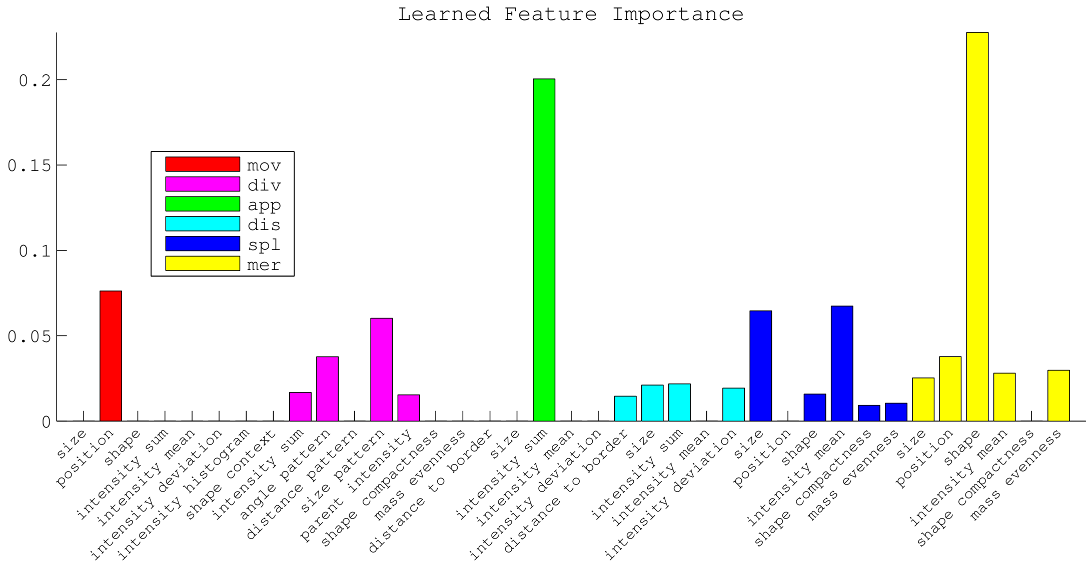


Figure 5.4.: Parameters $\boldsymbol{w}$ learned from the training data ($\Omega(\boldsymbol{w}) = \|\boldsymbol{w}\|_1$). Parameters weighing the features for different events are colored differently. The vector is normalized to unit 1-norm, i.e. $\|\boldsymbol{w}\|_1 = 1$.

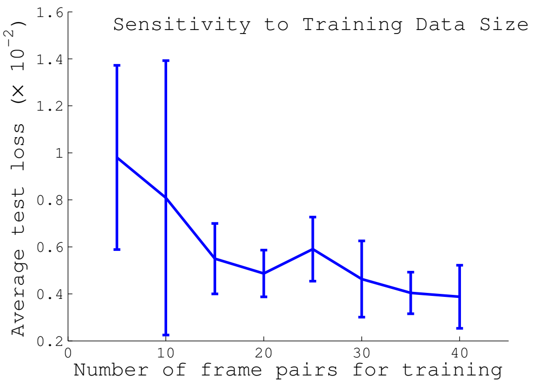


Figure 5.5.: Learning curve of structured learning (with L_2 regularization).

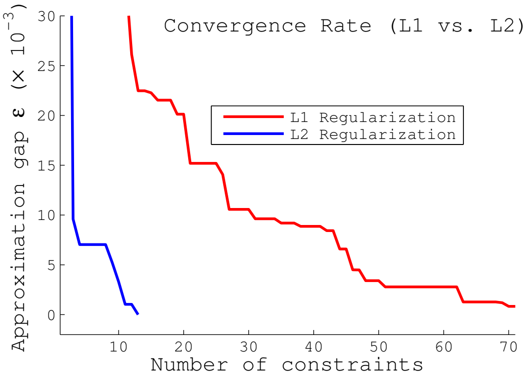


Figure 5.6.: Convergence rates of structured learning (L_1 vs. L_2 regularization).

5.3.5. Generalization Capabilities

The experiment described in the foregoing draws both training and test samples from the same time lapse experiment. This setting is frequently encountered in practice: the user simply wishes to obtain a good tracking for a given sequence with the smallest possible

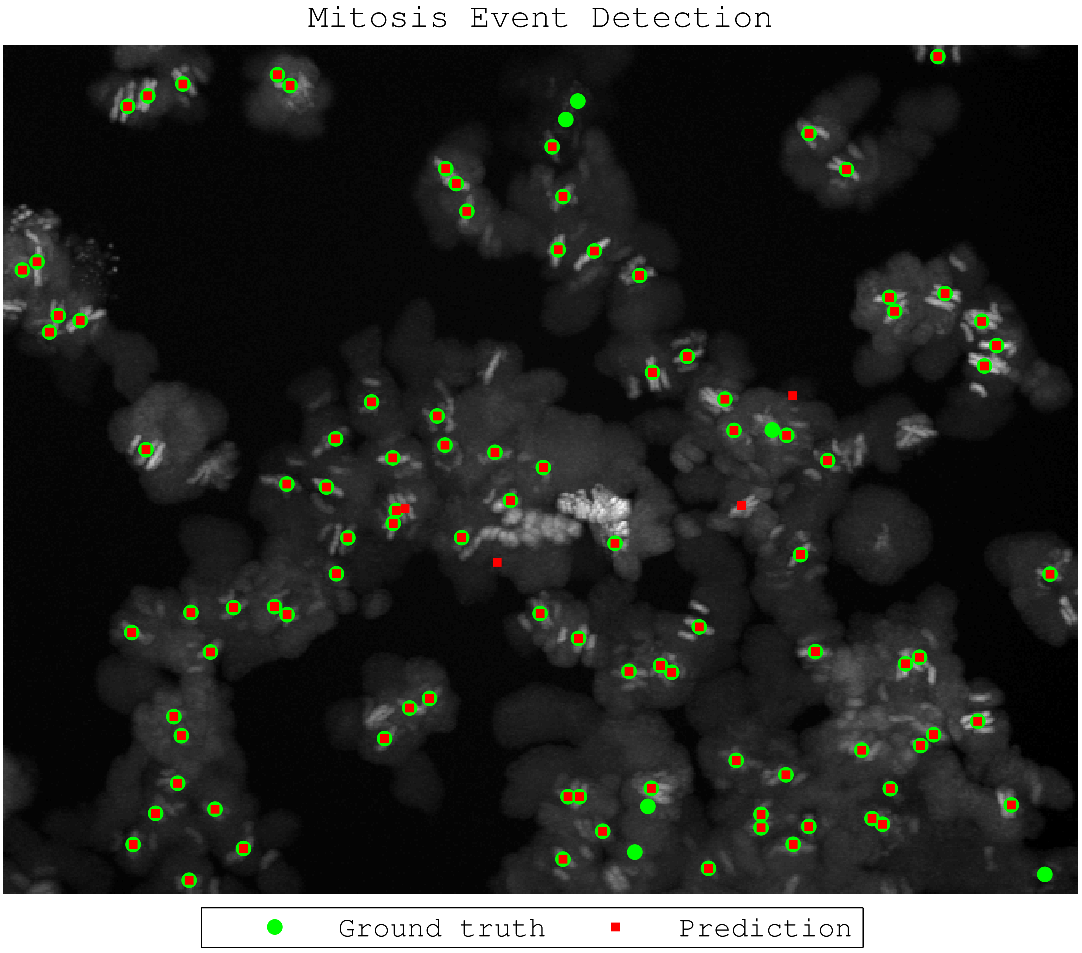


Figure 5.7.: Spatial distribution of predicted mitosis events vs. the ground truth. MIP can roughly tell the location and the pattern of mitosis events because the mitotic cells usually exhibit brighter appearance than the normal ones.

effort. However, in high-throughput experiments such as in the Mitocheck project [61], it is more desirable to train on one or a few sequences, and make predictions on many others. To emulate this situation, we have used the parameters $\boldsymbol{w}$ trained in the foregoing on the DCellIQ sequence [87] and used these to estimate the tracking of a sequence[4] from the Mitocheck project [61]. The sequence contains 94 frames and close to 24000 cells in total. The main focus of the Mitocheck project is on accurate detection of mitosis (cell division). Even though illumination and cell density are different from the training data,

[4] http://www.mitocheck.org/cgi-bin/mtc?action=show_movie;query=243867

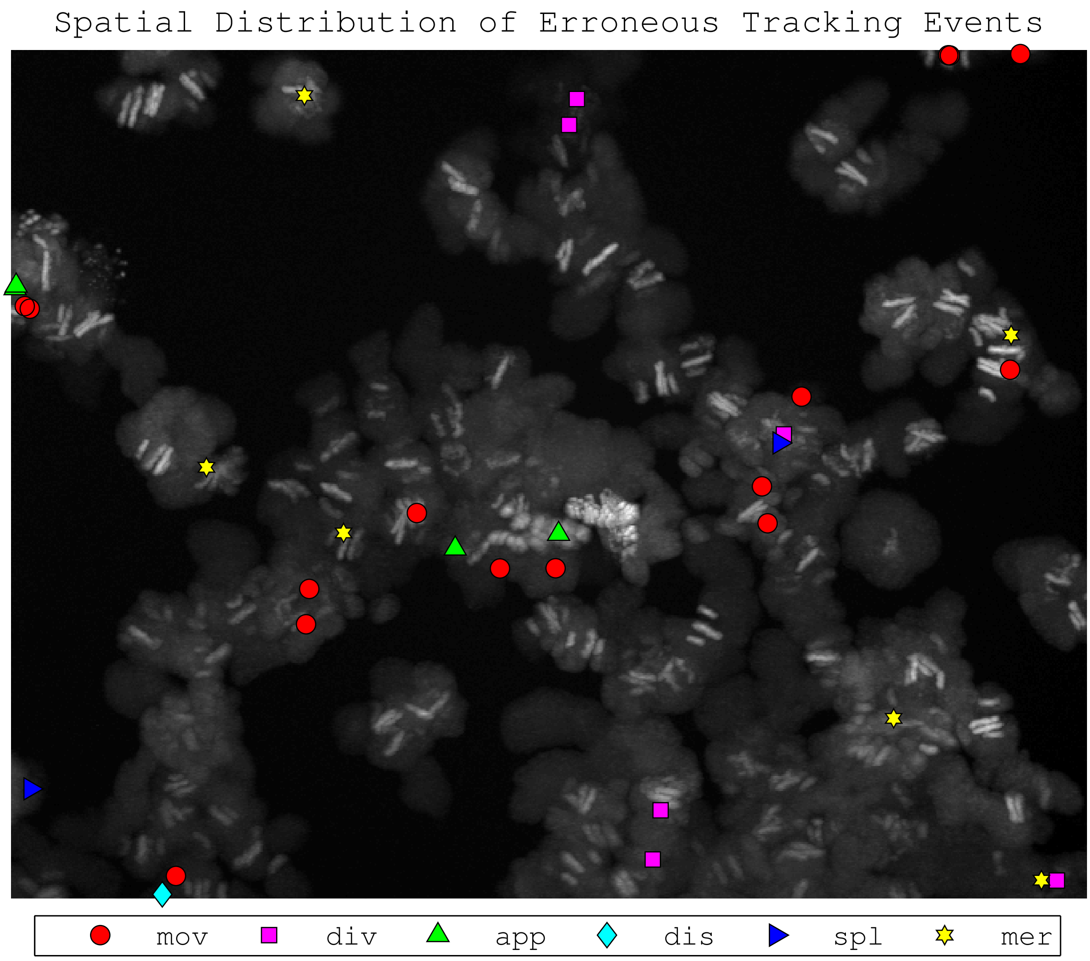


Figure 5.8.: Spatial distribution of tracking errors by our method.

our method detects 394 out of 412 mitotic events with 15 false positives, which gives an F-measure of 96.0%. Fig. 5.3.5 shows a few examples of detected mitosis events.

5.4. Conclusion and Future Work

We present a new cell tracking scheme that uses more expressive features and comes with a structured learning framework to train the larger number of parameters involved. Comparison to related methods shows that this learning scheme brings significant improvements in performance and, in our opinion, usability.

We currently work on further improvement of the tracking by considering more than two frames at a time, and on an active learning scheme that should reduce the amount of required training inputs.

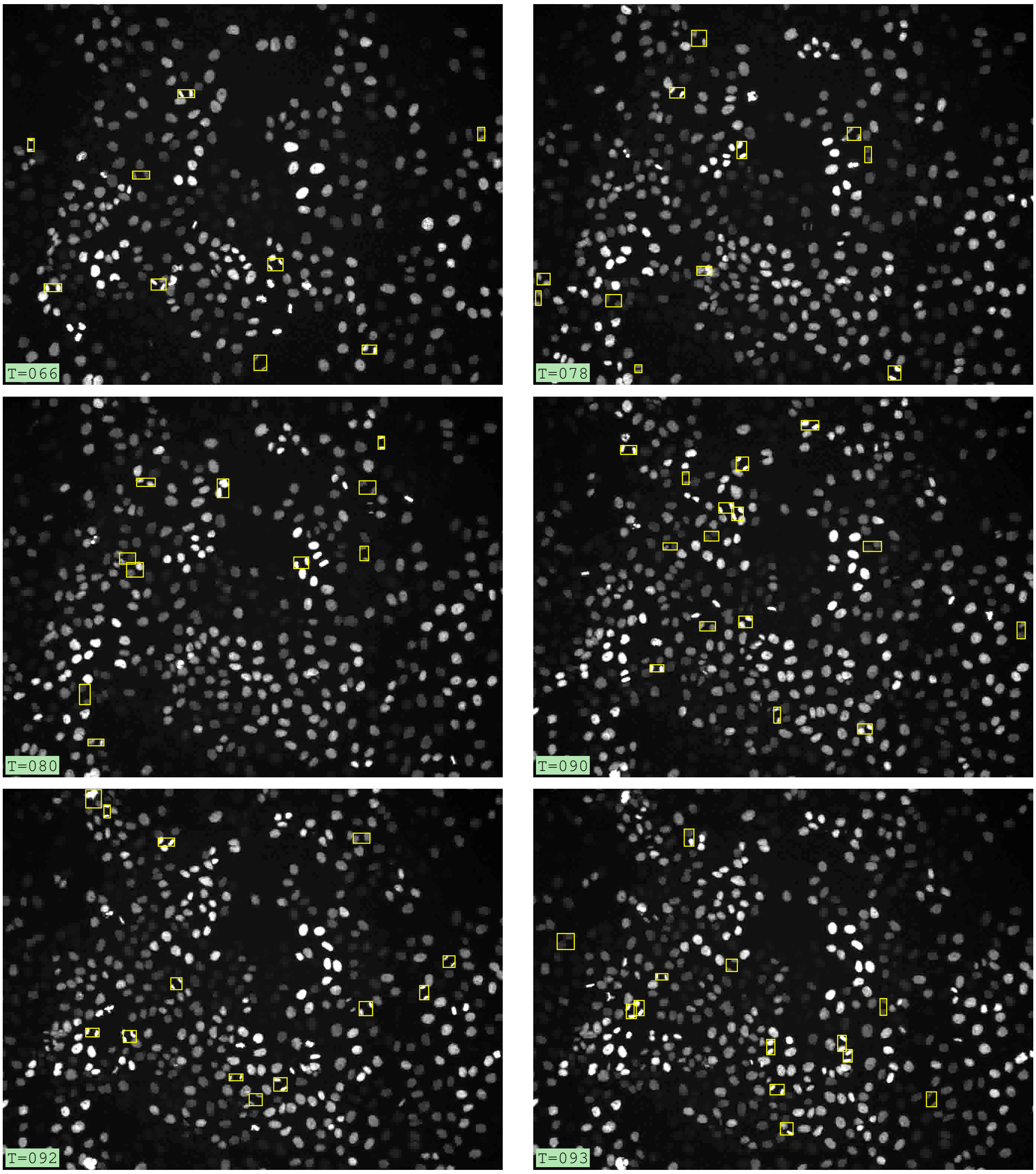


Figure 5.9.: Examples of detected mitosis events. The two daughters cells are highlighted with a yellow bounding box.

Chapter 6

Conclusions and Outlook

In this thesis we explored constrained modeling and learning, two paradigms for incorporating prior knowledge into biomedical data analysis. The significance of prior knowledge in improving the interpretation of complex datasets has been convincingly demonstrated by four diverse applications. For constrained modeling, Chapter 2 presented the use of sparsity prior (implemented by $L1$-regularization) that leads to physically reasonable estimation of deuteration distribution for HX/MS experiments; Similarly, Chapter 3 showed the extension of shape prior to an MRF energy model such that the segmented cell nuclei are smooth and regular. Regarding learning, Chapter 4 presented Rand Index with cost learning such that the resulting scoring reflects the true segmentation quality; Chapter 5 explored a structured learning strategy for cell tracking which removes an important bottleneck (i.e. parameter tuning) in real-world tracking problems.

A wide spectrum of computational methods have been employed or extended. To make the work in this thesis tangible for scientists from biology or medicine, all methods we developed have been or are being packaged into easy-to-use software tools, such as the HeXicon software[1] and structured learning for cell tracking in ilastik[2].

In the future, we are interested in the following developments:

- We would like to improve a few methods we developed such as HeXicon and cost-sensitive Rand Index (CSRI). In particular, we are interested in jointly solving the alignment and tracking problem in HeXicon using a sequential ordering prior (cf. Section 2.6 of Chapter 2). We are also interested in kernalizing the CSRI measure such that it can capture more non-linear functional dependencies between the pair-wise pixel label inconsistency and manual scoring.

- We will investigate a new paradigm which can be referred to as constrained optimization. This paradigm complements constrained modeling by directly restricting

[1] http://hci.iwr.uni-heidelberg.de/MIP/Software/hexicon.php

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

the solution space of the optimization rather than adding additional potential terms to the energy formulation. For example, authors in [107] proposed a novel approach to enforce connectivity of the segmentation by deriving *connected subgraph polytope*, a convex hull which rules out all solutions that fail to fulfil connectivity.

- Markov random field (MRF) and conditional random field (CRF) are the most commonly used models for segmentation problems. Several priors have been proposed to improve them such as connectivity, topology and shape. It is very challenging to study the possibility of unifying all these priors into a unique framework. Such a framework should provide high flexibility and satisfactory efficiency. Along the same line, it would be interesting to incorporate an adjacency prior such as "must touch", "must not touch" and "must be close". This prior seems primitive but is substantial for many real-world problems. For example of neural tissue analysis, given incomplete and corrupted information on synapses and mitochondria, an adjacency prior can help to detect both classes since synapses are packed with mitochondria.

- In response to one of the motivations for this thesis, i.e. manual tweaking data analysis methods in a feedback loop (Fig. 6.1) are stressful, inefficient and suboptimal, we intend to fully embrace the advances in machine learning (e.g. graphical models and structured learning) to design more intelligent systems that actively and directly exploit prior knowledge as in Fig. 6.2.

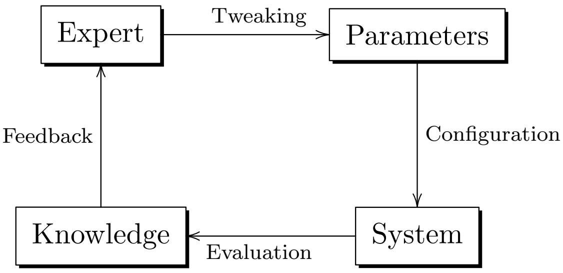


Figure 6.1.: Systems based on parameter tweaking.

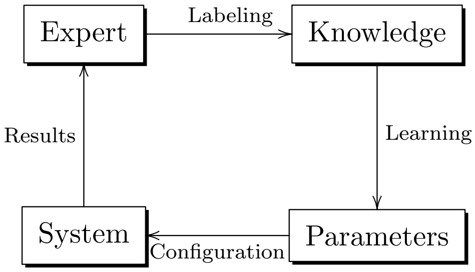


Figure 6.2.: Systems based on parameter learning.

- We also plan to investigate the possibility of developing a system that can solve cross domain problems. For example, in microscopic image analysis, cell tracking and neuron tracing can be formulated and solved in a similar object association formulation. Such a system requires a very rich set of features to boost its capability. How to learn from such high dimensional feature space and how to invoke proper kernels are research questions of high interest.

Frequently used Abbreviations

bEDT	Binary Euclidean Distance Transform
BIC	Bayesian Information Criterion
CNN	Convolutional Neural Network
CSRI	Cost-Sensitive Rand Index
Da	Dalton
DSLM	Digital Scanned Laser Light-Sheet Microscope
EDT	Euclidean Distance Transform
EM	Expectation-Maximization
fEDT	Function Euclidean Distance Transform
GC	Graph Cut
GFP	Green Fluorescent Protein
GFT	Gradient Flow Tracking
GMM	Gaussian Mixture Model
GPU	Graphics Processing Unit
GUI	Graphical User Interface
GVF	Gradient Vector Field
HX	Hydrogen-Deuterium Exchange
ILP	Integer Linear Programming
KDE	Kernel Density Estimation
LAT	Local Adaptive Thresholding
LC	Liquid Chromatography
LP	Linear Programming
LS	Level Set

Table 6.1.: Frequently used Abbreviations.

MALDI	Matrix-Assisted Laser Desorption/Ionization
MB	Megabytes
MDS	Multidimensional Scaling
MPI	Message Passing Interface
MRF	Markov Random Field
MS	Mass Spectrometry
NITPICK	Non-greedy, Iterative Template-based peak PICKer
NMR	Nuclear Magnetic Resonance
PDE	Partial Differential Equation
PGM	Probabilistic Graphical Model
PSF	Peah Shape Function or Point Spread Function
QP	Quadratic Programming
RANSAC	Random Sample Consensus
RF	Random Forest
RI	Rand Index
SIFT	Scale-Invariant Feature Transform
SNR	Signal-Noise Ratio
SVM	Support Vector Machine
TOF	Time-of-Flight
VI	Variation of Information

Table 6.2.: Frequently used Abbreviations.

List of Tables

List of Figures

Bibliography

[1] V. C. Abraham, D. Taylor, and J. R. Haskins. High content screening applied to large-scale cell biology. *Trends Biotechnol*, 22(1):15–22, 2004.

[2] R. R. Abzalimov and I. A. Kaltashov. Extraction of local hydrogen exchange data from HDX CAD MS measurements by deconvolution of isotopic distributions of fragment ions. *J Am Soc Mass Spectrom*, 17(11):1543–1551, 2006.

[3] E. M. Airoldi. Getting Started in Probabilistic Graphical Models. *PLoS Comput Biol*, 3(12):e252, 2007.

[4] Y. Al-Kofahi, W. Lassoued, W. Lee, and B. Roysam. Improved Automatic Detection and Segmentation of Cell Nuclei in Histopathology Images. *IEEE T Bio-Med Eng*, 57(4):841 –852, apr. 2010.

[5] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic Local Alignment Search Tool. *J Mol Biol*, 215(3):403–410, 1990.

[6] A. H. P. America and J. H. G. Cordewener. Comparative LC-MS: A landscape of peaks and valleys. *Proteomics*, 8(4), 2008.

[7] B. Andres, U. Koethe, M. Helmstaedter, W. Denk, and F. A. Hamprecht. Segmentation of SBFSEM volume data of neural tissue by hierarchical classification. In *DAGM*, pages 142–152, 2008.

[8] B. Appleton and H. Talbot. Globally Minimal Surfaces by Continuous Maximal Flows. *IEEE T Pattern Anal*, pages 106–118, 2006.

[9] F. Aurenhammer. Voronoi Diagrams: A Survey of a Fundamental Geometric Data Structure. *ACM Comput Surv*, 23(3):345–405, 1991.

[10] G. Bakir, T. Hofmann, B. Schoelkopf, A. J. Smola, B. Taskar, and S.V.N. Vishwanathan. *Predicting Structured Data.* MIT Press, Cambridge, MA, 2006.

[11] P. Baldi, S. Brunak, and S. R. Brunak. *Bioinformatics: The Machine Learning Approach*. The MIT Press, 2001.

[12] P. Bamford. Empirical comparison of cell segmentation algorithms using an annotated dataset. In *ICIP*, 2003.

[13] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-Up Robust Features (SURF). *Compu Vis Image Und*, 110(3):346 – 359, 2008.

[14] J. Besag. On the Statistical Analysis of Dirty Pictures. *J Roy Statistical Society, Ser. B*, 48:259–302, 1986.

[15] W. Bialek and D. Botstein. Introductory science and mathematics education for 21st-Century Biologists. *Science*, 303(5659):788, 2004.

[16] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007.

[17] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive Image Segmentation Using an Adaptive GMMRF Model. In *ECCV*, pages 428–441, 2004.

[18] S. Boppel, B. Y. Renard, M. Kirchner, H. Steen, J. A. J. Steen, U. Koethe, and F. A. Hamprecht. Sparse Profile Reconstruction for LC/MS Feature Extraction. In *ASMS*, 2008.

[19] L. Bottou and O. Bousquet. The Tradeoffs of Large Scale Learning. volume 20, pages 161–168, 2008.

[20] Y. Boykov and G. Funka-Lea. Graph Cuts and Efficient N-D Image Segmentation. *Int J Comput Vision*, 70(2):109–131, 2006.

[21] Y. Boykov and M.P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in ND images. In *ICCV*, volume 1, pages 105–112, 2001.

[22] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *ICCV*, volume 1, pages 26–33 vol.1, 2003.

[23] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE T Pattern Anal*, pages 1124–1137, 2004.

[24] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE T Pattern Anal*, 23(11):1222–1239, 2001.

[25] L. Breiman. Random Forests. *Mach Learn*, 45(1):5–32, 2001.

[26] K. L. Briggman and W. Denk. Towards neural circuit reconstruction with volume electron microscopy techniques. *Curr Opin Neurobiol*, 16(5):562–570, 2006.

[27] C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Min and Knowl Disc*, 2(2):121–167, 1998.

[28] T. S. Caetano, J. J. McAuley, L. Cheng, Q. V. Le, and A. J. Smola. Learning Graph Matching. *Int J Comput Vision*, 31(6):1048–1058, 2009.

[29] J. S. Cardoso and L. Corte-Real. Toward a Generic Evaluation of Image Segmentation. *IEEE T Image Process*, 14(11):1773–1782, 2005.

[30] A. E. Carpenter, T. R. Jones, M. R. Lamprecht, C. Clarke, I. H. Kang, O. Friman, D. A. Guertin, J. H. Chang, R. A. Lindquist, J. Moffat, et al. CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biol*, 7(10):R100, 2006.

[31] R. Caruana and A. Niculescu-Mizil. An Empirical Comparison of Supervised Learning Algorithms. In *ICML*, pages 161–168, 2006.

[32] T. Chan and W. Zhu. Level set based shape prior segmentation. In *CVPR*, pages 1164–1170, 2005.

[33] T. F. Chan and L. A. Vese. Active Contours without Edges. *IEEE T Image Process*, 10(2):266–277, 2001.

[34] J. Cheng and J. C. Rajapakse. Segmentation of clustered nuclei with shape markers and marking function. *IEEE T Bio-Med Eng*, 56(2009):741–748, 2009.

[35] I. V. Chernushevich, A. V. Loboda, and B. A. Thomson. An introduction to quadrupole-time-of-flight mass spectrometry. *J Mass Spectrom*, 36(8):849–865, 2001.

[36] J. K. Chik, J. L. V. Graaf, and D. C. Schriemer. Quantitating the statistical distribution of deuterium incorporation to extend the utility of H/D exchange MS data. *Anal Chem*, 78(1):207–214, 2006.

[37] L. P. Coelho, A. Shariff, and R. F. Murphy. Nuclear segmentation in microscope cell images: a hand-segmented dataset and comparison of algorithms. In *ISBI*, 2009.

[38] F. S. Collins, M. Morgan, and A. Patrinos. The Human Genome Project: Lessons from Large-Scale Biology. *Science*, 300(5617):286, 2003.

[39] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling*. Chapman and Hall, London, 1994.

[40] P. Das, O. Veksler, V. Zavadsky, and Y. Boykov. Semiautomatic segmentation with compact shape prior. *Image Vision Comput*, 27(1-2):206–219, 2009.

[41] C. M. Dobson. Protein folding and misfolding. *Nature*, 426(6968):884–890, 2003.

[42] A. Dufour, V. Shinin, S. Tajbakhsh, N. Guillen-Aghion, J. C. Olivo-Marin, and C. Zimmer. Segmenting and Tracking Fluorescent Cells in Dynamic 3-D Microscopy With Coupled Active Surfaces. *IEEE T Image Process*, 14(9):1396–1410, 2005.

[43] O. Dzyubachyk, W. A. van Cappellen, J. Essers, W. J. Niessen, and E. Meijering. Advanced Level-Set-Based Cell Tracking in Time-Lapse Fluorescence Microscopy. *IEEE T Med Imag*, 29(3):852, 2010.

[44] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least Angle Regression. *Ann Stat*, 32(2):407–451, 2004.

[45] J. R. Engen. Analysis of protein conformation and dynamics by hydrogen/deuterium exchange MS. *Anal Chem*, 81(19):7870–7875, 2009.

[46] S. W. Englander. Hydrogen exchange and mass spectrometry: A historical perspective. *J Am Soc Mass Spectrom*, 17(11):1481–1489, 2006.

[47] R. Fabbri, L. D. F. Costa, J. C. Torelli, and O. M. Bruno. 2D Euclidean Distance Transform Algorithms: A Comparative Survey. *ACM Comput Surv*, 40(1):2, 2008.

[48] P. F. Felzenszwalb and D. P. Huttenlocher. Distance Transforms of Sampled Functions. Technical report, Cornell University, 2004.

[49] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient Graph-Based Image Segmentation. *Int J Comput Vision*, 59(2):167–181, 2004.

[50] D. Freedman and T. Zhang. Interactive graph cut based segmentation with shape priors. In *CVPR*, volume 1, page 755, 2005.

[51] J. Freixenet, X. Munoz, D. Raba, et al. Yet another survey on image segmentation: Region and boundary information integration. In *ECCV*, 2002.

[52] Y. Freund. Boosting a Weak Learning Algorithm by Majority. *Inform and Comput*, 121(2):256–285, 1995.

[53] G. Funka-Lea, Y. Boykov, C. Florin, M. P. Jolly, R. Moreau-Gobard, R. Ramaraj, and D. Rinck. Automatic heart isolation for CT coronary visualization using graph-cuts. In *ISBI*, pages 614–617, 2006.

[54] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE T Pattern Anal*, 6(6):721 –741, 1984.

[55] R. C. Gonzalez and R. E. Woods. *Digital Image Processing.* Prentice Hall, Upper Saddle River, N.J., 2008.

[56] L. Grady. Random Walks for Image Segmentation. *IEEE T Pattern Anal*, pages 1768–1783, 2006.

[57] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact Maximum a Posteriori Estimation for Binary Images. *J Roy Statistical Society, Ser. B*, 51(2):271–279, 1989.

[58] M. Guilhaus. Special feature: Tutorial. Principles and instrumentation in time-of-flight mass spectrometry. Physical and instrumental concepts. *J Mass Spectrom*, 30(11), 1995.

[59] I. Guyon and A. Elisseeff. An Introduction to Variable and Feature Selection. *J Mach Learn Res*, 3:1157–1182, 2003.

[60] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning.* Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[61] M. Held, M. H. A. Schmitz, et al. CellCognition: time-resolved phenotype annotation in high-throughput live cell imaging. *Nature Methods*, 7(9):747–754, 2010.

[62] S. W. Hell and J. Wichmann. Breaking the diffraction resolution limit by stimulated emission: stimulated-emission-depletion fluorescence microscopy. *Opt Lett*, 19(11):780–782, 1994.

[63] R. P. Hertzberg and A. J. Pope. High-Throughput Screening: New Technology for the 21st Century. *Curr Opin Chem Biol*, 4(4):445–451, 2000.

[64] A. N. Hoofnagle, K. A. Resing, and N. G. Ahn. Protein analysis by hydrogen exchange mass spectrometry. *Annu Rev Biophys Biomol Struct*, 32(1):1–25, 2003.

[65] M. Hotchko, G. S. Anand, E. A. Komives, and L. F. Ten Eyck. Automated extraction of backbone deuteration levels from amide H/2H mass spectrometry experiments. *Prot Sci*, 15(3):583–601, 2006.

[66] Q. Hu, R. J. Noll, H. Li, A. Makarov, M. Hardman, and R. Graham Cooks. The Orbitrap: A New Mass Spectrometer. *J Mass Spectrom*, 40(4):430–443, 2005.

[67] L. Hubert and P. Arabie. Comparing Partitions. *J. Classification*, 2:193–218, 1985.

[68] H. Ishikawa. Transformation of General Binary MRF Minimization to the First Order Case. *IEEE T Pattern Anal*, 2010.

[69] V. Jain, B. Bollmann, M. Richardson, et al. Boundary Learning by Optimization with Topological Constraints. In *CVPR*, 2010.

[70] V. Jain, J. F. Murray, F. Roth, S. Turaga, V. Zhigulin, K. L. Briggman, M. N. Helmstaedter, W. Denk, and H. S. Seung. Supervised Learning of Image Restoration with Convolutional Networks. *CVPR*, 2007.

[71] T. R. Jones, A. E. Carpenter, M. R. Lamprecht, et al. Scoring diverse cellular morphologies in image-based screens with iterative feedback and machine learning. *Proc Natl Acad Sci*, 106(6):1826, 2009.

[72] M. I. Jordan. *Learning in Graphical Models*. Kluwer Academic Publishers, 1998.

[73] T. Kanade, Z. Yin, R. Bise, S. Huh, S. E. Eom, M. Sandbothe, and M. Chen. Cell Image Analysis: Algorithms, System and Applications. In *WACV*, 2011.

[74] V. Kaynig, T. Fuchs, and J. M. Buhmann. Neuron Geometry Extraction by Perceptual Grouping in ssTEM Images. In *CVPR*, 2010.

[75] P. J. Keller, A. D. Schmidt, A. Santella, K. Khairy, Z. Bao, J. Wittbrodt, and E. H. K. Stelzer. Fast, high-contrast imaging of animal development with scanned light sheet-based structured-illumination microscopy. *Nat Methods*, 7(8):637–642, 2010.

[76] P. J. Keller, A. D. Schmidt, J. Wittbrodt, and E. H. K. Stelzer. Reconstruction of zebrafish early embryonic development by scanned light sheet microscopy. *Science*, 322(5904):1065, 2008.

[77] P. Kohli, M.P. Kumar, and P.H.S. Torr. P3 & Beyond: Solving Energies with Higher Order Cliques. In *CVPR*, pages 1–8, 2007.

[78] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, Cambridge, MA, 2009.

[79] V. Kolmogorov and Y. Boykov. What Metrics Can Be Approximated by Geo-Cuts, or Global Optimization of Length/Area and Flux. In *ICCV*, volume 1, 2005.

[80] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE T Pattern Anal*, 26(2):147–159, 2004.

[81] N. Komodakis, N. Paragios, and G. Tziritas. MRF Optimization via Dual Decomposition: Message-Passing Revisited. In *ICCV*, pages 1–8, 2007.

[82] N. Komodakis and G. Tziritas. Approximate Labeling via Graph Cuts based on Linear Programming. *IEEE T Pattern Anal*, pages 1436–1453, 2007.

[83] L. Konermann, X. Tong, and Y. Pan. Protein structure and dynamics studied by mass spectrometry: H/D exchange, hydroxyl radical labeling, and related approaches. *J Mass Spectrom*, 43(8):1021–1036, Aug 2008.

[84] A. Kreshuk, M. Stankiewicz, X. Lou, M. Kirchner, F. A. Hamprecht, and M. P. Mayer. Automated detection and analysis of bimodal isotope peak distributions in H/D exchange mass spectrometry using HeXicon. *Int J Mass Spectrom*, 2010.

[85] A. Kreshuk, C. Straehle, C. Sommer, U. Koethe, G. Knott, and F. A. Hamprecht. Automated segmentation of synapses in 3d em data. In *ISBI*, 2011.

[86] A. Lehmussola, P. Ruusuvuori, J. Selinummi, H. Huttunen, and O. Yli-Harja. Computational framework for simulating fluorescence microscope images with cell populations. *IEEE T Med Imag*, 26(7):1010, 2007.

[87] F. Li, X. Zhou, J. Ma, and S.T.C. Wong. Multiple Nuclei Tracking Using Integer Programming for Quantitative Cancer Cell Cycle Analysis. *IEEE T Med Imag*, 29(1):96, 2010.

[88] G. Li, T. Liu, J. Nie, L. Guo, J. Chen, J. Zhu, W. Xia, A. Mara, S. Holley, and S. T. C. Wong. Segmentation of touching cell nuclei using gradient flow tracking. *J Microsc*, 231(1):47–58, 2008.

[89] G. Li, T. Liu, A. Tarokh, J. Nie, L. Guo, A. Mara, S. Holley, and S. T. C. Wong. 3D cell nuclei segmentation based on gradient flow tracking. *BMC Cell Biol*, 8(1):40, 2007.

[90] K. Li, E. D. Miller, M. Chen, T. Kanade, L. E. Weiss, and P. G. Campbell. Cell population tracking and lineage construction with spatiotemporal context. *Med Image Anal*, 12(5):546–566, 2008.

[91] S. Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.

[92] Y. Li, C. Huang, and R. Nevatia. Learning to Associate: HybridBoosted Multi-Target Tracker for Crowded Scene. *CVPR*, 2009.

[93] G. Lin, U. Adiga, K. Olson, J. F. Guzowski, C. A. Barnes, and B. Roysam. A hybrid 3D watershed algorithm incorporating gradient cues and object models for automatic segmentation of nuclei in confocal image stacks. *Cytom Part A*, 56(1):23–36, 2003.

[94] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Springer, 1994.

[95] C. Lipinski and A. Hopkins. Navigating Chemical Space for Biology and Medicine. *Nature*, 432(7019):855–861, 2004.

[96] X. Lou, F. O. Kaster, M. S. Lindner, B. X. Kausler, Ullrich Koethe, H. Jaenicke, B. Hoeckendorf, J. Wittbrodt, and F. A. Hamprecht. DELTR: Digital Embryo Lineage Tree Reconstructor. In *ISBI*, 2011.

[97] X. Lou, M. Kirchner, B. Y. Renard, U. Koethe, S. Boppel, C. Graf, C. T. Lee, J. A. J. Steen, H. Steen, M .P. Mayer, and F. A. Hamprecht. Deuteration distribution estimation with improved sequence coverage for HX/MS experiments. *Bioinformatics*, 26(12):1535, 2010.

[98] X. Lou, U. Koethe, J. Wittbrodt, and F. A. Hamprecht. Globally Optimal Segmentation of Cell Nuclei with Multi-object Shape Regularization. 2011. (Preprint).

[99] D. G. Lowe. Object Recognition from Local Scale-Invariant Features. In *ICCV*, page 1150, 1999.

[100] J. H. Macke, N. Maack, R. Gupta, W. Denk, B. Schoelkopf, and A. Borst. Contour-propagation algorithms for semi-automated reconstruction of neural processes. *J Neurosci Methods*, 167(2):349–357, 2008.

[101] D. R. Martin, C. Fowlkes, D. Tal, et al. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In *ICCV*, 2001.

[102] S. G. Megason and S. E. Fraser. Imaging in Systems Biology. *Cell*, 130(5):784–795, 2007.

[103] E. Meijering, O. Dzyubachyk, I. Smal, and W. A. van Cappellen. Tracking in cell and developmental biology. *Semin Cell Dev Biol*, 20(8):894 – 902, 2009.

[104] M. Meila. Comparing Clusterings by the Variation of Information. In *COLT*, 2003.

[105] J. A. Nelder and R. W. M. Wedderburn. Generalized Linear Models. *J Roy Statist Soc Ser A*, 135(3):370–384, 1972.

[106] P. Nikamanon, E. Pun, W. Chou, M. D. Koter, and P. D. Gershon. "TOF2H": A precision toolbox for rapid, high density/high coverage hydrogen-deuterium exchange mass spectrometry via an LC-MALDI approach, covering the data pipeline from spectral acquisition to HDX rate analysis. *BMC Bioinformatics*, 9:387, 2008.

[107] S. Nowozin and C. H. Lampert. Global Connectivity Potentials for Random Field Models. In *CVPR*, 2009.

[108] N. Olivier, M. A. Luengo-Oroz, L. Duloquin, E. Faure, T. Savy, I. Veilleux, X. Solinas, D. Débarre, P. Bourgine, A. Santos, et al. Cell Lineage Reconstruction of Early Zebrafish Embryos Using Label-Free Nonlinear Microscopy. *Science*, 329(5994):967, 2010.

[109] J. C. Olivo-Marin. Extraction of spots in biological images using multiscale products. *Pattern Recogn*, 35(9):1989–1996, 2002.

[110] N. Otsu. A thresholding selection method from gray-level histogram. *IEEE T Syst Man Cy*, 9(1):62–66, 1979.

[111] D. Padfield, J. Rittscher, and B. Roysam. Coupled Minimum-Cost Flow Cell Tracking for High-Throughput Quantitative Analysis. *Med Image Anal*, 2010.

[112] M. Palmblad, J. Buijs, and P. Håkansson. Automatic analysis of hydrogen/deuterium exchange mass spectra of peptides and proteins using calculations of isotopic distributions. *J Am Soc Mass Spectrom*, 12(11):1153–1162, 2001.

[113] B. D. Pascal, M. J. Chalmers, S. A. Busby, and P. R. Griffin. HD Desktop: An integrated platform for the analysis and visualization of H/D exchange data. *J Am Soc Mass Spectrom*, Dec 2008.

[114] B. D. Pascal, M. J. Chalmers, S. A. Busby, C. C. Mader, M. R. Southern, N. F. Tsinoremas, and P. R. Griffin. The Deuterator: software for the determination of backbone amide deuterium levels from H/D exchange MS data. *BMC Bioinformatics*, 8(1):156, 2007.

[115] J. Pearl. *Causality: Models, Reasoning, and Inference.* Cambridge University Press, 2000.

[116] S. Pellegrini, A. Ess, and L. V. Gool. Improving Data Association by Joint Modeling of Pedestrian Trajectories and Groupings. In *ECCV*, 2010.

[117] H. Peng. Bioimage informatics: a new area of engineering biology. *Bioinformatics*, 24(17):1827–1836, 2008.

[118] H. Peng, Z. Ruan, F. Long, J. H. Simpson, and E. W. Myers. V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets. *Nat Biotechnol*, 28(4):348–353, 2010.

[119] M. Polak, H. Zhang, and M. Pi. An evaluation metric for image segmentation of multiple objects. *Image Vision Comput*, 27(8):1223–1227, 2009.

[120] W. M. Rand. Objective Criteria for the Evaluation of Clustering Methods. *J Amer Statist Assoc*, 66(336):846–850, 1971.

[121] B. Y. Renard, M. Kirchner, H. Steen, J. A. J. Steen, and F. A. Hamprecht. NITPICK: peak identification for mass spectrometry data. *BMC Bioinformatics*, 9:355, 2008.

[122] E. Rittweger, K.Y. Han, S.E. Irvine, C. Eggeling, and S.W. Hell. STED microscopy reveals crystal colour centres with nanometric resolution. *Nature Photon*, 3(3):144–147, 2009.

[123] J. B. T. M. Roerdink and A. Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Mathematical Morphology*, 41:187–228, 2001.

[124] Y. Rubner, C. Tomasi, and L. J. Guibas. A Metric for Distributions with Applications to Image Databases. In *ICCV*, 1998.

[125] M. J. Rust, M. Bates, and X. Zhuang. Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM). *Nat Methods*, 3(10):793–796, 2006.

[126] H. Samet and M. Tamminen. Efficient component labeling of images of arbitrary dimension represented by linear bintrees. *IEEE T Pattern Anal*, 10(4):586, 1988.

[127] A. Sarti, R. Malladi, and J. A. Sethian. Subjective surfaces: A method for completing missing boundaries. *Proc Nat Acad Sci*, 97(12):6258, 2000.

[128] R. E. Schapire. The Strength of Weak Learnability. *Mach Learn*, 5(2):197–227, 1990.

[129] B. Schoelkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* The MIT Press, Cambridge, MA, 2002.

[130] G. Schwarz. Estimating the dimension of a model. *Ann Stat*, 6(2):461–464, 1978.

[131] L. Shamir, J. D. Delaney, N. Orlov, D. M. Eckley, I. G. Goldberg, G. Chalancon, M. Kosloff, H. U. Osmanbeyoglu, S. Saraswathi, P. Cossio, et al. Pattern Recognition Software and Techniques for Biological Image Analysis. *PLoS Comput Biol*, 6(11), 2010.

[132] S. Singh, S. Raman, J. Rittscher, et al. Segmentation Evaluation for Fluorescence Microscopy Images of Biological Objects. In *MIAAB*, 2009.

[133] G. Slabaugh and G. Unal. Graph cuts segmentation using an elliptical shape prior. In *ICIP*, volume 2, pages 1222–1225, 2005.

[134] G. W. Slysz, C. A. H. Baker, B. M. Bozsa, A. Dang, A. J. Percy, M. Bennett, and D. C. Schriemer. Hydra: software for tailored processing of H/D exchange data from MS or tandem MS analyses. *BMC Bioinformatics*, 10(1):162, 2009.

[135] C. Sommer, C. Straehle, U. Koethe, and F. A. Hamprecht. ”ilastik: Interactive learning and segmentation toolkit”. In *ISBI*, 2011.

[136] J. C. Spall. *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control.* LibreDigital, 2003.

[137] G. Srinivasa, M. C. Fickus, Yusong Guo, A. D. Linstedt, and J. Kovacevic. Active Mask Segmentation of Fluorescence Microscope Images. *IEEE T Image Process*, 18(8):1817 –1829, aug. 2009.

[138] P. Strandmark and F. Kahl. Parallel and Distributed Graph Cuts by Dual Decomposition. In *CVPR*, 2010.

[139] R. Szeliski. *Computer Vision: Algorithms and Applications.* Springer-Verlag New York Inc, 2010.

[140] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors. *IEEE T Pattern Anal*, 30(6):1068–1080, 2008.

[141] A. L. Tarca, V. J. Carey, X. Chen, R. Romero, and S. Draghici. Machine Learning and its Applications to Biology. *PLoS Comput Biol*, 3(6):e116, 2007.

[142] B. Taskar, C. Guestrin, and D. Koller. Max-Margin Markov Networks. NIPS, 2003.

[143] C. H. Teo, A. Smola, S. V. N. Vishwanathan, and Q. V. Le. A scalable modular convex solver for regularized risk minimization. In *SIGKDD*, 2007.

[144] C. H. Teo, S. V. N. Vishwanthan, A. J. Smola, and Q. V. Le. Bundle methods for regularized risk minimization. *J Mach Learn Res*, 11:311–365, 2010.

[145] R. Tibshirani. Regression shrinkage and selection via the Lasso. *J Roy Statist Soc Ser B*, 58(1):267–288, 1996.

[146] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large Margin Methods for Structured and Interdependent Output Variables. *J Mach Learn Res*, 6(2):1453, 2006.

[147] R. Unnikrishnan, C. Pantofaru, and M. Hebert. Toward Objective Evaluation of Image Segmentation Algorithms. *IEEE T Pattern Anal*, 29(6):929–944, 2007.

[148] A. Vasilevskiy and K. Siddiqi. Flux Maximizing Geometric Flows. *IEEE T Pattern Anal*, pages 1565–1578, 2002.

[149] O. Veksler. Star shape prior for graph-cut image segmentation. In *ECCV*, volume 7, page 8. Springer, 2008.

[150] J. C. Venter, M. D. Adams, E. W. Myers, et al. The Sequence of the Human Genome. *Science*, 291(5507):1304, 2001.

[151] S. Vicente, V. Kolmogorov, and C. Rother. Graph Cut based Image Segmentation with Connectivity Priors. *CVPR*, 2008.

[152] N. Vu and B. S. Manjunath. Shape prior segmentation of multiple objects with graph cuts. In *CVPR*, volume 1, page 8, 2008.

[153] T. E. Wales and J. R. Engen. Hydrogen exchange mass spectrometry for the analysis of protein dynamics. *Mass Spectrom Rev*, 25(1):158–170, 2006.

[154] D. D. Weis, J. R. Engen, and I. J. Kass. Semi-automated data processing of hydrogen exchange mass spectra using HX-Express. *J Am Soc Mass Spectrom*, 17(12):1700–1703, 2006.

[155] M. N. Wernick, Y. Yang, J. G. Brankov, G. Yourganov, and S. C. Strother. Machine Learning in Medical Imaging. *IEE Signal Proc Mag*, 27(4):25 –38, july 2010.

[156] C. Zanella, M. Campana, B. Rizzi, C. Melani, G. Sanguinetti, P. Bourgine, K. Mikula, N. Peyrieras, and A. Sarti. Cells Segmentation from 3-D Confocal Images of Early Zebrafish Embryogenesis. *IEEE T Image Process*, 19(3):770–781, 2010.

[157] Y. Zeng, D. Samaras, W. Chen, and Q. Peng. Topology Cuts: A Novel Min-Cut/Max-Flow Algorithm for Topology Preserving Segmentation in N-D Images. *Compu Vis Image Und*, 112(1):81–90, 2008.

[158] Z. Zhang, S. Guan, and A. G. Marshall. Enhancement of the effective resolution of mass spectra of high-mass biomolecules by maximum entropy-based deconvolution to eliminate the isotopic natural abundance distribution. *J Am Soc Mass Spectrom*, 8(6):659–670, 1997.