

Bioinformatics image analysis
reveals cell graphs and community structures
in malignant cell populations of
classical Hodgkin lymphoma

Dissertation
zur Erlangung des Doktorgrades
der Naturwissenschaften
im Fach Bioinformatik

vorgelegt beim Fachbereich Mathematik und Informatik
der Johann Wolfgang Goethe-Universität
in Frankfurt am Main

von
Hendrik Schäfer
aus Bad Karlshafen

Frankfurt (2018)
(D 30)

vom Fachbereich Mathematik und Informatik der

Johann Wolfgang Goethe-Universität als Dissertation angenommen.

Dekan: Prof. Dr. Andreas Bernig

Gutachter: Prof. Dr. Ina Koch
Prof. Dr. Dr. h.c. Martin-Leo Hansmann

Datum der Disputation: 19.12.2018

Danksagung

Während meiner Doktorandenzeit durfte ich mit einer Vielzahl engagierter Menschen zusammenarbeiten, denen ich an dieser Stelle meinen Dank aussprechen möchte.

An erster Stelle möchte ich mich bei Prof. Dr. Ina Koch bedanken, für die Betreuung des Projektes und meiner Doktorarbeit. Sie ermöglichte ein angenehmes Arbeitsklima, in dem ich die Möglichkeit hatte mich frei zu entfalten, und förderte den Austausch mit anderen Wissenschaftlern.

Desweiteren danke ich Prof. Dr. Dr. h.c. Martin-Leo Hansmann vom Dr. Senckenbergisches Institut für Pathologie für die vielen Ideen und die gute und enge Kooperation. Während der regelmäßigen Treffen mit ihm und seiner Gruppe profitierte ich unter anderem von seiner Expertise zu Hodgkinlymphomen und deren Diagnostik, sowie von den vielen Anregungen zur Analyse und Interpretation der Bilddaten.

Auch danke ich allen Mitgliedern der Arbeitsgruppe *Molekulare Bioinformatik* der Johann Wolfgang Goethe-Universität für das produktive, kollegiale Umfeld.

Zusammenfassung

Die digitale Pathologie ist ein neues, aber stetig wachsendes, Feld in der Medizin. Die kontinuierliche Entwicklung von verbesserten digitalen Scannern erlaubt heute das Ab-scannen von kompletten Gewebeschnitten und *Whole Slide Images* gewinnen an Bedeutung. Sie ermöglichen den Einsatz computergestützter Methoden. Ziel dieser Arbeit ist die Methodenentwicklung zur Analyse von *Whole Slide Images* des klassischen Hodgkin Lymphoms. Ein Datensatz von 35 Gewebeschnitten wurde bezüglich Zellpositionen und Zellmorphologie graphentheoretisch ausgewertet.

Lymphknoten sind ein wichtiger Bestandteil der adaptiven Immunabwehr. Sie besitzen eine feste Struktur, die es Immunzellen, wie zum Beispiel T-Lymphozyten, erlaubt mit antigenpräsentierenden Zellen, den dendritischen Zellen und B-Lymphozyten, zu interagieren.

Das Hodgkin-Lymphom, oder auch Morbus Hodgkin, ist eine Tumorerkrankung des Lymphsystems, bei der die monoklonalen Tumorzellen in der Regel von B-Lymphozyten im Vorläuferstadium abstammen. In einigen sehr seltenen Fällen stammen die Tumorzellen von T-Lymphozyten ab.

Etwas mehr als 9.000 Hodgkin-Lymphom-Fälle werden jährlich in den USA diagnostiziert. Zwar ist die 5-Jahre-Überlebensrate für Hodgkin-Lymphome mit 85,3 % vergleichsweise hoch, dennoch werden etwa 1.100 Todesfälle pro Jahr in den USA registriert. Desweiteren ist das Risiko für Nachfolgeerkrankungen aufgrund von Chemo- und Strahlentherapie stark erhöht.

Hodgkin-Lymphome lassen sich gemäß der Weltgesundheitsorganisation weiter untergliedern. Die am häufigsten vorkommende Ausprägung ist das klassische Hodgkin-Lymphom, welches wiederum in Subtypen unterteilt werden kann. Hier sind der Mischtyp und die noduläre Sklerose zu nennen, welche den größten Anteil der klassischen Hodgkin-Lymphomfälle darstellen. Auf mikroskopischer Ebene sind die Hodgkin-Reed-Sternberg-Zellen (HRS-Zellen) typisch für das klassische Hodgkin-Lymphom. Hodgkin-Zellen sind große, einkernige Zellen. Der Zellkern ist stark vergrößert, und weist eine grobe Chromatinstruktur auf. Reed-Sternberg-Zellen besitzen ein vergleichbares Erscheinungsbild, haben aber zwei oder mehr Zellkerne. Immunhistologisch gibt es für HRS-Zellen charakterisierende Marker, so sind HRS-Zellen positiv für den Aktivierungsmarker CD30. Die Immunhistologie ist ein wichtiges Standbein der heutigen Pathologie und wird routinemäßig zur Diagnose von Lymphomen eingesetzt.

Neben der konventionellen Mikroskopie, ermöglichen digitale Scanner das Digitalisieren von ganzen Objektträgern (*Whole Slide Image*). Die digitale Pathologie gewinnt

zunehmend an Bedeutung. Zum einen ermöglicht die Digitalisierung es, die Daten einfacher und platzsparender aufzubewahren, zum anderen bieten die *Whole Slide Images* die Möglichkeit für eine computergestützte Auswertung. Ein weiterer Vorteil ist für die Telepathologie zu erkennen. Dieser Begriff bezeichnet die Vorgänge der Pathologie, welche über Fernkommunikationswege abgewickelt werden, wie z.B. Telefon oder Internet. Digitale Bilder können schnell über das Netzwerk versendet werden und ermöglichen so Ferndiagnosen oder die Konsultation weiterer Fachärzte bei komplizierten Krankheitsbildern.

Trotz Fortschritten computergestützter Methoden, ist die digitale Pathologie erst in den Anfängen. *Whole Slide Images* werden so gut wie nicht in der Routinediagnostik eingesetzt, der Einsatz beschränkt sich hauptsächlich auf die Forschung und Lehre. Letztere kann zum Beispiel durch den Einsatz von virtuellen Mikroskopen verbessert werden und den Studenten so Krankheitsbilder veranschaulicht werden, ohne die Verwendung der physischen Präparate.

Mit der Digitalisierung der Bilder gehen neue Anforderungen an die Bilddaten und die computergestützten Analysemethoden einher. Hochauflösende digitale Aufnahmen, die qualitativ nah an die manuelle Betrachtung der Bilder unterm Lichtmikroskop heranreichen, besitzen eine entsprechende Datengröße. Ein großer Kritikpunkt an digitalisierten histologischen Bildern ist die Tatsache, dass das Bild nur in einer Fokusebene betrachtet werden kann. Das ist gegenüber der konventionellen Mikroskopie ein Nachteil. Hier können Pathologen während der Betrachtung nachjustieren und so gegebenenfalls zusätzliche Details sichtbar machen. Die Entwicklung moderner digitaler Scanner könnte diese Nachteile allerdings in der näheren Zukunft beseitigen.

Für die herkömmliche Diagnostik bringt die Digitalisierung bisher nur wenige Vorteile. Die effizientere Lagerung der Bilder und die Möglichkeit Ferndiagnosen zu stellen wiegen die zur Zeit noch vorhandenen Nachteile und die zusätzlichen Kosten sowie die damit verbundene Umstellung für die Routineprozesse noch nicht vollständig auf.

Ein großer Vorteil von digitalisierten Gewebeschnitten bietet sich bei der computergestützten Analyse. Automatisierte Bildanalyseverfahren wie Zellerkennung können Pathologen bei der Diagnose unterstützen, indem sie umfassende Statistiken zur Anzahl und Verteilung von immungefärbten Zellen bereitstellen. Es ist außerdem möglich die Diagnose zu beschleunigen, zum Beispiel durch die automatisierte Auswahl von Gewebesausschnitten und vereinfachter Handhabung, beispielsweise müssen Objektträger nicht gewechselt und transportiert werden.

Ziel meiner Arbeit ist die Methodenentwicklung zur quantitativen, systematischen Analyse von *Whole Slide Images* für klassische Hodgkinlymphomgewebeschnitte. Die immunohistologischen Bilder wurden hierfür vom Dr. Senckenbergisches Institut für Pathologie des Universitätsklinikums Frankfurt bereit gestellt. Die betrachteten Gewebeschnitte sind gegen CD30 immungefärbt, einem Membranrezeptor, welcher in HRS-

Zellen und aktivierten Lymphozyten exprimiert wird. Dazu wurden die Gewebeschnitte zunächst mit Hämatoxylin gefärbt, welches an die negativ geladenen Moleküle der Zelle bindet und sich daher hauptsächlich im Zellkern ansammelt, da sich hier das negativ geladene Grundgerüst der DNS befindet. Der zweite Farbstoff, Fuchsin, ist an einen Antikörper gekoppelt, der spezifisch an CD30 bindet. Die Gewebeschnitte wurden mit einem *Aperio ScanScope slide scanner* digitalisiert und liegen mit einer hohen Auflösung von 0,25 μm pro Pixel vor. Bei den vorliegenden Gewebeschnittgrößen ergeben sich Bilder mit bis zu 90.000 x 90.000 Pixeln. Unkomprimiert ergeben sich damit Einzelbilder von einer Größe bis zu 30 GB.

Der untersuchte Bilddatensatz umfasst 35 Bilder von Lymphknotengewebeschnitten der drei Krankheitsbilder: Gemischtzelliges klassisches Hodgkinlymphom, noduläres klassisches Hodgkinlymphom und Lymphadenitis. Letztere ist die Kontrollgruppe, die CD30-positiven Zellen sind hier keine malignen Zellen, sondern aktivierte Lymphozyten, welche aufgrund der Entzündungsantwort vom Lymphknoten gegen Krankheitskeime bei einer viralen oder bakteriellen Infektion gebildet werden.

Für den Umgang mit dem Dateiformat und die Auswertung der Bilder wird die eigens hierfür implementierte Software *Impro* verwendet. Einige Teile der Bildverarbeitungs-pipeline wurden neu implementiert, so zum Beispiel die Bestimmung einer *Region of Interest* und die Dekonvolution der Farbkanäle. Für weitere Funktionen wie das Festsetzen eines Schwellenwertes für die Segmentierung und die Berechnung von Morphologiedeskriptoren sind etablierte Bilderkennungssoftware und -bibliotheken wie *CellProfiler* und *Java Advanced Imaging* eingebunden.

Für die Bildverarbeitung werden die Bilder zunächst gekachelt und in Standard TIF-Dateien konvertiert. Anschließend wird die *Region of Interest* bestimmt. Es werden alle Kacheln bestimmt, die eine größere Menge an Gewebe enthalten. Dazu wird ein *Minimale Distanz zum Mittelwert*-basiertes Clusterverfahren verwendet. Jedes Pixel wird entweder der Klasse *Gewebe* oder der Klasse *Hintergrund* zugeordnet und alle Kacheln, die nur einen sehr kleinen Prozentsatz an Gewebepixeln enthalten werden für die nachfolgende Objekterkennung verworfen. Anschließend wird durch eine Dekonvolution der Farbkanäle die relative Menge an Hämatoxylin und Fuchsin bestimmt. Für die Intensität an Fuchsin wird ein Schwellenwert gesetzt, um das Bild in CD30-Pixel und Hintergrundpixel zu segmentieren. Nachfolgend werden benachbarte CD30-Pixel zu potentiellen CD30-positiven Objekten zusammengefasst, welche anschließend einen Größenfilter durchlaufen, um kleine Zellfragmente und Färberückstände zu eliminieren. Für die Segmentierung und das Erfassen der Zellobjekte kommt in unserer Bildverarbeitungs-Pipeline *CellProfiler* zum Einsatz. Für alle CD30-positiven Zellobjekte werden neben der globalen Position im *Whole Slide Image* weitere Morphologiedeskriptoren berechnet, wie Fläche, Feret-Durchmesser, Exzentrizität und Solidität.

Um die Qualität der automatischen Detektion von CD30-Zellen beurteilen zu können,

gibt Impro die Möglichkeit Zellpositionen manuell zu annotieren. Dies wurde für einen zufällig gezogenen Satz an Bildkacheln für sechs *Whole Slide Images* durchgeführt. Die Methode zeigt mit 84 % eine hohe Präzision. Das heißt, ein großer Prozentsatz der sich im Bild befindenden CD30-Zellen wurde erkannt. Die durchschnittlich erzielte Sensitivität von 95 % zeigt zudem, dass es sich bei den von der *Pipeline* vorgeschlagenen Objekten mit hoher Genauigkeit wirklich um CD30-positive Zellen handelt.

Die Anzahl der CD30-positiven Zellen in den Gewebeschnitten variieren von Fall zu Fall. Es konnte allerdings gezeigt werden, dass in Lymphadenitisfällen im Schnitt deutlich weniger CD30-positive Zellen präsent sind als in klassisches Hodgkinlymphom. Während hier im Schnitt nur rund 3.000 Zellen gefunden wurden, lag der Durchschnitt für das Mischtyp klassisches Hodgkinlymphom bei rund 19.000 CD30-positiven Zellen. Für Subtypen vom klassischen Hodgkinlymphom wurden Gewebeschnitte ausgewertet, die mehr als 50.000 Zellen enthielten. Von den Zellpositionen ableitend, lässt sich die Zelldichtevertelung direkt auf den Gewebeschnitten als Heatmap darstellen. Während die CD30-positiven Zellen in Lymphadenitisfällen relativ gleichmäßig verteilt sind, bilden diese in klassischen Hodgkinlymphom-Fällen Zellcluster höherer Dichte. Gerade beim nodulären klassischen Hodgkinlymphom ist die Ballung der CD30-positiven Zellen am größten und tritt auch bei Gewebeschnitten auf, die nur eine geringe Anzahl CD30-positiver Zellen aufweisen.

Die berechneten Morphologiedeskriptoren bieten die Möglichkeit die Gewebeschnitte und den Krankheitsverlauf näher zu beschreiben. Zudem sind bisher Größe und Erscheinungsbild der HRS-Zellen hauptsächlich anhand manuell ausgewählter Zellen bestimmt worden. Im Gegensatz dazu, beschreibt der hier vorgestellte Ansatz die Morphologie von einem großem Datensatz und für ganze Gewebeschnitte. Hierbei ist zu beachten, dass die automatisierte Zellerkennung weniger selektiv als die manuelle Auswahl von HRS-Zellen ist. Die Bildverarbeitungspipeline unterscheidet nicht zwischen HRS-Zellen und aktivierten Lymphozyten, die ebenfalls CD30-positiv sind. Die Zellprofile werden zwar nach Größe gefiltert, jedoch nur mit dem Ziel sehr kleine Zellfragmente zu entfernen. Für eine Unterscheidung der CD30-positiven Zellen sind komplexere Deskriptoren nötig, wie zum Beispiel Anzahl, Form und Struktur der Zellkerne. Ein Maß für die Ausdehnung der Zellen ist der maximale Feret-Durchmesser. Bei CD30-Zellen im klassischen Hodgkinlymphom liegt dieser im Durchschnitt bei 20 μm und ist somit deutlich größer als die durchschnittlich gemessenen 15 μm in Lymphadenitis.

Neben der statistischen Auswertung der Zellerkennung ist ein Ziel der Arbeit die Modellierung des Lymphoms als komplexes System. Systembiologische Ansätze erlauben es, Zellen und deren Beziehungen zueinander darzustellen. Es wurde ein graphentheoretischer Ansatz gewählt, um die CD30-positiven Zellen und ihre räumliche Nachbarschaft zu modellieren. Die entsprechenden CD30-Zellgraphen sind so definiert, dass jeder Knoten im Graphen einer Zelle entspricht, und Kanten räumliche Nähe widerspiegeln. In CD30-

Zellgraphen von klassischen Hodgkinlymphom-Gewebeschnitten ist der durchschnittliche Knotengrad gegenüber den von Lymphadenitis-Bildern stark erhöht. Das heißt, dass hier die Mikroumgebung stärker mit CD30-positiven Zellen durchsetzt ist. Auffallend im Mischtyp klassischen Hodgkinlymphom war die hohe Varianz bei den durchschnittlich beobachteten Knotengraden. In diesen Gewebeschnitten sind teilweise sehr hohe Knotengrade zu finden, es gibt aber auch Fälle, in denen nur wenige CD30-positive Zellen in der Mikroumgebung vorkommen. Mischtyp klassisches Hodgkinlymphom ist also sehr variabel in Bezug auf die Anzahl und Verteilung von CD30-positiven Zellen im Lymphknoten. Neben den direkten Unterschieden zwischen den drei Diagnosen zeigt der Vergleich mit Zufallsgraphen, dass die beobachteten Knotengradverteilungen nicht für eine zufällige Verteilung der Zellen im Gewebeschnitt sprechen. Die gemessenen Knotengrade unterscheiden sich stark von einer theoretischen Poissonverteilung, welche bei einer zufälligen Verteilung der CD30-positiven Zellen zu erwarten wäre. Die Knotengrade sind, verglichen mit einem zufallsverteilten geometrischen Graphen, deutlich erhöht. Dies spricht für ein Clustern der CD30-positiven Zellen im Lymphknotengewebe.

Um das Lymphom und das Zusammenwirken mit der Lymphknotenstruktur als Ganzes zu verstehen, sind nicht nur die einzelnen CD30-positiven Zellen und ihre direkten Nachbarn wichtig, sondern auch die Verteilung, also zum Beispiel der Zusammenschluss in größere Zellgruppen. Die Untersuchungen der Zelldichte zeigen bereits, dass es in einigen Bereichen des Lymphknotens Ballungsgebiete mit besonders vielen CD30-positiven Zellen gibt. Graphentheoretisch werden diese Gruppen von in Beziehung stehenden Objekten auch als *Communities* bezeichnet. Cliques-basierte Algorithmen erkennen *Communities* in Zellgraphen aufgrund der hohen Konnektivität ihrer Knoten. Die so erzielte Untergliederung der Zellen kann als *Overlay* über den Gewebeschnitt gelegt werden. Eigenschaften und Verteilung der *Communities* können hinzugenommen werden, um klassisches Hodgkinlymphom-Gewebeschnitte näher zu charakterisieren.

Diese Arbeit zeigt, dass die Auswertung von *Whole Slide Image* und die Nutzung der daraus gewonnenen zusätzlichen Informationen unterstützend zur Verbesserung der Diagnose möglich ist. Eine speziell angepasste Pipeline wurde in Impro erstellt, um CD30-positiv Zellen zu erfassen. Es wurden 35 *Whole Slide Images* ausgewertet und eine Zellerkennung auf höchster Auflösungsstufe durchgeführt. Die mehr als 400.000 CD30-positiven Zellobjekte wurden morphologisch beschrieben, und zusammen mit ihrer Position im Gewebeschnitt ist die Betrachtung wichtiger Eigenschaften des klassischen Hodgkinlymphoms, beispielsweise die Verteilung der malignen Zellen, realisierbar. Dennoch ist die hier vorgestellte Implementierung eher als Prototyp zu verstehen. Für den direkten Einsatz im pathologischen Alltag müsste die Bildverarbeitungspipeline effizienter gestaltet werden, sodass die Ergebnisse in wenigen Minuten vorliegen.

Die Zellgraphen beschränken sich bisher auf CD30-positiv Zellen. Die Methode kann erweitert werden, um das komplexe Zusammenspiel in der Tumormikroumgebung

zu modellieren. Weitere Zelltypen, wie zum Beispiel Lymphozyten, können in Zukunft integriert werden und Wechselwirkungen zwischen Tumor und Zellen der Immunabwehr untersucht werden. Das Konzept der Zellgraphen ist allgemein anwendbar und kann zur Darstellung und Modellierung anderer Krankheitsbilder übertragen werden.

Abstract

Digital pathology becomes more and more important nowadays. The field is growing, as continuous development of modern digital scanner allows the scanning of whole slides in high resolution. The whole slide images enable the application of computer-aided methods to support pathologists with additional quantitative information. This work aims to develop methods for the analysis of whole slide images. 35 tissue sections of classical Hodgkin lymphoma were analyzed regarding CD30 cell positions and their morphology using graph-theoretical methods.

Lymph nodes are small, bean-shaped structures and a major part of the lymphatic system of mammals. The human body contains about 500-700 lymph nodes of different size. Foreign particles are transported via the lymph fluid to be filtered and recognized efficiently in the lymph nodes. This enables a quick immune response in case of an infection.

Hodgkin lymphoma (HL), also called Morbus Hodgkin, is a malignant tumor of the lymphatic system. In general, the malignant cells in HL derive from B-cell pre-cursor cells of the germinal center. In some exceptional cases, they have been found to originate from T-Cells.

A little more than 9,000 new cases are diagnosed annually as Hodgkin lymphoma in the United States of America. Despite the high 5-years survival rate of 85.3 %, HL is still a severe disease and causes the death of about 1,100 people per year in the USA. Additionally, the treatment with chemo- and radiotherapy can cause subsequent diseases.

According the World Health Organisation (WHO), HL can be differentiated in subgroups. The most frequent one is the classical Hodgkin lymphoma (cHL). Cases diagnosed as cHL can be further characterized. The two most common types of cHL are nodular sclerosis cHL and mixed cellularity cHL.

Histopathologically, Hodgkin lymphoma is characterized by the occurrence of Hodgkin and Reed-Sternberg (HRS) cells. In microscopic images, these cells appear enlarged, and have one or more large cell nuclei with a coarse-grained chromatin structure. HRS cells are known to be positive for the activation marker CD30, and thus can be highlighted by immunostaining. Immunohistological staining is a commonly used method in pathology and is, in combination with microscopy, routinely used for diagnosis of lymphoma.

Beside traditional microscopy, digital scanners nowadays allow the digitization of whole object slides to whole slide images (WSIs). The storage efficiency and the usability in telepathology are the most distinguished advantages of WSIs. Nevertheless, WSIs are currently not used in routine pathology. The effort and costs to change the established

routine processes in pathology and to integrate digital object slides are still high.

Computer-aided analysis of tissue sections may lead to additional, more sophisticated, information to describe the diseases' progress. Statistical data gained from the whole tissue section can support pathologists to characterize disease patterns in more detail and may assist the diagnosis.

The aim of this work is the exemplary analysis of WSIs of tissue sections diagnosed as cHL. The immunohistological images were provided by the Dr. Senckenberg Institute of Pathology at the Goethe-University Frankfurt. The examined images are immunostained against CD30. CD30 is a membrane receptor, which is expressed by HRS cells, and therefore labels the malignant cells in cHL. A second staining, hematoxylin, is applied to highlight the nuclei of all cells in the tissue section. The WSIs were captured with an *Aperio ScanScope slide scanner* at a high resolution of 0.25 μm per pixel. The size of the tissue sections resulted in images with up to 90,000 x 90,000 pixels. Without compression the files can reach sizes of 30 GB.

The pre-selected image set consists of 35 WSIs with tissue sections diagnosed as mixed cellularity cHL, nodular sclerosis cHL and lymphadenitis. The latter is examined as a control group. Here, the CD30-positive cells are not tumor cells, but activated lymphocytes. They are part of an immune response against a viral or bacterial infection.

To deal with the non-standard SVS file format of Aperio and for the analysis of the images, we implemented an in-house software called Impro. The software is written in Java. The required image processing methods were implemented from scratch. Examples are the detection of a region of interest (ROI) and the color deconvolution to separate the stains. Additional methods, like the thresholding for the image segmentation and the computation of morphological cell descriptors, were integrated using established imaging software and libraries like *CellProfiler* and the Java Advanced Imaging API (JAI).

The cell detection pipeline identified more than 400,000 cells in the 35 WSIs. The number of CD30-positive cells varied among the 35 cases. Overall, the cell count is lowest in lymphadenitis cases. While lymphadenitis tissue sections had on average 3,000 CD30-positive cells, in mixed cellularity cHL tissue sections, the average count was 19,000. A few cHL cases existed for which the number of HRS cells exceeded 50,000. The cell density can be displayed in Impro as a heat map overlay on top of the tissue section. In lymphadenitis, the CD30-positive cells were distributed evenly throughout the tissue section. In contrast, the cells formed dense groups in cHL cases. Especially in nodular sclerosis cHL, the CD30-positive cells formed dense clusters, even if the total number of cells in the tissue section was low.

For each CD30-positive cell, the imaging pipeline computed morphological descriptors like eccentricity, solidity and area size. The descriptors allow a more detailed view of the disease pattern. Up to now, form and size of HRS cells have been only described on single, manually selected HRS cells. The presented approach allows a statistical view

on the cell shape. It is noteworthy that the cell detection pipeline does not differentiate between HRS cells and CD30-positive cells in general. The Feret diameter describes the extent of an object. CD30-positive cells in cHL cases have an average Feret diameter of 20 μm . The CD30-positive cell population in lymphadenitis has a diameter of 15 μm on average.

Beside the statistical analysis of single cells, the aim of this work is to model the lymphoma as a complex system. We applied system biology methods to depict the relations of neighboring cells. The cell positions are used to build up cell graphs. Neighborhood relations are modeled according to the unit disk graph formalism. Typical graph properties can be computed to characterize the different tissue sections. The cHL cases show an increased average vertex degree compared to lymphadenitis, meaning that the microenvironment consists of more CD30-positive cells. In mixed cellularity cHL, we also see a high variability for the vertex degree distributions. Compared to random geometric graphs, the analyzed cell graphs have an increased average vertex degree. Even in lymphadenitis, where the CD30-positive cells are more evenly distributed than in cHL cases, the average degree is higher than one would expect from randomly distributed cells. Lymphadenitis is a controlled immune response and may be limited to the parts of the lymph node where the foreign particles were recognized.

Many graphs exhibit a hierarchical structure, in which highly connected vertex groups exist. In graph theory, these groups with vertices sharing a high number of relations, are called communities. Clique-based algorithms recognize communities in cell graphs. The partitioning in cell groups can be displayed as overlay. The number and the size of communities can be used to characterize cHL tissue sections.

The presented results illustrate that the analysis of WSIs and the additional information gained from the image processing pipeline can be used to support the diagnosis of lymphoma. 35 WSIs in total were examined, and a cell detection was performed on the image layer with highest resolution. More than 400,000 CD30-positive cell objects were morphologically described. In combination with their position in the tissue section, important features of disease patterns, e.g. the distribution of malignant cells, becomes possible. Nevertheless, the proposed imaging pipeline is more or less a prototype. The application in routine pathology requires a response in a few minutes to be efficient.

Currently, the proposed cell graphs only consist of CD30-positive cells. The method can be extended in the future and other cell types, e.g. cells that are part of an immune response, can be integrated. This will allow to analyze the interaction of the tumor cells and their microenvironment. The cell graph approach can be generalized and allows the modeling of other disease patterns.

Contents

Zusammenfassung	IV
Abstract	X
1 Motivation	1
2 Introduction	4
2.1 Lymph Node: Structure and Function	4
2.2 Hodgkin Lymphoma	6
2.3 State of the Art	7
3 Methods	13
3.1 Immunohistological Images	14
3.1.1 Preparation of Tissue Sections	14
3.1.2 Image Digitization	15
3.2 Imaging Pipeline	18
3.2.1 Color Deconvolution	18
3.2.2 Minimum Distance to Mean Clustering	19
3.2.3 Multi-Resolution Clustering	20
3.2.4 Region of Interest	20
3.2.5 Third Party Software	21
3.2.6 Segmentation	22
3.2.7 Cell Shape Descriptors	22
3.2.8 Validation: Precision and Sensitivity	23

3.3	Graph Theory	24
3.3.1	Graph File Formats	24
3.3.2	Unit Disk Graph	27
3.3.3	Random Geometric Graph	27
3.3.4	Graph Partitioning	29
3.3.5	Communities	29
3.3.6	k -Clique Percolation	30
4	Results	33
4.1	Imaging Pipeline	33
4.1.1	Impro Software	35
4.1.2	Pre-Processing	40
4.1.3	Object Detection	54
4.1.4	Validation	81
4.2	Graph-Based Analysis	84
4.2.1	CD30 ⁺ Cell Graphs	84
4.2.2	CD30 ⁺ Community Structures	101
5	Conclusion	113
6	Supplement	118
	List of Figures	127
	List of Tables	130
	List of Abbreviations	131
	List of Authors	134
	Lebenslauf	145

Chapter 1

Motivation

In the United States about 185,000 people were diagnosed with Hodgkin lymphoma in 2011 and currently, more than 9,000 new cases are estimated per year. The probability to be diagnosed with Hodgkin lymphoma (HL) for a single person during the life time is 0.2 %. The 5-year survival rates are, compared to other cancer types, high. After being diagnosed with HL 85.3 % of the patients survive five years or more. Nevertheless 1,100 people die per year in the United States, because of HL*. HL is still a severe disease.

Digital pathology [2] is an emerging field that evolved rapidly over the last years. Improved optics allow a detailed visual exploration of diseased tissues and novel biomarkers raise new possibilities for diagnosis. Since digital cameras are used to acquire biopsy images, tissue sections can be digitally annotated by pathologists. The image slides are also archived and shared among pathologists, allowing for online consultation and revision by several experts, the process is also called telepathology [3]. Today, whole slide images (WSIs) are routinely used for research purposes.

A lot of WSIs are produced during the standard diagnose procedure, but up to my knowledge this is the first attempt to analyze complete WSIs of HL using image processing approaches. The size of an uncompressed single image is up to 30 GB and numbers of malignant cells differ from a few hundred up to 50,000. A computer-aided approach is necessary to handle the huge data input.

*[1] provides detailed statistics of the occurrence of HL and cancer in general in the US

Computer-aided methods for diagnostics enter more and more the field of digital pathology. For frequent cancer diseases like breast cancer or prostate cancer many approaches were developed to segment and classify tissue samples into healthy tissue and cancerous areas [4] [5]. In final decision making for diagnosis and treatment the expert knowledge of pathologists is indispensable, but image processing can provide helpful tools to improve the efficiency. Especially simple, but time consuming tasks can be worked off by computers. As an example only 20 % of the 1 billion breast biopsies taken annually in the United States are identified as cancerous. Thus much time is spend to categorize benign tissue samples. Counting cell nuclei and the quantification of the diseased tissue are used to determine the grade of cancer, which plays an important role for the prognosis of the outcome of the disease. Both can be measured by image segmentation and classification.

In this cooperative project we aim to get a deeper understanding of the distribution of Hodgkin and Reed-Sternberg cells in the lymph node. Therefore we analyze WSIs of tissue sections to get objective measurements of cell distances, densities and cell shape descriptors of patients diagnosed with classical Hodgkin lymphoma (cHL).

Our exhaustive image analysis aims to understand the development of HL. The spreading of Hodgkin and Reed-Sternberg cells within the lymph node and the processes which lead to different cell distributions in the different subtypes of cHL are not yet completely understood. This also includes interactions between Hodgkin and Reed-Sternberg cells and their environment, including cells of the immune system. The complex, heterogeneous micro-environment and its role for the development of the disease still needs further investigation.

The first aim of this study was the creation of an imaging pipeline to handle and process immuno-stained WSIs of cHL. We wanted to extract crucial information like positions and shape descriptors of the malignant cells. Therefore we created an automated imaging pipeline adapted to the needs of this project. First of all the pipeline should be able to handle big data so that we can employ it to sets of WSIs to get statistical relevant results. This included the re-implementation of existing methods to deal with the image format, to reduce the input data and to run on a computer cluster. Next, we wanted to

recognize malignant cells, which is why we performed an object detection. The extracted data should be accessible quickly for further analysis. Thus all information gathered from the imaging pipeline were stored into a database.

The second aim was the statistical analysis of the collected data. As an efficient data structure which reflects the local neighborhood of the cells we defined *Cell Graphs* [6]. Graph theoretical methods were used to measure typical properties of the distribution of malignant cells.

Furthermore we were interested in getting more abstract information than single cell positions and cell-cell communication. The *Cell Graphs* presented in this work, had typical sizes of 2,000-12,000 vertices and 50,000-2,000,000 edges. They allowed a detailed exploration of the neighborhood of detected single cells. More general structures like formation of cell clusters and their location in the lymph node or their co-location to specific lymph node structures may also lead to new insights for the diagnosis and the subsequent treatment of cHL. We applied the concept of community structure to *Cell Graphs* to find meaningful cell groups. Communities were used to describe and measure cHL sub-types on a higher abstraction level. This enabled an analysis of cell clustering and the formation of typical cell pattern within the lymph nodes.

Chapter 2

Introduction

2.1 Lymph Node: Structure and Function

Lymph nodes are small, oval structures of the lymphatic system. Their size varies from few millimeters up to two centimeters. The human body contains about 500 - 700 lymph nodes, located particularly in the neck, breast and abdomen. Lymph nodes are connected via lymph vessels. Lymph circulates through the network of lymph nodes and can be seen as a second transportation system besides blood circulation. Foreign bodies, which have infiltrated the body, are gathered by the lymph fluid and are directed into nearby lymph nodes. Here, they are filtered and initiate the immune response. Thus, lymph nodes are vital elements of the immune system.

Lymph nodes have a complex structure. Figure 2.1 schematically illustrates the different components. One or multiple lobules separate the lymph node into multiple sub units, one lobule for each afferent lymphatic vessel. The lymph enters through the afferent lymphatic vessel and flows into the subcapsular sinus. It passes the lobules through lateral transverse sinuses and exits the lymph node through medullary sinuses into the efferent lymphatic vessel. During the passage, the lymph is filtered by two processes [7]. The first barrier for potentially harmful particles is located in the subcapsular sinus. Various cells of the innate immune system form a network and passing cells are recognized by different combinations of pattern recognition receptors. Harmless molecules or cells can pass and reenter the blood stream after surpassing multiple lymph nodes, while possibly

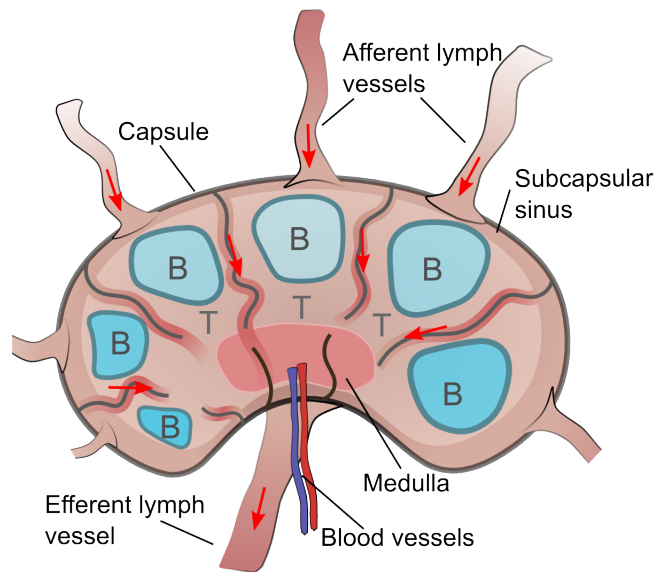


Figure 2.1: Lymph node structure Lymph nodes are bean-like secondary lymphatic organs, located throughout the human body. The general structure of a single lymph node is depicted. Lymph flows from afferent lymphatic vessel to the efferent vessel and foreign particles are recognized and filtered during its passage. The direction of flow is indicated by red arrows. Within the lymph node certain sub-structures are present. The B and T cell zones play an important role for the immune response.

dangerous objects are held back. Second, a selective system allows only certain cells to enter the lymph node parenchyma. Only small molecules, less than 70 kilodaltons of size are allowed to enter the conduit system. Cells on the other side can only penetrate into the parenchyma if specific receptors are present on their surface.

Lobules, the functional sub units of the lymph node, also have a clearly defined sub-structure. The area next to the afferent lymphatic vessel is called superficial cortex, or B cell zone. It includes one or multiple spheroid follicles and the surrounding tissue, the interfollicular cortex. The paracortex (or: deep cortex, T cell zone) is located adjacent to the superficial cortex. It encloses the deep cortical units (DCUs), which consists of the central DCU and the peripheral DCUs. The modular composition of each lobule supports the processing of different kinds of signals, i.e. the communication between homing T- and B-cells and their corresponding antigen presenting cells. Therefore, the lymph node structure has direct impact to the bodies immune response.

B cells, as well as T cells undergo a maturing process, before they can trigger an

immune response. B cells can differentiate into plasma cells or memory cells. The activation of B cells requires two signals. First, the naive B cell needs to bind to its corresponding antigen. Second, the B cell passes the T cell zone and has to be activated by a corresponding T helper cell, which in turn, also has been activated by the same antigen. The activation of the T cells also takes place in the lymph node. Dendritic cells present foreign antigens and cause the maturing of the T cells. Both, B cells and regulatory T cells play a crucial role in the anti cancer cell immune response and thus, influence the progression and the proliferation of diseases like Hodgkin lymphoma.

2.2 Hodgkin Lymphoma

Hodgkin lymphoma is a cancerous malignancy of the lymph node. The name originates from Thomas Hodgkin, who described the disease in the 1832 published article 'On Some Morbid Appearances of the Absorbent Glands and Spleen.' [8]. About 11 % of all malignant lymphomas are HL. Nowadays the disease is categorized into multiple subgroups. The WHO classification differentiates Hodgkin lymphoma into two main groups. Classical Hodgkin lymphoma (cHL), the most common type with about 95 % of all cases and the less frequent nodular lymphocyte predominant type. In addition classical Hodgkin lymphoma can be further subdivided into the following four types. Nodular sclerosis type (NS cHL), where at least one nodule formed by sclerotic bands is present which contains malignant tissue. It is the most common type with 60-70 % of all cHL cases. With roughly 25 %, mixed cellularity cHL (MC cHL) is the second frequent type of cHL. The two other subtypes are rare. They are called lymphocyte-depleted and lymphocyte-rich cHL.

The occurrence of the different subtypes correlates with the socioeconomic status and the ethnic background of patients [9]. E.g. for the US, studies show an increased incidence of NS cHL in young adults positively correlated to the socioeconomic status. On the other hand mixed cellularity and lymphocyte depleted subtypes are known to be predominant in developing regions or in general for people with low socioeconomic status. They are also more often associated with Epstein bar virus. There is evidence that HL

might at least partly be caused by a rare effect during the very common Epstein bar virus infection. A two time period study in Boston found a highly increased risk for HL for people who went to the same school for at least one year with other diagnosed HL cases. But the results couldn't be confirmed by other studies. A genetic predisposition for HL might also be possible. First-degree relatives of HL patients show a threefold increased risk for the disease. High concordance have been found for HL in monozygotic twins, while no cases of concordance has been found for heterozygotic twins. Those findings are still controversial, as the number of test cases were rather low and the selection process of the participants (public advertising in the medias) might have caused bias.

On cellular level, Hodgkin lymphoma has some unusual characteristics compared to other tumors. The malignant cells, also called Hodgkin and Reed-Sternberg (HRS) cells, originate almost always from B-cell pre-cursor cells of the germinal center, but in rare cases they originate from T cells. HRS cells make up about 1 % of the tumor tissue. The majority of the tumor micro environment is build up by reactive cells of the immune system, including lymphocytes, macrophages, eosinophiles, mast cells, plasma cells, and stromal cells. In consequence no solid tumor is present, but an inflammatory heterogeneous micro-environment. HRS cells actively form their surrounding environment via cytokines and chemokines [10].

2.3 State of the Art

The two main subjects of my thesis are cell detection and the subsequent graph-based analysis of the data. The cell detection part includes multiple steps, beginning with the conversion of the image format, image (pre-) processing, thresholding and object detection. The analysis describes the creation of graphs using geometric properties, the calculation of general graph properties like node degree, or cluster coefficient and advanced properties, e.g. community structures.

Both fields, image processing and graph analysis developed over years. Multiple methods and many different approaches exist. Here, I give a short overview of the current state to illustrate the context of our work and its implementation.

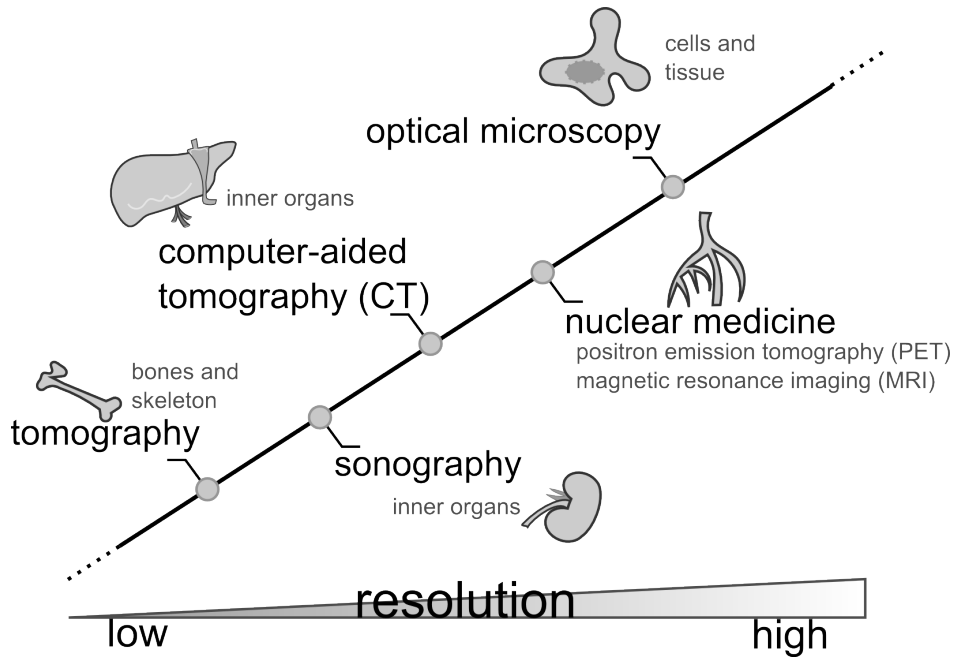


Figure 2.2: Medicinal image sources Imaging procedures that are commonly used in medical diagnosis and research. The methods differ in resolution and their applications. The resolution capability increases from left to right (scale does not reflect real data). Typical, but not exclusive, applications are depicted.

An exemplary list of image sources, which are important in medical research and diagnosis is given in Figure 2.2.

The resolution and the resulting level of detail of the images differ for the proposed methods. X-ray for general body structures like the skeleton. Ultra sound, magnetic resonance imaging/tomography (MRI/MRT), computer tomography (CT), positron emission tomography (PET), can be used to display inner organs, parts of organs, or tumor metastasis. Conventional light microscopy enables the exploration of tissue on the cellular level. An overview of the imaging related to cancer is given in [11].

Conventional light microscopy (CLM) is broadly used in primary diagnosis and is routinely used in pathology. Charge-coupled device (CCD) based digital cameras rapidly developed over the last decades and digital imaging became more and more important. Yet, whole slide images (WSI) are rarely used in diagnostic work-flows. Digitizing tissue section for diagnostics needs a costly computer infrastructure. It also requires to automate the whole image retrieval process to avoid additional delays during the diag-

nosis. Some studies targeting the inclusion of digital imaging into the standard routines exist. [12] and [13] give an overview of the perspectives of digital imaging in diagnostic work-flows. Some feasibility studies have been published to validate the application of WSIs in diagnostics pointing out that diagnosis based on WSIs yield comparable good results as CLM [14] [15]. Only few discrepancies were observed, only a few of which had prognostic impact [15] [16]. Nevertheless, single cases have been reported where image quality, e.g. the color rendering and false auto focus, are responsible for wrong diagnosis. WSIs have been tested for primary diagnostics for gynecological pathology [17], breast pathology [18], and teledermatopathology [19]. [20] notes, that even though there are studies examining the assignment of WSIs in routine diagnostics, none of them is based on a sufficient sample size to provide a statistically significant evaluation of the use of WSIs in routine diagnostics. The College of American Pathologists Pathology and Laboratory Quality Center reviewed various publications and provides a general guideline for validation of WSIs for diagnostic purpose in [21]. Currently, WSIs are most prominently used in research and teaching. WSIs can be examined using virtual microscopes [22] [23] [24]. This allows to access the tissue slides from different locations, as it is needed in telepathology [25]. Students can be trained by examining WSIs and tissue samples can be presented to teach the features of a disease. Virtual microscopes simplify the navigation through tissue sections, allow the pathologist to label the images and overlays can visualize additional information like patient data. For further information about digital pathology in general I here refer to three reviews: [2], [26], and [27].

For image processing, that is pre-processing (e.g. color deconvolution, definition of a region of interest, and normalization), pixel separation, and labeling of the images, many algorithms exist. An exhaustive overview of image processing would go beyond the scope of this work, therefore I refer to [28], [29], and [30] for further reading to get an overview of image processing in general. Here, I will only point out recent studies closely related to my thesis. This includes image processing of histological images and especially WSIs, cancer diseases and cell detection including morphological descriptors.

Two crucial steps in object detection in histopathological images are color deconvolution (also called color unmixing or color decomposition) [31] [32] and normalization.

Color deconvolution is needed to convert the red, green, and blue values of a pixel into the corresponding staining vector. A commonly used method is matrix inversion to transform the colors into a new color space, where the major axes represent the staining dyes. Non Negative Matrix Factorization has been applied as an unsupervised color decomposition approach in [33]. In this work the two dyes hematoxylin and DAB were separated. An evaluation of the usage of Non Negative Matrix Factorization and Non Negative Least Squares to perform a color deconvolution has been made in [32]. The proposed method is able to separate more than three different stains and can be applied to whole slide images.

A fast method to gather information from an image is a quantitative pixel analysis. Applied to histological images we can measure the amount of staining for the whole image or within sub-areas of the imaged tissue section. A comparison between the computer-aided quantification of immunohistochemical staining and the manual pathologist visual scoring has been made in [34]. The authors have found a high correlation between the manual annotation and the results from the digital image analysis. But in addition they also report the need of further investigations to handle artifacts like tissue folds and the presence of admixed tissue elements. The image analysis is also adapted to a specific tissue type and immuno histological staining (IHC staining) meaning that it can not be used in general, but has to be adapted.

A quantitative analysis of WSIs has been applied, as a case study, to kidney renal clear cell carcinoma cases [35]. The authors propose a general concept adaptable to different cancer endpoints and discuss the demands for the method when used in diagnostics, e.g. the quality control of the images and the presence of public available reference data bases. In an earlier work based on the diploma thesis of Alexander Schmitz, we published results of the quantification of CD30 positive pixels in cHL WSIs [36]. The tissue sections showed a high variability regarding the CD30 positive pixel count. The variability was mainly caused by the different progression states within the observed lymph node sections. Therefore classification based on pixel quantification was not able to distinguish between the three disease types MCcHL, NScHL, and Lymphadenitis.

More advanced imaging methods are required to classify tissue sections and to recog-

nize typical patterns of a disease. The approaches can be texture-based methods and/or object-based methods.

Texture-based methods consider the local neighborhood. Common texture descriptors are the gray scale co-occurrence matrix, the wavelet transformation, Fourier transformation, see [37] for a detailed overview. Texture-based methods have been used to classify malignant versus benign tissue in multiple studies, recently published studies are [38] and [39]. Image texture analysis also has been applied to breast tissue images to determine the region of interest for follow up microarray experiments [40]. Heterogeneity assessment of tissue sections in WSIs has been used to label breast cancer histopathological digital images [41]. The proposed method enabled displaying heterogeneity for the tested image slides as false color maps and it has been shown that the three heterogeneity measurements correlate with manual annotations made by pathologists.

Object-based methods are also widely used to perform computer-aided diagnosis. Multiple studies have been targeted to segment cell nuclei in microscopic images [42], or small structures in general [43], depending on the staining of the images small structures can be cells or cell nuclei. Pixel-based and line-based descriptors have been combined to gain a more '*efficient nucleus detector in histopathology images*' [44]. Cell nuclei and their shape play an important role in detecting malignant tissue. Malignant cells undergo a remodeling and the functionally different behavior also can be, at least partly, reflected in the shape of the nucleus. The observed differences of nuclear structures in cancer cells are reviewed in [45] and [46]. As reported in those reviews, the alterations of the cell nucleus may include the number of nuclei per cell, the appearance of the nucleolus and the appearance of chromatin. Cell shape descriptors have been used as a basis for automated learning approaches and to classify malignant tissue. In [47] features from multiple studies have been combined to detect and classify cancer in various tissues. In the study, the k-nearest neighborhood classifier were found to perform best compared to the four other tested classifiers. Accuracy and specificity were above 0.86 and 0.80 respectively. Fourier shape descriptors have been applied to histological images and have been tested for their biological interpretation [48]. Combined with a directed acyclic graph classifier the overall classification accuracy for renal tumor subtypes yielded 77 %.

Graphs have been used for different applications. A general overview of graph theoretical methods is given in [49] and [50]. Unit disk graphs [51] are graphs with special geometric constraints, that have been widely used to model wireless communication networks, e.g. in ad hoc networks [52]. Their properties and fast algorithms for typical graph problems in unit disk graphs are well studied [53]. In graph theory graph partitioning is a common problem, especially in biological and social networks [54]. Various methods have been developed to find meaningful partitions, a comprehensive review of the topic has been presented in [55]. One of most commonly used methods is k-clique percolation [56], other, mostly clique based methods have been developed (see e.g. [57] and [58]). Clique-based partitioning has been applied to unit disk graphs [59].

Graph partitioning plays also an important role in data visualization. Complex graphs contain too much information to be displayed for human eyes. Therefore, reduced graphs, which reflect the global structure of the original graph, have been used for visualization [60]. For validation of different partitions and for optimization based partitioning methods, quality measurements are required. A very popular one is Q-modularity [61], an additional score for the modularity is proposed by Lázár *et al.* [62].

Graph structures have been used for image segmentation in general [63], but also graph-based approaches tailored for histopathological application exist. In multiple studies graphs have been applied to represent malignant tissue, see [64], [65], and [66]. Cell objects have been extracted from small image sections, 384×384 pixels and 1024×1024 pixels respectively. Identified cells have been represented as vertices and edges have been added according to the Waxman model. The authors report differences in the graph metrics of malignant and benign cells [64] and use machine learning to perform a computer-aided grading of the tissue samples with an overall accuracy of 98.5 %.

Although some graph-based applications to histopathological images exist, there are very few studies which address cell recognition in WSIs at high resolution. Up to my knowledge, no analysis of WSIs of lymph nodes, and HL in particular, have been made. The extraction of objective data, like cell count, cell distribution, and cell co-localization can provide valuable information for understanding and for modeling the lymph node as a complex biological system.

Chapter 3

Methods

The aim of this study is to statistically analyze the distribution of HRS cells in tissue sections of human lymph nodes. Multiple steps are necessary to gather and process the required information from the 2D image data we used as input. The following section will give an overview of the methods and software tools we applied. The ordering of topics in this section conforms to the chronological order in the imaging pipeline, depicted in Figure 3.1.

As starting point for the analysis, WSIs of human lymph node sections were prepared by pathologists from the Dr. Senckenbergisches Institute of Pathology of the Goethe University in Frankfurt. A whole slide scanner digitized the glass slides to produce WSIs. The preparation and WSI format is described in sections *Immunohistological Images* and *Image Digitization*.

The imaging pipeline extracted cell positions from the digitized images. The detailed

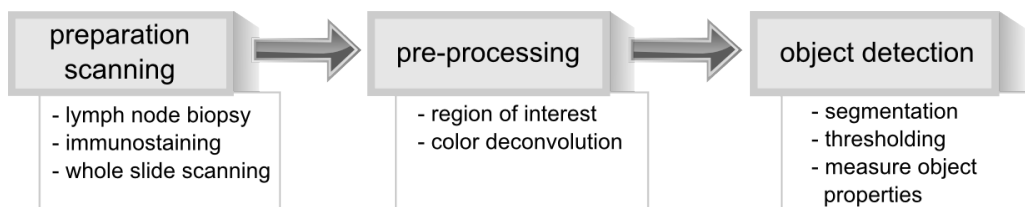


Figure 3.1: Imaging pipeline Images are prepared and digitized by pathologists. Multiple pre-processing steps are needed to enhance image quality and to deal with common image artifacts. Images are segmented to identify and label objects.

pre-processing steps are presented in the Sections 3.2.1 to 3.2.3. After the pre-processing, the object detection takes place. We made use of the open source software CellProfiler [67]. The tool provides many high level functions for image processing and for object detection in biological images. To get information about the software, see section *Third Party Software*.

For the representation of cells, their positions, and their neighborhood graphs were used. Section 3.3 explains the graph theoretical background of this work.

3.1 Immunohistological Images

In pathology, microscopic sections are routinely used for the diagnosis. After the fixation and dehydration, samples are embedded in paraffin. A microtome slices the paraffin block into thin sections. Additional staining highlights specific cell components or proteins.

3.1.1 Preparation of Tissue Sections

The images processed in this study were stained twice. Hematoxylin, the standard stain, binds rather unspecific to all negatively charged components of the cells. Since the backbones of the DNA and RNA constitute the main fraction of negatively charged biomolecules within the cell, cell nuclei are stained most, but also the cytosol appears in bright blue. The hematoxylin staining increases the contrast of the image. Cell nuclei can be distinguished, such that the cell density becomes visible. A second stain was added to identify specific cell types. The second staining can be applied to target a broad number of proteins, for example eosin binds to positively charged proteins, whereas immunostaining targets specific proteins. In immunostained images the dye is attached to an antibody, binding to the target protein. For samples embedded in paraffin, two antibodies are required. The first antibody binds to the target protein and the second antibody, which is targeted against the first antibody, is labeled with the dye (e.g. new fuchsin, DAB).

We used the magenta colored dye new fuchsin for the immunostaining. For diagnosis of cHL, multiple marker proteins are commonly examined. For identification of Reed-

Sternberg cells, CD30 and CD15 are typical marker proteins, they are part of the cluster of differentiation (CD) protocol.

The cluster of differentiation protocol was proposed and founded at the *1st International Workshop and Conference on Human Leukocyte Differentiation Antigens* (HLDA) in 1982. The original goal of categorizing monoclonal antibodies targeted against surface molecules of leukocytes was expanded. Now, the list of CD molecules consists of more than 300 unique cell surface proteins that can be targeted by monoclonal antibodies. The list of CD molecules is maintained and updated by the Human Cell Differentiation Molecules group. The current CD nomenclature is published in [68].

The expression profile of CD molecules gives information of the cells function and helps to distinguish between different cell types, also called immunophenotyping. In cancer therapy the CD protocol gives valuable information about the origin of the malignant cells. The presence or absence of specific surface proteins allows a categorization of the malignant cells and is important for diagnosis and treatment. The identification of cell type specific surface proteins is an ongoing work. An example is the human cell-surface immunome database [69], which contains cell types and their corresponding surface proteins.

3.1.2 Image Digitization

Image data shifted from analogue media to digital representations recently. Digital images have multiple advantages. They are easier to store and can be shared and thus have a better accessibility. Computer-aided methods can be applied to analyze whole tissue sections. In the case of WSIs, there are also downsides. First, scanning the tissue slides requires additional hardware. The prizes for whole slide scanner dropped over the last years, but the purchase is still expensive. Second, the WSIs have a fix focus plane and fine adjustments may be needed when fine structures, e.g. the chromatin in the nucleus, are examined.

An Aperio Scanscope device digitized the tissue samples and created SVS files in Aperio's proprietary image format. The format is an altered large-tiff format, containing multiple layers for versions of the image with different downsample rates. Figure 3.2

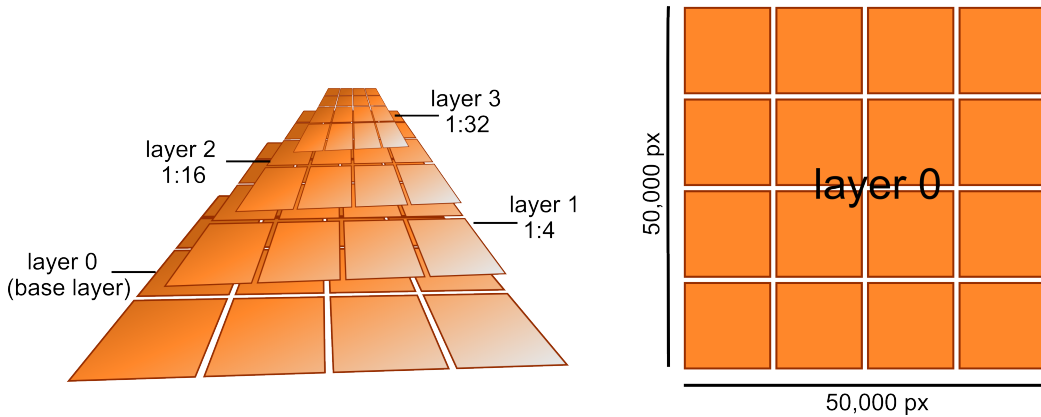


Figure 3.2: Aperio SVS image format Each SVS file consists of multiple layers to store multiple versions of the whole slide scan at different resolutions. The downsample rate increases by factor four per layer. Only layer 3 is downsampled by two compared to layer 2. It is optional and only present in images with a large base layer. The image of a single layer is stored in a tiled image format, which allows quick access to image parts without loading the whole image.

depicts the pyramidal SVS format. The base layer called layer 0 provides a resolution of $0.25 \mu\text{m}$ per pixel. The image sizes of our samples vary from $10,000 \times 10,000$ pixels up to $90,000 \times 90,000$ pixels. The downsample rates for all four layers are presented in Table 3.1.

Table 3.1: Downsample rates of the SVS image layers

	downsample	μm per px
layer 0	1	0.25
layer 1	4	1.0
layer 2	16	4.0
layer 3	32	8.0

Figure 3.3 demonstrates the four layer resolutions stored in an SVS file. In the third layer, which is the one with lowest resolution, a single pixel has the size of $8 \mu\text{m}$. The resolution is high enough to be able to identify the lymph node structure, i.e. sclerotic bands or germinal B-cell centers, but single cells can not be detected. At layer 2 the stained CD30^+ cells can be identified, but close objects cannot be separated. Finer structures like cell nuclei become visible at layer 1 with a resolution of $1 \mu\text{m}$ per pixel. The cell shapes can be computed at this resolution. The base layer provides the highest resolution and enables the observation of very fine structures, like the shape of the nuclei

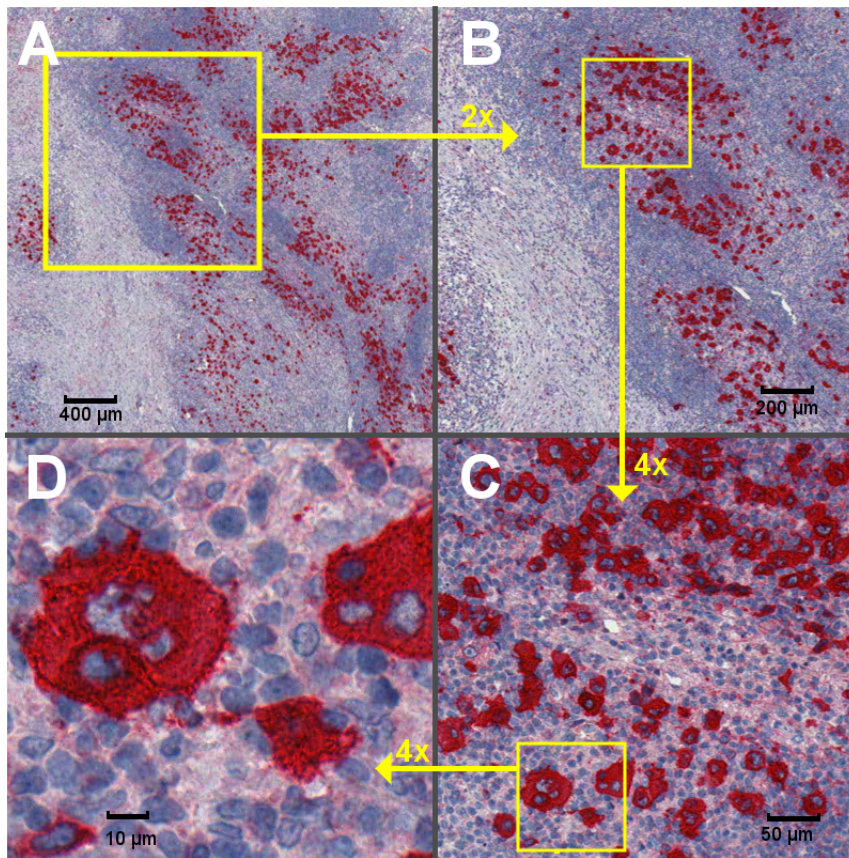


Figure 3.3: Resolution of the four SVS image layers The image depicts an exemplary section of an NScHL WSI. The four sub-images demonstrate the level of detail in each layer. The white rectangles mark the field of view shown in the next image.

or the nucleolus.

Each layer is stored as tiled image, allowing to load only parts of the images. The layer image is separated into equally sized tiles, each of which can be loaded independently. At highest resolution the uncompressed layer 0 reaches more than 20 GB and thus, can not be loaded into the memory of current desktop computers. The image format also includes a JPEG2000 compression. JPEG2000 is a lossless compression, resulting in SVS files with a size of 0.5 - 4 GB for images in this work.

3.2 Imaging Pipeline

3.2.1 Color Deconvolution

In immuno-histological images or digital images, the colors of pixels are commonly stored in the RGB color space. The color value is composed of the three values for red (R), green (G), and blue (B), reflecting the receptors of the human eye. The RGB color space does not refer to the actual staining in the images. The tissue sections are immunostained. Thus, the image captured by the scanner is mainly composed of the used dyes. Color deconvolution can be applied to split the color information into the stains. Pixels are then expressed as the relative amount of dye. For the RGB we get the following equation:

$$P_{RGB} = M_{ABS} * P_{stainings}. \quad (3.1)$$

$P_{stainings}$ is a vector containing the values of the amount of the dyes. M_{ABS} is the absorption matrix, describing the relative absorption rates for red, green, and blue light for the different dyes. Equation 3.1 can be transposed and the stain concentration can be calculated by:

$$P_{stainings} = P_{RGB} * M_{ABS}^{-1}. \quad (3.2)$$

Most image processing programs provide default absorption vectors for the most common dyes. Table 3.2 gives an example for an H&E staining, taken from the color deconvolution tool implemented in Fiji [70].

Table 3.2: Default absorption matrix for a H&E staining from Fiji

	R	G	B
hematoxylin	0.644	0.717	0.267
eosin	0.093	0.955	0.283
rest (orthogonal)	0.636	0.001	0.772

If only two dyes are present in the histochemical image, a third absorption vector is chosen which is orthogonal to both vectors of the color stains. The third channel represents everything within the image that does not originate from the staining process

and mainly includes artifacts like air bubbles or damaged tissue sections.

Table 3.3 shows an example absorption matrix used in this work. In contrast to the Fiji matrix, the second stain is new fuchsine. It is also a red dye, but compared to eosin, the blue color component is lower. The absorption matrix for a dye can differ, depending on the fabricator and the staining procedure in the laboratories. We decided to compute the absorption matrix directly from sample images from the Dr. Senckenberg Institute of Pathology. Manually selected image sections were taken as training data labeled either as hematoxylin or new fuchsine. The ratio of the three color components red, green, and blue were calculated and averaged for all pixels of the two dyes. A third, orthogonal vector was added to determine the absorption matrix depicted in Table 3.3.

Table 3.3: An absorption matrix for the two dyes, new fuchsine and hematoxylin, and an orthogonal rest vector

	R	G	B
hematoxylin	0.632	0.623	0.461
new fuchsine	0.360	0.689	0.629
rest (orthogonal)	0.232	-0.72	0.655

3.2.2 Minimum Distance to Mean Clustering

Minimum distance to mean clustering is a supervised standard clustering method. As first step, the cluster heads are calculated from pre-labeled test data. The cluster head is represented by a p -dimensional vector. The entries of the p -dimensional vector are calculated by averaging the value of a single property over all instances of the predefined test class. On the contrary to k -means clustering, the cluster heads do not change during the clustering iterations, since they are pre-calculated from the training data. For our approach, we classify the pixels according to their properties, which are color-coded pixel descriptors. Examples of descriptors are the intensity, the saturation and the brightness. The test samples are assigned to the cluster head with the minimum distance. Figure 3.4 gives an overview of the whole process.



Figure 3.4: Minimum distance to mean clustering The algorithm can be divided into three steps: First an annotated set of training samples is chosen. They are used to calculate the cluster heads in the second step. Third, test samples are classified, by choosing the cluster head with the minimum distance. Here, two pixel descriptors are considered, the saturation and the brightness of the pixel.

3.2.3 Multi-Resolution Clustering

Multi-resolution clustering is an extension of the minimal distance to mean clustering. The main idea is to take the level of detail in the image into account. WSIs may contain multiple versions of a tissue section image at multiple resolutions. The segmentation of tissue and background pixels does not need a high level of detail and can be performed at low resolution. If we want to classify more detailed structures we would need to process the image at a higher resolution. For example we can divide tissue pixels further into nuclei, cytoplasm, artifact, and $CD30^+$ pixels. Because of the hierarchical structure, we need only to consider pixels within areas which contain tissue pixels in the above layer. Thus, a region of interest can be defined by choosing a pixel class of interest. As a result, we exclude regions without information and thus reduce the amount of input data.

3.2.4 Region of Interest

The region of interest (ROI) defines a filtered subset of samples within the data set that is needed to answer a specific question. For image processing, this means to separate the image into background and the area containing the relevant information. The ROI can be expressed as a binary image or matrix. Entries with value 1 mark pixels or areas which are of interest and need to be processed further, while entries with value 0 represent background. Multiple methods exist to define ROIs. In our investigation, we are interested in the distribution of $CD30^+$ cells in the lymph node. Therefore, we have to separate the pixels into tissue and background. We use a minimum distance to mean

clustering approach to identify pixel classes. For more detailed questions, the ROI can be further limited. Because of the hierarchical composition of organs, i.e. different tissue types composed by specific cell types, a hierarchical approach can be used to refine the ROI stepwise. Tissue pixels classified in the low resolution image, e.g. layer 3, can be separated further into sub-classes like cell nuclei, cytoplasm and CD30⁺ stained cells. The sub-classes describe finer tissue structures, and the separation of the pixels requires a higher resolution.

3.2.5 Third Party Software

CellProfiler CellProfiler [67] is an open-source image processing software focused on cell detection and images with biological background. Since version 2.0, the core program is written in Python. CellProfiler also makes use of Cython [71] and the Java/Python bridge to allow the execution of algorithms in C/C++ or Java. The software is modular and extensible by CellProfiler or ImageJ plugins. The resulting flexibility and the provided graphical user interface allows a quick creation, testing and adaption of imaging pipelines. CellProfiler supports a majority of currently used image formats via the Bioformats library [72], which is provided by the Open Microscopy Environment (OME) project [73].

CellProfiler is designed to perform image processing tasks on high throughput data. Manually created pipelines can be applied to big data sets using the batch mode. Here, multiple instances of CellProfiler are created to run on multiple CPU cores or cluster nodes to gain a higher computing power.

For the cell detection, we use some pre-implemented methods in CellProfiler, therefore we implemented an interface to run automatically generated CellProfiler pipeline files in batch mode.

JAI and Fiji Impro has interfaces to other third party libraries and software. We use the java advanced imaging API (JAI) for simple imaging operations, like loading and saving in standard image file formats, rotation and cropping, and standard filters. Some plugins also make use of the more sophisticated functions of Fiji [70], a derivative of

ImageJ [74].

3.2.6 Segmentation

Image segmentation is a major step for object detection. The aim of a segmentation is to separate the image's pixels into separate, meaningful segments. Basic segmentation approaches can be divided into three steps. First, all image pixels are divided into classes during the pixel labeling. Second, a connected component analysis is performed to get labeled connected regions. Third, during the post-processing the initial segmentation gained from the connected component analysis is refined or corrected. The correction may include further knowledge about the target objects. Touching objects in the image might have been recognized as a single region and have to be split by the post-processing taking information, like size and morphology of the objects, into account. In most cases, pixels are labeled with two classes: the object class, containing the information of interest, and the background class, which is rejected for further analysis steps. Multiple segmentation methods exist, most of which are based on thresholding. In general, thresholding means to binarize a grey-scale image given a certain threshold. All pixels equal or above the threshold are converted to one, the rest is considered to be zero. For segmentation based on thresholding, grey-scale images can represent intensity values of the original image, but also more complex image properties are used for the pixel labeling. The image is then converted into a grey-scale image based on the specific property. For texture-based segmentation, the contrast, the brightness, or the co-occurrence of pixels within the locale neighborhood can be considered.

3.2.7 Cell Shape Descriptors

Cell shape descriptors measure morphological cell features like roundness. They can be used to classify different cell types or to gain further information about the state and function of a cell.

Here, we will consider morphological features. A more detailed description and additional shape descriptors can be found in the CellProfiler manual [75].

Table 3.4 gives a short explanation of common cell shape descriptors.

Table 3.4: Common cell shape descriptors

Shape descriptor	Explanation
area	number of pixels of the object
solidity	the object area divided by the convex hull of the object
eccentricity	the distance of the two foci divided by the major axis length of a fitted ellipse
mean radius	mean distance of all object pixels to their closest non-object pixel
minimum Feret diameter	minimal possible distance between two parallel tangents placed on opposite sides of the object
maximum Feret diameter	maximal possible distance between two parallel tangents placed on opposite sides of the object

3.2.8 Validation: Precision and Sensitivity

The validation of an automated imaging pipeline requires some sort of scoring. Two commonly used scores are the precision, or recall, and the sensitivity, or true positive rate. They are calculated according to Equation 3.3 and Equation 3.4, respectively. Both equations are based on the number of true positives (TP), false positives (FP), and false negatives (FN). TPs are objects correctly detected by the pipeline. FPs are detected as objects by the pipeline, but are wrongly classified as an object. FNs are existing objects in the image that were missed by the pipeline. The precision and sensitivity range between zero and one. Values close to one mean we have a high precision or sensitivity respectively. The precision and sensitivity are defined as:

$$Precision = \frac{TP}{TP + FP} \quad (3.3)$$

and

$$Sensitivity = \frac{TP}{TP + FN} \quad (3.4)$$

For cell detection, the two values can be interpreted as follows: The precision is the probability to draw a real cell when randomly picking one of the detected objects from the

automated imaging pipeline. The sensitivity is the percentage of identified cell objects by the automated pipeline compared to the actual number of real cells within the image.

3.3 Graph Theory

Graphs are widely used data structures and are applied to issues in multiple research areas like mathematics, computer science, and sociology. The terminology differs between the different areas, but as a common description we can say: a graph consists of a set of entities and a set of links. A link represents the relation between two entities. Here we use the mathematical terminology, according to which entities are called vertices and the links edges.

We apply graphs to model the spatial distribution of CD30⁺ cells in tissue sections of Hodgkin lymphoma. Vertices represent CD30⁺ cells and edges the spatial proximity between cell pairs. The cell graphs were generated as unit disk graphs.

The implemented software, Impro, supports multiple graph file formats as input and output formats, see Section 3.3.1.

The generated cell graphs consist of 500 to 50,000 vertices and can have 1,000,000 edges and more. Communities are sub-graphs with a high inter-connectivity. They allow a more generalized view and can be used to reduce the graph structure by merging cells into biologically meaningful cell groups. Graph partitioning methods and community detection are two commonly used methods to analyze the graph structure. Both can reveal highly connected sub graphs. In cell graphs a highly connected sub-graph might be a cluster of cells. The two methods will be discussed in Section Graph Partitioning and the following sections.

3.3.1 Graph File Formats

Graphs have many different applications and numerous implementations are available. For storage purposes and data exchange multiple file formats have been established. We discuss four common formats that are also supported by Impro. All presented file formats are plain text file formats (ASCII).

1. Edge list format The edge list format is a very common graph file format, which is supported by many graph-related programs. Graphs are represented as a list of edges. The format is often used to import data from external resources, e.g. experimental data, that can be interpreted as relationships. This includes tabular files like Excels XLS format or comma separated values (CSV) files. Listing 3.1 demonstrates a small sample in CSV format. The example shows a small part of a geometric graph, each line represents one edge. The first two columns are the indices of the connected vertices. The third column is the edge weight, i.e. the Euclidean distance between the two vertices. The following columns are specific for the graphs introduced in this work and contain additional information, such as the global x and y coordinates of the two vertices and their morphological classification.

Listing 3.1: Edge list format CSV example

```

...
5,1,253.0,12473,25819,12613,25608,Small_Cut,Small_Cut
5,3,301.0,12473,25819,12767,25751,Large_Cut,Small_Cut
6,7,481.0,5385,17162,5866,17134,Small_Round,Small_Cut
6,9,638.0,5385,17162,6023,17188,Large_Round,Small_Round
7,10,168.0,2907,19456,2781,19568,Small_Cut,Small_Cut
8,null,infinity,15232,14734,null,null,Small_Cut,null
9,12,672.0,16252,13264,15360,14394,Small_Cut,Large_Cut
...

```

In Impro we additionally allow null pointer edges to support isolated vertices, so we can add an edge where only the first vertex is defined. They are normally not part of an edge. In the null pointer edge, all values, referring to the second vertex, are set to 'null', see row 7 in Listing 3.1.

2. Graph Modelling Language - GML The Graph Modelling Language [76] was first introduced by the consortium of the GD '95, the Symposium on Graph Drawing. The file format follows a simple grammar, and mainly consist of a list of entries, each of which has a key and a value. Values can be numbers, strings or a list of entries, the latter allows nested structures. In principle, the grammar is capable to encode arbitrary

structures, but the GML format has mainly been developed for graphs. A new format based on GML is the Graph Markup Language (GraphML).

Listing 3.2 presents a simple graph consisting of two vertices, connected by one directed edge. Additional properties, for example edge weights, can be added as entries.

Listing 3.2: Graph Markup Language example graph

```
graph [  
  comment "GML format sample graph"  
  directed 1  
  node [  
    id 1  
    label "Node1"  
  ]  
  node [  
    id 2  
    label "Node2"  
  ]  
  edge [  
    source 1  
    target 2  
    weight 72  
  ]  
]
```

3. Trivial Graph Format - TGF The Trivial Graph Format is similar to the edge list format. The file contains a list of all vertices, followed by a list of the edges, see Listing 3.3.

Listing 3.3: Trivial Graph Format - TGF

```
1 "Node1"  
2 "Node2"  
#  
1 2 72
```


The rows after the '#' are interpreted as edges. In the example only one edge with an edge weight of 72 exists between vertex 1 and vertex 2.

4. KAVOSH The KAVOSH format is a very simple edge list format. Each row represents one edge defined by the two vertex indices separated by a blank character. Isolated vertices are not supported.

3.3.2 Unit Disk Graph

Unit disk graphs are both, geometric graphs and intersection graphs [51]. In geometric graphs, vertices and/or edges are associated with geometric objects. In unit disk graphs each vertex is linked to a circle in the plane, the unit disk, defining a threshold. The size of the circles within in a single graph is fixed for all vertices, given by a radius r . The edges are set according the intersection graph definition. Vertices are pairwise connected with an edge if their circles intersect and we call the corresponding set of circles the intersection model. The edge constraint can also be formulated as a dependency of the Euclidean distance of two vertices. For a given disk radius r , all pairs of vertices $(u, v) | u \neq v$ are connected by an edge, if the Euclidean distance between u and v is below $2 \times r$. This can be modeled as containment model by circles with a radius r_2 . Edges connect vertices, if the corresponding circle of a vertex contains the center of the circle of the other vertex. Here, we measure cell positions and geometric distances between cell objects. Therefore, we refer to the containment model, when we use the terms radius or disk.

A typical application of unit disk graphs is modeling of broad cast networks [77] and mobile phone systems. The vertices are transmitter and receiver of signals and the network models a communication network.

3.3.3 Random Geometric Graph

A geometric graph is a graph $G = (V, E)$ where each $v \in V$ is assigned to a geometric object. According to cell graphs, which are defined as unit disk graphs, this geometric object is a disk. The disk can be described by the x and y coordinate of its center and a fixed radius r . A random geometric graph $G_{rnd}(n)$ is generated by placing n vertices in

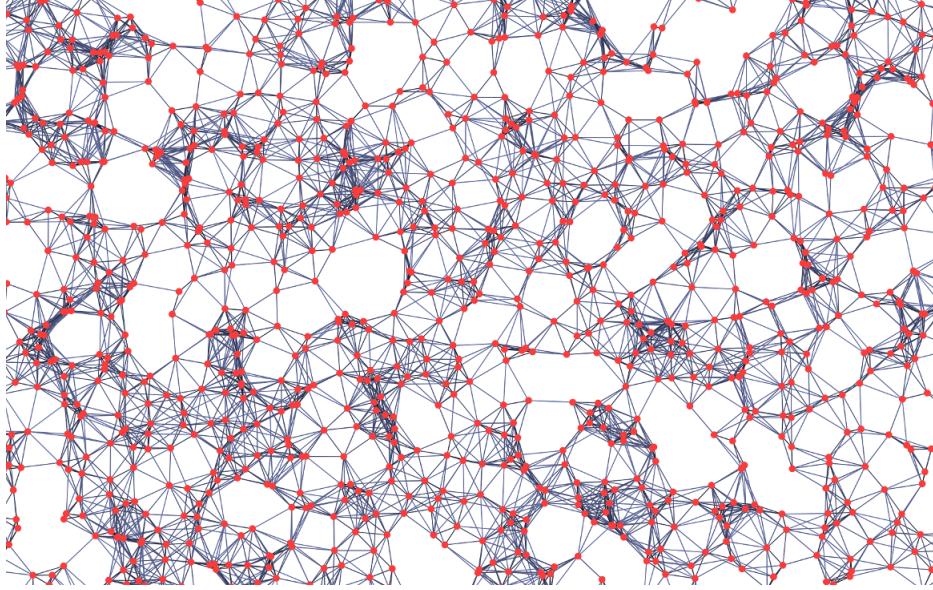


Figure 3.5: Random geometric graph The vertex positions of the random geometric graph are randomly distributed. Edges are defined by a distance threshold r . The cell density and distance threshold are chosen to match the real data obtained from a cHL tissue section.

a 1×1 plane. The center coordinates are chosen randomly. Vertices are defined by the containment model.

In this work, random geometric graphs are used as null model to compared cell graphs computed from WSIs to randomly distributed cells. The n and r parameter are fitted to match the corresponding cell graph. n equals the vertex count in the original cell graph. The disk radius r equals t , the distance threshold of the cell graph, scaled by the tissue area A of the image the cell graph is computed from, see Equation 3.5.

$$r = \frac{t}{\sqrt{A}} \quad (3.5)$$

Figure 3.5 presents a example section of a random geometric graph. Typically the graph does not contain highly connected areas and the vertices are evenly distributed over the 2D-plane.

3.3.4 Graph Partitioning

Multiple methods for graph partitioning exist. Many of them are based on cliques, which represent complete subgraphs and thus have the highest possible connectivity. One commonly used method is the community detection by k -clique percolation. The basic idea is to start with a k -clique as seed and then merge neighboring k -cliques into the community if they have a sufficient overlap. The method is discussed in Section *k-clique Percolation*.

Another possible way to generalize the graph is to merge vertices. In graph theory, this process is called *edge contraction*. The resulting graph has a lower amount of vertices and edges and it is called the minor of the original graph. This method can be applied similar as community detection. The method and its application to geometric graphs is explained in Section *Graph Reduction by k-Clique Contraction*.

Community detection and edge contraction reduce the graph and enable a higher abstraction level. Cells are clustered or merged based on a similarity measurement. In this study the reciprocal distances within the tissue were used. The method can be adapted by applying different distance measurements as edge weight, e.g. the difference between shape descriptors.

3.3.5 Communities

In network theory, the term community is synonymously used for a cluster or a group of vertices. The definition of communities is mostly problem specific. The decision which vertices build a community depends on the applied clustering algorithm. Fortunato *et al.* provide an extensive description of the basic concept of communities and graph clustering in general and discusses state of the art algorithms for community detection. In this work, we use the terminology and definitions described by Fortunato *et al.* [55], if not otherwise mentioned.

One goal of graph partition methods is to segment the vertices of a graph into groups, in such a way that we get a high connectivity within such a group, but a lower connectivity to vertices outside the group. A community has a high intra-cluster density, but a low inter-cluster density compared to the overall edge density of the graph. In graphs,

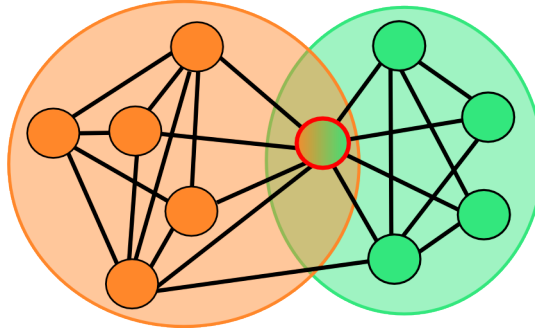


Figure 3.6: Overlapping community structure The depicted graph can be separated into two highly connected subgroups. The red marked vertex has edges to both groups and thus is a part of both communities. An example in social networks could be a person who shares two independent interests, like soccer and dancing, the communities could be a club or association the person participates in.

where edges often refer to some sort of communication between the connected vertices, a community means a group of vertices sharing more information to each other than they share on average to other vertices in the graph.

The coverage of a graph by overlapping communities is an alternative concept. It allows an overlap of communities, meaning that a single vertex in the graph can be part of multiple communities. Those vertices may be seen as an intermediate link between two communities. An example is depicted in Figure 3.6. The red bordered vertex is highly connected to both vertex groups, while only a few edges are present between both groups.

3.3.6 k -Clique Percolation

k -clique percolation is a common method to define communities within a graph. The basic idea is to use a k -clique as seed for a community and then to iteratively percolate into neighboring k -cliques, expanding the community. The criterion for two cliques to percolate into each other is the adjacency. Two k -cliques are adjacent if they have $k-1$ vertices in common. As this criterion is very strict for high k values, we use a relaxed definition for percolation:

Definition 1. *Two k -cliques l -percolate into each other if they share at least $k-l$ vertices.*

Various implementations for clique percolation exist, an example is CFinder [56] [78]. For the application to cell graphs, we adapted two different algorithms. The first

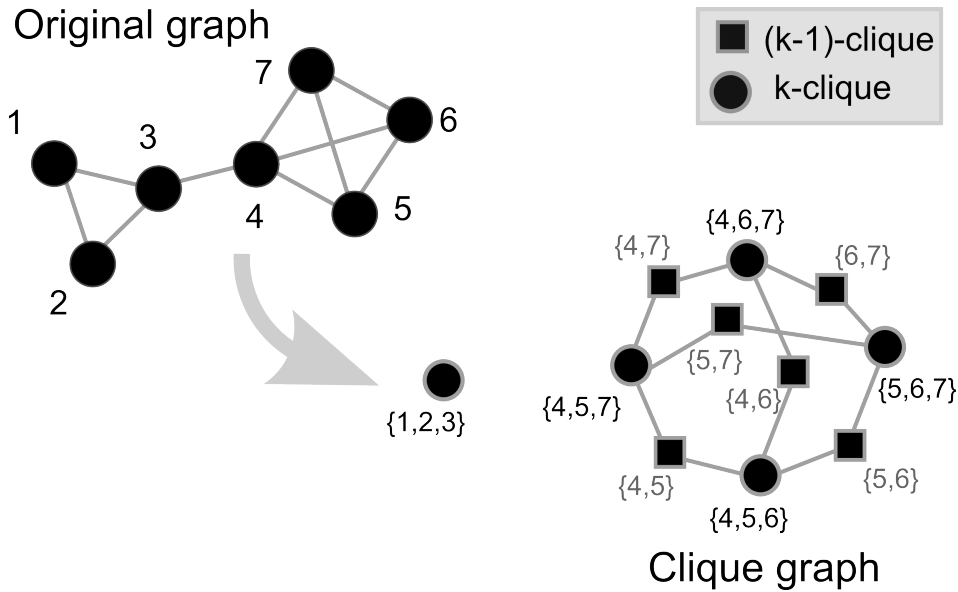


Figure 3.7: Clique graph creation From the original graph all k -cliques and all $(k - 1)$ -cliques are computed. They represent vertices in the clique graph. Edges connect k -cliques with all contained $(k - 1)$ -cliques.

algorithm enumerates the whole search space by computing all k -cliques in the graph. Then, all $(k - l)$ -cliques contained in each k -clique are enumerated. The two clique types are used to build a bipartite graph, where cliques are the vertices and edges connect k -cliques with all $(k - l)$ -cliques which they contain. In the resulting graph, all neighbors of a $(k - l)$ -clique vertex are k -cliques which can l -percolate into each other, see Figure 3.7 for an example. To obtain the communities of the graph a breadth-first-algorithm is performed to determine all connected components in the clique-graph. The union of the vertices of all cliques within a connected component can be considered as community.

The second algorithm of clique percolation focuses on low memory usage. For large and especially for dense graphs, the enumeration of all k - and $(k - l)$ -cliques results in memory usage, which exceeds the hardware specification of current desktop computers. Thus, the second implementation performs a local search to find and expand communities. It is noteworthy that communities, which are build up with k -clique percolation, always contain at least one vertex, which is distinct to all other communities in the graph. That is because detected communities are not further extensible. If a community is completely contained in another community or in multiple other communities this means

that it can percolate into those communities and thus is only a part of a community. Therefore, the local search algorithm only remembers, which vertex is already part of a community. For each vertex, which is not already part of a community, a random k -clique is computed and considered as seed for a new community. The clique is then expanded by l -percolation. As we do not remember any of the already computed k -cliques it is necessary to test all k -cliques for all new member of the community. The k -cliques have to be computed for each new added member and the overlap to the current community has to be checked against all its member vertices. The test against the whole community is computationally expensive, but on the other hand, the global search implementation has to check the existence of $(k - l)$ -cliques in the vertex list of the bipartite graph to correctly set edges. While the maximum size of a community equals the number of vertices of the original graph, the number of possible $(k - l)$ -cliques is much higher.

Chapter 4

Results

The following sections discuss the outcome of the image analysis applied to the set of 35 WSIs of lymphadenitis and cHL tissue sections. The results can be divided into two major parts. The first Section, *Imaging Pipeline*, includes all results related to the image processing. For some steps of the imaging pipeline, multiple approaches and multiple parameters were tested. Therefore, the first sections presents the different, implemented methods in Impro, followed by a comparison of their results and a discussion of the final outcome. The pixel-based methods and results of the quantification of the CD30 and hematoxylin staining are published in [36]. A statistical overview of the identified cell objects is given.

The second part is the *Graph Analysis*. The section covers the creation of graphs as a representation for the distribution of HRS cells in lymph node sections and the analysis of standard graph properties. The application of CD30⁺ cell graphs to cHL tissue sections is published in [6]

4.1 Imaging Pipeline

Object detection in digital images is a complex task that requires multiple steps. Therefore, a pipeline is needed to perform and to automate the image processing steps. A new software, Impro, was developed to combine and validate those steps. The software is introduced in Section *Impro Software*. The image processing steps of the pipeline

will be discussed in chronological order. They can be divided into three major subjects: pre-processing, object detection, and validation.

The Section *Pre-processing* combines all tasks needed for the preparation of the image, e.g. dealing with the file format, determining the regions of interest, eliminating artifacts, and improving the quality of the image. The requirements for the pre-processing depend on the image data and also on the question. The study aimed to analyze WSIs of lymph node sections to gain information on the distribution of CD30⁺ cells in cHL. For this application, we faced multiple challenges. First, the input images consisted of large tissue sections of high resolution. One point of the pre-processing was to reduce the input data. This was achieved by the identification of image areas containing tissue. Other parts of the image were skipped during the further processing. Second, the histological images were double stained with hematoxylin and fuchsin. To identify the CD30⁺ cells, the stains needed to be separated. This was achieved by color deconvolution. A side effect is that the processed images only contain a single color channel, resulting in a reduction of data before the computationally expensive object detection takes place. The third challenge was the quality of the staining. Even though images with high quality were preselected for the study, image quality varied within the image set. In most images, we also found small staining artifacts. Moreover, the intensity of the staining was not consistently good in all areas of the image.

The *Object Detection* requires the segmentation of the image and the labeling of the segments. Morphological properties of each detected object were computed. The thresholds for the segmentation were adapted for each image separately due to the different intensities of the staining. As the result of the object detection, CD30⁺ cell positions were extracted from the images. We will give a statistical overview of cell numbers and densities for the three disease types lymphadenitis, MCcHL, and NScHL. The measured cell objects were used to build a database for a set of 35 WSIs.

The last step of the imaging pipeline is the *Validation* of the detected cell objects. The outcome of the automated cell detection was compared with manually annotated cell positions to verify the overall accuracy. A graphical user interface (GUI) was implemented to allow the direct annotation of cell positions. Compared to manual cell positions, the

pipeline achieved a high precision and sensitivity.

4.1.1 Impro Software

For the cell detection, a new software, Impro, was implemented. The aim was to create a tool to perform the complete image processing. Impro focuses on two points. First, the software should be independent of the input image format. Figure 4.1 schematically depicts the structure of Impro. We have the core software (`impro.jar`), which enables the loading and managing of multiple WSIs. Besides, the jar file contains Java classes for basic functionality like import of WSI formats and their conversion to standard TIF files. Thus, all following image processing steps are independent of the input file format. The core software and its main components are described in more detail in paragraph *Impro core components*.

Second, the imaging pipeline should be adaptable, e.g. for a different staining or other cell types. To be more flexible and to allow the inclusion of additional features, Impro can be extended by plugins. Plugins are accessed via the graphical user interface of the core software. Extensive features, e.g. the pre-processing of the images and the cell detection, were implemented as plugins. An overview of the already existing plugins is listed in paragraph *Plugins overview*, including an overview of the provided functions.

In this project, the focus was on the identification of HRS cells in cHL tissue. Thus, the cell detection is tweaked for double stained histological WSIs with a fuchsine immuno staining targeted against CD30. For the purpose of clarity we will use the term *CD30 pipeline* to refer to this specific pipeline.

Impro core components The core program of Impro is `impro.jar`. The software is implemented in Java 1.8. Thus, it is platform independent and can be run on Desktop-PCs as well as on a computer cluster. General functions and data structures are implemented. The first core function is loading and managing of WSIs in different file formats. Impro makes use of the third party library `bioformats` [79], which is part of the open microscopy environment project [73] (OME). `Bioformats` allows read-access to established WSI formats of multiple scanner vendors like Aperio, Hamamatsu, Leica, and Zeiss. For further

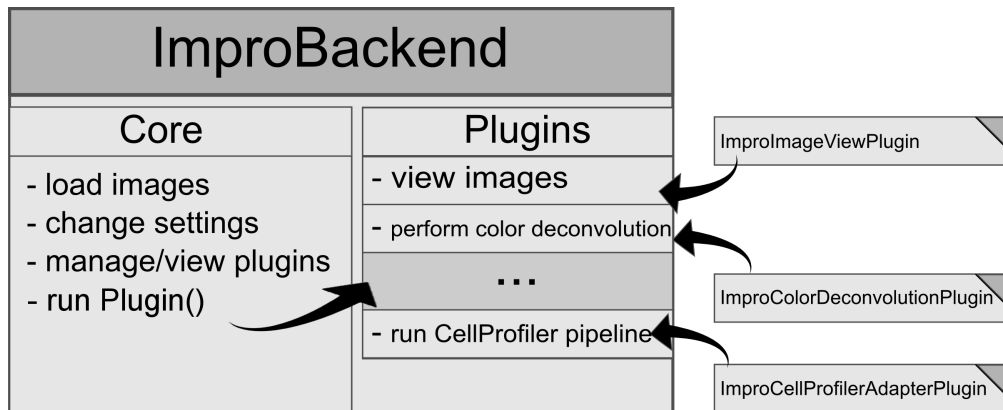


Figure 4.1: General structure of Impro The main program contains some core functionality to load images and set program properties. It loads and manages plugins, which can extend Impro. The additional functions for the image processing are provided by the separately implemented plugins. Plugins can be easily exchanged or added, allowing an adaptation of the pipeline and the implementation of new features. All plugins with graphical user interface can be accessed via the menu bar in Impro.

processing, the software deflates the images and exports image tiles as single image files in standard TIF format. The conversion to a standard format makes it possible to share and process the images in multiple third party programs. The management of the separated image tiles is done by *ImproImage*, a core class of Impro. The *ImproImage* class will be described in more detail in Section *Image Format*.

Impro provides a graphical user interface allowing the user to access single parts of the pipeline. In addition, it is also possible to use a headless mode. In this case, the complete pipeline runs without further actions of the user, e.g., for computations on a computer cluster.

The graphical user interface of Impro has three main components, displayed in Figure 4.2. The menu bar (1), is at the top, left corner of the GUI. It enables core functions, e.g., the loading of images. All installed plugins are listed in the menu bar and can be executed. The second important GUI element is the image list (2). The image list displays all loaded images and meta information, like the original file name, the image ID, and the resolution in microns per pixel (mpp). The bottom panel is called the plugin panel (3). Here, the active plugin is displayed. Multiple extensive features of Impro, e.g., processing images or performing a graph-based analysis can be accessed via this panel. The view in Figure 4.2 shows the *ImageViewerPlugin* exemplarily.

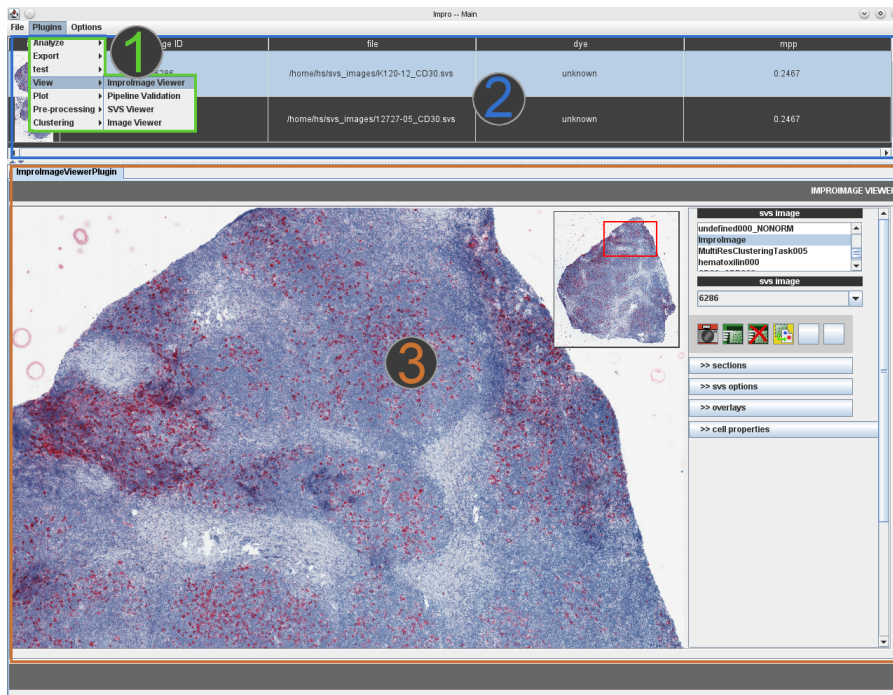


Figure 4.2: Graphical user interface (GUI) of Impro The main window contains three components 1) menu bar: configuration of Impro and access to plugins. 2) image list: displays loaded SVS images and their properties 3) the plugin view: displays the GUI of the currently selected plugin, here the ImprImageViewPlugin.

Beside the graphical mode, Impro and all its plugins can run in headless mode. The non-graphical mode can be used by starting the program with the additional parameter `--exec-commandfile <file>` (or the shortcut `-x <file>`). In this mode, Impro performs all tasks specified in the command file, which is given as `<file>`. Each line of the file represents an operation. The general structure of a command is:

Listing 4.1: Example command in Impro pipeline.

```

ImprPluginName;TaskName:property1=strValue;property2= \\
    intValue;property3=strValue2

```

First, the *ImprPluginName* specifies the plugin to be executed. Second, the *TaskName* selects the corresponding task to perform. This is followed by a list of properties, which is needed to run the task with specific settings. The latter depends on the requested task and is different for each plugin and command. Most operations provided by Impro or

as a plugin can also be executed using a corresponding text command. The command file enables the definition of a complete pipeline run including specific settings for each operation. The pipeline can run independently without further manual inputs of the user. Considering the running time of single operations of a few minutes up to several hours, most parts of the analysis have to be done on a computer cluster controlled by a command file.

Plugins Overview

Table 4.1 and Table 4.2 list the plugins and their features. Each plugin will be described in more detail in the following sections. For the sake of clarity, the plugins are divided into two groups. The first group consists of all plugins that take part in the image processing, see Table 4.1. The focus of the work is not the implementation, but rather the evaluation of cell detection in cHL WSIs. Therefore, we restrict the description of the implementation to a short discussion in the sections *Pre-processing*, *Object Detection*, and *Validation*.

Table 4.1: Overview of the plugins related to image processing

Plugin name	Feature description
ImageViewer	Visualization of ImproImages and additional overlays for displaying pipeline results
SVSViewer	Views images in Aperio's SVS format
ImageMultiResClustering	Image segmentation with minimal distance to mean clustering approach - defines a ROI based on pixel classes
ColorDeconvolution	Color deconvolution based on an absorption matrix or training images for up to three stains
CellProfilerAdapter	Interface for running CellProfiler pipelines
Registration	Rigid registration of whole slide images
PipelineValidation	Allows manual annotation of randomly sampled tissue section and performs the comparison of pipeline results with manually detected cells

The second group are plugins for the following analysis of the detected cell objects, including plugins for the graph-based and the statistical analysis. Table 4.2 shows the plugins of the second group with a short explanation of their function. The graph-based analysis is described in the Sections *CD30⁺ Cell Graphs* and *CD30⁺ Community*

Structures.

Table 4.2: Overview of the plugins for statistical analysis and graph-based methods

Plugin name	Feature outline
CellCSVExporter	Exports the results of the cell detection as CSV files
CellClassMap	Creates heatmaps based on cell classes, which are provided by the cell detection approach
Graph	Computation of several graph properties for cell graphs
GraphReducer	Finding community structures, using clique reduction or clique percolation
Heatmap	Creates 2D matrices, describing properties of the cell distribution, which can be displayed as heatmaps
Statistics	Provides descriptive statistics for distribution of cell graph properties (e.g. node degree and cluster coefficient)
Quant	Quantification of different pixel classes (e.g. the amount of CD30 ⁺ pixel)

Overview: Imaging Pipeline For CD30⁺ Cells

The identification of CD30⁺ cells in lymph node sections requires multiple steps. Here, we introduce the specific pipeline, the CD30 pipeline, see Figure 4.3. In Impro, a pipeline is defined as an ASCII file, each line schedules a command to be executed by Impro or by one of the plugins. The parameters for each processing step are also defined by the Impro command file. The input images are provided in the SVS format. The first step of the pipeline is the conversion into standard TIF files. The tissue section is recognized, using a pixel-wise minimum distance to mean classification, and defines the ROI for all further processing steps. Tissue tiles of the image are exported with an overlapping border, to avoid cutting of cell profiles. Color decomposition splits each image tile into two intensity images of the two stains hematoxylin and the CD30-targeted fuchsin. A threshold is applied to segment the CD30 intensity image into object and background pixels. The thresholding and object detection are done by a small CellProfiler pipeline, containing additional features like filtering of objects of small size and an intensity-based approach to separate touching cell objects. A post-processing step is needed to remove duplicated cells in the object detection result, because of the overlapping of the tissue tiles. Properties of each detected cell object are calculated and stored into a MySQL database. Based on cell positions, graphs are created that represent the spatial distribution of CD30 cells

in the lymph node sections. The graphs are stored in an edge list format into ASCII files. For a detailed overview of the specific parameter settings of the CD30 pipeline, see Listing S1. The following sections discuss the image processing steps in more detail and present intermediate results that are important to verify the performance and accuracy of the CD30 pipeline. The processing steps are presented in chronological order of the complete pipeline run.

4.1.2 Pre-Processing

Before the cell detection takes place, multiple pre-processing steps are required. First, the WSIs image have to be read. The following pre-processing steps are dependent on the image acquisition method. The color deconvolution splits the original RGB image into the two stainings. The definition of a ROI reduces the input for the cell recognition by separating the tissue tiles from background tiles.

Image Format

The image analysis should be mostly independent of the image format. The format of the digitized tissue sections differs for multiple scanner devices. Some parts of the cell recognition pipeline were implemented from scratch, but we also used third-party software like CellProfiler, which did not support WSI formats, like Aperio's SVS format, back in 2011 when the project started. Nowadays, software tools like CellProfiler and Fiji include the Bio-Formats library [80] [79] to enable read-access to WSI formats. Nevertheless, this direct import restricts the user to the pre-implemented methods of the software tools or at least requires the implementation of new plugins for the corresponding software. Thus, we kept the initial approach, to deflate and convert the WSIs. This was done by splitting each image into multiple files. Each image layer was split into single tiles. The default size of a tile is 3072 x 3072 for layer 0, the width and height of the low resolution layer depend on the downsample rate of the layer. Each tile was saved as a separate TIF image file. TIF is a standard image format, supported by most imaging software.

In Impro, the image is still treated as a single image. The class, which organizes the single image tiles, is called ImproImage. It only buffers currently needed image tiles.

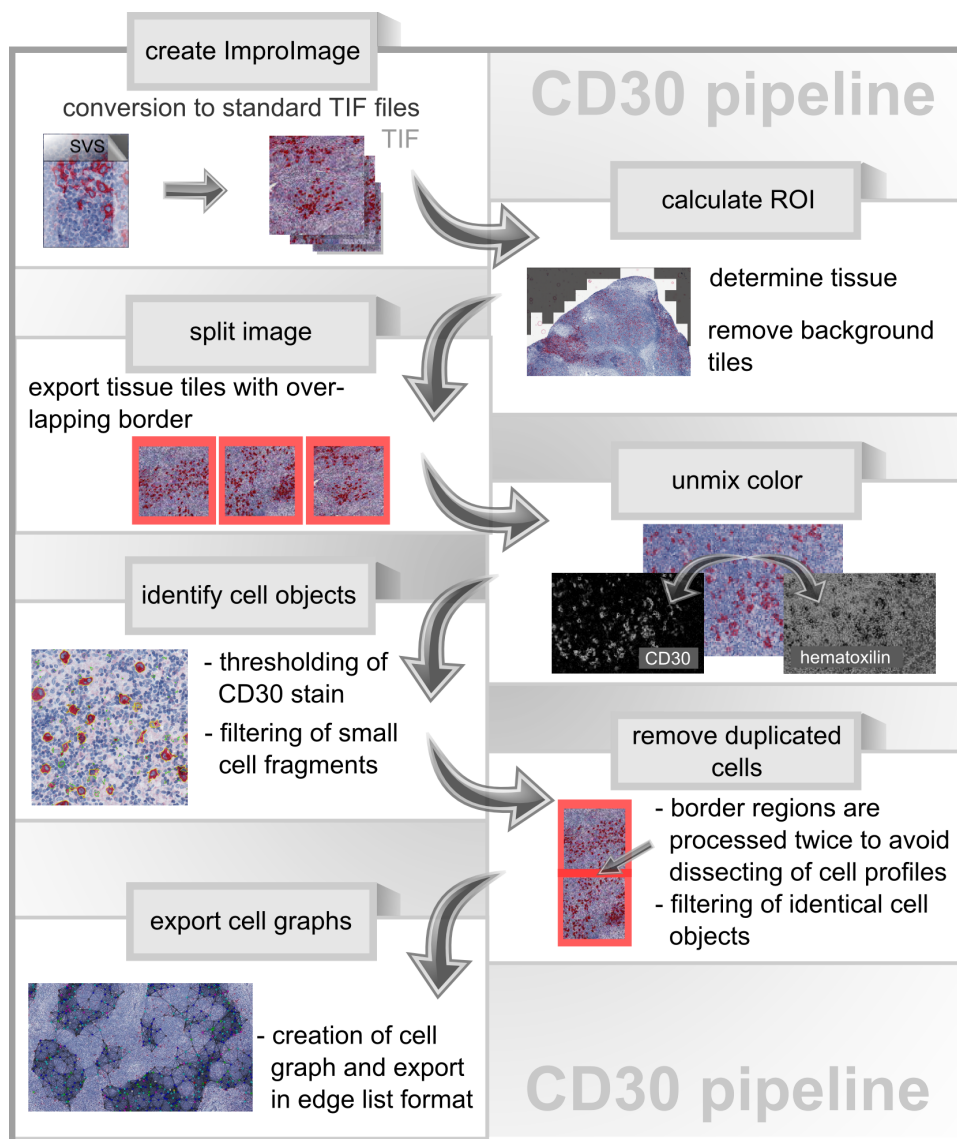


Figure 4.3: CD30 pipeline The CD30 pipeline is an adapted imaging pipeline, aiming for the identification of CD30⁺ cells in WSIs of lymph node sections. All major intermediate steps are depicted. Images are loaded in the SVS file format. Tissue areas in the image are determined and split into single, overlapping image tiles. The following image processing tasks can be executed in parallel. The RGB images are unmixed and separated into the CD30 and the hematoxylin stain. The CD30 stain image is thresholded and cell profiles are determined. Small cell fragments and duplicated cells are filtered out and cell properties are stored into a MySQL database. The recognized cell objects are used to build graphs, which are stored in standard graph file formats.

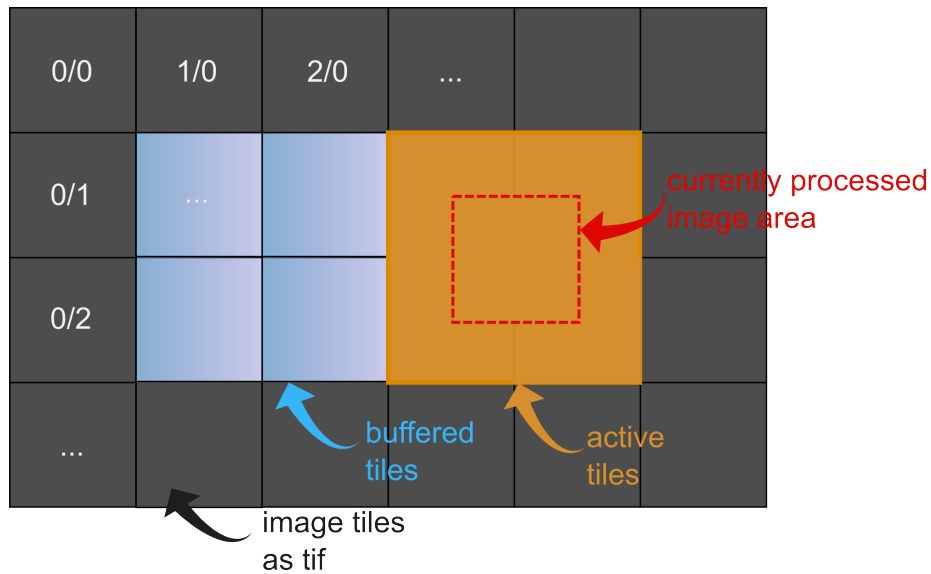


Figure 4.4: ImproImage: internal image format All image tiles are stored on hard disk in TIF file format. The tiles are active if they contain parts of the currently requested sub area of the image. The tiles are then buffered until the maximum tile number is achieved. The maximum can be altered in the configuration file. When the buffer is full, image tiles, which have not been requested for the longest time, are discarded.

Thus, if a sub-image is requested, only tiles containing parts of the requested area are loaded. The size of the buffer can be set via the configuration file. On normal desktop PCs, where only a small amount of memory is available, only a small number of tiles are buffered, while on a computer cluster nearly the whole image can be loaded into the RAM. ImproImage sorts loaded tiles in the order of their last requests. If the specified maximum number of loaded tiles is exceeded, the tiles which were not requested for the longest time are discarded. As a result, the memory usage remains on the same level during the run time. Often requested tiles are kept in the memory, minimizing the number of time-consuming I/O requests when working in local neighborhood of image tiles.

The advantages of ImproImage are the constant memory load and the ability to work with small parts of the WSI. On the other hand there is a higher number of I/O requests. Two classes were implemented to allow the processing of the whole image and to minimize the I/O for tile loading, called CompleteImproImageTask and CompleteImproImageGridTask. Most image tasks do not only require the processed image part itself,

but also additional surrounding pixels, e.g. filters like the Gaussian filter or the Sobel operator. Therefore, `ImproImage` always loads four image tiles, see the yellow area in Figure 4.4. The active area, which is currently processed, has the size of a single tile and is within the loaded tiles (red square). This ensures that one can access the neighborhood for each active pixel without loading tiles from the hard disk. The `CompleteImproTask` shifts the active area through the whole image. This is done row by row, but it switches the direction in each line. Each second line is parsed from the end to the start. As we shift a 2×2 tile area through the image, each inner tile is requested at least twice. By changing the read direction, the request tiles are always neighbored and thus most requested tiles are already buffered.

`CompleteImproImageGridTask` is an extension of `CompleteImproImageTask`. The tiles are parsed in the same way, but it allows to specify smaller tiles sizes. The image tiles are then separated into smaller sub-regions, which can be processed independently. A further advantage of splitting the image into small tiles is that it makes the image processing task parallelizeable. The specific task, e.g., the color deconvolution, can be executed by multiple processes, each of which handles another part of the image.

Image Viewer

In digital pathology, virtual microscopy has become more important in the last years. Digital slides can be accessed by multiple expert pathologists, allowing additional consultation and annotation by telepathology. Virtual microscopes read WSIs and mimic the user experience of optical microscopes. For example, standard operations like navigation through the image slides and zooming into specific image areas are supported. Visualization of the virtual slides can help to validate and to tune an automated imaging pipeline and to visualize the results directly on the tissue section. Currently, two visualization plugins are included.

The first plugin, `SVSViewer`, allows the inspection of images in Aperio's SVS format. The user can navigate through the digitized tissue slide, continuously zoom in and out, and select image regions (rectangular and free-hand), which can then be exported to separate image files. The viewer uses the `Openslide` library to read the image files.

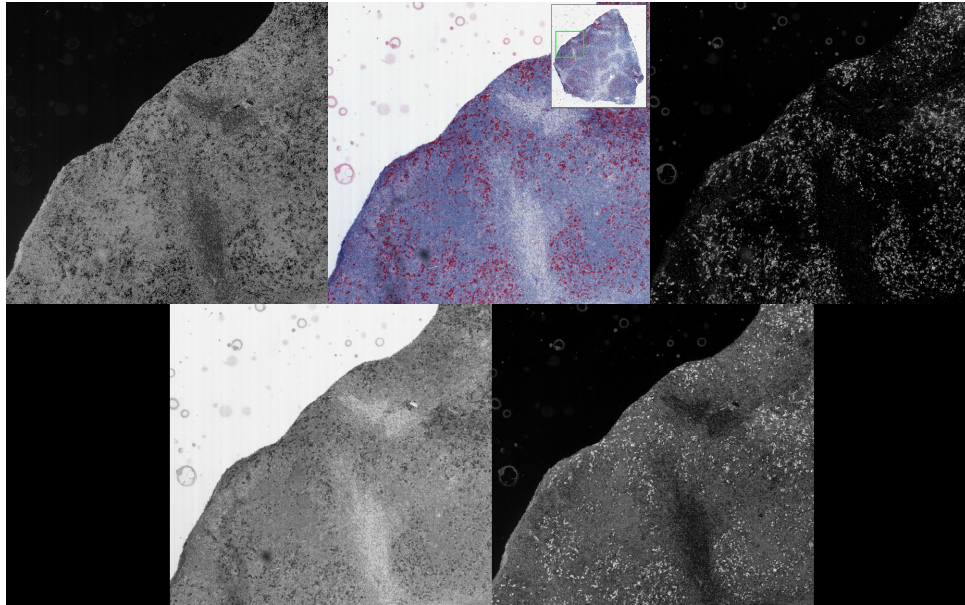


Figure 4.5: Different image properties displayed in the ImageViewer plugin The original image (colored) can be processed, resulting in different grey-scale images. Here, exemplary results from the color deconvolution (fuchsine, hematoxylin) and intensity values for the two color descriptors, brightness and saturation, are depicted.

The second viewer, ImageView, gives the user access to the exported image tiles created from the ImproImage class. In addition to the navigation within the image, it also provides features for the annotation of the tissue section. Different overlays can be added to visualize multiple properties, e.g., results of the imaging pipeline. The properties include different views of the histopathological image, like grey-scale versions of the intensity of the different stains, or image properties, e.g. the saturation or brightness of the pixel values. An overview can be seen in Fig. 4.5.

In addition, results gained by the cell recognition can be visualized. Cell positions and properties can be added as an overlay, and a comparative view of the impro pipeline cell detection and manually annotated cells for the validation is provided. Figure 4.6 demonstrates two available overlays. The cell overlay on the left visualizes the detected cell positions and their morphology classes. Other properties, like the cell density, can be displayed as heatmaps, see the right image in Figure 4.6. The density per tile is plotted. The color ranges from black, low cell density, to red for the highest cell density values. Zero values are not drawn.

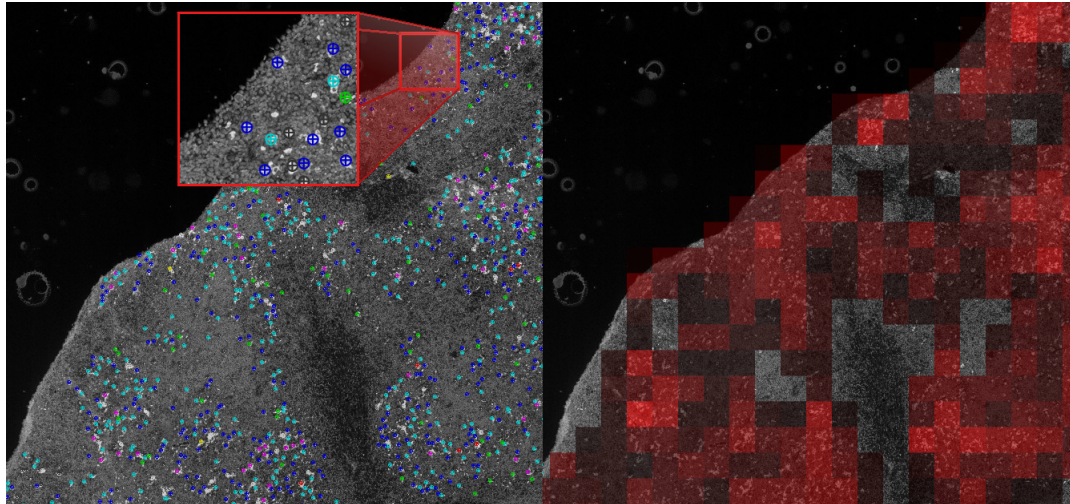


Figure 4.6: Results of the cell detection pipeline In the left image, the overlay displays cell positions, and the coloring is based on cell classes computed from multiple shape descriptors. Heatmap overlays, like the one shown in the right images, can be used to display general image properties. In this example the cell density is plotted.

Region of Interest

The ROI is limited to the part of the image covered by tissue. Therefore, the first step is the separation of the image pixels into background and tissue. The whole procedure, including the pixel classification and the definition of the ROI, is implemented in the ImageMultiResClustering plugin. The used multi-resolution clustering approach is described in Section 3.2.3. The plugin takes multiple parameters as input. First, a training image set has to be specified. The separation into pixel classes is done for each layer. Thus, training images and pixel classes have to be set separately for all layers. Second, the user has to determine the pixel signature. The signature is a set of descriptors, e.g., the BrightnessDescriptor, SaturationDescriptor, ContrastDescriptor, and VarianceDescriptor. In the CD30 pipeline, we defined the ROI using the two pixel classes *Background* and *Tissue*. The pixel class signature considered two descriptors, the BrightnessDescriptor and the SaturationDescriptor. Figure 4.6 summarizes the result of the training step. The mean signatures of the *Background* and the *Tissue* pixel classes are plotted. The error bars illustrate the standard deviation within the training image set. *Background* pixels are brighter than *Tissue* pixels, while *Tissue* pixels have a higher

saturation. Both, the differences in brightness and saturation, are caused by the hematoxylin and fuchsin staining. The blue line in Figure 4.7 shows the class border for the minimum distance to mean clustering approach used in this study. New test samples were either classified as *Background* if their pixel signature was located above, or as *Tissue*, when the pixel signature was below the class border line. The background of the plot also visualizes the occurrence of the combination of saturation and brightness values for an exemplary WSI. The intensity value of a pixel in the plot represents the count of pixels in the WSI in logarithmic scale. The plot shows a high and narrow peak near the *Background* signature. A large fraction of pixels in the WSI have a similar signature and can be considered to be unstained *Background* pixels. *Tissue* pixels have a higher variation regarding saturation and brightness. Differences are reasoned by the two stains and the diversity within the tissue area. In cell nuclei, there are more negatively charged molecules and more hematoxylin can bind compared to cytoplasmic image areas. The signature of a cell nuclei pixel is dark and highly saturated. In contrast, sclerotic bands are barely stained, and the pixel signatures in such areas are rather unsaturated and bright. The segmentation into the two pixel classes is depicted in the second step of Figure 4.8.

After the assignment of a pixel class to each image pixel, the ImageMultiResClustering plugin is able to define the ROI. Two additional parameters are needed. First, one of the pixel classes has to be chosen as the class of interest (COI). That is the pixel class that contains the information to be extracted from the image. Here, the COI is the *Tissue* pixel class. The second parameter specifies a threshold. The image is processed in tiles and only tiles whose amount of COI pixels is above the threshold are considered to be part of the ROI. The tiling and the resulting ROI can be seen in the right image of Figure 4.8.

When dealing with WSI data, one faces multiple challenges. The tissue preparation and staining process is very sensitive and often leads to artifacts in the image. Unspecific staining can pollute the glass slide in some areas. The slicing of the tissue section stresses the tissue and may result in tearing up fragments. In many cases, smaller fragments are clinched or even folded. Cell positions in this area are displaced. To avoid such artifacts,

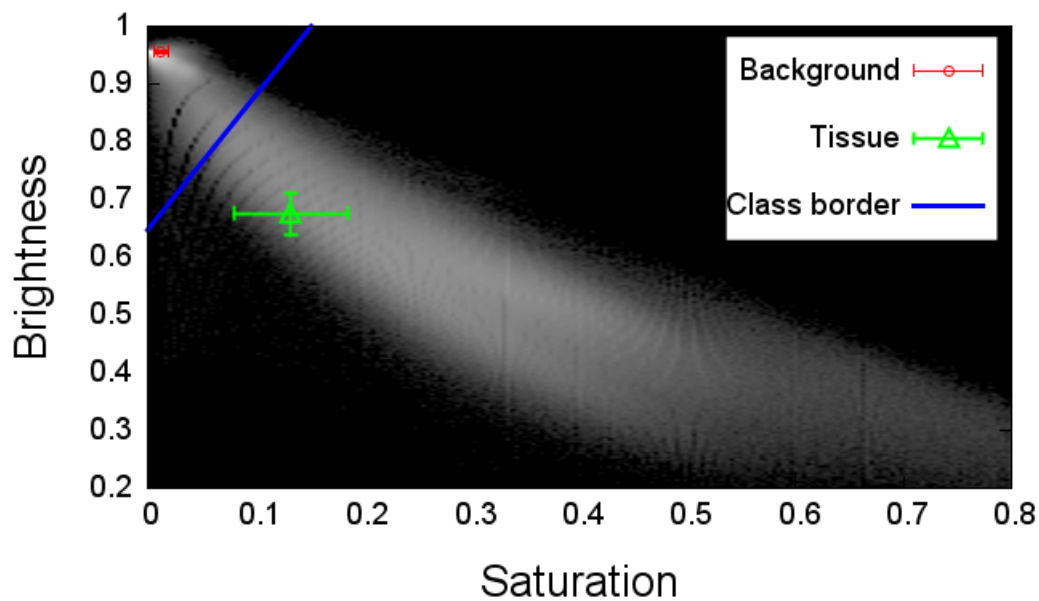


Figure 4.7: Class signatures for the multi-resolution clustering Two pixel classes were considered: *background* pixels and *Tissue* pixels. The signatures include the brightness and saturation value. Beside the cluster heads, which are the average vectors of the training data set, the standard deviation for both parameters are plotted. *Background* pixel have high brightness values and a very low saturation. *Tissue* pixels are more saturated and the variation of brightness and saturation values is much higher within the class. The border between both pixel classes is drawn in blue. The intensity values in the background of the plot represent the pixel signature distribution in one example NScHL WSI in logarithmic scale.

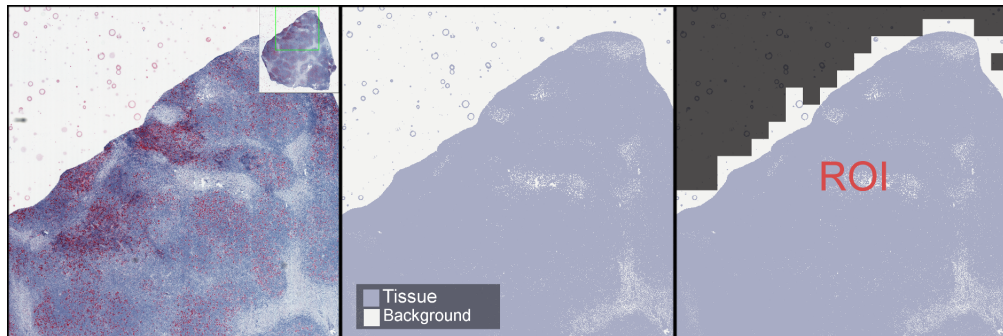


Figure 4.8: Multi-resolution clustering pipeline steps The original image (left) is segmented into the two pixel classes *Background* pixels and *Tissue* pixels (middle). The segmented image is used to create the ROI (right), only tiles are considered with a certain amount of *Tissue* pixels.

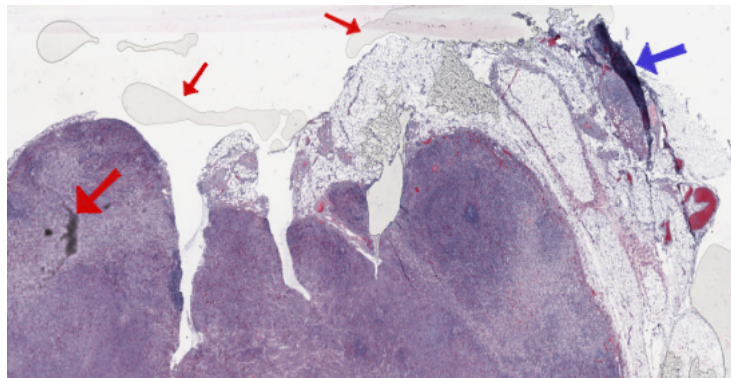


Figure 4.9: Artifacts in WSIs Slicing and staining of the tissue samples can produce artifacts in WSIs. The red arrows mark a typical pollution of the glass slide caused by air bubbles and dried liquid. The blue arrow marks a clinched and folded tissue fragment. Here, the displacement of the cells does not allow to determine the cell distribution correctly. We defined the ROI in such a way that staining artifacts and fragmented tissue were ignored in the following pipeline steps.

the ROI was additionally post-processed.

A region growing algorithm was applied to the detected ROI to identify large tissue fragments within the WSI. Small fragments, which most likely are artifacts, are discarded.

The aim of the definition of the ROI is to reduce the input data. Image 4.10 depicts the percentage of tissue tiles in the three image groups lymphadenitis, MCcHL, and NScHL. Within the image sets, tissue covered 35 % to 80 % of the image area. In most cases, 40 % or more of the initial data was identified as background and was discarded.

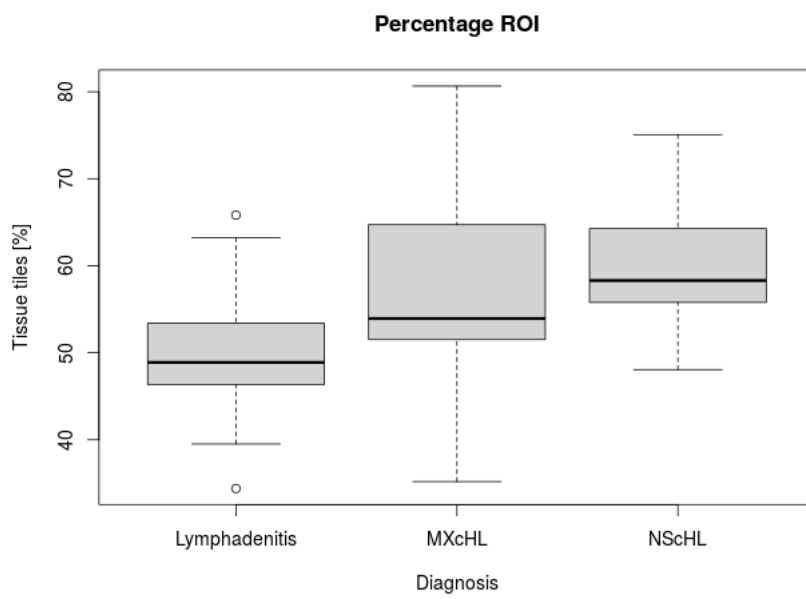


Figure 4.10: Region of interest The relative amount of tissue tiles for the three diagnoses, lymphadenitis, MCcHL, and NScHL are plotted. The images have at least 35 % tissue tiles, but most commonly the amount of tissue tiles range between 50 and 60 %. The definition of the ROI reduces the input data in those cases by 40 %.

Color Deconvolution

For the color deconvolution, pre-selected image sections were used as training set to define the absorption matrix for the two color dyes, hematoxylin and fuchsin. An overview of the graphical user interface is given in Figure 4.11. Area 1 describes the color dyes, including the name, a list of the training images for the computation of the absorption vectors, and a gradient, which represents the dye in RGB color space for different intensity values. On the right side, two dyes can be selected and displayed as a gradient to demonstrate mixed pixel values. In our work, only two dyes are used: hematoxylin and fuchsin (CD30 staining). The third absorption vector is calculated as an orthogonal vector and represents the staining, which is not caused by the dyes.

The plugin allows processing of images in TIF file format and Aperio's SVS format. Properties for the input and output of the color deconvolution can be set by the user, see area 2 and 4 in Figure 4.11.

The sections are manually chosen from the set of double stained images. The samples chosen from the histological image are always a mixture of the dyes. To obtain the absorption matrix for a single dye, it would be necessary to prepare mono-stained images. Figure 4.12 depicts an exemplary result of the color deconvolution, for a small sub-region of a WSI.

The dyes were not separated in all image regions. The test samples for the dye fuchsin always contained a small amount of hematoxylin, shifting the absorption vector slightly. Thus, the intensity of hematoxylin is zero in CD30⁺ areas. The cytoplasm of CD30⁺ cells appears black in the hematoxylin image, see red arrows in Figure 4.12. Here, the pixels were divided into CD30⁺ and CD30⁻. The exact amount of dye was not of interest, thus the error during the stain separation did not affect the cell detection.

Normalization

WSIs of histopathological tissue slides can vary by multiple factors. Sources of variance are the staining process, possible pollution of the slides, and changes in the illumination caused by the optical scanning device. Nowadays, we can overcome some of those artifacts, e.g., modern scanning devices automatically post-process the captured images

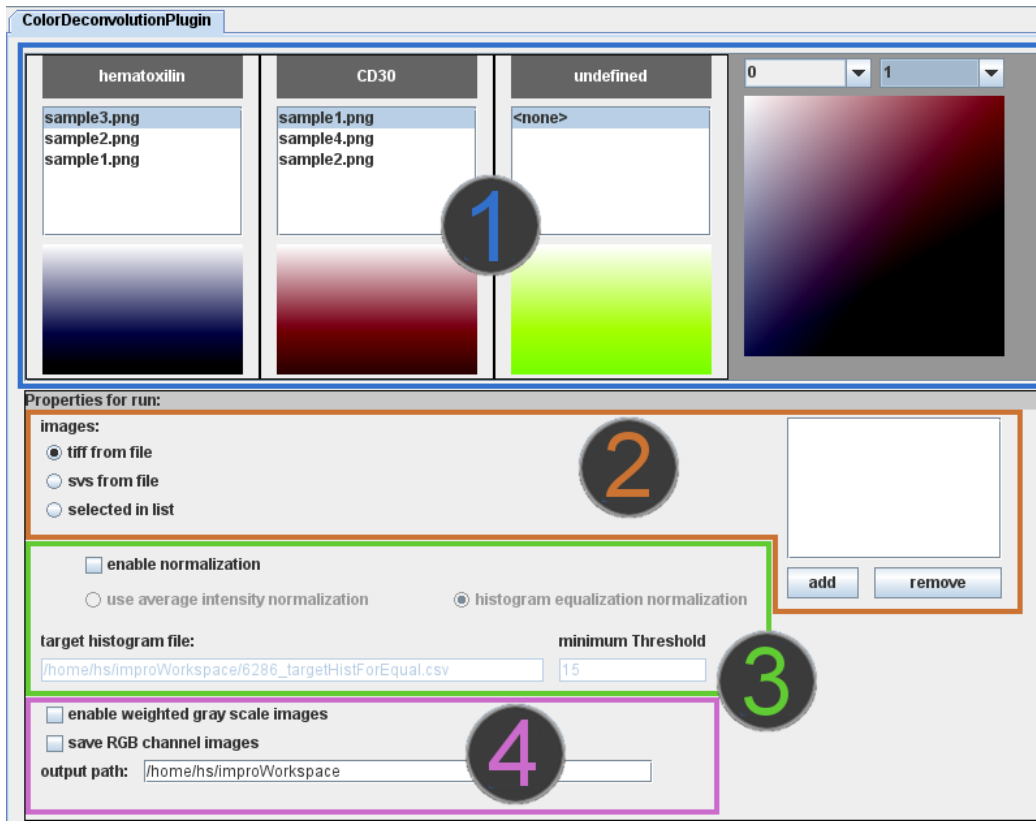


Figure 4.11: GUI of the ColorDeconvolutionPlugin 1) The calculated color vectors for different intensity values and a list of the input sample files. 2) Selection of the input files for the color deconvolution step. 3) Normalization of the resulting grey scale images. 4) Output target directory and optional output images.

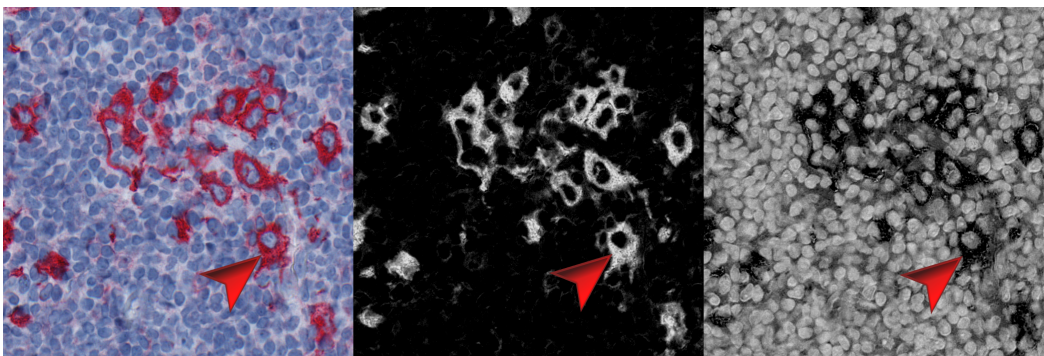


Figure 4.12: Output of the color deconvolution step The separation of the color stain into the two channels hematoxylin and fuchsin. The original image is processed, the resulting images show the intensities of fuchsin (middle) and hematoxylin (right). The dyes are not completely separated, thus areas that are CD30⁺-stained (see red arrow) have zero intensity in the hematoxylin channel.

to eliminate illumination differences, caused by the optics of the scanner. But, *color nonstandardness* [81] is still challenging when dealing with histopathological WSIs.

For the CD30 pipeline, two normalization approaches were tested and implemented. They were provided by the ColorDeconvolution plugin and were executed directly after the separation of the color stains. The first method, average intensity normalization, scales all intensity values by a constant factor s . The user specifies the target average intensity for the image. The factor s is then calculated according to Equation 4.1.

$$s = \frac{intensity_{target}}{intensity_{observed}} \quad (4.1)$$

$Intensity_{target}$ is the average intensity value that shall be reached in the image, the $intensity_{observed}$ is the measured average intensity in the image. The user can also specify a minimum intensity value. Pixels below that value are excluded and are not taken into account for the $intensity_{observed}$ calculation. By default, this threshold is one, and only zero valued pixels are excluded. The exclusion of low intensity values is intended to specify a threshold separating pixels with unspecific staining from pixels with specific staining, e.g., CD30⁺ pixels. The value has to be chosen manually because the quality of the images differs, and intensity values for specific and unspecific staining vary from slide to slide. We tested multiple automatic methods to determine the minimum threshold, e.g., based on the intensity histogram, but due to the heterogeneous distribution of CD30⁺ cells and the occurrence of staining artifacts within the images the minimum threshold could not be determined automatically for the WSIs. An exemplary result of the average intensity normalization with a minimum threshold of 0.059 and $intensity_{target}$ set to 0.294 is depicted in Figure 4.13 C. The target intensity value was determined, using manually selected sub-images with a staining of high quality, meaning a high intensity of the CD30⁺ specific staining and nearly no unspecific staining.

The second normalization method provided by Impro is based on histogram equalization. The algorithm gets one parameter as input, the target histogram. The intensity value histogram of the image is then computed. The intensity values of the test image histogram are shifted to match the intensity value distribution in the target histogram. Each intensity value of the test image histogram is mapped to a new value. In contrast

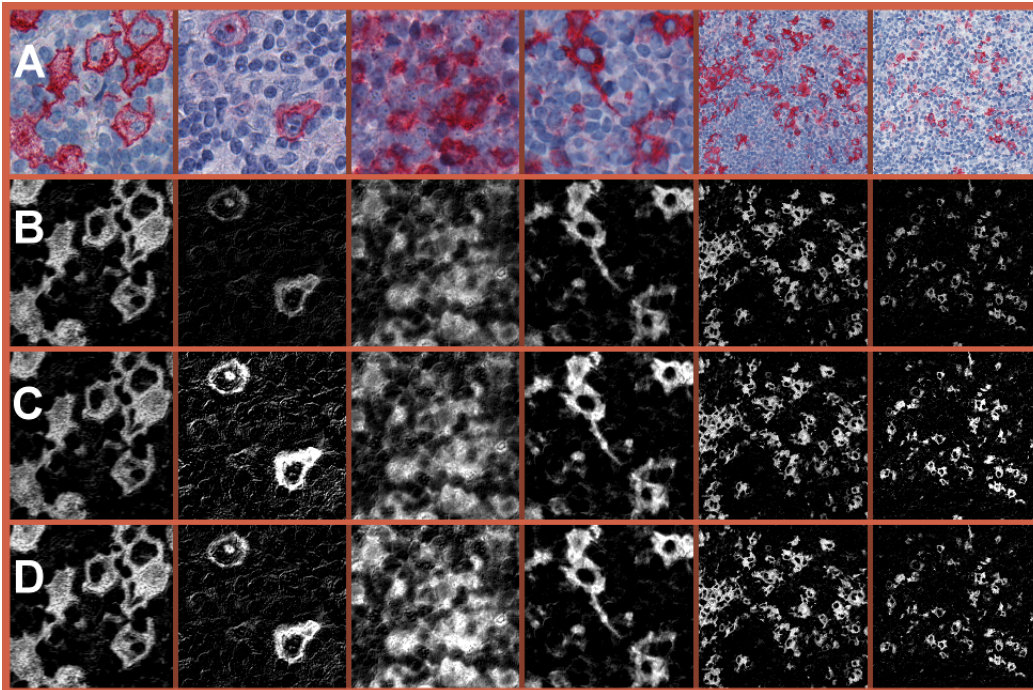


Figure 4.13: Normalization in Impro The columns represent six tissue samples taken from cHL WSIs. The last two columns are printed with a lower magnification. The samples vary in the quality of the CD30 staining. Row A depicts the original histopathological images in RGB color space. The samples in column two and six have an overall lower CD30 staining intensity. For the sample in column 2, we can observe a high variance regarding the CD30 staining. Sample 3 has a high CD30 staining intensity, but the surrounding area of the CD30⁺ cells is polluted by unspecific staining. Row B displays the CD30 channel obtained by color deconvolution. In row C and D, intensity values are normalized. The average intensity method is shown in C. The result of histogram normalization is depicted in row D.

to the average intensity method, the factor applied to each pixel value is different for each intensity value.

Figure 4.13 shows six tissue samples, four of which are close-up views, the last two tissue sections are presented with a lower magnification. Row A presents the original tissue samples directly cropped out from the histopathological image. The samples represent typical conditions that can be found in WSIs of cHL. The sample in the first column has a broad range of intensity values. The CD30⁺ cells have highly saturated red areas. The cytosolic areas are brighter and less saturated. Column two and six are examples for low intensity CD30⁺ staining. Column three, four, and five have high intensity values for CD30 staining, but column three additionally shows a lot of unspecific staining.

4.1.3 Object Detection

The last section focuses on the pre-processing. To extract information about the distribution of CD30⁺ cells in the tissue sections, cells have to be detected as objects. In the current setup, three plugins provide functions for the cell detection. The first plugin, *CellProfilerAdapter*, is an interface to CellProfiler. CellProfiler is used to perform the image segmentation and the object detection. The second plugin, *CellCSVExporter*, allows to export results from the *CellProfilerAdapter* to files in CSV format. The last plugin, *PipelineValidation*, provides methods to validate the quality of the cell detection.

CD30⁺ Pixel Segmentation

The *CellProfilerAdapter* plugin is an adapter for CellProfiler. For the processing in CellProfiler the images need some preparation. The first step is the adaptation of the image format. The initially used CellProfiler version (svn:9661) did not support WSI image formats like the SVS format. In newer versions the bioformats library is included in CellProfiler, and all common WSI formats are supported. However, the managing of the different image layers and image tiles is not done automatically. Thus, we decided to keep our own image format to be more flexible and to have a better control regarding memory usage. The handling of the WSIs and image tiles is described in detail in Section 4.1.2. We use a tile size of 1,024 x 1,024 pixels and all tiles are treated with a border of 100 pixels. The reason for the border is the cutting of cells profiles as shown in Figure 4.14.

The treatment of the WSI as separate tiles may result in splitting single cell objects if they are located on tile borders. A measured cell profile touching the border of a tile might be incomplete. To avoid detecting just parts of a cell, those objects are rejected. The overlap of 200 px (100 px per border) makes it possible to detect cells within the border area. The border area is handled multiple times resulting in a longer running time because of the increased area, see Table 4.3. For small tile sizes, the influence of the border is rather high. If a small running time is required and the memory resources are high enough, the overall overlap can be reduced by expanding the tile sizes. For image tiles with a size of 4,096 x 4,096 px and a border of 100 px, the running time is assumed

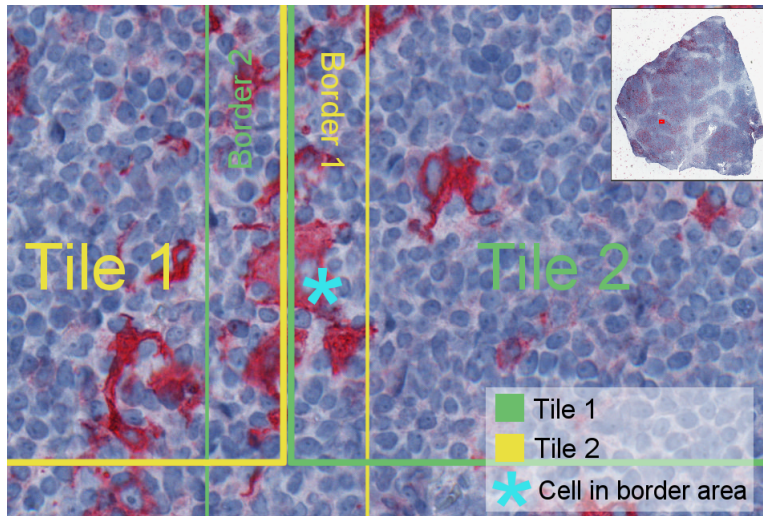


Figure 4.14: Image border for cell detection The image shows two neighboring image tiles (thick lines) and their extension of 100 pixels (thin line). The border ensures that cells at the border of two tiles (see blue mark) can be detected by CellProfiler. The overlapping area is processed multiple times. Thus, a single cell might be detected twice. To remove duplicates, the results have to be post-processed.

to be 1.078-fold.

Table 4.3: Relative size of an image for different tile sizes and a fixed border of 100 px

Tile size	Relative image area
1,024	1.4288
2,048	1.2048
4,096	1.0781

Some cells might be detected multiple times. The marked cell in Figure 4.14 is located within the overlapping part of two neighboring tiles. As the cell does not touch the border, the cell profile is detected in both image tiles. To correctly handle duplicates, the results of the CellProfiler pipeline are post-processed. The local cell positions detected by CellProfiler in single image tiles are used to calculate the global cell positions. An additional filtering step removes duplicates by comparing the centers of mass of each pair of cell objects. If the distance of two centers of mass is below a threshold of five pixels, one of the objects is discarded. The error margin of five pixels is necessary if adaptive thresholds are used for the cell detection. The different thresholds in neighboring tiles could lead to slightly changed cell profiles and thus to a shift of the center of mass.

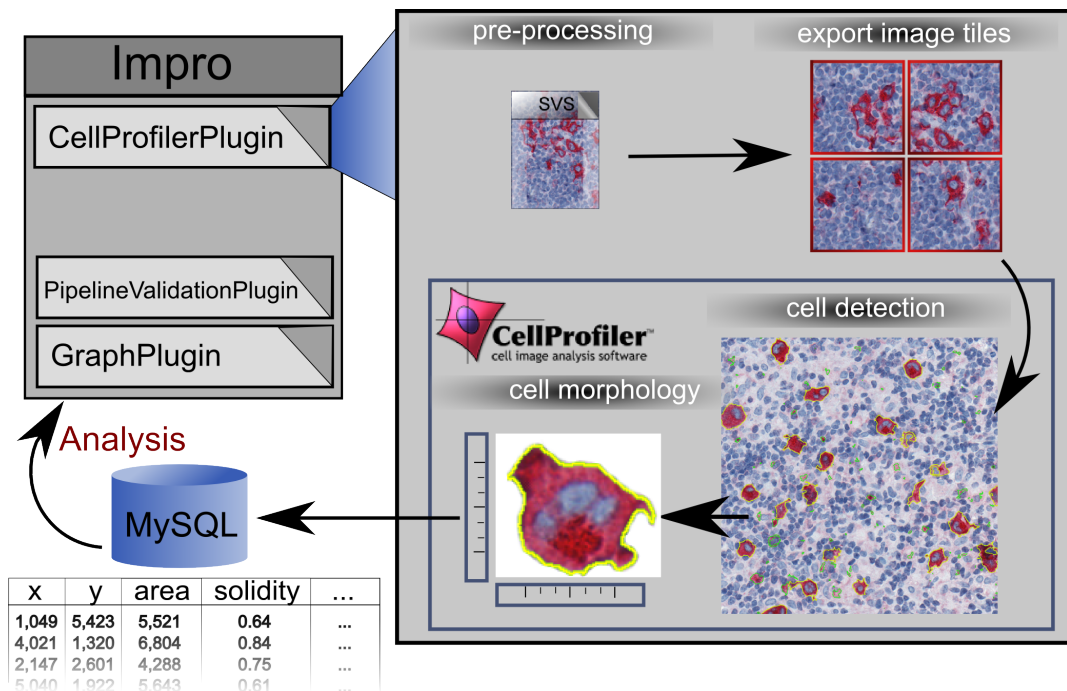


Figure 4.15: CellProfilerAdapter overview The diagram summarizes the pipeline steps done by the CellProfilerAdapter plugin. The image is first pre-processed and converted into overlapping image tiles. Then, the cell recognition is done using CellProfiler. The cell position and other properties, e.g. cell morphology descriptors, are measured and saved into a MySQL database. Afterwards, the collected data is analyzed and processed further in Impro.

The actual object detection step is done in CellProfiler. Therefore, Impro executes CellProfiler as a new process in batch mode. Table 4.4 summarizes the CellProfiler pipeline. We developed the customized CellProfiler pipeline for the detection of CD30⁺ cells. All modules are listed, and the used parameters are printed.

First, the LoadImages module is used to load single image tiles exported by the *CellProfilerAdapter* plugin in Impro. The LoadImages module also reads meta data from the file name. This includes the SVSID of the original image and the global X and Y coordinates of the upper left corner of the processed image tile. The meta data is needed to put the image tiles, which are treated separately, into the context of the WSI images. The additional X and Y coordinates allow the calculation of the global cell positions, using the local cell positions in the specific image tile. After loading, the images are processed by the IdentifyPrimaryObjects module. The module provides multiple thresholding methods including global and adaptive approaches. Global thresholds are

Table 4.4: The CellProfiler modules and parameters used for the object detection

Module name	Parameters
LoadImages	Extract metadata from where? File name Regular expression...: <i>see below</i>
IdentifyPrimaryObjects	Typical diameter of objects: 60 to 150 Discard objects outside the diameter range? No Discard objects touching the border of the image? Yes Threshold strategy: Manual Manual threshold: < <i>dependent on the input image</i> > Method to distinguish clumped objects: Intensity Fill holes in identified objects? Yes
MeasureObjectSizeShape	Calculate the Zernike features? No
FilterObjects	Select the filtering method: Limits Select the measurement to filter by: AreaShape/Area Filter using a minimum measurement value? Yes Threshold strategy: Manual Minimum value: 1750.0 Filter using a maximum measurement value? No
MeasureObjectSizeShape	Calculate the Zernike features? No
ExportToDatabase	Add a prefix to table names? Yes Table prefix? %%%PREFIX%%%
CreateBatchFiles	Store batch files in default output folder? Yes

applied equally for all pixels within an image, while adaptive approaches are locally adapted, thus regions of one image are treated differently. In our current setup, we use the Manual threshold option. The threshold is set by the user and is applied globally to the complete image. In addition, the threshold is the same for all image tiles when CellProfiler is used in batch mode. The global threshold allows no adaptation to local differences in the staining quality, but the results turned out to be more consistent. The adaptive thresholds, which are calculated for each single image tile separately, can also generate wrong results.

The algorithms to automatically adjust the threshold rely on assumptions, e.g. the percentage of objects pixels within an image. The expected percentage number of object

pixels has to be set for the Mixture of Gaussian (MoG) thresholding method by the user manually. Other thresholds like the Otsu threshold work best if 50 % of the image is covered by foreground pixels. The fraction of foreground pixels can not be predefined for our WSIs. The total CD30⁺ cell number within different tissue sections is highly variable. The separate treatment of the image tiles causes additional issues, as the CD30⁺ cell density is not even in the whole lymph node section. In some areas, cells cluster and 50 % or more of the areas are covered by foreground pixels. On the contrary, there are also benign areas, which not have yet been infiltrated by any malignant cells. Besides, the immuno staining quality can differ within a single image, resulting in unspecific staining in some regions. The adaptive thresholds tend to misinterpret such unspecific staining and set low intensity thresholds, meaning that the unspecific stained pixels are considered to be foreground pixels.

Figure 4.16 shows example images for CD30 staining with different qualities (row A) and the resulting segmentation for three different thresholding methods, provided by CellProfiler (B-D). The colored outlines mark image pixels considered to be foreground pixels, all other pixels are treated as background. The tested thresholding methods are Maximum correlation thresholding (MCT), MoG, and RobustBackground *. Yellow outlined objects also surpass the size threshold and are thus classified as cell objects, see also Section *CD30⁺ Objects*. Green outlined objects are too small to be recognized as whole cell profiles. They are probably cell fragments or caused by unspecific staining.

Figure 4.16 exemplarily depicts results for the adaptive thresholds. The thresholds for specific staining with high intensity values results in correct segmentation. Here, MCT and MoG perform best. The RobustBackground method computes a higher threshold, and small cells with lower intensity are not detected. Even for lower intensity images (second column in Figure 4.16), all three methods calculate the threshold correctly and the segmentation is similar. Thus, for input images, where the majority of pixels are actual foreground pixels, the adaptive thresholds perform well. Column three is an example image containing a high amount of unspecific staining. Even though the intensity of the specifically stained cells is much higher, the quality of the segmentation decreases.

*<http://www.cellprofiler.org/CPmanual/>

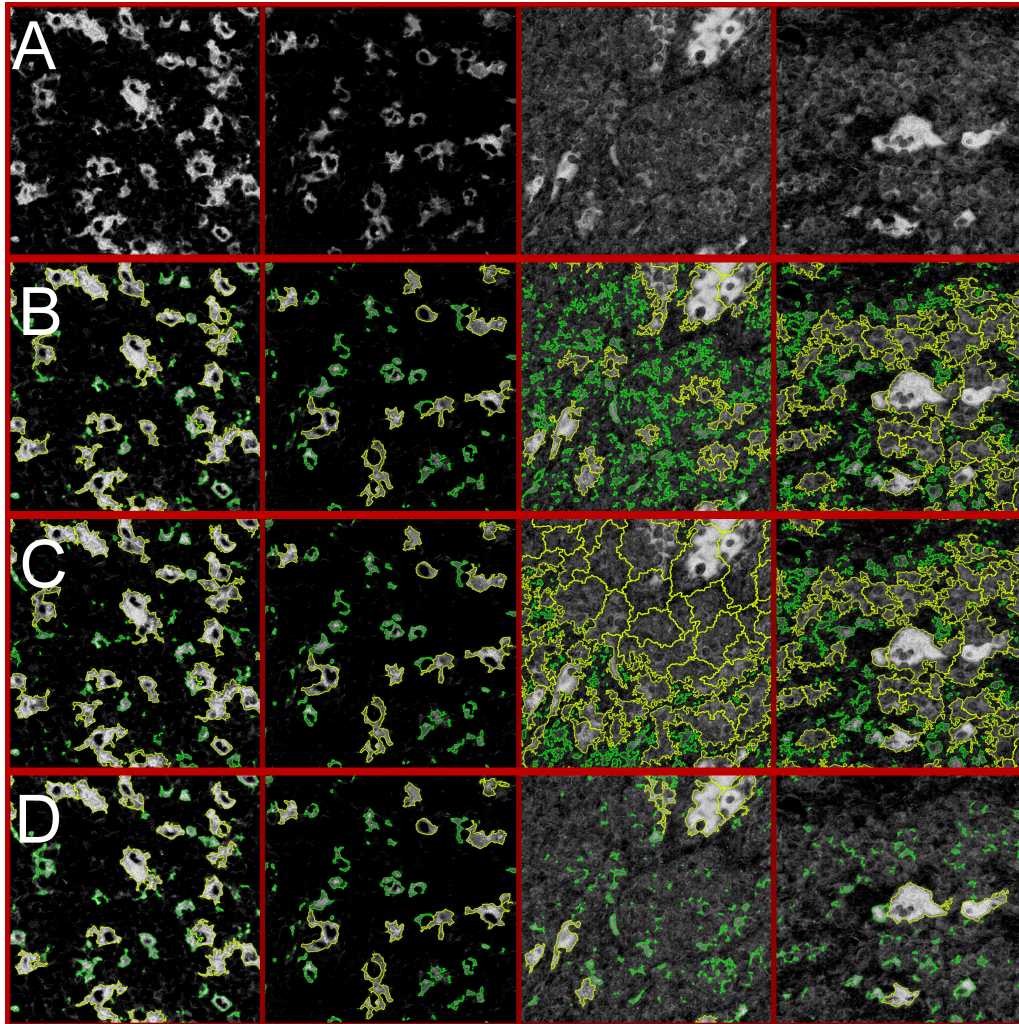


Figure 4.16: Object detection with automatic thresholds provided by CellProfiler The image shows different approaches to determine thresholds for the image segmentation. Each row depicts the results for one method applied to four sample images. The first and second column represents images with a very specific CD30 staining with high intensity and low intensity values. The third and fourth column represent image regions which also include unspecific CD30 staining.

Compared to the other methods, the MoG methods calculates a very small threshold. Here, almost all image pixels are considered to be foreground pixels and cause a large number of false positive objects. MCT calculates a more precise threshold, but still many unspecifically stained pixels are above the threshold. Most of those pixels are discarded during the object size filtering step. Nevertheless, multiple false positive cells remain.

Because of the high heterogeneity in the cHL image set, manual, global thresholds were applied. The thresholds were manually set by HS and TS by inspecting low resolution versions of the images. The results were validated. In some cases, mainly lymphadenitis cases with a low CD30 stain intensity, the results were refined to gain a higher quality in the object detection.

After the thresholding, the images are binarized. Pixels are either classified as *Foreground* or as *Background* pixels.

A similar approach was implemented by AS in the ImproQuant plugin. The classification was extended and both stainings, CD30 and hematoxylin, were used. Pixels were assigned to either being CD30_positive, hematoxylin_positive or unstained. Figure S1 illustrates the quantification results for the three disease types lymphadenitis, MCcHL, and NScHL. The pixel classes are distributed differently in the three diagnoses. The amount of CD30_positive pixels were low in all lymphadenitis images. In cHL the overall amount of CD30_positive pixels were higher. Single cases exist, where the relative occurrence of CD30 is comparably low as in lymphadenitis, but other cases contain 5 % and more CD30_positive pixels. The ratio of unstained and hematoxylin pixels also differs in the three image sets. In lymphadenitis, the unstained pixels are slightly more frequently found than hematoxylin pixels. The distribution of the two pixel classes highly differs in NScHL. Here, a high amount of unstained pixels is typical. 70-90 % of the pixels were classified as unstained. MCcHL shows a high variability. On the one hand, unstained pixels are on average much more frequent than hematoxylin pixels, which is similar to NScHL. On the other hand, some images had an increased fraction of hematoxylin pixels. Overall, the two image sets lymphadenitis and NScHL have a very small overlap according to the analyzed pixel class distributions. MCcHL has some characteristics that are similar to NScHL, but the distribution of pixel classes can not

be used to fully separate MCcHL from lymphadenitis cases. The results of the the pixel quantification were published in [36].

CD30⁺ Objects

The last step of the object detection was the object labeling. From the thresholding we obtained the foreground pixels. The labeling assigns each foreground pixel to one object. In case of the CD30⁺ objects, we faced some requirements to precisely assign the foreground pixels. Additional processing steps, also provided by the IdentifyPrimaryObjects module in CellProfiler, were needed.

CD30, the target protein for the immuno staining, is located mainly in the membrane. The staining only highlights the membrane and the cytoplasm of the cell, while no CD30 is present in the cell's nucleus. As a result, pixels representing the nucleus are not treated as *Foreground* pixels. The *Fill holes in identified objects?* option in CellProfiler was enabled to overcome the misclassification. The algorithm searches for pixels classified as *Background* that are enclosed by *Foreground* pixels and reclassifies them as *Foreground* pixels. In case of the malignant CD30⁺ HRS-cells, the method performs well. The malignant cells are enlarged and large cytoplasmic areas surround the cell nuclei pixels. In rare cases, the cell nuclei are located close to the border of the cell. It is possible that not all membrane pixels are classified as foreground pixels and the nucleus is not fully surrounded by *Foreground* pixels. Here, the correction step does not work and the nucleus is not counted as part of the cell object. The effects of different thresholds and the fill whole operation can be seen in Figure 4.17.

Touching Cell Objects Another issue for the labeling are touching or clumped objects. After the labeling, objects are treated differently depending on the classification in single or clumped objects by CellProfiler. The latter are post-processed to separate the touching cells into single cell objects. In the CD30 pipeline, clumped objects are separated based on intensity. The second method, separation based on shape, works best if the objects have a regular, e.g., circular, shape. Here, clumped objects tend to have cleavages, where the single objects are touching each other. In cHL images the

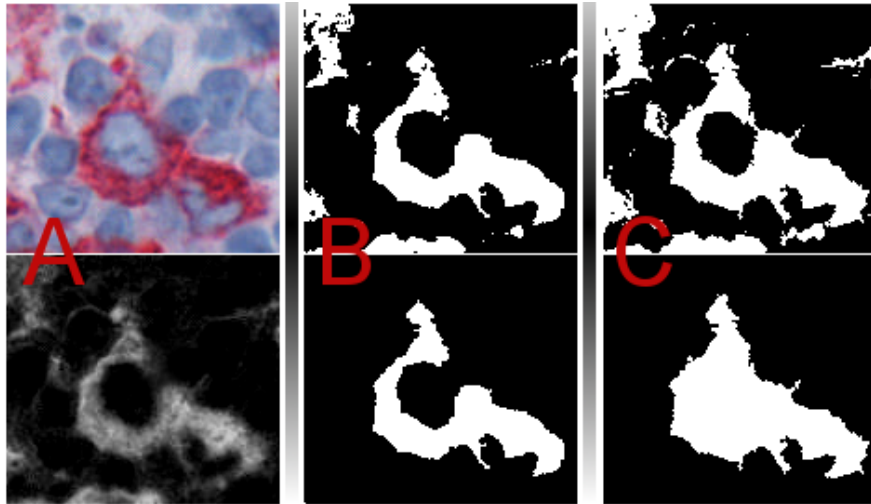


Figure 4.17: Thresholding and fill whole operation Column A shows a single CD30⁺ cell object in the original RGB image and the corresponding fuchsin stain after the color deconvolution. The nucleus of the cell is located near the upper right bond of the cell and the fuchsin staining has a relative low intensity compared to the rest of the cell object. Column B and C demonstrate two intensity thresholds in the upper images and the application of the fill whole operation in the bottom image. In B, the nucleus is not enclosed by object pixels. As a result, the nucleus is not detected to be part of the CD30⁺ cell object. The slightly decreased threshold in C fills the gap and the cell nucleus is annotated correctly. Such slight shifts in the intensity threshold can have a high impact to the morphology detected by the imaging pipeline. The cell object in B has a decreased area and a lower solidity.

malignant cells can build large cell clusters. As the shape-based method just considers the outline of such a cell cluster, there is no information about the cell borders within the clumped object, leading to a relatively random separation. The intensity-based method in CellProfiler uses local intensity maxima as seeds. The border line between the touching objects is detected by applying a watershed algorithm. While both methods showed good results for small clumped objects with a cell count of two or three, the intensity-based method yielded better results for large cell clusters. However, even though the recognized cell objects approach better to manually fitted cell outlines, the detection of fine structures like micro spikes, which are important for cell migration, cannot be separated in clumped objects. In the case of very dense cell groups, such fine structures also cannot be identified by human eye.

The accuracy of the results depends on the level of detail we are looking at. While the general cell count in an image area does not require the accurate cell position of each single cell, advanced analyses, e.g. the nearest neighbor distances, are more influenced by

incorrectly determined cell positions. Within large cell clusters, the cell positions might be slightly shifted, but even though the borders of the cells are not accurately determined, distances between the cells' centers of weight are only influenced by a few pixels, and the overall error is small. However, detailed properties like the cell morphology might be highly dependent on the correct cell outlines. The solidity for example is decreased if fine structures like micro spikes are not detected. This issue is related to the *coastline paradox*. Similar to coastlines, cell outlines can have a fractal-like structure and the measured length depends on the used level of detail. Such properties might still be used to classify the cell profiles in the image, but one needs to keep in mind that they might be caused by the image processing method, and the morphological property does not fully reflect the biological state of the cell.

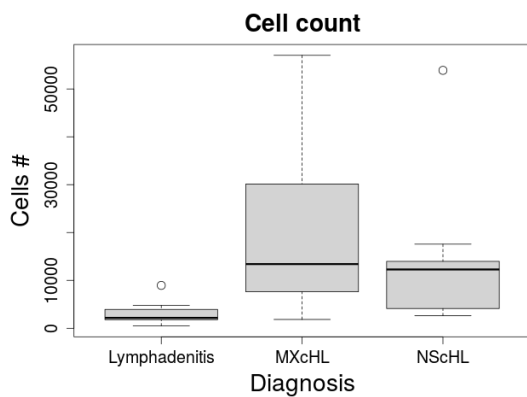


Figure 4.18: Cell count The count of detected CD30⁺ cell objects for the three image types lymphadenitis, MCcHL, and NScHL. The count ranges between 470 and 57,033 cells.

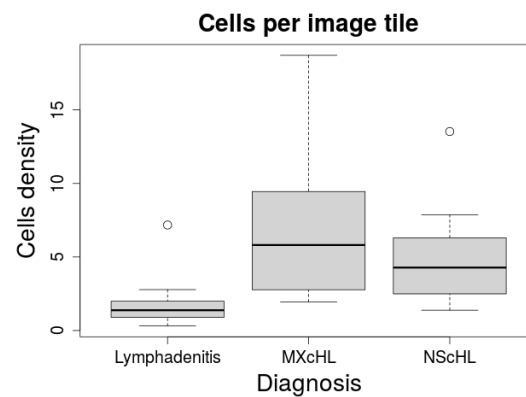


Figure 4.19: Cell density per image tile The cell density in the CD30⁺ image set. The density corresponds to the area of a single ROI tile. The count of cells per 0.59 mm² tissue is depicted.

Filtering Of Small Cell Fragments A third factor for labeling the objects in the cHL images set arises from the sectioning of the lymph node and the fact that we deal with 2D image data. Each cell profile we recognize is a 2D profile gained by sectioning a 3D object, the malignant cell. The cutting plane of the tissue determines the observed cell profiles. Some objects might be cut through their center, but others might only be cut at the border. As we are also interested in the cell morphology, we want to exclude small CD30⁺ cell fragments obtained by the sectioning process. A filtering step

is applied, excluding all CD30⁺ objects with a size below a certain threshold. The *MeasureObjectSizeShape* module of CellProfiler is used to calculate the area size of the potential cell objects. The *Filter using a minimum measurement value?* property in the FilterObjects module is enabled to apply the filtering step, see Table 4.4. In the CD30 pipeline, the lower bound is set to an area size of 1,750 pixels. In the tissue section, this corresponds to an area of about 110 μm^2 . For a perfect circular profile this would mean a diameter of 5.917 μm . In comparison, an erythrocyte, one of the smallest cell types in the human body, has a typical diameter of 7-8 μm . Smaller objects are considered to be cell fragments, and the actual cell is not part of the 2D plane captured by the microscope.

CD30⁺ Cell Count Figure 4.18 depicts the cell counts for the CD30⁺ image set. We found the lowest count of CD30⁺ cells in the lymphadenitis cases. Ten out of eleven lymph node sections contain less than 5,000 cell objects. The average cell count for the diagnosis lymphadentis is $\sim 3,034$. Image 5161 has an untypical high cell count within the lymphadenitis set, about 9,000 cells were detected.

In the two cHL image sets, which have a higher cell count compared to the lymphadenitis image set, we detected the most CD30⁺ cells in MCcHL with $\sim 19,000$ cells on average. The MCcHL lymph nodes had a broad range of different cell counts. While the cell count ranges between 2,600 and 18,000 for eleven out of twelve NScHL cases, we found four MCcHL cases with a cell count greater than 20,000. In both cHL image sets, a single case is present showing a cell count greater than 50,000.

CD30⁺ Cell Density Figure 4.19 plots the cell density expressed as the cell count per image tile. One tile corresponds to a tissue area of 0.59 μm^2 . If we compare the three image sub sets, we get similar results compared to the cell count. Overall, the density of CD30⁺ cells is increased in both cHL image sets. Except one case, the density values range between 0.31 and 2.78 CD30⁺ cells per image tile. In contrast to the cell count, the cell density is more symmetrically distributed. This might be due to the fact that the total cell count is also influenced by the size of the lymph node section, while this does not effect the cell density. Figure 4.20 depicts the correlation between cell count and cell density. Here, the influence of the tissue section size becomes clearly visible.

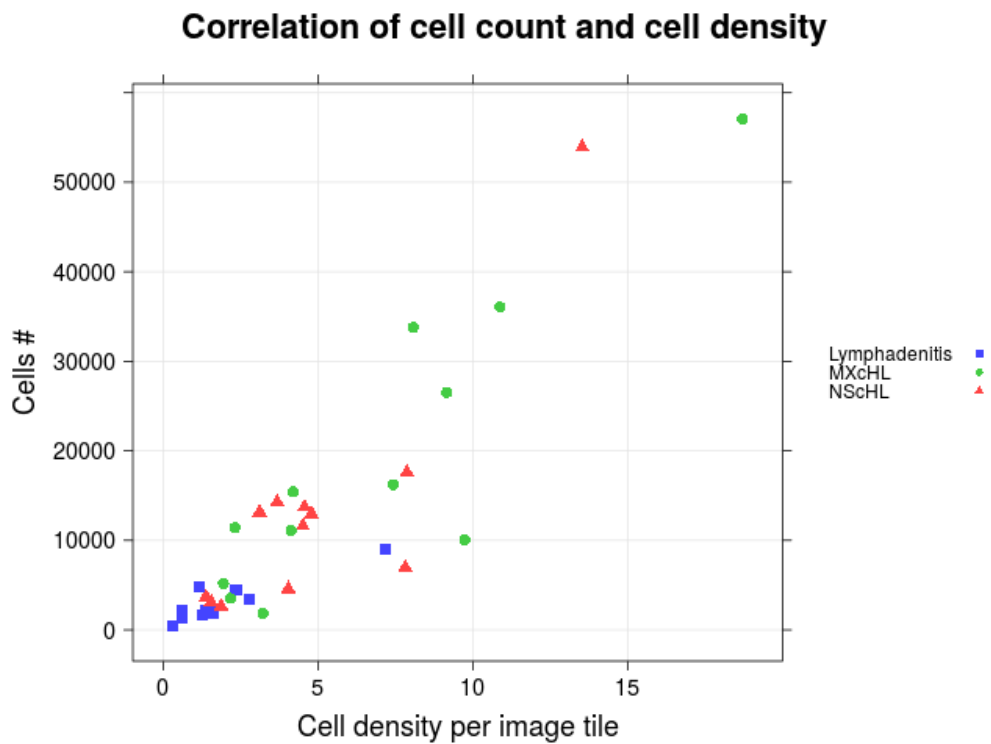


Figure 4.20: Correlation of cell count and cell density The plot depicts the three image sets, the cell count, and cell density of each case. The plot visualizes the variation caused by the size of the analyzed lymph node sections. In some cases, the cell count is threefold higher for cases with comparably high cell density. For example, one MCcHL case with a cell density of ~ 10 has about 10,000 cells, while cases with a similar cell density exhibit a cell count of 28,000 and more.

There is a high variance in the tissue section size, for example, observable for the two MCcHL cases 2949 and 5822. Case 2949 had a cell density of 9.75, and $\sim 10,000$ cell objects were detected. Image 5822 had a decreased cell density of 9.16, but the detected cell count, $\sim 26,500$, was increased by a factor of 2.64. The high difference is caused by the lymph node sizes of 0.68 cm^2 and 1.90 cm^2 , respectively. The global structure of a lymph node has a high variability due to anatomical location, age, diet, and antigen exposure [82]. The lymphatic system adapts to and memorizes infections, which is also partly reflected in the variability of the lymph node structure. In this study, lymph nodes were not labeled according to their anatomical location. Nevertheless, we need to keep this in mind as one factor for the variability in the data.

Figure 4.21 visualizes some typical density distributions for MCcHL cases (A) and NScHL cases (B). The white lines mark the tissue sections detected by the CD30 pipeline. For the calculation of the cell density, only the highlighted region is taken into account. To be representative for the broad ranges of cell counts in the samples, the cases were selected and ordered by the total cell count. The top tissue sections contain more than 15,000 CD30⁺ cells. The cell count decreases towards the bottom of the image. The last two tissue sections have less than 5,000 CD30⁺ cells each. In NScHL, we observed a clustering of cells, meaning that there are spots with a high CD30⁺ cell density, while other areas in the tissue section are not populated by malignant cells. This is the case for all four depicted tissue samples. Images three and four, both tissue sections with a low cell count, show only a small number of densely packed cell clusters.

In MCcHL, the cell density distribution is more diverse. The first and the last image, which depict tissue sections with the highest and with the lowest cell count, show a density pattern similar to the NScHL cases. The CD30⁺ cells cluster in some regions, and spots with a high density exist. Those spots of high cell density are missing in images two and three. Here, the cells are distributed more evenly throughout the tissue section. In image two, the cells are spread through the whole lymph node section, and only few areas exist, where no CD30⁺ cells are present.

In the NScHL image set, ten out of twelve tissue sections contain cell clusters with a high cell density, fully or partly separated by regions without malignant cells. From

the MCcHL image set, only three tissue sections were identified to have a similar cell density distribution. The other nine cases had either no regions with a high cell density, or the cells were distributed evenly, lacking the separating areas without CD30⁺ cells, or both. Two MCcHL cases had a very high overall cell density, but in contrast to NScHL cases with similarly high densities, the malignant cells were spread through the whole lymph node section. The differences of locale cell densities were relatively low. One possible explanation are the sclerotic bands, present in all NScHL cases, but not necessarily present in MCcHL. To directly show the correlation between the cell density of CD30⁺ cells in the NScHL lymph node sections, the clustering of the cells and the presence of sclerotic bands, an additional imaging pipeline would be required to identify sclerotic regions. This could be done using a texture based method to separate regions with a low cell nucleus density, which are likely sclerotic, and other regions representing tissue.

Two figures, supporting the differences in the cell density distribution, are shown in the supplement. Figure S2 depicts the fraction of high cell density image tiles with neighboring high cell density image tiles for each image of the cHL image set. On average, this fraction is higher for NScHL cases, but half of the images of the MCcHL image set also have high values of about 80 % and more.

The MCcHL set contains six images with a low number of adjacent high cell density image tiles. I.e., the high cell density image tiles are isolated, meaning that no, or at least only small cell clusters exist. Figure S3 lists the average occurrence of one up to eight adjacent high cell density image tiles for the two cHL sub-types NScHL and MCcHL. In MCcHL, the 8-neighborhood mostly consisted of a single high cell density image tile. The occurrence of multiple high cell density tiles is increased in NScHL compared to MCcHL. In NScHL, we observe larger dense cell clusters compared to MCcHL.

CD30⁺ Cell Morphology The CD30 pipeline in Impro runs the MeasureObjectSize-Shape module in CellProfiler for the labeled cell objects. The module calculates common shape descriptors for the detected cell objects. Typical cell shape descriptors that are important for this work are listed in Table 3.4. As the descriptors are calculated from

2D image data, we will refer to cell morphology as the shape of the cell profile given by the cutting plane of the tissue section.

The plot in Figure 4.23 describes the relative amount of different cell area sizes for the three image sets lymphadenitis, MCcHL, and NScHL. The area size is plotted in number of pixels of the cell objects. 1,600 px correspond to $100 \mu\text{m}^2$. For comparison, a perfectly circular cell profile with a diameter of $10 \mu\text{m}$ has an area of $78.54 \mu\text{m}^2$. According to the literature Reed-Sternberg cells reach a diameter of $30 \mu\text{m}$ and more. A circular cell object with such a diameter has an area of $706.86 \mu\text{m}^2$, or 11,309 px, respectively. The two cHL images sets have a similar cell area distribution. All three curves have their maximum at the range of 1,000-2,000 px. Compared to the 46 % in lymphadenitis, the relative amount of cells in this range is low in cHL. A big fraction of cells lies in the range of 4,000-15,000 px. Those rather large cells make up 40 % of the cells in MCcHL and 44 % in NScHL. In lymphadenitis, cells with an area above 4,000 px are uncommon, here we see 15 % of the cells in the same range. The observed cell area sizes comply with literature data, for a detailed study of the cell morphology see [83]. The fraction of large cells in cHL and the higher variability of the area sizes in both cHL images sets originate from the malignant CD30^+ HRS cells. Hodgkin cells are mononucleated and are slightly enlarged with a diameter of 20-30 μm . Reed-Sternberg cells are multinucleated and with a diameter of up to $100 \mu\text{m}$ they are enormously enlarged. We still observed the same maximum in lymphadenitis and cHL cases for the cell area size. One suggestion would be, that the control mechanisms for the cell size is still partly intact in HRS cells, leading to a typical cell size. The enlarged multinucleated cells emerge from incomplete cytokinesis events, resulting in re-fusion of the daughter cells [84]. Another possible explanation is that active lymphocytes which are the CD30^+ cells in lymphadenitis are also present in cHL tissue sections. From the CD30 stained image data alone it is not possible to conclude which mechanism leads to the observed cell area size distribution.

2 % of the cells had a cell area size of $>20,000$ px. While this is possible for HRS cells, such a cell area size is untypically high for the activated lymphocytes in lymphadenitis. The area size suggests, that those cell profiles are artifacts caused by wrong detection by the CD30^+ pipeline. Here, either the separation of multiple cell objects failed or

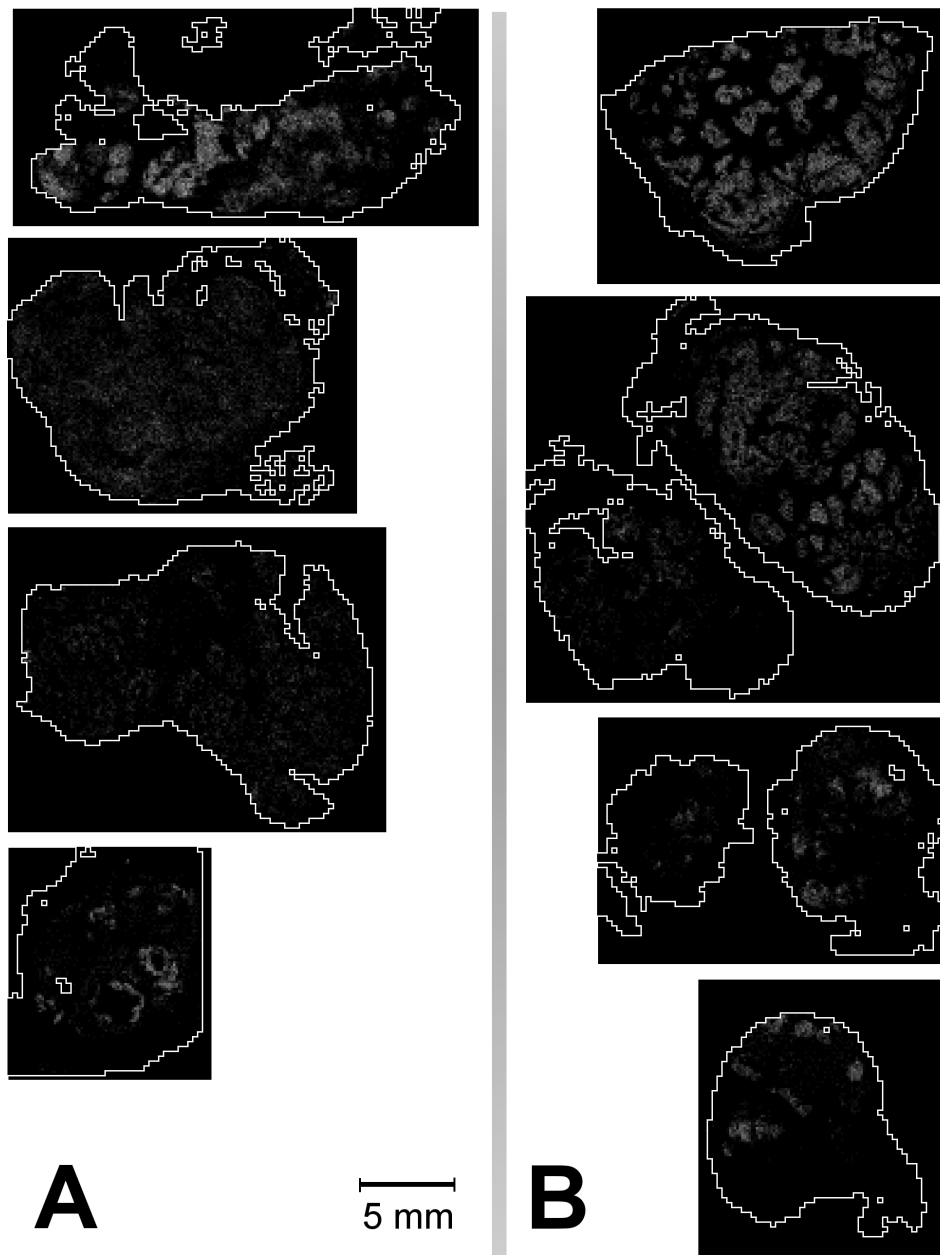


Figure 4.21: Cell density in cHL Typical cell distributions in the two cHL subtypes, MCcHL **A** and NScHL **B**. The depicted cases are sorted by the total cell count. Images in the top row had the highest cell count ($> 15,000$ cells), images in the bottom had the lowest cell count ($< 4,000$ cells). The white line marks the region of interest, i.e., the region of the image that is interpreted as tissue in the CD30 pipeline. In NScHL, the cells cluster in all images. Even in images with low cell count, we observed spots with high cell density. In contrast, MCcHL had different patterns of cell density. The top image and the bottom image show a cell density distribution similar to the NScHL cases. Cells cluster in certain regions with a high cell density. The other two cases lacked regions with densely packed cells. Instead, the CD30 cells were distributed more evenly throughout the lymph node section.

unspecific staining was falsely classified as a cell object.

Figure 4.24 depicts the distribution of the maximum Feret diameter averaged for the three image sets. The distributions are similar to the cell area size distributions. Most of the cells in cHL cases have a maximum Feret diameter of 80 px, or 20 μm , respectively. In lymphadenitis the maximum is found at 60 px. In general, the maximum Feret diameter is shifted to higher values in cHL. In lymphadenitis 50 % of the cells have a maximum Feret diameter above 80 px. This fraction is rather low compared to the 82 % in cHL.

The fact that for cHL cells the maximum in the Feret diameter distribution is shifted compared to lymphadenitis, which is not the case for the area size, suggests that HRS cells have a less even and less round shape than the activated lymphocytes in lymphadenitis. To get additional information about the general shape of the cell profiles, two typical morphology descriptors were included. The eccentricity is based on an oval that is fitted to the cell outline. An eccentricity of 1.0 means that the object is narrow and elongated, an eccentricity of 0.0 means that the cell is not stretched in any particular direction. The solidity describes the ratio between the area of the object and the convex hull of the object. If the outline of the cell profile is convexly shaped the solidity is 1. Concave cell profiles result in low values close to 0. For both morphology descriptors, the distributions of the two cHL image sets were closer to each other than to the lymphadenitis image set. Figure 4.25 depicts the relative count of cells with a specific eccentricity. Most cells have an eccentricity of 0.65 to 0.9. As the eccentricity value is relatively abstract, Table 4.1.3 lists some ratios between the major axis and the minor axis of an oval with a specific eccentricity. The examples in Figure 4.22 illustrate the shape of cells with a given eccentricity. It is noteworthy that the ratio between the major axis of the object and its minor axis is not linearly correlated to the eccentricity. Eccentricities within the range of zero to 0.50 are found for objects that appear round and their major axis is no more than 15 % elongated compared to the minor axis. Cells whose dimensions are twofold increased towards the major axis have an eccentricity of > 0.85 .

Eccentricity	major : minor
0.20	1.02
0.35	1.07
0.50	1.15
0.65	1.32
0.80	1.66
0.95	3.20

Table 4.5: The ratio of the major axis and the minor axis depending on the eccentricity

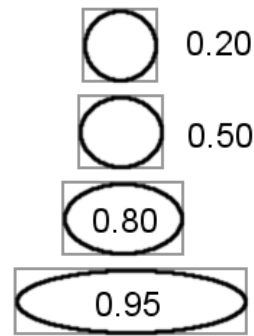


Figure 4.22: Ovals with specified eccentricities. Samples of ovals with different eccentricities.

Compared to the cHL eccentricity distributions, the eccentricity in lymphadenitis is shifted to lower values. Elongated cells are less frequent, while the occurrence of cells with an eccentricity below 0.6 is increased. The cells tend to be slightly stretched in one direction, but overall the difference between the major axis and the minor axis is not as high as in cHL.

The solidity, see Figure 4.26, is higher in lymphadenitis. The distribution has one narrow peak, with a maximum at 0.85. In MCcHL and NScHL the maxima can be found at 0.8 and 0.75. Both distributions are also broader and solidity values between 0.45 and 0.75 are much more frequent in cHL than in lymphadenitis. CD30⁺ cells are solid and convex in lymphadenitis. The lower solidity in cHL indicates that cells are shaped less regularly, meaning that the cell profiles have more recesses or outgrowths.

From the energetic point of view, irregular or stretched cell outlines are inefficient. The cell consumes energy to organize the cytoskeleton, which enables the polarization of the cell. The cells shape is highly influenced by cell adhesion and cell movement. While those factors are no exclusive reasons for the cell morphology, they are possible cellular functions for which stretched, irregular cell shapes might be indicators.

The correlation between cell movement and 2D cell morphology descriptors needs further verification by time-lapse data. One possible way would be to relate the displacement of cells with their average eccentricity, to check whether the eccentricity indicates

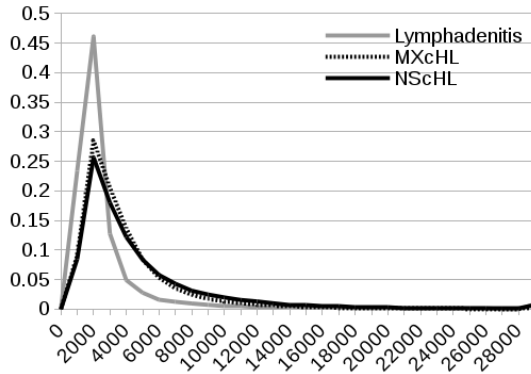


Figure 4.23: Morphology descriptor area The area computed by CellProfiler in relative numbers.

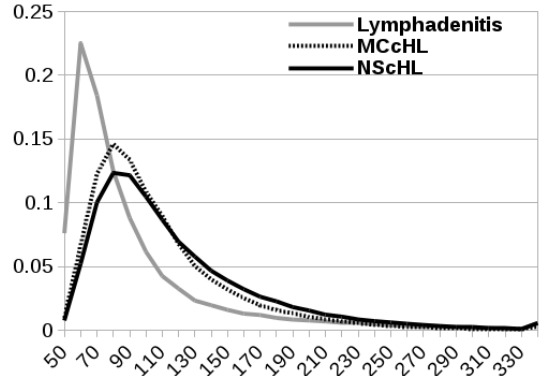


Figure 4.24: Morphology descriptor maximum Feret diameter The maximum Feret diameter computed by CellProfiler in relative numbers.

cell movement and possibly the movement direction of the cell.

3D image data might be used to validate the effects of the sectioning and the use of 2D image data. The cell profiles currently used in the study are 2D cross sections of the original cell and the loss of information in Z-axis needs to be kept in mind. Cell profiles with a low eccentricity value in our study might be stretched in Z direction. A low solidity of the cells could indicate cell adhesion and possibly cell-cell contacts. To further verify this relation to cell function, high resolution 3D images can be used. In addition, markers for cell adhesion and for the formation of ion channels, which play an important role in cell-cell communication, are required.

If the cell profiles of standard immuno histological 2D images are representative for the dissected lymph node, the morphology distribution of CD30⁺ cell profiles can be integrated in the standard diagnosis of cHL and can be used as an additional source of information about the progression of the disease.

Neighborhood Analysis

In the last sections, cells were analyzed according to their spatial global positions and the cell morphology. We also investigated the relationship between the relative cell position and cell shape. Looking at the local neighborhood of a single cell, are there any preferences regarding the cell morphology? To simplify the comparison of cells, we

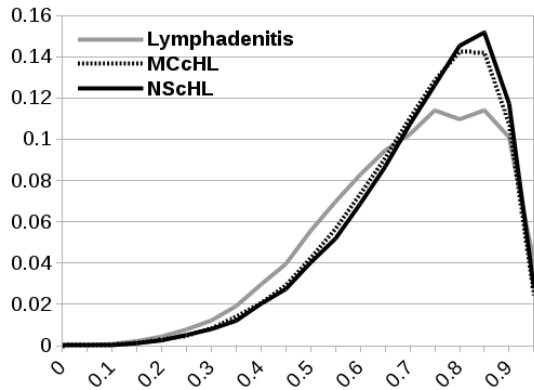


Figure 4.25: Morphology descriptor eccentricity The eccentricity computed by CellProfiler in relative numbers.

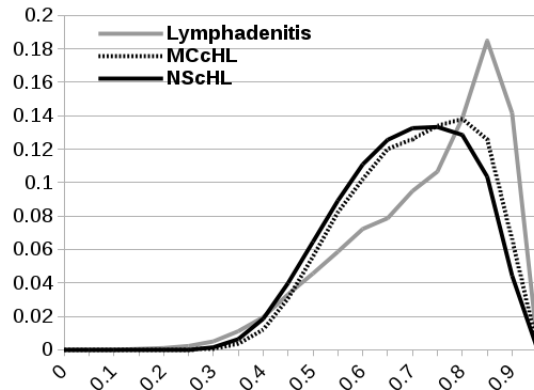










Figure 4.26: Morphology descriptor solidity The solidity computed by CellProfiler in relative numbers.

assigned cell classes, depending on morphology descriptors. Three values were taken into account: area, solidity, and eccentricity. A certain threshold was determined for all three descriptors. The thresholds were set whilst taking into account the overall distribution of each morphology descriptor and were refined after consulting pathologists of the Dr. Senckenberg Institute of Pathology. Cells were divided into *small* or *large* cells. *Large* cells have an area of $\geq 500\mu\text{m}^2$ ($\sim 8,000\text{ px}$). Cells with a high eccentricity are elongated in one direction. All cells with an eccentricity above the threshold of 0.85 are labeled as *elongated*. The solidity descriptor was used to determine if cells are *frayed* by setting the threshold < 0.75 . Cells that are neither classified as *elongated*, nor as *frayed*, are labeled as *round* cells. Table 4.6 gives an overview of the eight cell classes. Comparing the manually set thresholds for the three classes with the descriptor distributions visible in Figures 4.23 - 4.26, it is obvious that the cell classes are not equally frequent in the image set. Only a small fraction of cells are classified as *large*. Figure 4.27 depicts the frequencies of the cell classes for the whole image set (**A**) and separately for the different diagnoses (**B-D**). To be independent of the total number of cells in each image, the image sets were averaged by calculating the cell class distribution for each image separately and averaging afterwards. Otherwise, images with a high cell count would have been weighted much higher than images with a low cell count.

Table 4.6: The morphological cell profile classes defined by the three descriptors area, eccentricity, and solidity. The attributes *Large*, *Elongated*, and *Frayed* are computed by setting a fixed threshold for each morphology descriptor.

Symbol	Index	Name	Large	Elongated	Frayed
	0	small_round	-	-	-
	1	large_round	+	-	-
	2	small_elongated	-	+	-
	3	large_elongated	+	+	-
	4	small_frayed	-	-	+
	5	large_frayed	+	-	+
	6	small_elongated_frayed	-	+	+
	7	large_elongated_frayed	+	+	+

Small cells are much more frequent than *large* cells. Most cells were classified as *small_round* cells. Averaged over all images, 39 % of the cell profiles belong to this cell class. Although *small_round* is the most frequent class in all three diagnose image sets, it is most predominant in lymphadenitis. Large morphological cell classes (1, 3, 5, 7) are rare in lymphadenitis and make up 8.1 % of all cells. In cHL, the percentage of large cells is increased to 11.06 % in MCcHL and 17.36 % in NScHL. The relative amount of frayed cells is slightly increased in cHL. It is on average 20.15 % higher than in lymphadenitis.

To investigate the relation of morphological cell features and the geometrical closeness between cells, we investigated the occurrence of morphological cell classes in the local neighborhood. Here, we defined the local neighborhood as nearest neighbor pairs. For each image we computed the fractions of the eight morphological cell classes. We also

determined the conditional property, $P(\text{NCP} = j \mid \text{CP} = i)$, given that a cell CP has the cell class i the probability to have a nearest neighbor NCP with cell class j .

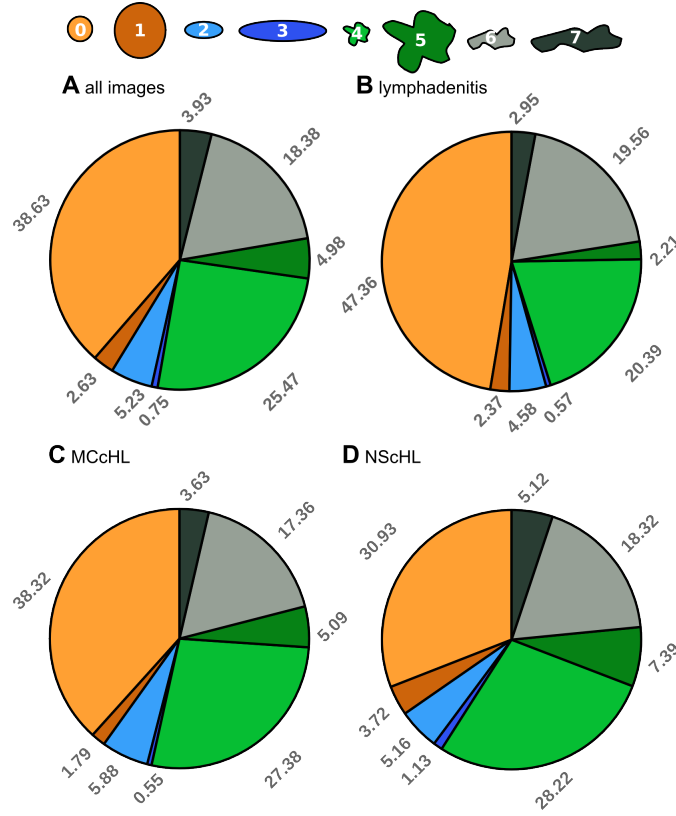


Figure 4.27: Fractions cell classes Distribution of cell classes averaged for all images and for the three diagnosis image sets.

To verify a correlation of the cell classes between nearest neighbor cells, we compared the observed occurrences to reject the null hypothesis that the cell classes are random and thus are uncorrelated. If the morphological cell classes of a pair of nearest neighbor cells are random, the probability to get k nearest neighbor pairs of two certain morphology classes can be modeled as a binomial distribution. From this we can compute the prediction interval $[k_{low}, k_{up}]$:

$$\max k_{low} \in \mathbb{N} : \sum_{k=0}^{k_{low}} p^k (1-p)^{n-k} \binom{n}{k} \leq \alpha/2 \quad (4.2)$$

and

$$\min k_{up} \in \mathbb{N} : \sum_{k=k_{up}}^n p^k (1-p)^{n-k} \binom{n}{k} \leq \alpha/2. \quad (4.3)$$

If the number of counted nearest neighbor pairs is measured to be above or below the prediction interval, the null hypothesis can be rejected within a error tolerance of α . In this study, α was set to 1 %. Morphological pairs counted above the prediction interval are with high certainty over-represented in the data set under the assumption of a random distribution of the morphological classes. Morphological pair counts below the prediction interval are underrepresented. To compare the image sets, we counted for each morphological class pair the number of cases showing an over- (NR_{high}) or under-representation (NR_{low}). The results are depicted in Figures 4.28 to 4.31. Preferences of the neighborhood relations are symbolized as arrows, when the relation score, see Equation 4.4, is ≥ 50 % of the maximal possible score.

$$relation\ score = |NR(CP, NCP)_{high} - NR(CP, NCP)_{low}| \quad (4.4)$$

Grey dotted arrows represent favored relations of two morphology cell classes, black solid arrows symbolize neighborhood relations that are avoided. The thickness of the line correlates with the relation score. Thin lines depict a relation that equals 50 % of the maximal relation score, thicker lines symbolize a relation that is present in more than 50 % of the cases. Figure 4.28 depicts the relations found in all 35 images independent of the diagnosis. While the morphology classes 1 to 4 do not show significant preferences regarding their nearest neighbor, morphology classes 0, 5, 6, and 7 are dependent on the neighboring morphology class. *Small_round*, *small_elongated_frayed*, and *large_frayed* cells have an increased chance to be neighbored by cell profiles having the same morphology class. The relation score for two neighboring cells of the class *small_round* is found to be the highest and it reaches 83 % to 92 % in all image sets. The diagram in Figure 4.28 also highlights that frayed morphology classes avoid nearest neighbor cells with the attributes small and round.

Figure 4.29 summarizes the neighborhood relations of the lymphadenitis image set. Compared to the two cHL image sets *small_round* cells are more isolated. The direct

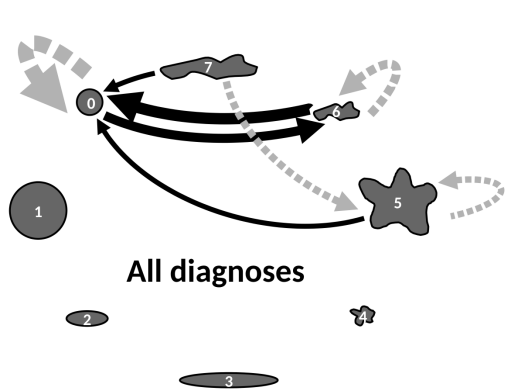


Figure 4.28: All diagnoses The neighborhood relations of the whole image set. The three morphology classes 0, 5, and 6 favor nearest neighbors of the same morphology class. The two classes *small_round* and *small_frayed* avoid the neighborhood of each other.

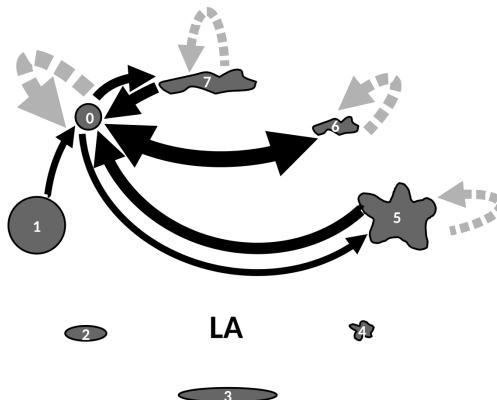


Figure 4.29: Lymphadenitis The neighbor relations in the lymphadenitis image set. The self-preferences of the classes 0, 5, and 6 are also present in most of the lymphadenitis images. In addition, all frayed classes, except the *small_frayed* cells, avoid the neighborhood of *small_round* cells.

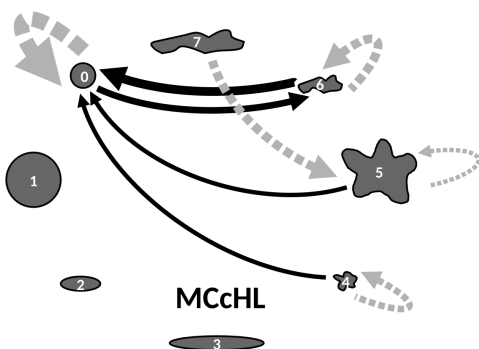


Figure 4.30: MCcHL diagnosis In MCcHL, the relation between *small_round* and other classes is less strong. The morphology classes 4, 5, and 6 avoid the direct neighborhood of *small_round* cells, but the percentage of cases is less.

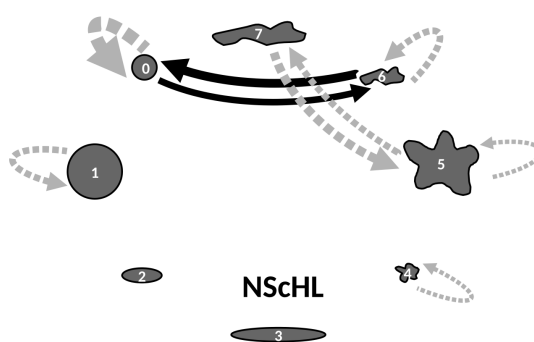


Figure 4.31: NScHL diagnosis In NScHL, only the two morphology classes *small_round* and *small_elongated_frayed* avoid each other as a direct neighbor. In contrast to lymphadenitis, the cHL image sets show an attraction between the two morphology classes *large_elongated* and *large_frayed*.

neighborhood of *small_round* cells is avoided by the three morphology classes 5, 6, and 7. The preference of *large_elongated_frayed* to be in the neighborhood of *large_frayed* cells is a relation that only can be found in cHL.

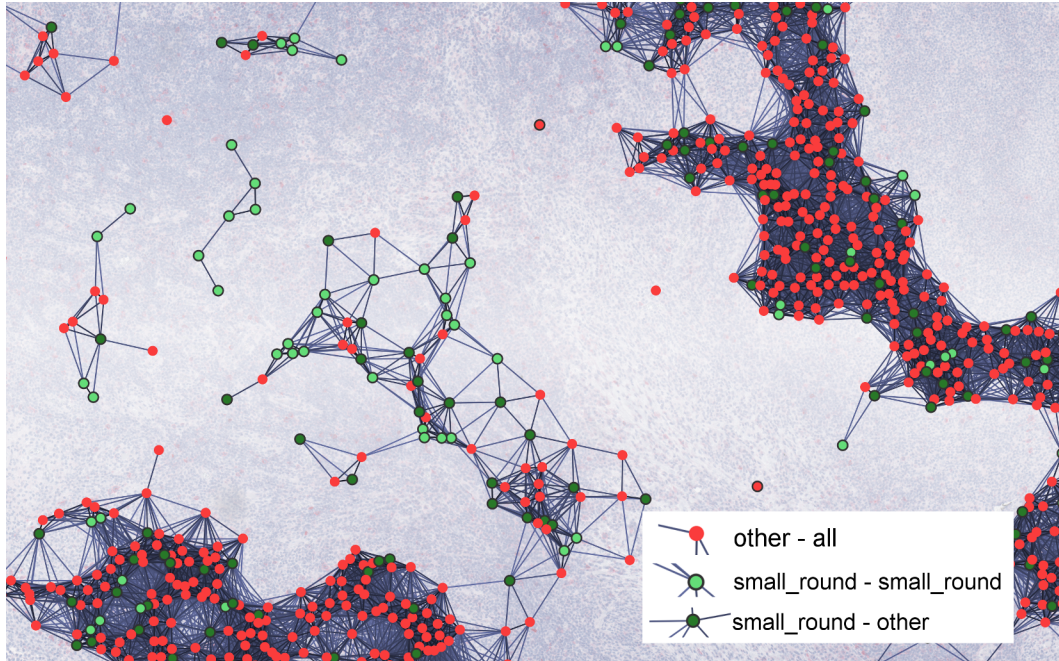


Figure 4.32: Nearest neighbor pairs *small_round* The distribution of the morphology class *small_round* in a sample section of a MCcHL case. Red circles are all cells with a morphology class other than *small_round*. Green circles mark cells with the attribute small and round. *Small_round* cells with a nearest neighbor of the morphology class *small_round* are highlighted in a brighter green.

The neighborhood relation of the eight morphology classes can be visualized and directly mapped on the tissue section. Figure 4.32 depicts the cell distribution in a MCcHL tissue section. The colors separate *small_round* cells, green, from other morphology classes, shown in red. Surprisingly, although *small_round* cells showed a preference to neighbor cells with the same morphology class, we found a higher average distance between *small_round* - *small_round* nearest neighbor pairs than the overall average distance between nearest neighbor cells. The higher distances of the *small_round* cells contradict the hypothesis that the nearest neighbor preferences between *small_round* and other *small_round* cells originate from an attraction of this morphology class. Instead, we found the relative amount of *small_round* cells increased in areas with a lower CD30⁺ cell density. Figure 4.32 depicts a tissue section supporting this observation. In

high density regions of CD30⁺ cells the relative amount of *small_round* cells is low (green circles) and only few of them are neighbored by another *small_round* cell (bright green circles). In the center of the image and the upper left part, where the CD30⁺ cell density is lower, we can observe more *small_round* - *small_round* nearest neighbor pairs.

The nearest neighbor analysis is based on the master thesis of JS.

Statistics

Impro has additional plugins that are focused on the statistical analysis and on visualization of the results. The two plugins *Quant* and *Heatmap* are both implemented by AS. Both plugins are accessible by GUI only. The *Quant* plugin performs a pixel quantification on the low resolution layer of the WSIs. Pixels are classified by a minimum distance to mean classifier. The training images and the background pixel class can be set by the user. The plugin provides a statistical overview of the relative amount of each pixel class. The method and results of the application to cHL tissue sections were published in [36]. The Heatmap plugin can plot the results of the Quant plugin directly on the tissue section. The results are plotted as heatmaps to visualize the relative amount of pixels of a certain pixel class. Both plugins, the Quant and Heatmap plugin, are no longer maintained and do not work in recent snapshots of Impro.

The CellClassMap plugin is similar to the Heatmap plugin, but it processes the results of the cell detection from the CellProfilerAdapter plugin. It creates and stores heatmaps for the complete ROI of all selected images. Heatmaps for the *cell count* are computed for all cells of an image and for multiple morphological subsets. A heatmap for each cell class and each meta class is created. For the computation of the heatmaps, first, a grid is created. The default size of each grid tile is 512 x 512 pixels. All cells within a distance of 700 pixels measured from the center of the grid tile are counted. The computed heatmaps can be visualized as an overlay in the ImageViewer plugin. Two examples of such overlays are shown in Figure 4.33, the *average node degree* heatmap is created by the *Graph* plugin.

The *Statistics* plugins create a statistical overview for each image. The plugin can be accessed via GUI and with an Impro command. First, general properties of the image

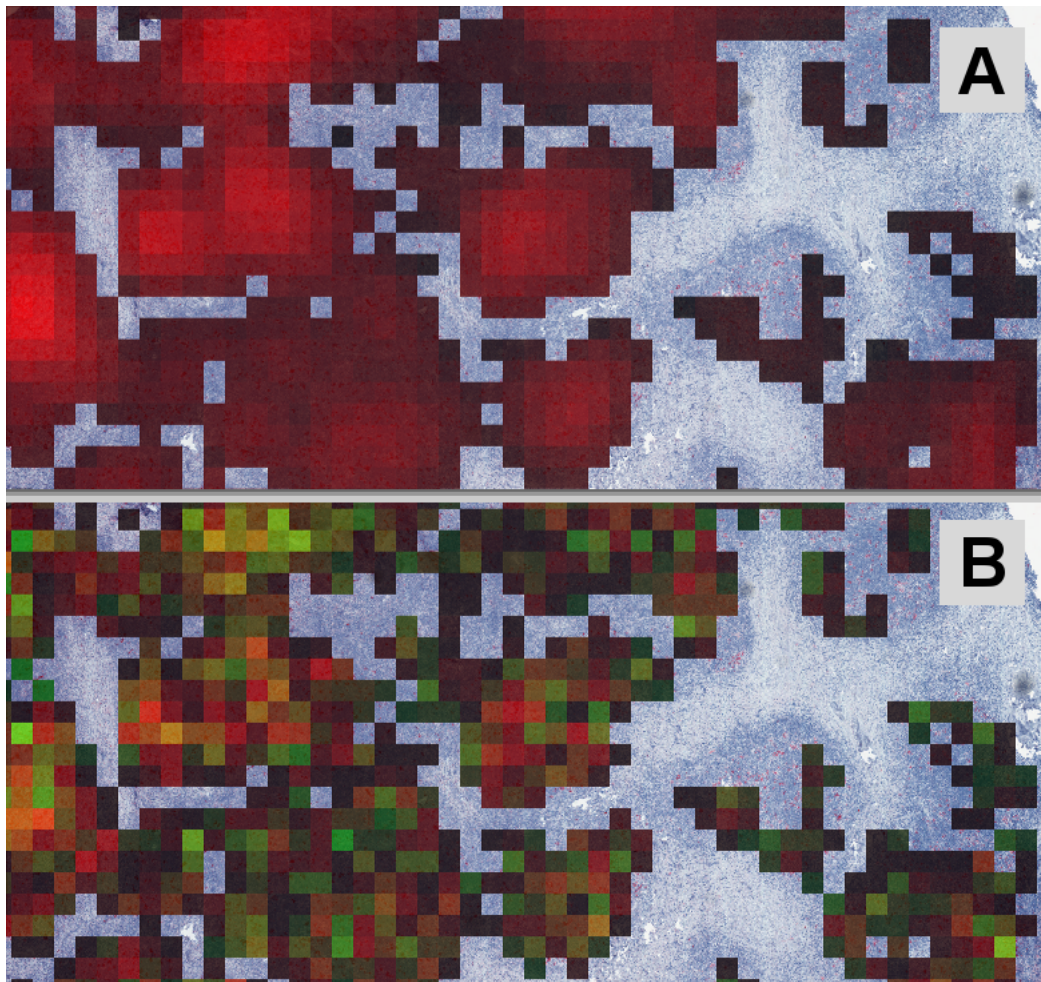


Figure 4.33: Cell class maps of the NScHL image 6286 The ImproViewer plugin displays heatmaps, e.g. created by the CellClassMap plugin, as overlay on top of the original tissue section. Here, two examples are visible. In A, the displayed property is the *average node degree*. Dark and unsaturated values represent a low node degree, high node degrees are displayed bright and saturated. Areas without a valid value, i.e. areas where no cells are found by the pipeline, appear transparent in the heatmap overlay. Sub-image B plots the relative *cell count* for the two meta classes *round cells* (red) and *cut cells* (green). Note, that the mixture of red and green values in RGB color space results in yellow, thus yellow values can be interpreted as areas, where the relative cell count of both meta cell classes is similar.

and its CellGraph are created. Table 4.7 gives a short overview of the statistics and a description of each property.

Table 4.7: Properties of CD30⁺ WSIs calculated by the *Statistics* plugin in Impro

Property	Description
Image_EdgeNumAll	Edge count in cell graph.
Image_IsolatedVerticeNumAll	Number of vertices without neighbors.
Image_VerticeNumAll	Vertex count in cell graph.
Image_TissuePercent	Percentage of tiles that belong to the ROI.
Image_Diagnosis	Integer value representing the disease type of the image.
ClusterCoeff_mean	Average cluster coefficient in the cell graph.
ClusterCoeff_variance	Variance in the distribution of cluster coefficients.

4.1.4 Validation

The *PipelineValidation* plugin can be accessed via the menu, see Plugins → View → Pipeline Validation. The plugin provides an image viewer similar to the ImproImage-Viewer plugin. The user can manually annotate images and mark all cell positions by hand. A subset of image tiles is selected randomly as a sample. The image size and the number of contained objects is too large to be annotated manually in a reasonable time. The number of sampled tiles can be set in the configuration file of Impro. For our work, values between 5-10 % suited best. This sample size is big enough to be representative for the image, but still small enough to be annotated manually. To ensure that only image tiles with relevant information are selected, the sample set is generated as a subset of the ROI image tiles. Thus, all tiles used for the validation contain tissue.

The user can manually annotate cells in the randomly sampled image tiles. In addition to single cell objects, which are represented only by their position, the plugin allows the annotation of multi-cell objects. This allows a less strict annotation in densely packed regions, where the separation of cells is not possible even by human eye. A multi-cell object is defined by its area and a number range that specifies the upper and lower bound of expected cell objects in the area. An example of a manual annotation is given in Figure 4.34 A.

For the validation, we compare the result of the automated pipeline with the manually

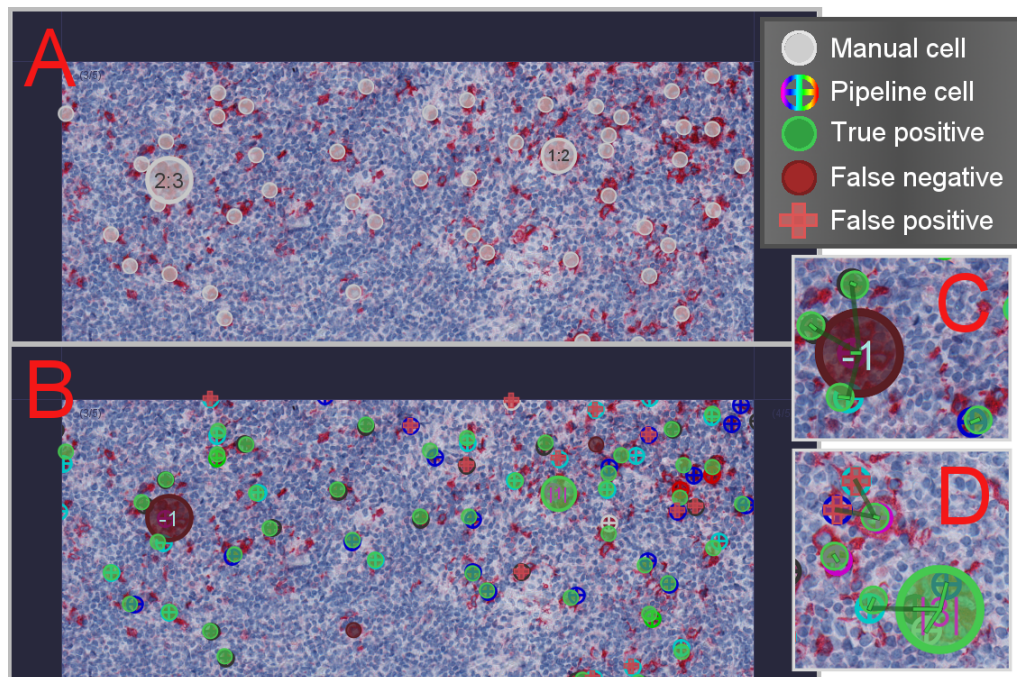


Figure 4.34: Pipeline validation and manual annotation **A:** A part of a manually annotated image tile is depicted. The surrounding tiles are not sampled for the validation and thus, are blended out (dark blue area). White circles represent manual cell objects. Multi-cell objects are drawn with their corresponding size and the range of the possible cell count, within the circle. **B:** The result of the validation. Manual cells and pipeline cells are depicted in the overlay. All manual cells mapped to a pipeline cell are colored green, they are true positives. Manual annotated cells, which were not detected by the pipeline are colored dark red. They mark false negatives. If a cell is detected by the pipeline, but is not manually annotated, the cell is considered to be false positive. Those cells are marked with red cross-like symbols. **C and D:** Both images show a small validated region with higher resolution. To show the mapping of manually annotated and pipeline cells, all possible cell pairs are connected by green lines. The actual mapping is highlighted by bright green lines. Manual multi-cell objects can be mapped to multiple pipeline cell objects.

annotated cells. For comparing both sets of cell objects, the first step is to map the results of the automated pipeline to the corresponding manual cell objects. The mapping is needed because cell objects are saved by their position, that is the center of the cell object. As the outline of the detected cell object might differ and as the cells center can not be precisely annotated manually, it is necessary to allow a margin between a pair of cell objects. Therefore, for each cell object found by the pipeline, the neighboring manually annotated cells within a specified threshold distance are determined. The paired cell objects are added to a list and sorted by their Euclidean distance. The cells are then mapped, beginning with the pair with lowest distance. All other pairs

containing at least one of the two mapped cells, are removed from the list. The mapping is iteratively continued until the list of paired cell objects is empty. For multi-cell objects, the distance is not calculated as euclidean distance. Here, the distance is set to the radius of the multi-cell object. Thus, single cell objects are always preferably mapped compared to neighboring multi-cell objects. The special treatment of multi-cell objects is also necessary because no cell center positions are determined for the contained cells. The distance to the center of the multi-cell object does not make sense if multiple cells are present. Possible mapping pairs of a multi-cell are also not rejected from the pair list until the upper bound of mapped pairs are reached.

Image 4.34 C and D show the mapping process. Manual cells as well as pipeline cells are depicted. All potential cell pairs are visualized by green lines. The resulting mapping is highlighted with bright green lines. Both images, C and D, contain one multi-cell object. In C, it is mapped to only one pipeline cell. The multi-cell minimum cell count property is set to two, meaning that at least two cells are present in the defined area. Here, the mapped cell is treated as true positive, but the missing second cell counts in as false negative. In image D, the multi-cell object is mapped to three pipeline cells. The cell count is within the manually cell count range and all three cells are treated as true positives. Image D also contains two false positive cells. The pipeline cells marked with the red crosses could not be mapped to any of the manual cells. That means the automated cell pipeline detected a cell object, which does not exist according the manual annotation.

After mapping all pipeline cell objects to manual annotated cells, the automated detection results can be classified as true positives, false positives and false negatives. True positives are all cell objects found by the automated pipeline and the manual annotation. All cells detected by the automated pipeline that can not be matched to a manual cell objects are false positives. Unmatched manual cells are false negative cells in the pipeline. For multi- cell objects we use the lower and the upper bound to calculate the number of TPs, FPs and FNs. If the number of matching cells within the specified area is below the minimum cell count, all missing pairs are counted as false negatives. The existing pairs are true positives.

To get a scoring for the imaging pipeline result for single images, we calculate the precision and the sensitivity based on the TPs, FPs, and FNs (see Section 3.2.8). The score is calculated for the whole image and for each single sampled tile. The latter is useful to determine in which regions of the image the object detection was successful and beside this we also can find regions with results of lower quality. This helps to find the limits of the imaging pipeline and may reveal potential refinements for the image processing.

The CD30 cell count in cHL tissue WSIs range from several hundred up to ten thousand in most of the cases. If 10 % of the image is sampled for the validation, this means about 100 up to 1000 cells need to be manually annotated. Depending on the cell density and the size of the tissue section, the validation of a single image takes about 15 – 60 minutes.

Figure 4.35 depicts the FPs, TPs, and FNs for multiple sampled tiles for a WSI. The overall fraction of TPs is 73.4 % meaning a high number of correctly detected cells. The quality of the detection pipeline differs among the selected tiles. In the tiles 8:6, 9:4, and 9:5, the cell detection pipeline struggled most and about 70 % of the cells were not detected. The image tiles were manually inspected. The staining of the three tiles had a much lower intensity compared to the other sampled image tiles. Here, a locally lowered threshold might have increased the number of detected cells.

Table 4.8 summarizes the validation output of three WSIs. In total, 2,670 cells were manually annotated. For all validated images, the cell detection pipeline achieved a precision of 84 % and a sensitivity of 96 %. For all three diagnoses, the precision exceeded 80 %. The sensitivity were also high for each diagnosis, in MCcHL and LA less than 2 % of cells in the image were not detected.

4.2 Graph-Based Analysis

4.2.1 CD30⁺ Cell Graphs

Section *CD30⁺ Objects* discussed results of the detected CD30⁺ cell objects and their spatial distribution in the tissue sections. In Section *Neighborhood Analysis*, nearest neighbor



Figure 4.35: Pipeline validation for a single WSI The TPs, FPs, and FNs for a single WSI are plotted. The results are depicted for single tiles and the total result for all tiles. Overall, most cell objects were detected by the pipeline and the amount of FP cell objects is low. The precision and sensitivity are higher than 80 %. But for single tiles the quality of the detection drops down. The three image tiles 8:6, 9:4, and 9:5 have a high amount of FNs. For the three tiles we also observed a low quality of staining. The low intensity of the fuchsin stain is a likely explanation for the high FN-rate.

Table 4.8: Validation results of cell detection.

Diagnosis	Cell #			Precision	Sensitivity
	TP	FP	FN		
MCcHL	952	189	12	0.83	0.99
NScHL	686	155	81	0.82	0.89
LA	921	137	18	0.87	0.98
all	2,559	481	111	0.84	0.96

pairs and their morphological properties were analyzed. Here we relax the neighborhood definition, using unit disk graphs to model the local neighborhood of $CD30^+$ cells.

The object detection, described in Section *CD30⁺ Objects* results in a set of cell positions and for each detected object an additional list of measured properties. The list currently includes all descriptors provided by the MeasureObjectSizeShape CellProfiler plugin. Additional properties can be added via the MeasureObjectNeighbors and the MeasureObjectIntensity plugins. For a full list see Section *Measurement* of the CellPro-

filer documentation[†]. For further analysis we use graph-based methods. The output data of the CellProfilerAdapter plugin can be accessed from a MySQL database and is used by the Graph plugin to create cell graphs, the major parts of the plugin are implemented by TS.

Cell Graphs - Theoretical Background

Digital light microscopic images are a good source to gain detailed information about biological processes on cellular level. Nowadays, accurate staining of specific proteins can visualize cells, their compartments and further details like the expression of certain proteins. However, depending on the information one wants to access, images can be a rather inefficient representation. The data is stored pixel wise, by intensity values for one or multiple color channels. Therefore, abstract objects, e.g. cells and cell nuclei, need to be identified by segmentation and labeling. Another drawback regarding WSIs is that among the object pixels containing the required information, also background pixels and many unnecessarily detailed information are stored, leading to large files. Sizes of one or multiple Giga-byte are common, even with high compression rates, e.g. provided by JPEG2000 compression. In our work we focus on CD30⁺ cells and their distribution. To model the spatial distribution we introduce *cell graphs*. The following paragraphs give a theoretical definition of cell graphs and explain the construction from imaging data.

Definition 2 gives a formal description of a cell graph. The graphs vertices refer to cells and their position in space. Depending on the image data, a position may be a 2D or 3D coordinate, but in this thesis all cells objects are gathered from 2D histologically stained images.

Definition 2. *A cell graph is a graph $G = \{V, E\}$. Each vertex $v \in V$ represents a cell object having a position in space. Edges are defined according the containment model of unit disk graphs. Two vertices u, v are connected by an edge only if the geometric distance of v is below a given threshold t .*

The resulting cell graphs have some defined properties and the geometric construction of the graphs has a high impact. Some typical graph properties can be expressed depen-

[†]<http://www.cellprofiler.org/CPmanual/>

dent on the local geometric distribution of cells. For example, the cell density influences most local graph properties. The vertex degree of a single cell can be expressed as the cell density within the unit disk of the specific cell. A local cell density of 6 in the unit disk of one specific vertex x results in the $\text{vertex_degree}(x) = 6 - 1 = 5$. The vertex x has edges to all vertices contained in its unit disk, except x itself.

In our work we focus on CD30⁺ cells and the resulting cell graphs are restricted to cells expressing CD30. Thus, we call such a graph CD30⁺ cell graph. The constructed graphs model the spatial distribution of reactive lymphocytes and/or HRS cells. In lymphadenitis the population of CD30⁺ cells consists of reactive lymphocytes. In cHL cases the tissue mainly consists of HRS cells, but the presence of reactive lymphocytes is not excluded.

The distance threshold t was set to 700 px (175 μm). This is roughly the distance of ten cell diameters and the value was chosen to define the local neighborhood. The distance is low enough that neighboring cells are within a distance that makes cell-cell communication possible. We need to keep in mind that long distance communication, e.g. by some chemokines, are not taken into account. They are not represented as edges in the CD30⁺ cell graphs. An exemplary tissue section and the corresponding CD30⁺ cell graph is depicted in Figure 4.36.

Graph Plugin

Here, the usage of the Graph plugin, including both, the graphical user interface and the headless, text command based mode will be explained.

Figure 4.37 illustrates the graphical user interface of the graph plugin. Graphs can be created from the raw cell positions, identified by the CellProfiler pipeline (see Section *Object Detection*) and can be loaded from earlier stored graph files. Cell graphs can be exported and imported in multiple file formats. The graph plugin supports CSV, TGF, GML, and KAVOSH (for a detailed description of the file formats see Section 3.3.1).

For the initial cell graph generation from the MySQL database the user can specify a maximum distance for the edge calculation. The maximum distance is equivalent to the radius t specifying the size of the unit disks for the containment model. For a more

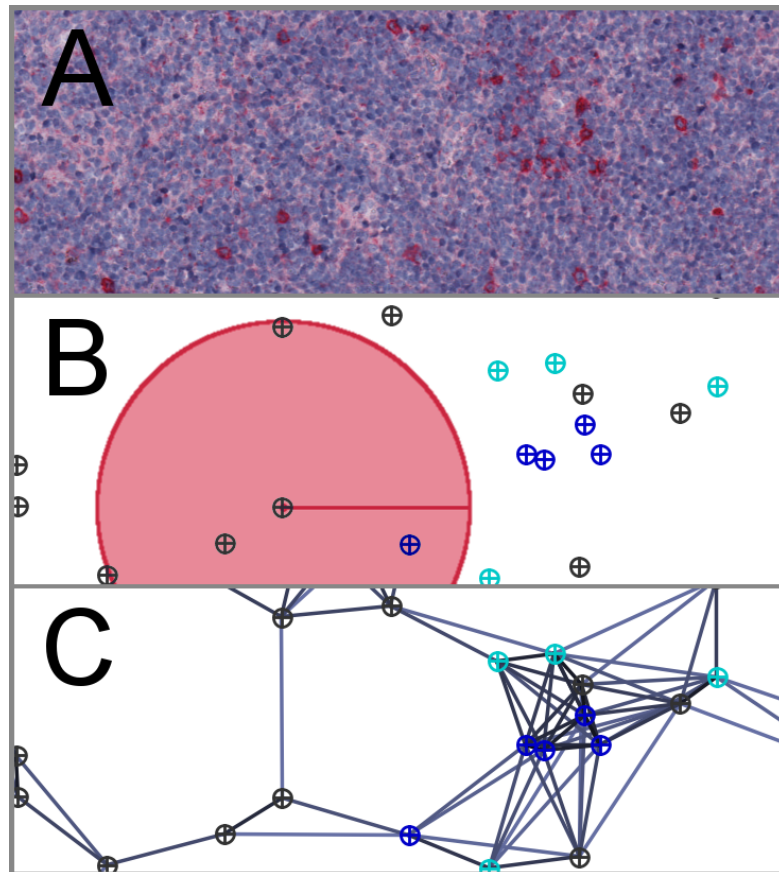


Figure 4.36: CD30⁺ cell graph with distance threshold 175 μm **A** depicts the original histological stained image. In **B** the cells are recognized and cell objects are marked. The coloring corresponds to the morphological properties. To illustrate the distance threshold of 175 μm , a unit disk of one of the cell objects is drawn. The final CD30⁺ cell graph is visible in **C**. Edges are added if the Euclidean distance between the centers of two cell objects is below the distance threshold t .

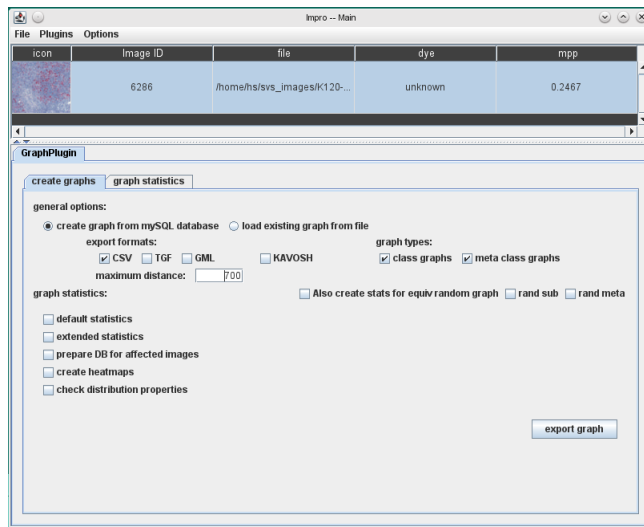


Figure 4.37: Graphical user interface of the Graph plugin Images can be selected in the image list, which is part of the Impro core program. The bottom panel contains additional functionality, provided by the Graph plugin. The GUI enables to export in multiple formats (CSV, TGF, GML, and KAVOSH), the creation of cell class specific graphs, the creation of corresponding random graphs, and further statistical analysis.

detailed description see Section *Unit Disk Graph*. Besides the complete cell graph the user can generate specific subgraphs defined by the cell morphology, that are cell class graphs and cell meta class graphs.

The user can also enable the generation of random graphs. An equivalent random graph with the same number of vertices compared to the original cell graph, but with new randomly chosen cell positions, is created. Random graphs and their definition are described in Section *Random Geometric Graph*.

Optionally, it is possible to compute typical graph statistics. The user can chose *default statistics* and the *extended statistics* option. The splitting of the provided statistics is reasoned by the computational demand. Default statistics include all graph properties that are fast to compute, e.g. all vertex properties that can be computed locally (node degree, cluster coefficient, and nearest neighbor type). The extended statistics option also enables extensible, time consuming tasks. At the current state, the plugin computes the eccentricity vertex centrality, which includes determining all shortest paths, and the connected components.

The options provided by the GUI also can be accessed via text command and can

be executed in headless mode. Listing 4.2 shows two exemplary commands. *ExportCellGraphs* loads a graph and exports it in the requested *output_formats*. Valid formats are GML, TGF, CSV, and KAVOSH. The second command, *CreateReducedCCCGraphs*, creates the corresponding cell class graphs and cell meta class graphs.

Listing 4.2: Supported commands Graph plugin

```
# create a cell graph from data base:
GraphPlugin;ExportCellsGraph:prepare_db_current=true;\
    igp_distribution_properties=true;threshold_distance=700;\
    create_heatmaps=true;default_statistics=true;\
    extended_statistics=false;output_format=CSV
# create sub graphs (cell class and cell meta class graphs)
GraphPlugin;CreateReducedCCCGraphs:create_heatmaps=true;\
    default_statistics=true;extended_statistics=false;\
    igp_distribution_properties=true;output_format=CSV;\
    rgtcat_save_reduced_graphs=false;
```

The *create_heatmap* enables the additional creation of heatmaps for all created graphs. A grid with a tile size of 512 x 512 is created and for each tile the contained cells are determined. The *cell count* and the *average node degree* is calculated. The heatmaps are stored as comma separated value files in the image directory within the impro workspace directory. The heatmaps can be viewed using the *ImageViewer* plugin, see Figure 4.33.

Cell graphs can be further processed via the *GraphReducer* plugin. It provides functions for graph reduction based on community structures. The original cell graphs are partitioned into cell groups and a new reduced graph is created, where each vertex represents a group of cells in the original graph. For the GraphReducer plugin no GUI is available, the plugin can only be accessed by impro text commands. The following Listing demonstrates the usage of the *GraphReducer* plugin.

Listing 4.3: Supported commands GraphReducer plugin

```
GraphReducerPlugin;ReduceGraph:workspace=$WORKSPACE$;\
```



```

    reducer=GeometricCliques;reducer_options=5,3/1,1;\
    max_level=10;
GraphReducerPlugin;ReduceGraph:workspace=$WORKSPACE$;\
    reducer=CommunityStructures;reducer_options=4;\
    max_level=10;
GraphReducerPlugin;CreateMultiLevelGraphStatistics:\
    dir=$WORKSPACE$/mgraph_stats_diss/

```

With the *ReduceGraph* command the plugin performs a graph reduction for the cell graphs of all loaded images. The *workspace* option lets the user chose a custom input directory. The *reducer* option specifies which method will be used for the graph reduction. The *GeometricCliques* reducer performs a clique-based reduction as described in 4.2.2. As additional parameter the user has to specify the *reducer_options*, in case of the *GeometricCliques* reducer a single integer value to define the clique size k . Optional the *max_level* defines the maximum number of reduction steps. The second supported reducer, *CommunityStructures* reducer, performs a clique percolation, which is explained in Section *k-Clique Percolation* (3.3.6). Here, the *reducer_options* needs to be two same sized lists of integer values, separated by a '/'. The first list specifies the clique sizes for each graph level (parameter k). The second list defines the required overlaps between two cliques to percolate into each other for each graph level (parameter x). Optionally the user can limit the maximum graph level and thereby restrict the maximal number of reduction steps by setting *max_level*. The *max_level* can be set for both reduction methods.

The *GraphReducer* plugin also provides a statistical overview of the reduced graphs. The statistics are created by the *CreateMultiLevelGraphStatistics* command. All *.mgraph* files stored in the requested input directory *dir* are processed. Distributions of the vertex count and the area size of all communities are created.

Cell graph properties

The vertex count of the CD30 cell graphs is equal to the number of cell objects detected by the imaging pipeline. Figure 4.18 summarizes the results for the CD30 image set. In

lymphadenitis, the number of vertices is less than 5,000, except for a single case which reaches about 10,000 CD30⁺ cells. Compared to the LA cases, cHL cell graphs commonly have more than 5,000 vertices. Typical ranges are 8,000 to 30,000 vertices for MCcHL cases and 5,000 to 15,000 vertices for NScHL.

Table 4.9 summarizes the edge count per vertex of the cHL image set. Overall, cHL cell graphs are more connected than CD30 cell graphs from lymphadenitis cases. One lymphadenitis cell graph was examined having a average edge count of 0.67 per vertex. In this case, many vertices are isolated in the graph and CD30⁺ cells have large distances to neighboring CD30⁺ cells. The graph consists of many small connected components, which may reflect less communication between CD30⁺ cells. The average edge count per vertex in lymphadenitis was 3.27. The sample LA graph depicted in Figure 4.38 has a comparable edge count per vertex, 3.81. The graph is disconnected, but connected components of size of 100 vertices or above exist.

Table 4.9: Edge count per vertex in CD30⁺ cell graphs

Diagnosis	Average edge #	\pm Std. dev.	Minimum	Maximum
LA	3.27	2.97	0.67	11.40
MCcHL	10.51	5.89	2.35	18.68
NScHL	12.10	3.80	5.33	18.37
ALL	8.78	5.77	0.67	18.68

In MCcHL and NScHL cases, the average edge count per vertex is 10.51, and 12.10, respectively. While NScHL has on average more edges, the number of edges vary more in MCcHL. The edge per vertex count in MCcHL ranges from a minimum of 2.35, which is comparably high as some of the lymphadenitis cases, to a maximum of 18.68, the overall highest edge count. To get a reference value for the edges per vertex count, we compared the observed edge counts to random geometric graphs. Figure 4.38 depicts exemplary CD30⁺ cell graphs in the row A and corresponding random geometric graphs in row B. The random geometric graphs were created to match the cell density of the original CD30⁺ cell graph. Note, that the overall cell density in the ROI of each WSI was taken into account. The image illustrates a small part of the tissue section and the

cell density within one column, showing the original cell graph and the corresponding random graph, may differ slightly. The vertices of the random graphs are distributed more evenly throughout the plane. The CD30⁺ cell graphs all have locally dense CD30 cell regions and within the tissue sections there are areas without any CD30⁺ cells.

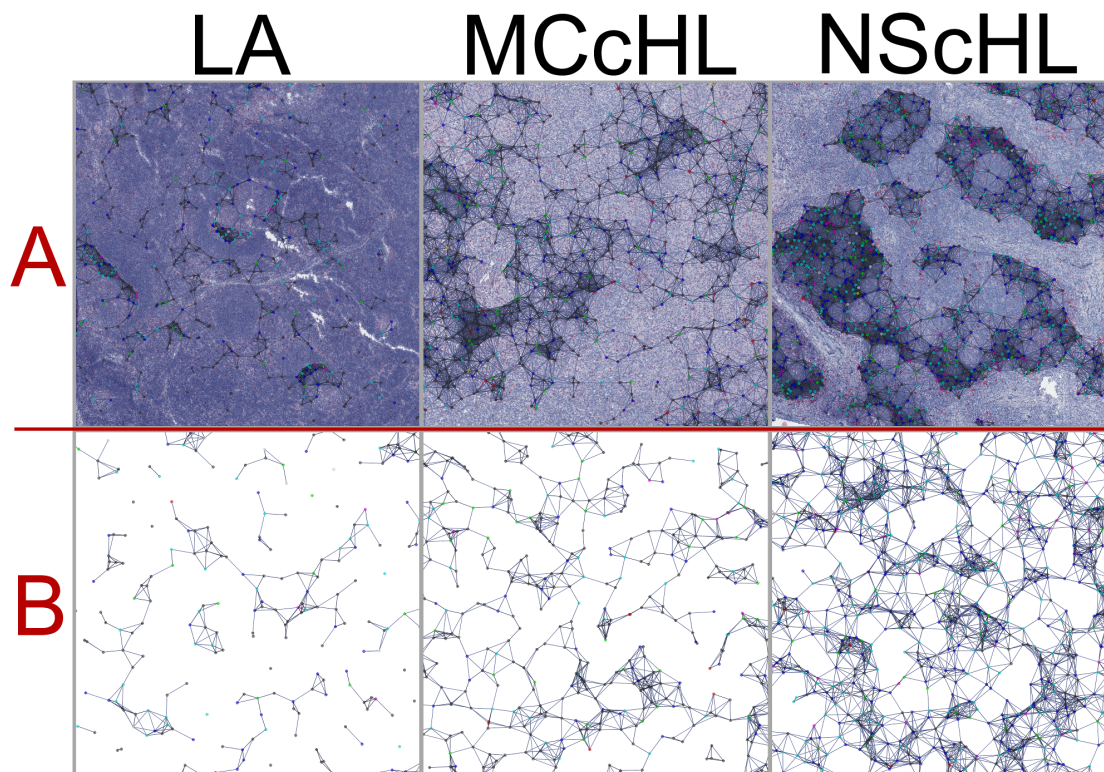


Figure 4.38: Sample sections of CD30⁺ cell graphs for all three diagnoses Row **A** depicts cell graphs computed from the cell detection. Row **B** shows random geometric graphs with a density equal to the sample in row **A**. The densities are calculated for the whole tissue section. The CD30⁺ cell graphs contain denser areas and for higher densities they still have areas depleted of cells.

We compared the edge per vertex count of the CD30⁺ cell graphs with simulated random geometric graphs. Figure 4.39 illustrates the comparison with the simulated data. The 35 images are sorted by the cell density on the x axis. On the y axis the edge per vertex count is shown as relative value compared to the average edge per vertex count of the simulated graphs of the same density. The bottom part of the figure gives a detailed view to the value range near 1. The average of the simulated data is always 1, see red line. The plot also shows the minimum and maximum values reached within 10 simulated random geometric graphs. All 35 CD30 images had an increased edge per

vertex count. The vast majority of cases were increased 1.3-fold up to 4-fold. In five images, the relative edge per vertex count were clearly increased. The relative edge count of those five cases were in the range of 6.4 and 14.6. All of the five images were cHL cases, four of which were NScHL.

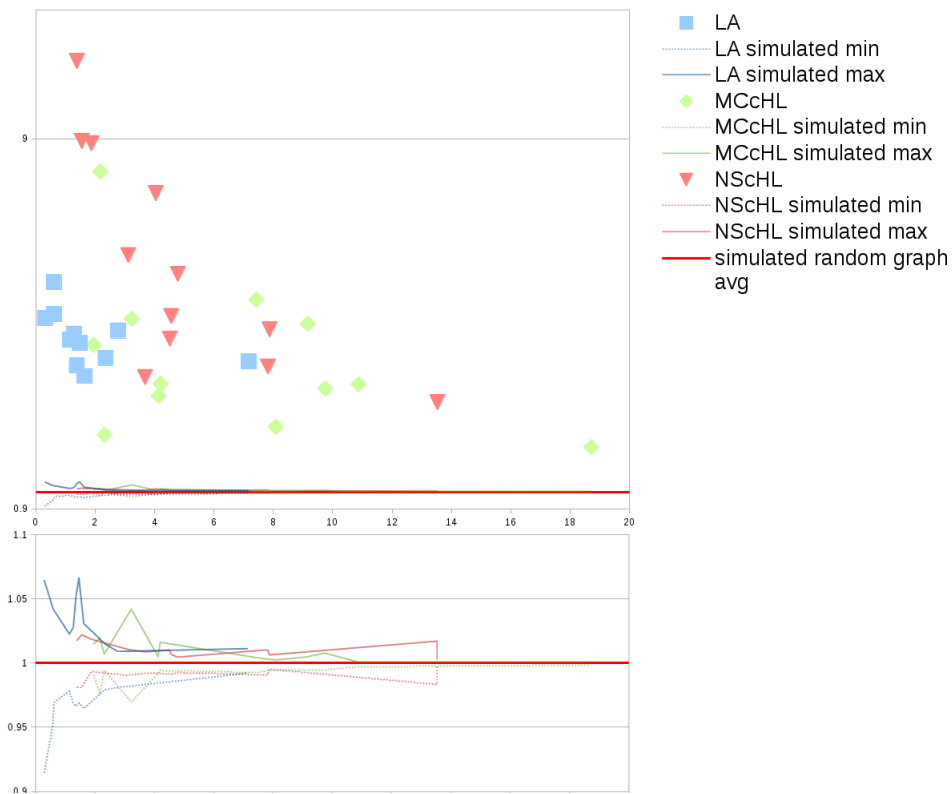


Figure 4.39: Relative edges per vertex counts Data for 35 images are plotted. They are sorted by the cell density on the x axis. The y value is calculated from the average edge count per vertex, divided by the average edges per vertex count of simulated random geometric networks with equally high cell density. The CD30 cell graphs have a much higher edge count. The edges per vertex count is typically increased by a factor of 1.2 to 4.8. The lower image depicts the ranges of the simulated data.

Interestingly, the relative edge count can vary a lot for images with similar cell density. Looking at MCcHL cases, we observed three image with a cell density of roughly 2. While one case had a relative edge count of 1.4, a value very close to the simulated random geometric graphs, another case had a value of 7.3 and thus differed substantially from the simulated data. The different cell distributions of both cases are illustrated in Figure 4.40. The image shows the ROI in white and the relative amount of CD30⁺ cells.

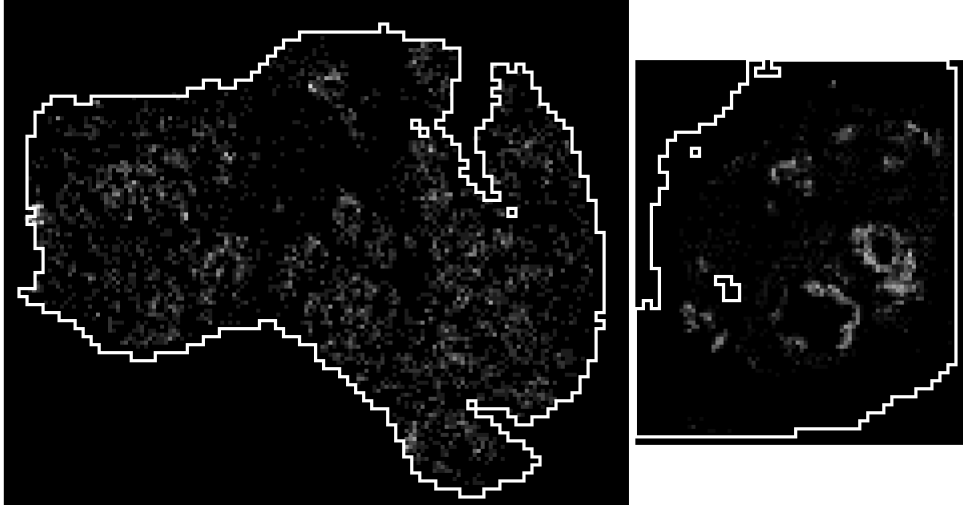


Figure 4.40: High and low edge count comparison The $CD30^+$ cell distribution of two MCcHL cases. The left case had a low relative edges per vertex count, the right one a high relative vertex count compared to the simulated random geometric graphs. The left tissue section has a more even $CD30$ cell distribution and only small parts of the lymph node are depleted by $CD30$ cells. In contrast, the second tissue section contains locally restricted clusters of $CD30$ cells of high density.

The two images were post-processed by increasing the contrast and brightness. Thus, the pixel brightness demonstrate the relative cell density within one tissue section. Intensity values are not comparable between the two tissue sections. The $CD30^+$ cells of the two lymph node samples highly differ. On the left side, the malignant cells are distributed equally through out the whole lymph node section. The whole section is affected and only few areas exist that are not infiltrated by the $CD30^+$ cells. The equal distribution in the whole lymph node section leads to the low relative edge per vertex count. In comparison, the right sample has large areas depleted of $CD30^+$ cells. The malignant cells instead are located in small, but dense areas. The densely packed neighborhood results in an accordingly high number of edges in the cell graph. Even though both cases were classified as MCcHL, the spreading of the malignant cells seem to follow different mechanisms. A follow-up question is which biological processes are the driving forces behind those mechanisms. The lymph node structure, cell-cell communication and cell movement may play an important role in the development of the disease.

Vertex degree distribution In cell graphs, the vertex degree describes the number of CD30⁺ cells in the local neighborhood. Due to the Euclidean distance based construction of the CD30 cell graphs, the vertex degree reflects the cell density per unit disk area. Accordingly, the vertex degree distribution gives information about the cell distribution in the tissue section. Low vertex degrees mark CD30⁺ cells in regions sparsely populated by CD30⁺ cells, high vertex degrees mark cells clustering with many other CD30⁺ cells. The vertex degree distribution differed for the three diagnoses, see Figure 4.41 and Figure 4.42. A low average vertex degree was found in lymphadenitis cases. The vertex degree was shifted towards high values in cHL.

In lymphadenitis, most vertices have a vertex degree of less than 10. The cHL cases had on average higher vertex degrees, but we also found a high variability in the image sets. Figure S9 summarizes the vertex degree distribution for all three diagnoses. For the majority of cases the vertex degree distributions are typical for one specific diagnosis. In lymphadenitis, the distribution has a high and narrow peak between 0 and 10. Vertex degrees of 20 or above are an exception.

The exemplary MCcHL vertex degree distribution in Figure 4.41 has its maximum at 16. Here, most CD30⁺ cells are distributed in areas with a higher cell density compared to CD30⁺ cells in lymphadenitis. The vertex degrees are distributed over a broad range. Cells most frequently have a vertex degree between 10 and 20, but still a large fraction of CD30⁺ cells are neighbored by more than 20 CD30⁺ cells.

The highest vertex degree maxima were found in NScHL cases. Figure 4.42 gives two example distribution of NScHL. **NScHL-I** has a relative low mean vertex degree compared to other NScHL images. With respect to the average vertex degree it is more similar to the MCcHL cases. The maxima of the curve is at vertex degree 12, but still many high degrees (> 30) were observed. **NScHL-II** demonstrates the vertex degree distribution of a NScHL case with highly clustered CD30⁺ cells. The most frequent degree is 47. Overall, the vertex degrees were more evenly distributed in the range of 0 and 85.

Figure S9 compares the three image sets by plotting the average vertex degree distributions, and the respective maxima and minima for each diagnosis. The averaged

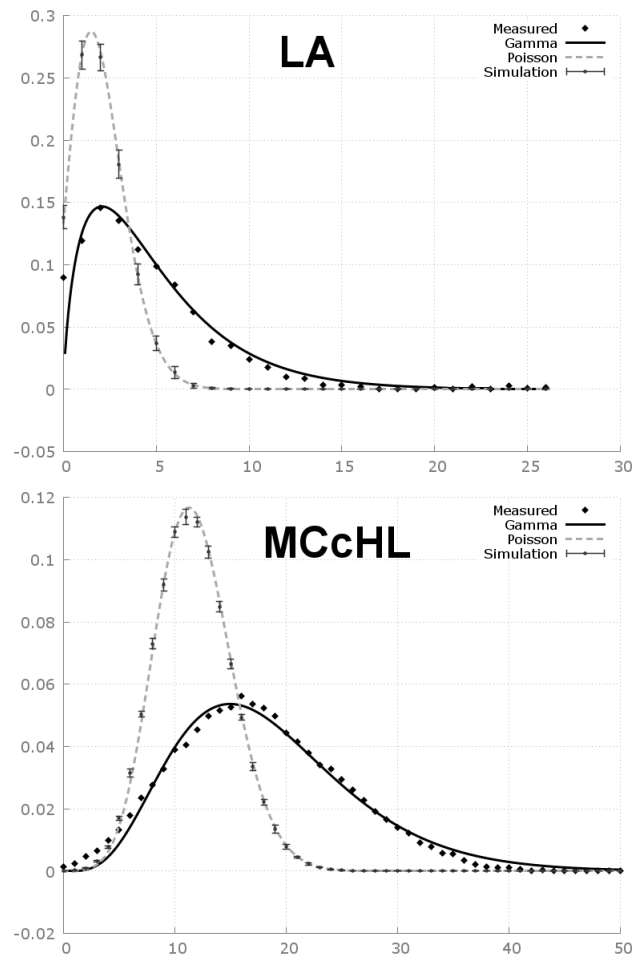


Figure 4.41: Vertex degree distribution of lymphadenitis and MCcHL The upper diagram depicts the typical vertex degree distribution of a lymphadenitis case. The distribution peaks at a vertex degree of 2. Vertex degrees higher than 20 are rarely seen in lymphadenitis. In 9 out of 11 lymphadenitis cases, less than 10 % cells had a higher vertex degree. The second diagram shows an exemplary vertex degree distribution of the diagnosis MCcHL. The distribution is shifted towards higher values compared to lymphadenitis and the number of high vertex degree values is significantly increased. Although the majority of MCcHL vertex distributions are similar, extreme distributions that are shifted to very low or to high vertex degrees also exist.

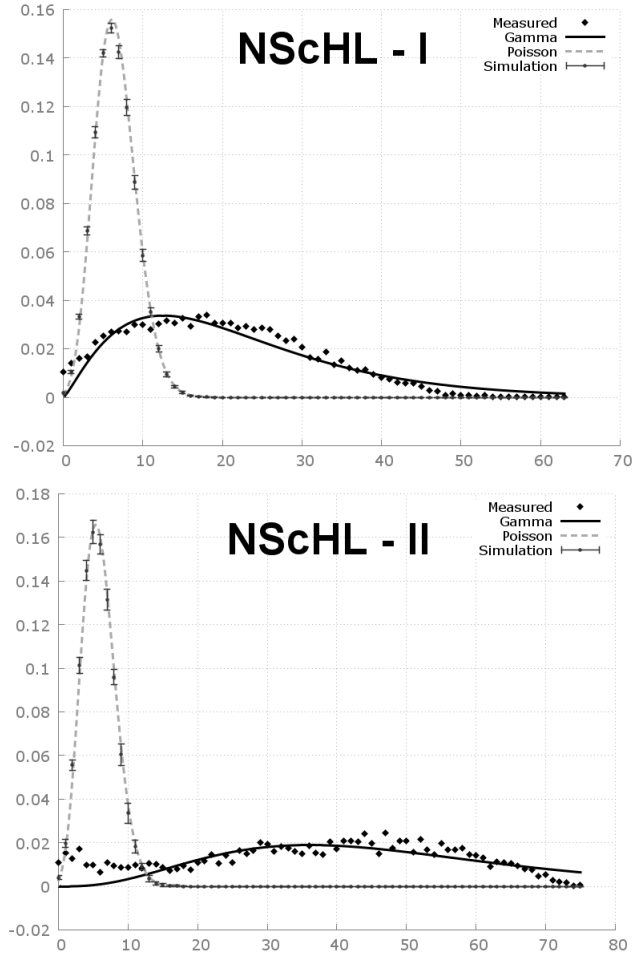


Figure 4.42: Vertex degree distribution of NScHL cases Both diagrams depict vertex degree distributions of NScHL diagnosed cases. While the first distribution has its maximum at a low vertex degree of 12, the second shows a more uniform distribution of vertex degrees with a maximum at 35. Even though the maximum value is reached for a lower vertex degree than in the MCcHL cases in Figure 4.41, the case NScHL I has many cells with a high vertex degree. Vertex degrees of 40 and above are more likely to be found in NScHL than in MCcHL.

curves approve the statements gained from the exemplary vertex degree distribution. In general, LA shows very low vertex degrees. While MCcHL has in many cases a high fraction of vertices with a degree of 5-15, NScHL is characterized by a flat vertex degree distribution with equally distributed vertex degrees between 5-40. However, those trends are not sufficient to distinguish between the three diagnoses. Exceptional cases exist resulting in similar vertex degree distributions across the three diagnosis image sets, see maxima and minima curve in Figure S9.

Randomness of cell distribution

To check whether the CD30⁺ cell distribution is caused by a directed driving force or is a rather random process, the measured vertex degree distributions were compared to random geometric graphs of same cell density, see Section 3.3.3 in **Methods**. With random geometric graphs as null model, we assume equally, randomly distributed CD30⁺ cells in the lymph node section.

As the positioning of each vertex in the 2D geometric random graph is randomly chosen in a plane, the likelihood of two vertices, v_1 and v_2 , to be neighbored, is only dependent on the distance threshold t of the graph in relation to the size of the plane. The probability $p_N(v_1, v_2)$ of v_1 being connected to v_2 with an edge can be expressed as:

$$p_N(v_1, v_2) = \frac{\pi t^2}{A_{plane}}, \quad (4.5)$$

where πt^2 is the area of a unit disk in the graph and A_{plane} is the area of the plane. The number of neighbors, i.e. the vertex degree distribution, can be modeled as Poisson distribution with $\lambda = E_{disk}$, the expectation number of cells per unit disk. Both, simulated geometric random graph data and the Poisson distributions are depicted in Figure 4.41 and 4.42. Compared to the null model, the vertex degree distributions of cell graphs are shifted to high values. The modeled Poisson distributions have a single, narrow peak. The maxima are 1.5 in LA, 5.5 and 6 in NScHL, and 15 in MCcHL for the four sample vertex distribution. Compared to the modeled distributions, the measured vertex degree distributions are asymmetrical. In most cases they have a long tail of higher vertex degrees, which occur with a low frequency. The differences of the observed distributions

and the expected Poisson distributions of random geometric graphs are clearly visible so that we can reject the null hypothesis of a random distribution of CD30⁺ cells in the lymph node.

CD30⁺ Cell Clustering

The shift to higher vertex degrees and the long tails of very high vertex degrees suggest a preference of clustering for CD30⁺ cells. This effect is visible in cHL as well as in LA cases. The nonuniform distributions of cells in the sections are partly caused by the lymph node structure itself. Lymph nodes are structured into several compartments with different functions. The inner parts are sectioned into multiple lobules, which are also organized into sections, the superficial cortex and the paracortex.

During infections only parts of lymph node might be active and CD30⁺ reactive immune cells, are only present in some of the compartments. As we measured the cell density within the whole lymph node, the CD30⁺ cluster more than expected. However, the clustering of cells is way higher in cHL than in LA. Multiple regions with high cell density are observable, while other regions are nearly depleted of CD30⁺ cells. The clustering can be reasoned by different processes. First of all, cell division plays an important role for the spreading of malignant cells in the lymph node. Ongoing cell division in combination with low cell movement leads to a clustering of the malignant cells. Another hypothesis would be active movement. The malignant cells may be attracted to specific regions. The nutrition, as well as the surrounding micro environment are possible influences.

CD30⁺ Cell Graphs Are Not Scale-free

In Section *Randomness of cell distribution* we described the differences between the measured vertex degree distributions of CD30⁺ cell graphs and simulated Poisson distributions. Except the shift to higher vertex degrees most measured distributions were asymmetrical. They were described best by a gamma distribution, see fitted gamma distribution in Figure 4.41 and 4.42. The gamma distribution is defined by two parameters, the rate and the shape parameter. As moments of arbitrary degree exist for the gamma

distribution we can conclude that CD30⁺ cell graphs are not scale-free.

The rate and shape parameter of the fitted gamma distributions of all 35 images are depicted in Figure 4.43. The gamma distribution gained from LA images tend to have a low shape parameter compared to the cHL images. On average LA images have a higher rate parameter than cHL images, but here the overlap between the image sets is much higher.

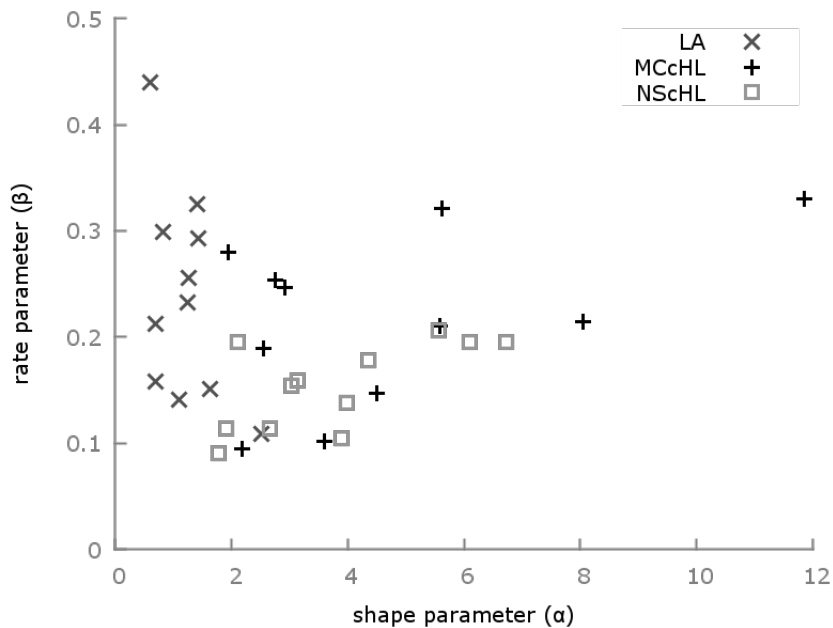


Figure 4.43: Gamma parameter overview All 35 images separated into the three diagnoses LA, MCcHL, and NScHL. The plot depicts the shape and the rate parameter of the gamma distribution fitted to describe the measured vertex degree distribution of the CD30⁺ cell graphs best. A low shape parameter is typical for LA cases.

The LA image set is separable from the cHL cases using the rate parameter. Only exceptional case overlaps with the MCcHL and NScHL image sets. However, the two cHL subtypes can not be distinguished based only on the gamma distribution parameters, shape and rate.

4.2.2 CD30⁺ Community Structures

Up to now we analyzed the CD30⁺ cells according their spatial position and the relation between locally neighbored cells. The goal of this rather limited view is to break down

the large system, CD30⁺ cell graphs of 2D lymph node sections reach more than 50,000 vertices, into smaller pieces. This makes it possible to measure properties of the system in low computation time. Local properties like vertex degree and nearest neighbor distance can be calculated without seeing the whole, complex system. Instead of treating single cell objects, this section focuses on cell groups. Here, the step is to identify meaningful groups. In graph theory, groups of vertices computed based on the graph topology, are called cluster or communities.

The following section discusses the results of two graph theoretical methods to define communities in CD30⁺ cell graphs. The first method is k -clique percolation, which has been used for various networks. The second method iteratively combines vertices of k -cliques using edge contraction.

k -Clique Percolation

k -clique percolation is a widely used method to determine groups of vertices in graphs [54][55]. In cell graphs, the community structures are based on the geometric property of the graphs that arises from the unit disk graph construction. Here, edges are generated according to the Euclidean distance of vertex pairs. A k -clique arises in unit disk graph if the locale vertex density is at least $\frac{k}{A_0}$, where A_0 is the area of a circle with a diameter of t , the distance threshold of the unit disk graph. The k sets a limit regarding the minimum cell density in the cell graph. All cells located in regions of lower density are not part of the community structure. Cells share the same community if they are connected via a chain of adjacent k -cliques.

Figure 4.44 depicts the community structure of a NScHL case, for $k = 3, 5, 10$. Communities are highlighted in different colors. The overlay, created by the *ImproImageViewer* plugin depicts the member cells of the communities, and the area of the community. The area is calculated by Delaunay triangulation, but invalid triangles, that are triangles with an edge length greater than the distance threshold of the unit disk graph, are filtered out. A more detailed view is visible in Figure S11 in the supplements.

For $k = 3$, there are only few, very large communities, see Figure 4.44 A. The communities are mainly limited by the lymph node structure, which is heavily altered

by the NScHL. Sclerotic bands section the lymph node into separated nodules and thus split the population of malignant cells into multiple large communities. The adjacency criterion of 3-cliques is rather low and in conclusion touching groups of CD30⁺ cells are combined into one community, e.g. see large brown community in Figure 4.44 A.

For higher k values, those large communities are split into multiple smaller communities, see beige and olive green community in Figure 4.44B. Such oval communities are split even more in the community structure gained by 10-clique percolation. Here, only very dense cell groups are connected to one community. In addition low CD30⁺ cell density regions are not covered by the community structure any more, because of the lack of 10-cliques in the corresponding part of the CD30⁺ cell graph.

Figure 4.45 summarizes the community counts, detected by the 3-clique percolation method. For most cell graphs, the community count ranges from 60 to 300. In NScHL the community count is slightly decreased compared to MCcHL and LA, suggesting that the CD30⁺ cells in NScHL are clustered to fewer, but larger communities.

The size of the CD30⁺ cell communities differs among the diagnoses lymphadenitis, MCcHL, and NScHL. It is also influenced by the chosen k value for the k -clique percolation. Figure 4.46 summarizes the community sizes, by plotting the relative amount of CD30⁺ cells in a community of size x . Because of the stricter condition, 10-clique percolation produces much smaller communities compared to 3-clique and 5-clique percolation. Here, only few cells were located in communities greater than 3,000.

In comparison, lymphadenitis cases contain much smaller CD30⁺ communities compared to cHL. Even for 3-clique percolation, which produces the largest communities, nearly no community were found in lymphadenitis having a size of $> 1,000$. A single, exceptionally large community (8,000 cells) was detected in a lymphadenitis image showing an untypically high CD30⁺ cell density. 5- and 10-clique percolation require much higher cell densities. In 10-clique percolation this requirement is so strict, that even in NScHL and MCcHL cases the cells are split into small communities, typically of size 3,000 and below. In the 10-clique percolation plot of Figure 4.46, only six communities greater 3,000 cells exist, four of which are MCcHL cases. In NScHL fewer large communities are found, because tissue sections with a high CD30⁺ cell content show a highly

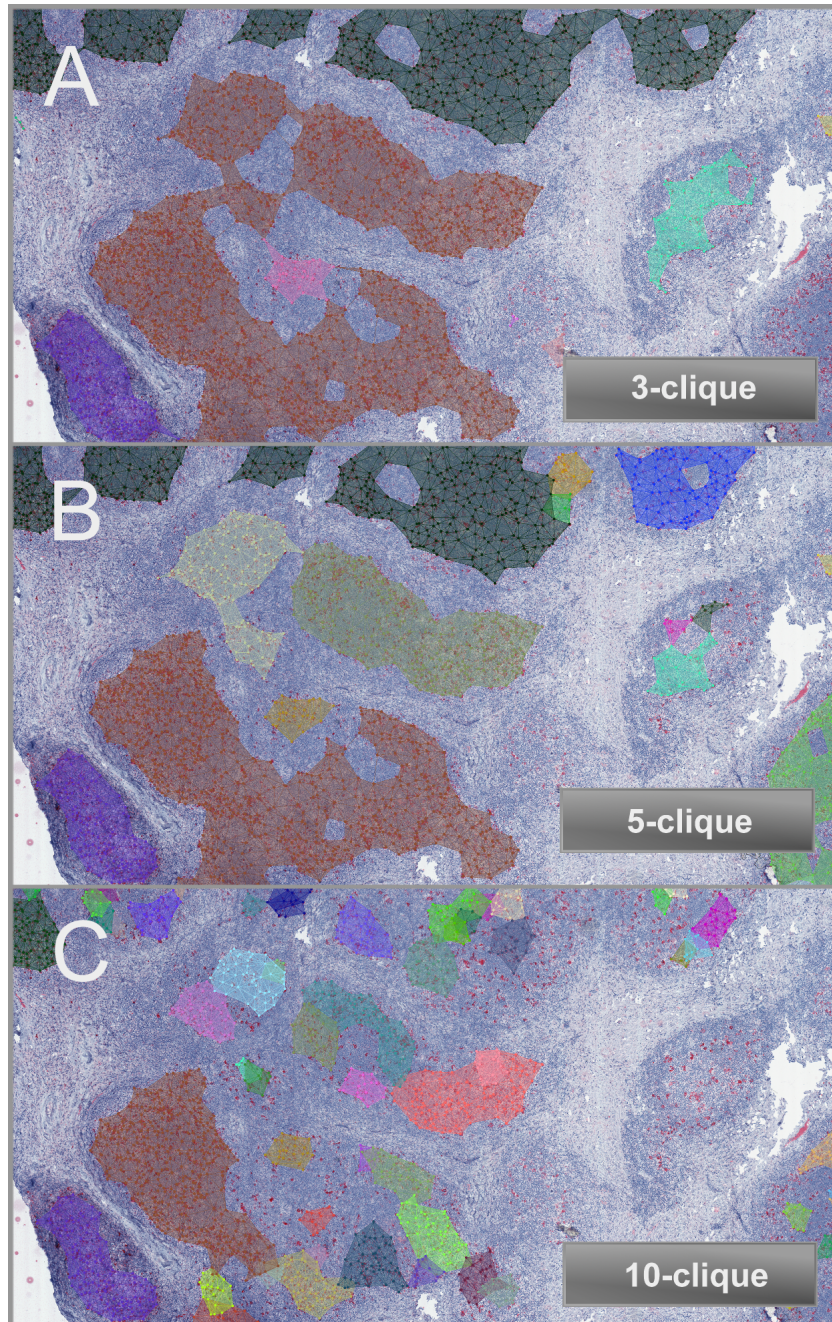


Figure 4.44: Communities in NScHL cell graph Results of the k -clique percolation for $k = 3, 5, 10$. The communities are highlighted in different colors. 3-clique percolation leads to a small number of large communities (A). With raising k values, the vertices are grouped into more, but smaller communities. 10-clique percolation only includes regions with a local CD30^+ cell density of $\frac{10}{A_0}$ or higher. Low density regions, e.g. the small tissue section within the sclerotic region on the right side, are not included in the community structure for high k values.

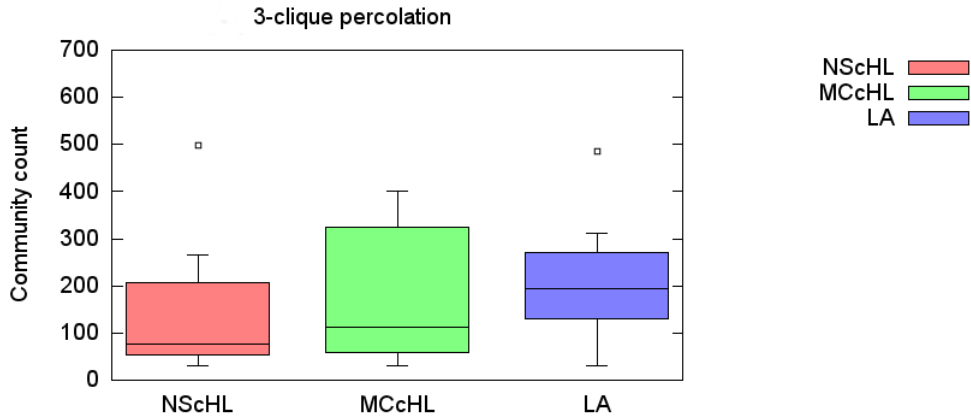


Figure 4.45: Community counts for the three diagnoses The 3-clique percolation reduces the graph and the majority of the cell graphs have less than 300 communities. The differences between the three diagnoses is low, but NScHL has on average a lower community count than the other diagnoses.

altered lymph node structure and the formation of nodules by sclerotic bands separates the malignant cell populations into smaller communities.

While Figure 4.46 plotted the size in number of cells, Figure 4.47 depicts the area in μm^2 . The area was calculated by Delaunay triangulation, but additional exclusion of triangles with a side length exceeding the distance threshold t of the cell graph. Most cells are located in communities with small expand. The percentage of cells drops for communities which have expanded wider in the tissue for all three diagnoses. In lymphadenitis the drop is steeper and only a small percentage of CD30^+ cells can be found in an community with an area of $> 100,000 \mu\text{m}^2$. For higher k s, the clique percolation algorithm produces a lower number of communities having a small area. The 10-clique percolation plot visualizes that no cells are located in communities having an area less than $7,000 \mu\text{m}^2$. From this we can conclude, that communities with a small area only exist in low cell density regions, which are excluded from the computation of the communities because of the high k value.

The overall smaller communities in lymphadenitis may have multiple reasons. First, the smaller number of CD30^+ cells per community can be, at least partly explained, by the overall lower density, see Section *CD30⁺ Cell Density*. The smaller area of communities in lymphadenitis may originate from the more controlled progress of the inflammation, compared to the rather uncontrolled spreading of malignant cells. In some of

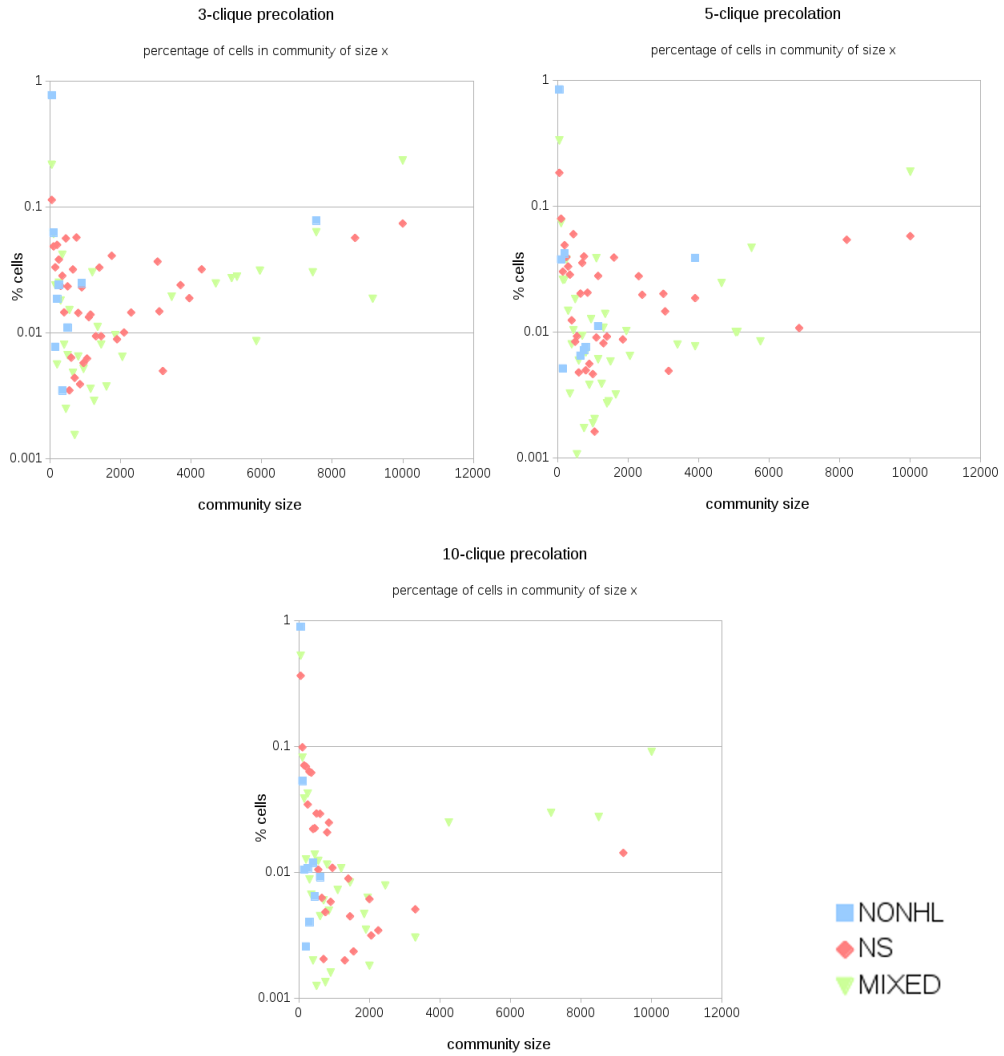


Figure 4.46: Average percentage of cells in a community of size x The plot depicts the relative cell counts located in communities of size x for the three diagnoses lymphadenitis, MCcHL, and NScHL. The size of the communities decreases for high k values. Overall, $CD30^+$ cells in lymphadenitis are located in communities of small size, while cHL cases contain larger communities.

the cHL tissue sections the malignant cells were spread through the whole lymph node section. In lymphadenitis the CD30⁺ cells were mostly limited to only parts of the tissue section.

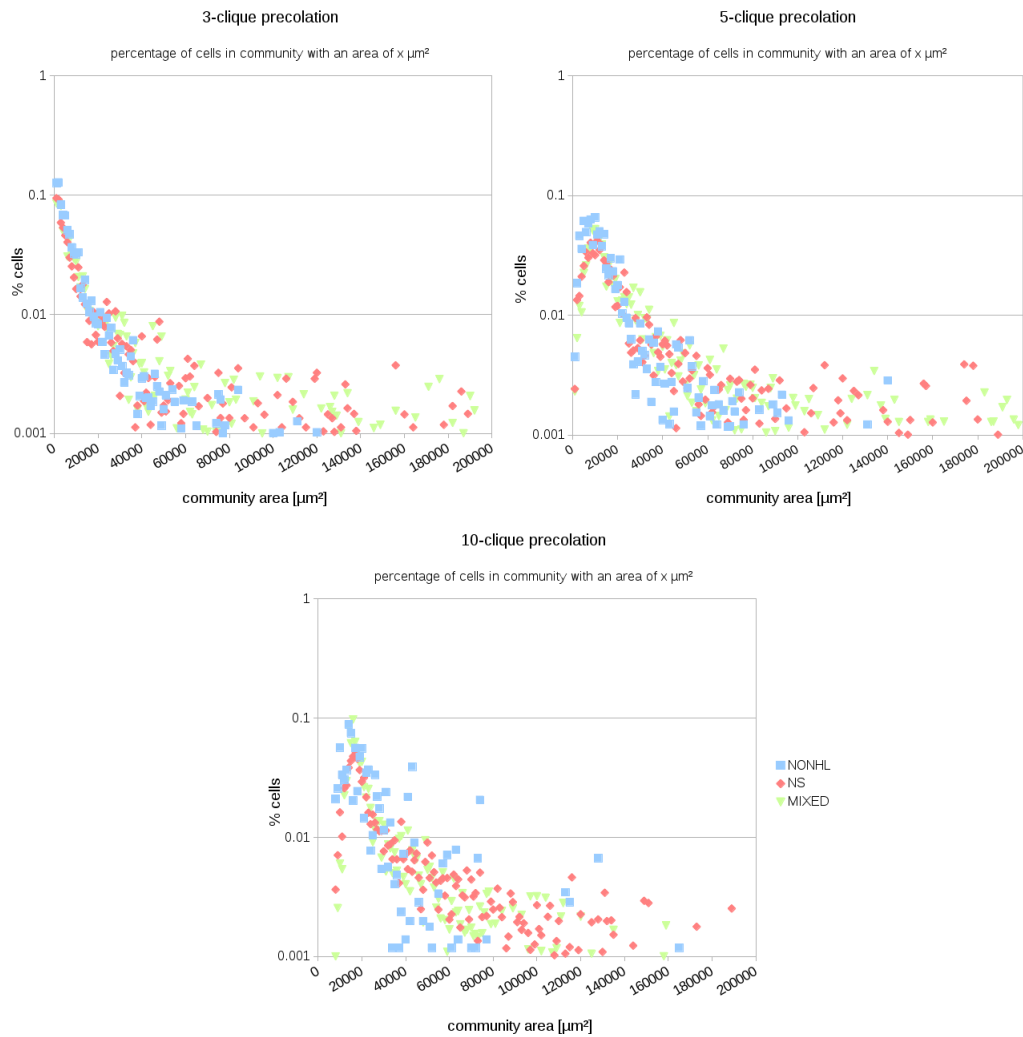


Figure 4.47: Average percentage of cells in a community with an area of $x \mu\text{m}^2$. The three plots depict the ratio of cells in communities of certain sizes for k -clique precolation ($k = 3, 5, 10$). Cases diagnosed as cHL have communities expanded over a bigger area than cases of lymphadenitis.

Graph Reduction by k -Clique Contraction

A second method to find communities in cell graphs is k -clique contraction. The idea is, to contract a set of non-overlapping k -cliques in each iteration. Therefore all current

k -cliques of the cell graph are computed and scored. In this work, we use a very basic score based on the Euclidean distances of the vertices in the k -clique. Beginning with the best scoring k -cliques, the algorithm contracts all edges of the selected, non-overlapping cliques. The process can be iteratively applied to the reduced graph, resulting in a set of graphs with increasingly high reduction level.

An example is depicted in Figure 4.48. The figure illustrates different graph levels[‡]. For each community, the center of mass is displayed and the expand is drawn with different colors. The method is a bottom-up approach, thus at the beginning each vertex lies in its own community. The k clique contraction merges k -communities to form a single, larger community. After multiple steps the communities consist of large cell groups, representing CD30⁺ cells clustering in the lymph node section, see the two bottom images in Figure 4.48. For the corresponding CD30⁺ cell graph, with different reduction levels we refer to Figure S10.

Graph Reduction Steps The number of iterations and the growth of the communities depend on the connectivity of the graph. In densely connected regions the communities grow exponentially with a rate of k . Thus, the number of the communities drops quickly in early reduction steps. Figure 4.50 depicts the fraction of vertices of the original cell graph that is located in a community of size x or higher. The diagram plots the relative counts for ten iteration steps. The curve is shifted towards higher community size values the more iteration steps were performed. We can also see that the size of the communities is limited by the number of iterations. As only non-overlapping cliques are reduced in on step, the maximum vertex count of a community can be k^i , where i is the number of iterations. While during the first iterations many small communities are merged to form a larger community, later iteration steps affect rather large communities. For the example graph, there are only little changes between reduction level nine and ten. This is also true for most of the CD30⁺ cell graphs in this study. From visual inspection we found the communities in reduction step six to nine, to be most similar to communities we would select manually.

[‡]Some reduction level are not shown.

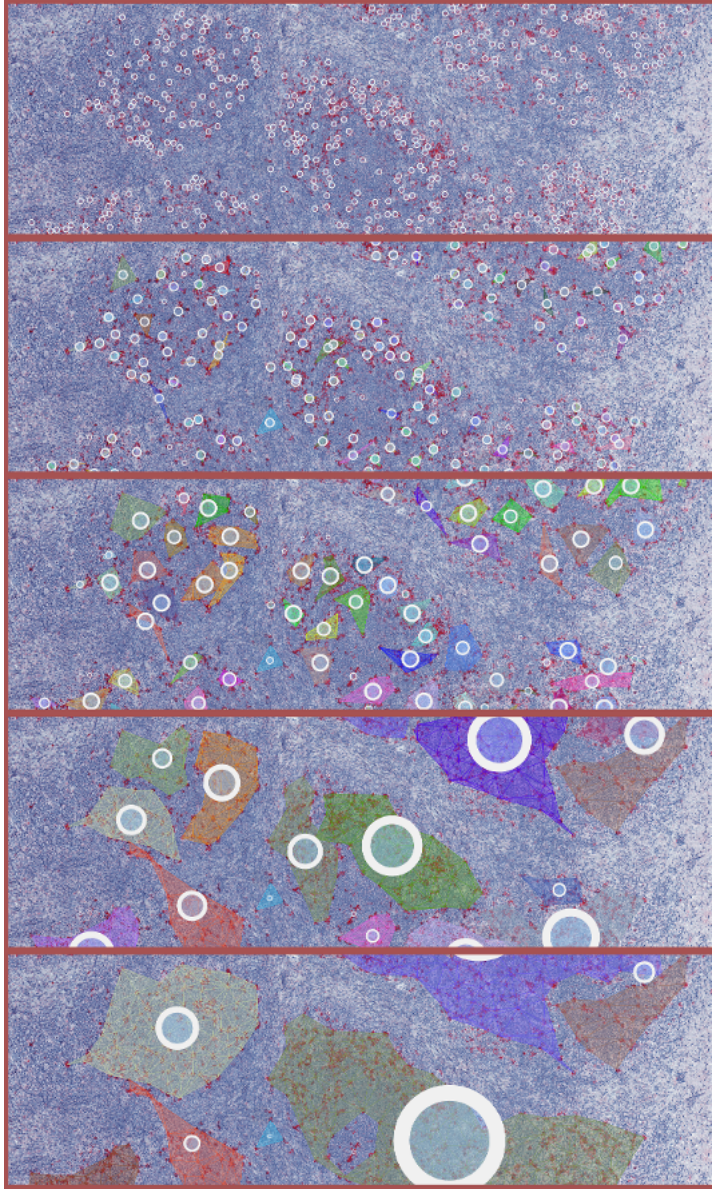


Figure 4.48: Communities gained by clique contraction The figure depicts a small section of a NScHL case. The communities of the $CD30^+$ cell graph are drawn for multiple steps of the clique contraction method. The area of each community is highlighted in a single color. The white circles represent the center of mass' and their size is proportional to the number of vertices within the community. The figure also illustrates the bottom-up approach. First, each vertex for its own is one community. The k -clique contraction merges k of those small communities into one larger community. The bottom picture shows a late step with very few, large communities.

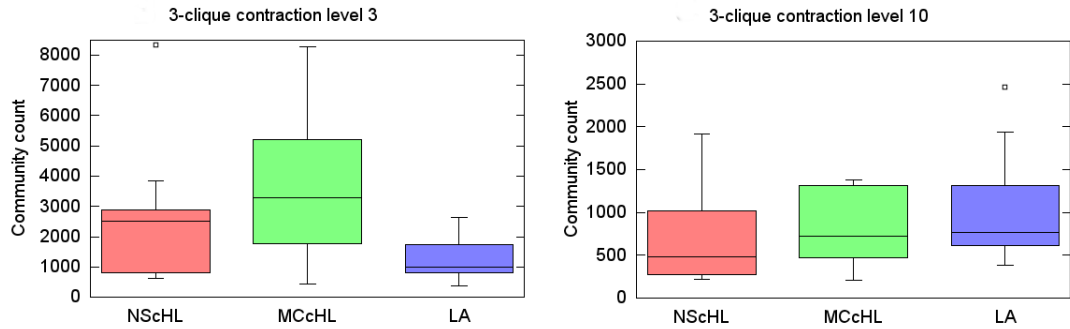


Figure 4.49: Community counts 3-clique contraction The 3-clique contraction iteratively combines cliques of the graph to larger communities. While at reduction step 3 many communities exist, the number decreases further to less than 1,300 after 10 reduction steps. The number of communities is still high, compared to the 3-clique percolation, see Figure 4.45.

Overall, the clique contraction methods produces more communities, compared to the clique percolation. Figure 4.49 gives an overview of the community counts per diagnosis. Two iteration steps are plotted. After three iterations many small communities exist, which are further combined to larger communities in the following reduction steps. After three iterations we observe a much higher number of communities in MCcHL compared to the other two diagnoses. The lower community count in LA is expected as LA $CD30^+$ cell graphs have a lower number of vertices. The cell count in NScHL cases is similar high compared to MCcHL. Here, the clique contraction method reduces the graphs in early reduction steps more than in MCcHL, as $CD30^+$ cells cluster more in the tissue.

For comparison, Figures S12 and S13 show the fraction of vertices in communities of size x for two artificial graphs. In the random graph, in which vertices are randomly placed in space with same density as in the original graph, the communities tend to be smaller and cells cluster less than in the $CD30^+$ cell graph. In the regular graph, see Figure S13, the communities grow exponentially and all communities have nearly the same size. Slightly differences of the community sizes are caused by the sorting of the cliques done by the reduction algorithm. A measurement of the quality of the community structure is the modularity, Q . For the k -clique reduction method, we could see that, first, the Q value raises with each iteration step. Because of the scoring and sorting of the k cliques, dense cell cluster are merged first into communities. At some point the algorithms starts merging big cell clusters with only few connections and the Q value decreases. De-

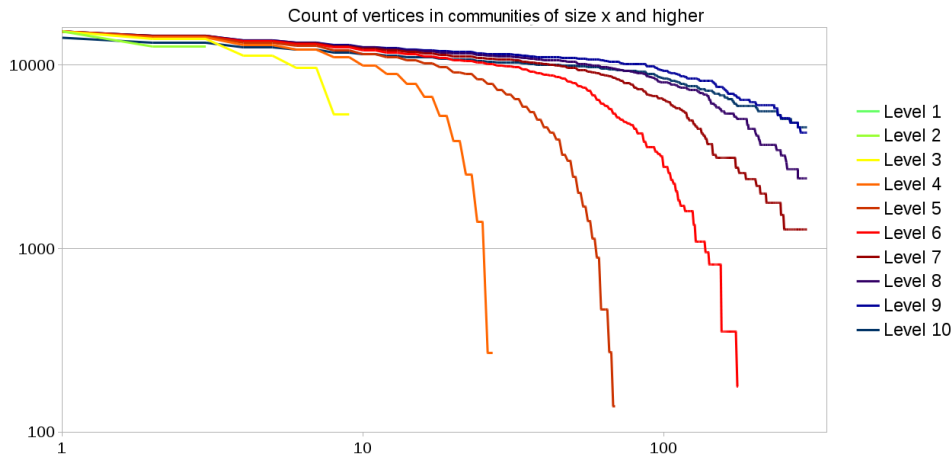


Figure 4.50: Communities gained by clique contraction Each curve represents one reduction step for an exemplary CD30⁺ cell graph. The y axis displays the total number of vertices, that located in a community of size x or higher. The community sizes are partly limited by the number of iterations, for a larger number of reduction steps, it is mainly caused by the graph structure.

pending on the CD30⁺ cell distribution, the Q value can also have local maxima for multiple iteration steps. For a more complete overview of the Q values of community structures in cell graphs see 'Bioinformatische Bewertung von Graphpartitionierungen durch cliquenbasierte Reduktion auf Gewebeschnitten des Hodgkin-Lymphoms', a bachelor thesis performed by TM.

Comparison of k -Clique Percolation and k -Clique Contraction

Clique percolation is a very frequently used method to find communities in graphs, e.g. in biological networks. However, the application to geometric graphs, a property which cell graphs have because of their construction as unit disk graphs, has some limits. The separation of cell clusters with clique percolation highly depends on parameter k . Two touching cell clusters are only separated if no k -clique exist, which percolates into k -cliques of both clusters. As cell cluster sizes differ within one tissue section and the touching area of two adjacent cell clusters may also differ in size, the required k value to separate those clusters may differ. Thus, a constant k value can be inaccurate in tissue slides where the cell density and/or the cell cluster size differ throughout the tissue section. For large k values only regions with a high CD30⁺ cell density, are

considered. The minimum cell density of a cell group to be part of the community structure is $CD30_{density} \geq \frac{k}{A_{o_t}}$, where A_{o_t} is the area of a circle with radius t , the distance threshold of the underlying cell graph. In contrary to the k -clique percolation, the clique contraction method allows smaller graduated reduction steps. The benefit of the finer reduction steps, comes at the cost of scoring the community structure in each iteration. Another complex task is the definition of a good score for the k -cliques. The currently used score, based of the vertices' distances, locally favors merging of vertices with a small distance. But the algorithm ignores global differences of the cell density in the tissue sections. In low CD30⁺ cell density regions, cliques with a large spreading of the cells are merged in early iteration steps, while similar cliques near dense CD30⁺ regions are merged later on. The score of the clique contraction method needs further tweaking. Currently, all non-overlapping cliques are reduced in each iteration. This also results in reducing cliques with different scores. In a single step, cliques with high scores as well as cliques with low scores are reduced. An adaptation would be a threshold relative to the best scoring clique. The algorithm stops when only cliques remain whose margin to the best scoring clique is above the threshold.

To make k -clique percolation better applicable to cell graphs, multiple k values need to be tested. Community structures found with low k values can be further divided by increasing k , resulting, similar to the the clique contraction method, into a hierarchical structure. The best fitting community structure then needs further information and a scoring, e.g. the Q modularity, is required to decide which partitioning best reflects the cell distribution.

Chapter 5

Conclusion

Digital pathology is a growing field. Whole slide scanners allow the digitization of complete tissue sections and make object slides accessible for computer aided analysis. We examined a set of 35 pre-selected WSIs with a high resolution. With 0.25 μm per pixel, the WSIs enable a detailed view of the cell. Even cell compartments, like the nucleus, and the outline of the cells are visible.

The image set consists of tissue sections with three different diagnoses, two of which are the cHL sub-types NScHL and MCcHL containing CD30⁺ malignant tumor cells. For comparison, we included also lymphadenitis cases as the third group. Here, the CD30⁺ cells are non-malignant cells. The CD30⁺ cell population is composed of activated lymphocytes and is involved in a normal inflammation of the lymph node.

The images have been provided by the Dr. Senckenbergisches Institute for Pathology. Thus, the staining quality conforms with object slides used in the routine diagnostic. The 35 pre-selected cases were of best quality, chosen from a set of 137 WSIs in total. About 36.5 % of the WSIs were considered to be of good quality. Unfortunately, the number of cases and the overall quality differed throughout the three diagnoses. While there was a large number of 57 MCcHL cases available, the image count for lymphadenitis and NScHL was ~ 40 each. To achieve similar image set sizes, we limited the data set to the 35 presented images. From the original image set, 23.4 % of the cases were rated to have a very low quality. While the image quality may be sufficient for an

expert pathologist to make a diagnosis, it may lead to highly inaccurate results with the computer-aided approach. Large areas show staining and preparation artifacts, such that tissue structures are barely recognizable, even for human eye. The diagnosis can be made using just selected parts of the WSI and thus regions with staining issues are neglected. For the computer-aided method the statistic is done for the whole tissue section and is more sensible for an insufficient image quality.

One of the main goals of the Impro software was dealing with the SVS whole slide image format. To integrate standard imaging software into the pipeline it was required to provide the images in a standard image format. This was achieved by splitting the WSI into image tiles in standard TIF format. During the last years, projects like the OME evolved, and accessing WSI formats became easier. Now, standard imaging programs allow reading such images, but the support for pyramidal tiled images is still low. Our approach to convert the images has the advantage of being independent of the input WSI format, and the separate treatment of the image tiles easily allows for a parallelization of the imaging tasks. Yet, the approach also comes at a cost. First, the image tiles are written to the hard disc to make them available to external programs like CellProfiler. This produces a high I/O load, slowing down the processing time of the images. For everyday usage of the computer-aided analysis in routine pathology, such a slow-down needs to be avoided. Second, the external program has no information about the whole image. For example, some automatic thresholding methods of CellProfiler cannot be applied to the separated image tiles, because the intensity value distribution of the whole image is necessary.

The conversion of the WSIs into separated image tiles is only one time-consuming task. Overall, we tried to fasten up the image processing time allowing to work off whole slides. On the one hand, this is done by reducing the input data, e.g., by defining a region of interest. On the other hand, most Impro plugins process tasks multi-threaded to make use of the computation power of modern multi-core CPUs. But still, the minimum processing time is more than 1 h. The software architecture of Impro can still be improved. The tasks of the pipeline run mostly separately for the whole image, leading to saving and loading the intermediate results to, or from, the hard disc. A way more

efficient design would be the single loading of an image tile and applying all imaging tasks at once. Beside the software architecture one also can improve the efficiency of the imaging pipeline using GPUs. They are better suited to apply a single task to a large number of values, e.g. all pixels of an image, than CPUs. For pattern detection, it is also worth noting neuromorphic processor units (NPU). As deep learning currently is a distinguished topic in computer science, NPUs may play an important role in the near future. To make use of the hardware, many image processing algorithms need a re-implementation though. The implementation requires a lot of effort, as standard image processing libraries currently do not fully support the hardware. Yet, the costs of the re-implementation will likely shrink in the near future, as computing on GPUs becomes more and more popular. The development will allow very quick and efficient image processing and will most likely reduce the running time, to a few minutes or less, even for WSIs. The fast computation is a key requirement to integrate the computer-aided analysis of WSIs into routine diagnostics.

This work demonstrates that the processing of WSIs and the detection of CD30⁺ cells in complete cHL tissue sections is possible. For the set of 35 WSIs, the pipeline detected more than 400,000 CD30⁺ cells. The large number of cells makes the need for a computer-aided approach evident. For comparison, the manual cell labeling used for the validation took about 15 - 60 minutes per image, but only 10 % of the image tiles were annotated. The validation also showed a high accuracy of the Impro pipeline. For the manually annotated WSIs the pipeline achieved a total precision of 84 % and a sensitivity of 95 %. The precision is the ratio of detected cells to the actual number of cells in the image. The sensitivity can be interpreted as the accuracy of the detected cell objects. Only a small fraction of detected cell objects were false positive.

The comparison of cell count and cell morphology descriptors gives additional information to characterize the tissue sections. The three diseases differed in the CD30⁺ cell numbers and cell morphology distribution. While NScHL and lymphadenitis cases can be separated based on these properties, MCcHL has a high variety. Overall, the cell morphology and number is more similar in MCcHL and NScHL compared to lymphadenitis. This was expected because lymphadenitis has a different disease pattern. In cHL, the

population of CD30⁺ mainly consists of HRS cells, which most likely have a differentiated behavior compared to the activated lymphocytes in lymphadenitis. Surprisingly, the area size distributions of cHL and lymphadenitis have similar maxima. Also cHL-CD30⁺ cells show an increased average area size compared to lymphadenitis. Most cells have an area size of 187.5 μm^2 in all three image sets. Either a large percentage of HRS cells are of the same size of activated lymphocytes or cHL cases also contain a fraction of activated lymphocytes. The first hypothesis is supported by the fact that HRS cells in most cases originate from precursor B lymphocytes. Additionally, the detected CD30⁺ objects are based on 2D cross sections of actual 3D objects, this may also influence the observed area sizes.

The statistical data already give valuable information of the disease pattern. Modeling the lymphoma at the system level also needs to include relations between single cells. Cell-cell communication plays an important role in the tumor micro environment. In cell graphs, the communication is modeled indirectly. Edges represent the spatial closeness and thus connect potentially communicating cells. Here, graphs represent an efficient data structure that allows a quick exploration of the local neighborhood. The nearest neighborhood analysis performed by JS is an example. The analysis highlights statistically over- or underrepresented morphology classes of CD30⁺-nearest-neighbor-pairs. The results illustrate differences for the three diagnoses not only in the distribution of the morphology descriptors, but also in the preferences of CD30⁺ cells to neighbor cells of a certain morphology class. For example, *small, round* cell profiles highly prefer *small, round* cell profiles as a direct neighbor.

A more global view to the cell graphs is achieved by community structures. The malignant cells are distributed unevenly in the tissue section and form, especially in NScHL, dense cell clusters. Communities, calculated by clique percolation, reveal the underlying hierarchical structure of the graph. Up to now, the computed communities are only based on the cell graph's topology. Additional cell properties like the morphology may allow to determine more specific communities that are also grounded on biological, functional properties. One interesting aspect for future work is the influence of the lymph node structure on the formation of CD30⁺ cell clusters. The micro environment,

including cell-cell communication, is another influence.

The presented approach focuses on the examination of CD30⁺ cells, but cell graphs can easily be extended to other cell types. Especially in cHL, where malignant cells do not form a solid tumor, the micro-environment is of interest. The lymph node tissue consists of a large number of immune cells, and malignant cells need a strategy to survive surrounded by an environment that is specialized in recognizing and destroying potentially harmful particles or cells. A large pool of routinely used antibodies exists to differentiate lymphocytes and other immune cells. The object detection needs to be adapted according to the altered immune staining.

Multi-stained images, e.g. by fluorescence microscopy, allow the recognition of different cell types in a single tissue section. The corresponding cell graph includes the malignant tumor cells embedded in their micro-environment. Local interactions of the malignant cells can be examined and the interplay of tumor and surrounding tissue becomes visible. Another possibility to extend cell graphs in the future are 3D data similar to the approach described in [85]. The unit disk graph formalism can be transferred to 3D space allowing for a less restricted view on the development of cHL.

At the current state, the computer-aided approach demonstrates the possibility to perform an automated cell recognition on whole slide images of cHL to support the diagnosis with additional statistical information. For the application in routine pathology, some criteria are not yet fully met. First, the efficiency of the image processing pipeline needs to be improved in the future allowing for a quick response to incoming patient cases. Second, further investigations can find new correlations between cell graph properties and disease progression, which may help to characterize diseases in more detail and to assist pathologists to make a diagnosis.

Chapter 6

Supplement

Listing S1: Imaging pipeline for WSIs of cHL lymph node sections.

```
## TEMPLATE/TEST FILE FOR impro PIPELINE
MainApp;Timestamp:name=StartTime
## add a run extension (multiple runs):
MainApp;ImproRun:improRunExtension=rezidiv
## create an ImproImage from svb file
ImproImageViewerPlugin;CreateImproImage
MainApp;Timestamp:name=TimeImproImageCreated
## compute ROI based on layer 3 and 2
ImproImageMultiResClusteringPlugin;IdentifyROI:minLayer=3;
    maxLayer=2;saveROI=true;onlyProcessROI=true;noImageOutput=true;
    fragmentFilterSizes=64,64,32,16;
    trainingDirectory=~improWorkspace/mult_res_classes/;
MainApp;Timestamp:name=ROICalculated
## perform color deconvolution:
ColorDeconvolutionPlugin;UnmixColor:minLayer=3;
    maxLayer=0;useNormalization=false;minimumNormalizationThreshold=255;
    colorVectorFile=$WORKSPACE$/color_stains.txt;
    subtractChannels=0,0,0,0,0,1,0,0,0;
    subtractChannelFactors=1.0,1.0,1.0,1.0,1.0,5.0,1.0,1.0,1.0
MainApp;Timestamp:name=TimeColorDeconvolutionDone
## split image tiles including border for cellprofiler (ROI only)
ImproCellProfilerAdapter;SplitImage:tileSize=1024;imgType=CD30000;
    border=100;useROI=true;createSubfolder=false;
    outputDir=$WORKSPACE$/tmpImages/
MainApp;Timestamp:name=SplittingGridTilesEnded
## run cellprofiler task / store cell objects in database
```

```

ImproCellProfilerAdapter;IdentifyCellObjects:
  pipelineFile=$WORKSPACE$/pipeline_odysseus_gcb2014.cp;
  dir=$WORKSPACE$/tmpImages/;useBatchFile=true;numberOfProcesses=8
## prepare database
MainApp;Timestamp:name=PrepareDatabaseStarts
ImproCellProfilerAdapter;PrepareDatabase
MainApp;Timestamp:name=PrepareDatabaseEnded
## remove duplicate cells
MainApp;Timestamp:name=RemoveDuplicateCellsStarts
ImproCellProfilerAdapter;RemoveDuplicateCells:threshX=20;threshY=20
MainApp;Timestamp:name=RemoveDuplicateCellsEnded
## graphs
MainApp;Timestamp:name=BeforeGraphExport
GraphPlugin;ExportCellsGraph:prepare_db_current=true;
  threshold_distance=700;create_heatmaps=false;default_statistics=true;
  igp_distribution_properties=true;extended_statistics=false;
  output_format=CSV,GML
MainApp;Timestamp:name=AfterGraphExport
## graphs
MainApp;Timestamp:name=BeforeCCCR0IGraphExport
GraphPlugin;CreateReducedCCCGraphs:create_heatmaps=true;default_statistics=true;
  extended_statistics=false;igp_distribution_properties=true;output_format=CSV;
  rgtcat_save_reduced_graphs=false;
MainApp;Timestamp:name=AfterCCCR0IGraphExport
MainApp;Timestamp:name=EndTime
## exit
MainApp;ExitProgram

```

Table S1: Edge count in CD30⁺ cell graphs in absolute numbers

Diagnosis	Average edge #	± Std. dev.	Minimum	Maximum
LA	15,595.55	29,270.59	313	102,081
MCcHL	251,409.67	304,677.49	12,135	1,017,703
NScHL	173,350.17	244,358.10	30,177	919,348
ALL	150,533.40	243,452.46	313	1,017,703

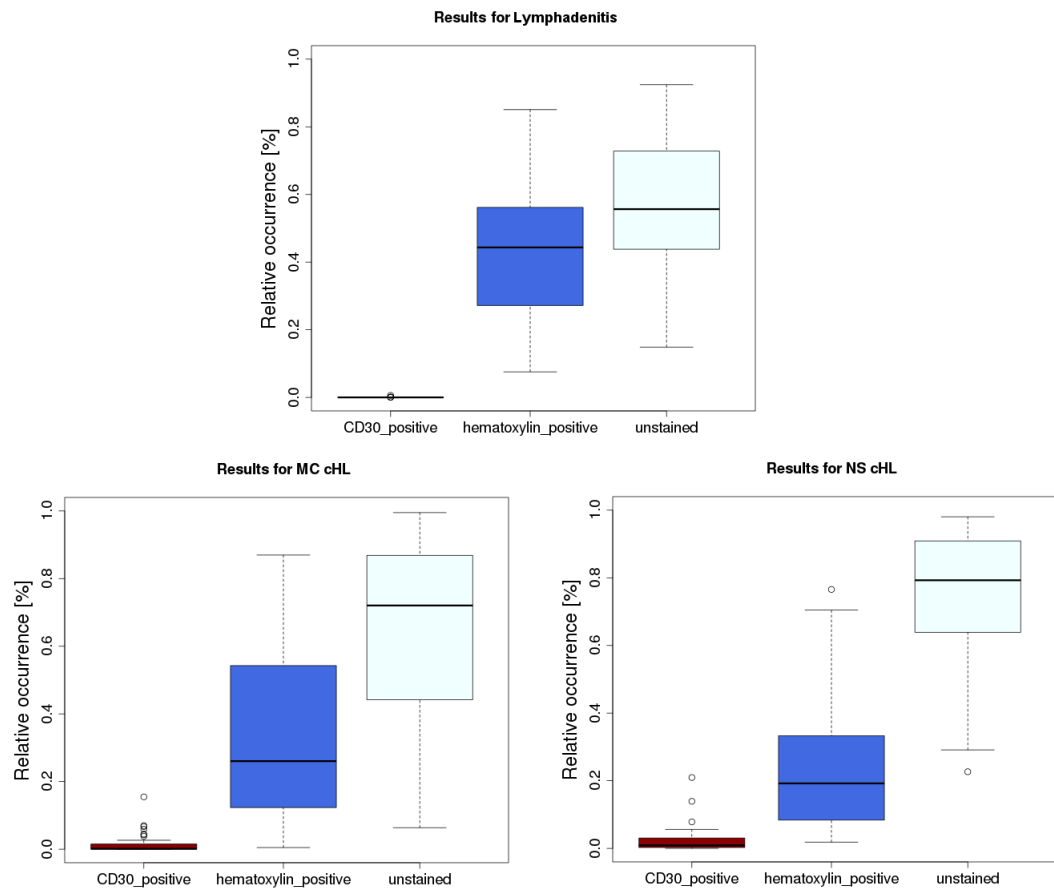


Figure S1: Pixel classification and quantification Each pixel were assigned to one of the three classes CD30_positive, hematoxylin_positive or unstained. In cHL cases the amount of CD30_positive pixels were increased. Compared to the other diagnoses, NScHL images consisted of more unstained pixels. Unstained pixels can be found in connective tissue and sclerotic bands. The latter are frequent in NScHL and may explain the high amount of unstained pixels.

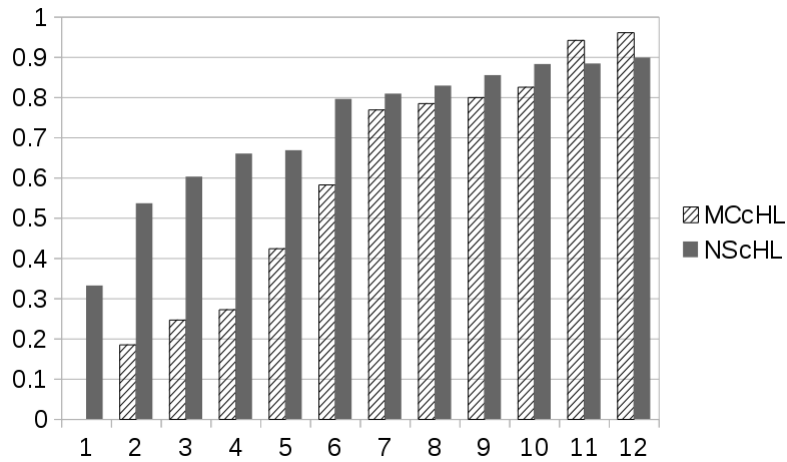


Figure S2: Percentage of image tiles with a high cell density having a high cell density neighborhood The twelve images of the MCcHL images set and the twelve images of the NScHL image set are depicted. On the y-axis the fraction of high density image tiles are plotted, having at least one high density image tile within their local neighborhood. In NScHL, this fraction is on average higher than in MCcHL. Ten out of twelve cases have 60 % and more high cell density image tiles neighbored by other high cell density tiles. Only half images of the MCcHL image set have similarly high fraction. Four of the MCcHL images have a low amount of high cell density image tiles located near high cell density image tiles. Here, most image tiles with a high cell density are isolated. A high number of isolated high density image tiles indicate that the cells do not form big cluster. Big cluster of densely packed cells would reach over multiple image tiles.

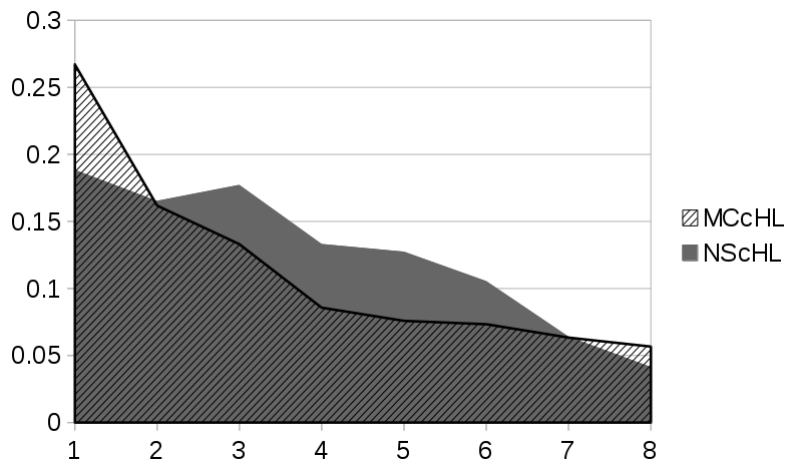


Figure S3: 8-neighborhood of high cell density image tiles The amount of high cell density image tiles that have x high cell density image tiles in their 8-neighborhood. The MCcHL image set contains more high cell density image tiles neighbored by a single high cell density image tile. In NScHL the relative amount of three up to six neighboring high cell density image tiles is increased compared to the MCcHL image set.

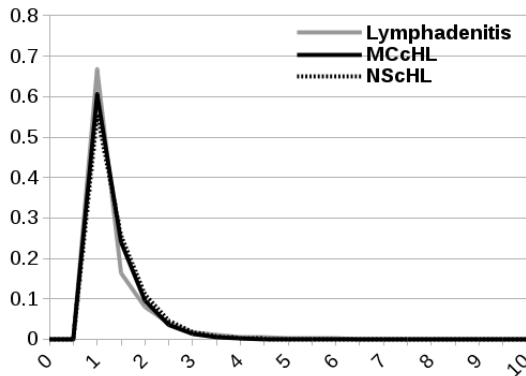


Figure S4: Morphology descriptor compactness The compactness computed by CellProfiler in relative numbers.

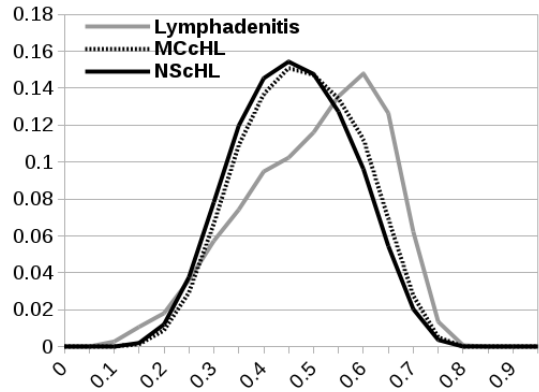


Figure S5: Morphology descriptor extent The extent computed by CellProfiler in relative numbers.

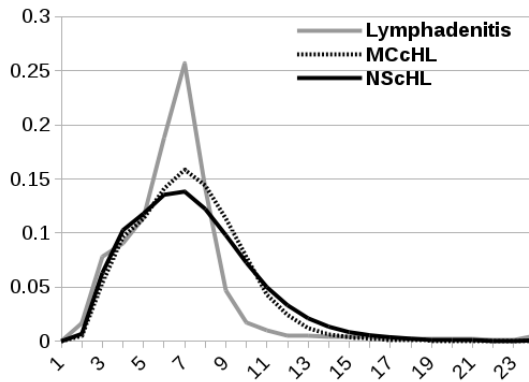


Figure S6: Morphology descriptor mean radius The mean radius computed by CellProfiler in relative numbers.

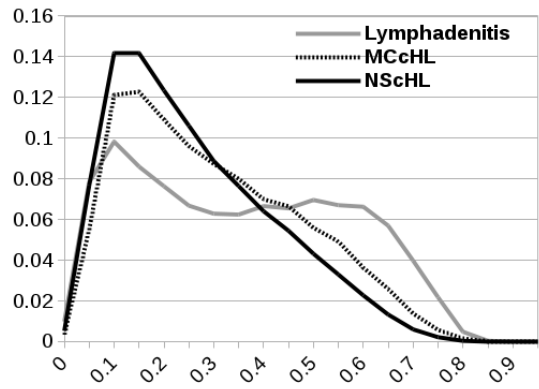


Figure S7: Morphology descriptor form factor The form factor computed by CellProfiler in relative numbers.

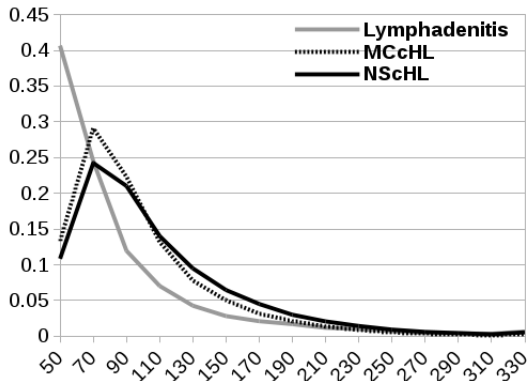


Figure S8: Morphology descriptor major axis length The major axis length computed by CellProfiler in relative numbers.

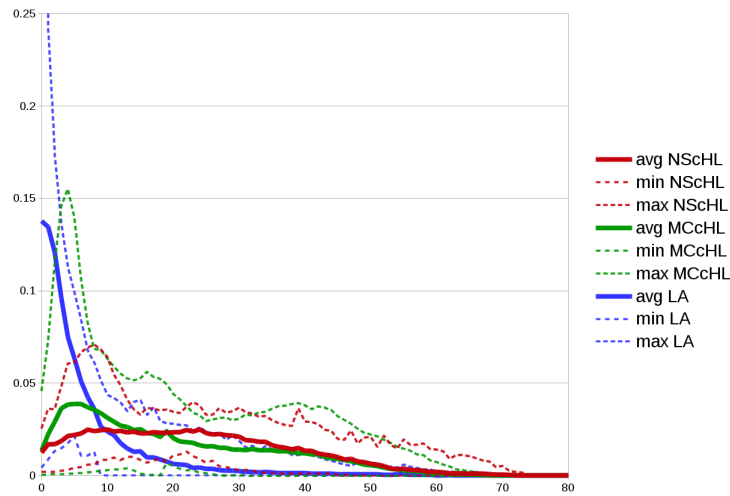


Figure S9: Vertex degree distribution: CD30 image set The average vertex degree distributions for the three diagnoses lymphadenitis, MCcHL, and NScHL. The dotted lines mark the minimal and maximal occurrences in the image set. Overall CD30⁺ cells tend to have a low vertex degree in lymphadenitis. Here, the majority of CD30⁺ cells have a vertex degree of less than 10. The highest average vertex degree was found in NScHL, a large fraction of CD30⁺ cells have a vertex degree between 10 and 50. The MCcHL vertex degree distribution is in between the two other diagnoses. Vertex degrees of 5 to 10 are observed mostly, but high degrees of 20 and higher are more likely in MCcHL than in lymphadenitis. The maximum curves illustrate the high variability in the image sets. In single cases the vertex distribution highly overlaps with distribution of an other diagnosis.

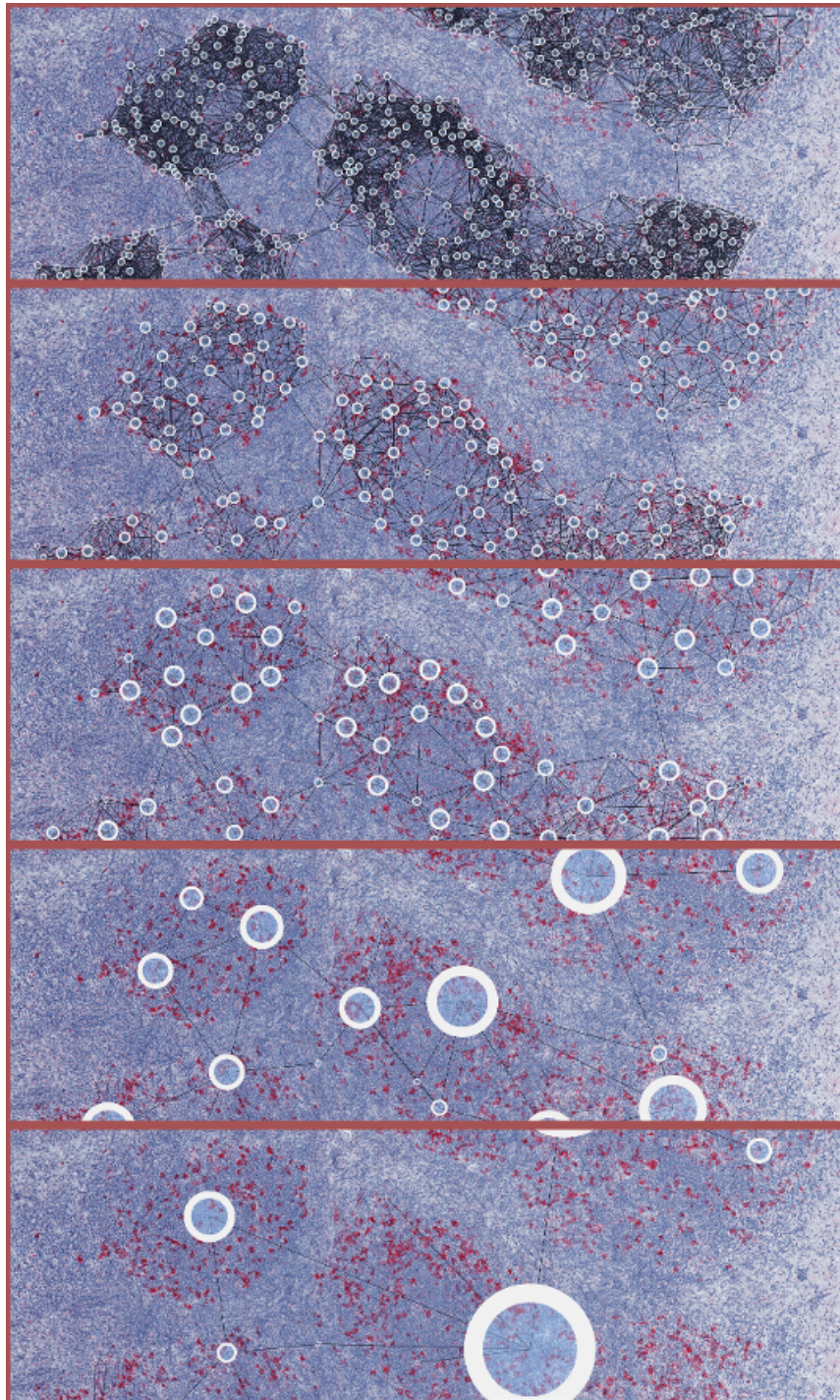


Figure S10: k-clique contraction graph levels The original graph, first row, is step-wise reduced. The vertex size scales with the number of member in the community represented by the vertex. Edges for the reduced graphs are calculated by single linkage.

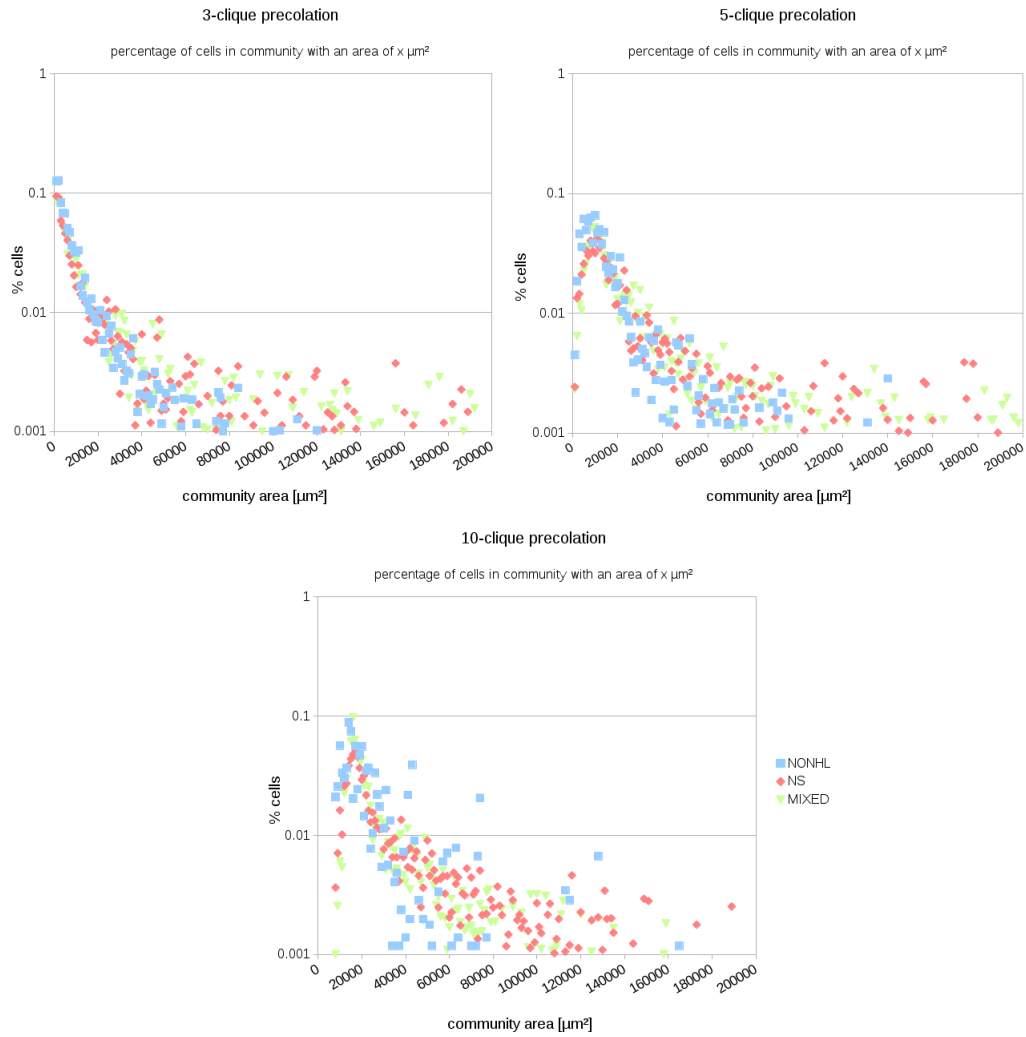


Figure S11: Community area The calculation of the community area is based on Delaunay triangulation (B). The resulting calculated area is always convex and some triangles calculated by the Delaunay triangulation do not reflect the actual area of the community. The area is post-processed by deleting all triangles with an edge that exceeds the edge threshold of the original unit disk graph. The resulting community area is depicted in C.

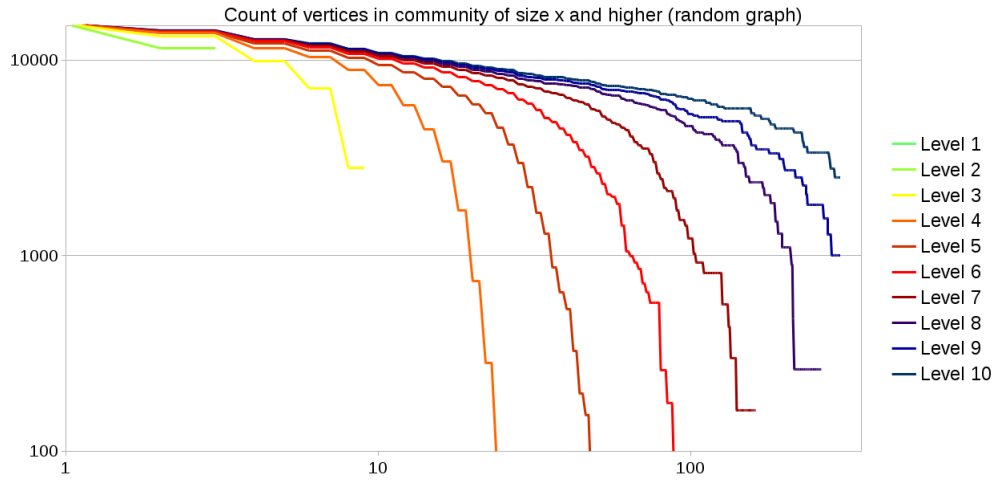


Figure S12: Clique contraction random graph The plot depicts ten reduction steps of a random geometric graph for the clique contraction method. The y axis shows the count of vertices in a community of x or higher. With increasing reduction level, the size of the communities raises.

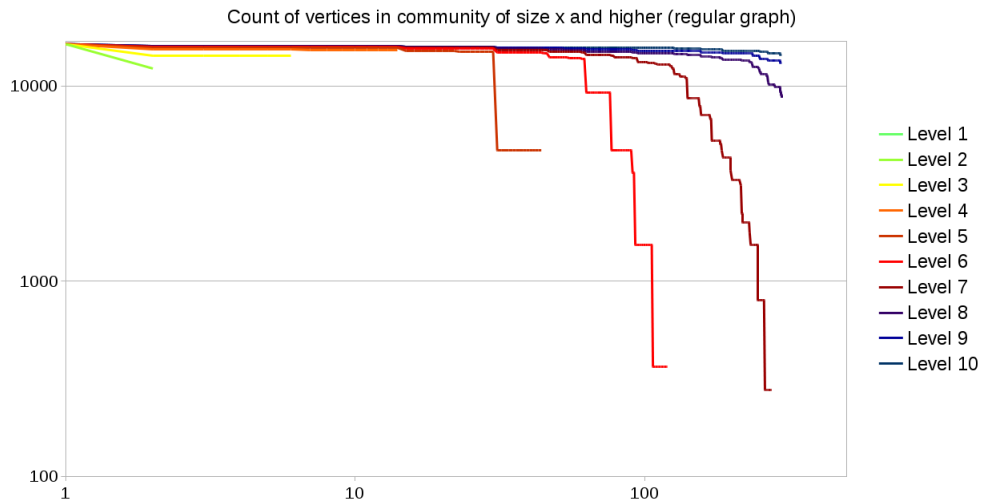


Figure S13: Clique contraction regular graph The plot depicts ten reduction steps of a random geometric graph for the clique contraction method. The y axis shows the count of vertices in a community of x or higher. Compared to cell graphs and random geometric graphs, most vertices are part of communities of the same size. The plot lines proceed nearly without changes and then drop down to zero within a narrow x margin.

List of Figures

2.1	Lymph node structure	5
2.2	Medicinal image sources	8
3.1	Imaging pipeline	13
3.2	Aperio SVS image format	16
3.3	Resolution of the four SVS image layers	17
3.4	Minimum distance to mean clustering	20
3.5	Random geometric graph	28
3.6	Overlapping community structure	30
3.7	Clique graph creation	31
4.1	General structure of Impro	36
4.2	Graphical user interface (GUI) of Impro	37
4.3	CD30 pipeline	41
4.4	ImproImage: internal image format	42
4.5	Different image properties displayed in the ImageViewer plugin	44
4.6	Results of the cell detection pipeline	45
4.7	Class signatures for the multi-resolution clustering	47
4.8	Multi-resolution clustering pipeline steps	48
4.9	Artifacts in WSIs	48
4.10	Region of interest	49
4.11	GUI of the ColorDeconvolutionPlugin	51
4.12	Output of the color deconvolution step	51

4.13	Normalization in Impro	53
4.14	Image border for cell detection	55
4.15	CellProfilerAdapter overview	56
4.16	Object detection with automatic thresholds provided by CellProfiler	59
4.17	Thresholding and fill whole operation	62
4.18	Cell count	63
4.19	Cell density per image tile	63
4.20	Correlation of cell count and cell density	65
4.21	Cell density in cHL	69
4.22	Ovals with specified eccentricity	71
4.23	Morphology descriptor area	72
4.24	Morphology descriptor maximum Feret diameter	72
4.25	Morphology descriptor eccentricity	73
4.26	Morphology descriptor solidity	73
4.27	Fractions of cell classes	75
4.28	Neighborhood: All diagnoses	77
4.29	Neighborhood: Lymphadenitis	77
4.30	Neighborhood: MCcHL	77
4.31	Neighborhood: NScHL	77
4.32	Nearest neighbor pairs small_round	78
4.33	Cell class maps of the NScHL image 6286	80
4.34	Pipeline validation and manual annotation	82
4.35	Pipeline validation for a single WSI	85
4.36	CD30 ⁺ cell graph with distance threshold 175 μ m	88
4.37	Graphical user interface of the Graph plugin	89
4.38	Sample sections of CD30 ⁺ cell graphs for all three diagnoses	93
4.39	Relative edges per vertex count	94
4.40	High and low edge count comparison	95
4.41	Vertex degree distribution of lymphadenitis and MCcHL	97
4.42	Vertex degree distribution of NScHL cases	98

4.43	Gamma parameter overview	101
4.44	Communities in NScHL cell graph	104
4.45	Community counts for the three diagnoses	105
4.46	Average percentage of cells in a community of size x	106
4.47	Average percentage of cells in a community with an area of $x \mu\text{m}^2$	107
4.48	Communities gained by clique contraction	109
4.49	Community counts 3-clique contraction	110
4.50	Communities gained by clique contraction	111
S1	Pixel classification and quantification	120
S2	Percentage of image tiles with a high cell density having a high cell density neighborhood	121
S3	8-neighborhood of high cell density image tiles	121
S4	Morphology descriptor compactness	122
S5	Morphology descriptor extent	122
S6	Morphology descriptor mean radius	122
S7	Morphology descriptor form factor	122
S8	Morphology descriptor major axis length	123
S9	Vertex degree distribution: CD30 image set	123
S10	k-clique contraction graph levels	124
S11	Community area	125
S12	Clique contraction random graph	126
S13	Clique contraction regular graph	126

List of Tables

3.1	Downsample rates of the SVS image layers	16
3.2	Default absorption matrix for a H&E staining from Fiji	18
3.3	Absorption matrix for new fuch sine and hematoxylin	19
3.4	Common cell shape descriptors	23
4.1	Overview of the plugins related to image processing	38
4.2	Overview of the plugins for statistical analysis and graph-based methods	39
4.3	Relative size of an image for different tile sizes and a fixed border of 100 px	55
4.4	The CellProfiler modules and parameters used for the object detection . .	57
4.5	The ratio of the major axis and the minor axis depending on the eccentricity	71
4.6	The morphological cell profile classes	74
4.7	Properties of CD30 ⁺ WSIs calculated by the <i>Statistics</i> plugin in Impro . .	81
4.8	Validation results of cell detection.	85
4.9	Edge count per vertex in CD30 ⁺ cell graphs	92
S1	Edge count in CD30 ⁺ cell graphs in absolute numbers	119

List of Abbreviations

Abbreviation	Term
ASCII	American Standard Code for Information Interchange
CCD	charged-coupled device
CD	cluster of differentiation
cHL	classical Hodgkin lymphoma
CLM	conventional light microscopy
COI	class of interest
CPU	central processing unit
CSV	comma separated values
CT	computer tomography
DAB	3'-Diaminobenzidine
DCU	deep cortical unit
DNA	deoxyribonucleic acid
FN	false negative
FP	false positive
GB	gigabyte
GML	Graph Modelling Language
GPU	graphics processing unit

Abbreviation	Term
GraphML	Graph Markup Language
GUI	graphical user interface
H&E	hematoxylin and eosin
HL	Hodgkin lymphoma
HRS cell	Hodgkin and Reed-Sternberg cell
IHC	immunohistochemistry
I/O	input, output
JAI	java advanced imaging
LA	lymphadenitis
MC cHL	mixed cellularity classical Hodgkin lymphoma
MCT	maximum correlation thresholding
mmp	microns per pixel
MoG	Mixture of Gaussian
MRI	magnetic resonance imaging
MRT	magnetic resonance tomography
NPU	neuromorphic processor unit
NS cHL	nodular sclerosis classical Hodgkin lymphoma
OME	Open Microscopy Project
PC	personal computer
PET	positron emission tomography
<i>px</i>	pixel
RGB	red, green, blue
RNA	ribonucleic acid
ROI	region of interest

Abbreviation	Term
SQL	Structured Query Language
svn	Subversion
TGF	Trivial Graph Format
TIF	Tagged Image File
TP	true positive
US	United States
WHO	World Health Organization
WSI	whole slide image

List of Authors

Abbreviation	Author
AS	Alexander Schmitz
JS	Jennifer Scheidel
TM	Timothy Mason
TS	Tim Schäfer

Bibliography

- [1] A. M. Noone, N. Howlader, M. Krapcho, D. Miller, A. Brest, M. Yu, J. Ruhl, Z. Tatalovich, A. Mariotto, D. R. Lewis, H. S. Chen, *et al.*, “SEER Cancer Statistics Review (CSR), 1975-2015.” https://seer.cancer.gov/csr/1975_2015/, 2018. Accessed: 2018-04-01.
- [2] S. Al-Janabi, A. Huisman, and P. J. Van Diest, “Digital pathology: current status and future perspectives,” *Histopathology*, vol. 61, no. 1, pp. 1–9, 2012.
- [3] R. S. Weinstein, K. J. Bloom, and L. S. Rozek, “Telepathology and the networking of pathology diagnostic services.,” *Archives of Pathology & Laboratory Medicine*, vol. 111, no. 7, pp. 646–652, 1987.
- [4] M. Veta, J. P. Pluim, P. J. van Diest, and M. A. Viergever, “Breast cancer histopathology image analysis: A review,” *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 5, pp. 1400–1411, 2014.
- [5] Y. Xu, J.-Y. Zhu, I. Eric, C. Chang, M. Lai, and Z. Tu, “Weakly supervised histopathology cancer image segmentation and classification,” *Medical Image Analysis*, vol. 18, no. 3, pp. 591–604, 2014.
- [6] H. Schäfer, T. Schäfer, J. Ackermann, N. Dichter, C. Döring, S. Hartmann, M.-L. Hansmann, and I. Koch, “CD30 cell graphs of Hodgkin lymphoma are not scale-free – an image analysis approach,” *Bioinformatics*, vol. 32, no. 1, pp. 122–129, 2015.
- [7] M. Hons and M. Sixt, “The lymph node filter revealed,” *Nature immunology*, vol. 16, no. 4, pp. 338–340, 2015.

- [8] T. Hodgkin, "On some morbid appearances of the absorbent glands and spleen.," 1832.
- [9] R. T. Hoppe, P. T. Mauch, J. O. Armitage, V. M. D. Diehl, and L. M. Weiss, *Hodgkin Lymphoma*. Lippincott Williams and Wilkins, 2 ed., 2007.
- [10] D. Aldinucci, A. Gloghini, A. Pinto, R. De Filippi, and A. Carbone, "The classical Hodgkin's lymphoma microenvironment and its role in promoting tumour growth and immune escape," *The Journal of Pathology*, vol. 221, no. 3, pp. 248–263, 2010.
- [11] L. Fass, "Imaging and cancer: a review," *Molecular Oncology*, vol. 2, no. 2, pp. 115–152, 2008.
- [12] L. Pantanowitz, "Digital images and the future of digital pathology," *Journal of Pathology Informatics*, vol. 1, no. 1, p. 15, 2010.
- [13] J. Gilbertson and Y. Yagi, "Histology, imaging and new diagnostic work-flows in pathology," *Diagnostic Pathology*, vol. 3, no. Suppl 1, p. S14, 2008.
- [14] S. Al-Janabi, A. Huisman, M. Nap, R. Clarijs, and P. J. van Diest, "Whole slide images as a platform for initial diagnostics in histopathology in a medium-sized routine laboratory," *Journal of Clinical Pathology*, vol. 65, no. 12, pp. 1107–1111, 2012.
- [15] J. Ho, A. V. Parwani, D. M. Jukic, Y. Yagi, L. Anthony, and J. R. Gilbertson, "Use of whole slide imaging in surgical pathology quality assurance: design and pilot validation studies," *Human Pathology*, vol. 37, no. 3, pp. 322–331, 2006.
- [16] L. J. Weinstein, J. I. Epstein, D. Edlow, and W. H. Westra, "Static image analysis of skin specimens: the application of telepathology to frozen section evaluation," *Human Pathology*, vol. 28, no. 1, pp. 30–35, 1997.
- [17] J. Ordi, P. Castillo, A. Saco, M. del Pino, O. Ordi, L. Rodríguez-Carunchio, and J. Ramírez, "Validation of whole slide imaging in the primary diagnosis of gynaecological pathology in a University Hospital," *Journal of Clinical Pathology*, vol. 68, no. 1, pp. 33–39, 2015.

- [18] S. Al-Janabi, A. Huisman, S. Willems, and P. Van Diest, “Digital slide images for primary diagnostics in breast pathology: a feasibility study,” *Human Pathology*, vol. 43, no. 12, pp. 2318–2325, 2012.
- [19] C. Massone, H. P. Soyer, G. P. Lozzi, A. Di Stefani, B. Leinweber, G. Gabler, M. Asgari, R. Boldrini, L. Bugatti, V. Canzonieri, *et al.*, “Feasibility and diagnostic agreement in teledermatopathology using a virtual slide system,” *Human Pathology*, vol. 38, no. 4, pp. 546–554, 2007.
- [20] D. Treanor, B. D. Gallas, M. A. Gavrielides, and S. M. Hewitt, “Evaluating whole slide imaging: A working group opportunity,” *Journal of Pathology Informatics*, vol. 6, 2015.
- [21] L. Pantanowitz, J. H. Sinard, W. H. Henricks, L. A. Fatheree, A. B. Carter, L. Con-tis, B. A. Beckwith, A. J. Evans, A. Lal, and A. V. Parwani, “Validating whole slide imaging for diagnostic purposes in pathology: guideline from the College of American Pathologists Pathology and Laboratory Quality Center,” *Archives of Pathology and Laboratory Medicine*, vol. 137, no. 12, pp. 1710–1722, 2013.
- [22] K. Saeger, K. Schlüns, T. Schrader, and P. Hufnagl, “The virtual microscope for routine pathology based on a PACS system for 6 Gb images,” in *International Congress Series*, vol. 1256, pp. 299–304, Elsevier, 2003.
- [23] R. K. Kumar, G. M. Velan, S. O. Korell, M. Kandara, F. R. Dee, and D. Wakefield, “Virtual microscopy for learning and assessment in pathology,” *The Journal of Pathology*, vol. 204, no. 5, pp. 613–618, 2004.
- [24] J. Feit, L. Matyska, V. Ulman, L. Hejtmánek, H. Jedličková, M. Ježová, M. Moulis, and V. Feitová, “Virtual microscope interface to high resolution histological images,” *Diagnostic Pathology*, vol. 3, pp. 1–3, 2008.
- [25] A. Afework, M. D. Beynon, F. Bustamante, S. Cho, A. Demarzo, R. Ferreira, R. Miller, M. Silberman, J. Saltz, A. Sussman, *et al.*, “Digital dynamic telepathology—the Virtual Microscope,” in *Proceedings of the AMIA Symposium*, p. 912, American Medical Informatics Association, 1998.

- [26] A. D. Belsare and M. M. Mushrif, “Histopathological image analysis using image processing techniques: An overview,” *Signal & Image Processing*, vol. 3, no. 4, p. 23, 2012.
- [27] T. J. Fuchs and J. M. Buhmann, “Computational pathology: Challenges and promises for tissue analysis,” *Computerized Medical Imaging and Graphics*, vol. 35, no. 7, pp. 515–530, 2011.
- [28] W. Burger and M. J. Burge, *Digital image processing: an algorithmic introduction using Java*. Springer Science & Business Media, 2008.
- [29] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 3 ed., 2007.
- [30] K. R. Castleman, *Digital Image Processing*. Prentice Hall, 2 ed., 1996.
- [31] A. C. Ruifrok and D. A. Johnston, “Quantification of histochemical staining by color deconvolution.,” *Analytical and Quantitative Cytology and Histology*, vol. 23, no. 4, pp. 291–299, 2001.
- [32] D. Carey, V. Wijayathunga, A. Bulpitt, and D. Treanor, “A Novel Approach for the Colour Deconvolution of Multiple Histological Stains,” in *Proceedings of the 19th Conference of Medical Image Understanding and Analysis*, pp. 156–162, BMVA, 2015.
- [33] A. Rabinovich, S. Agarwal, C. Laris, J. H. Price, and S. J. Belongie, “Unsupervised color decomposition of histologically stained tissue samples,” in *Advances in Neural Information Processing Systems*, p. None, 2003.
- [34] A. E. Rizzardi, A. T. Johnson, R. I. Vogel, S. E. Pambuccian, J. Henriksen, A. Skubititz, G. J. Metzger, and S. C. Schmechel, “Quantitative comparison of immunohistochemical staining measured by digital image analysis versus pathologist visual scoring,” *Diagnostic Pathology*, vol. 7, no. 42, pp. 1596–7, 2012.

- [35] S. Kothari, J. H. Phan, T. H. Stokes, and M. D. Wang, “Pathology imaging informatics for quantitative analysis of whole-slide images,” *Journal of the American Medical Informatics Association*, vol. 20, no. 6, pp. 1099–1108, 2013.
- [36] T. Schäfer, H. Schäfer, A. Schmitz, J. Ackermann, N. Dichter, C. Döring, S. Hartmann, M.-L. Hansmann, and I. Koch, “Image database analysis of Hodgkin lymphoma,” *Computational Biology and Chemistry*, vol. 46, pp. 1–7, 2013.
- [37] A. Materka and M. Strzelecki, “Texture analysis methods – a review,” in *COST B11 Report, Brussels*, 1998.
- [38] A. Gertych, N. Ing, Z. Ma, T. J. Fuchs, S. Salman, S. Mohanty, S. Bhele, A. Velásquez-Vacca, M. B. Amin, and B. S. Knudsen, “Machine learning approaches to analyze histological images of tissues from radical prostatectomies,” *Computerized Medical Imaging and Graphics*, vol. 46, pp. 197–208, 2015.
- [39] M. F. A. Fauzi, H. N. Gokozan, B. Elder, V. K. Puduvalli, C. R. Pierson, J. J. Otero, and M. N. Gurcan, “A multi-resolution textural approach to diagnostic neuropathology reporting,” *Journal of Neuro-Oncology*, vol. 124, no. 3, pp. 393–402, 2015.
- [40] B. Karaçalı, A. P. Vamvakidou, and A. Tözeren, “Automated recognition of cell phenotypes in histology images based on membrane- and nuclei-targeting biomarkers,” *BMC Medical Imaging*, vol. 7, no. 1, p. 7, 2007.
- [41] P. Belhomme, S. Toralba, B. Plancoulaine, M. Oger, M. N. Gurcan, and C. Bor-Angelier, “Heterogeneity assessment of histological tissue sections in whole slide images,” *Computerized Medical Imaging and Graphics*, vol. 42, pp. 51–55, 2015.
- [42] Y. Guo, X. Xu, Y. Wang, Y. Wang, S. Xia, and Z. Yang, “An image processing pipeline to detect and segment nuclei in muscle fiber microscopic images,” *Microscopy Research and Technique*, vol. 77, no. 8, pp. 547–559, 2014.
- [43] E. Bernardis and X. Y. Stella, “Pop out many small structures from a very large microscopic image,” *Medical Image Analysis*, vol. 15, no. 5, pp. 690–707, 2011.

- [44] J. P. Vink, M. Van Leeuwen, C. Van Deurzen, and G. De Haan, “Efficient nucleus detector in histopathology images,” *Journal of Microscopy*, vol. 249, no. 2, pp. 124–135, 2013.
- [45] D. Zink, A. H. Fischer, and J. A. Nickerson, “Nuclear structure in cancer cells,” *Nature Reviews Cancer*, vol. 4, no. 9, pp. 677–687, 2004.
- [46] P. Dey, “Cancer nucleus: morphology and beyond,” *Diagnostic Cytopathology*, vol. 38, no. 5, pp. 382–390, 2010.
- [47] R. Kumar, R. Srivastava, and S. Srivastava, “Detection and Classification of Cancer from Microscopic Biopsy Images Using Clinically Significant and Biologically Interpretable Features,” *Journal of Medical Engineering*, vol. 2015, 2015.
- [48] S. Kothari, J. H. Phan, A. N. Young, and M. D. Wang, “Histological image classification using biologically interpretable shape-based features,” *BMC Medical Imaging*, vol. 13, no. 1, p. 9, 2013.
- [49] D. Reinhard, *Graphentheorie*. Springer, 2006.
- [50] M. Newman, *Networks: an introduction*. OUP Oxford, 2010.
- [51] B. N. Clark, C. J. Colbourn, and D. S. Johnson, “Unit disk graphs,” *Discrete Mathematics*, vol. 86, no. 1-3, pp. 165–177, 1990.
- [52] Y. Fernandess and D. Malkhi, “K-clustering in wireless ad hoc networks,” in *Proceedings of the second ACM international workshop on Principles of mobile computing*, pp. 31–37, ACM, 2002.
- [53] M. V. Marathe, H. Breu, H. B. Hunt, S. S. Ravi, and D. J. Rosenkrantz, “Simple heuristics for unit disk graphs,” *Networks*, vol. 25, no. 2, pp. 59–68, 1995.
- [54] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, “Uncovering the overlapping community structure of complex networks in nature and society,” *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

- [55] S. Fortunato, “Community detection in graphs,” *Physics Reports*, vol. 486, no. 3, pp. 75–174, 2010.
- [56] I. Derényi, G. Palla, and T. Vicsek, “Clique percolation in random networks,” *Physical Review Letters*, vol. 94, no. 16, p. 160202, 2005.
- [57] F. Reid, A. McDaid, and N. Hurley, “Percolation computation in complex networks,” in *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 274–281, IEEE, 2012.
- [58] M. R. Fellows, J. Guo, C. Komusiewicz, R. Niedermeier, and J. Uhlmann, “Graph-based data clustering with overlaps,” *Discrete Optimization*, vol. 8, no. 1, pp. 2–17, 2011.
- [59] A. Dumitrescu and J. Pach, “Minimum clique partition in unit disk graphs,” *Graphs and Combinatorics*, vol. 27, no. 3, pp. 399–411, 2011.
- [60] W. Cui, H. Zhou, H. Qu, P. C. Wong, and X. Li, “Geometry-based edge clustering for graph visualization,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1277–1284, 2008.
- [61] M. E. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Physical Review E*, vol. 69, no. 2, p. 026113, 2004.
- [62] A. Lázár, D. Ábel, and T. Vicsek, “Modularity measure of networks with overlapping communities,” *EPL (Europhysics Letters)*, vol. 90, no. 18001, pp. 1–6, 2010.
- [63] P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient graph-based image segmentation,” *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.
- [64] C. Gunduz, B. Yener, and S. H. Gultekin, “The cell graphs of cancer,” *Bioinformatics*, vol. 20, no. 1, pp. i145–i151, 2004.
- [65] C. C. Bilgin, P. Bullough, G. E. Plopper, and B. Yener, “ECM-aware cell-graph mining for bone tissue modeling and classification,” *Data Mining and Knowledge Discovery*, vol. 20, no. 3, pp. 416–438, 2010.

- [66] B. Oztan, K. R. Shubert, C. S. Bjornsson, G. E. Plopper, and B. Yener, “Biologically-driven cell-graphs for breast tissue grading,” in *2013 IEEE 10th International Symposium on Biomedical Imaging (ISBI)*, pp. 137–140, IEEE, 2013.
- [67] M. R. Lamprecht, D. M. Sabatini, A. E. Carpenter, *et al.*, “CellProfilerTM: free, versatile software for automated biological image analysis,” *Biotechniques*, vol. 42, no. 1, p. 71, 2007.
- [68] G. Clark, H. Stockinger, R. Balderas, M. C. van Zelm, H. Zola, D. Hart, and P. Engel, “Nomenclature of CD molecules from the Tenth Human Leucocyte Differentiation Antigen Workshop,” *Clinical & Translational Immunology*, vol. 5, no. 1, p. e57, 2016.
- [69] M. C. Díaz-Ramos, P. Engel, and R. Bastos, “Towards a comprehensive human cell-surface immunome database,” *Immunology Letters*, vol. 134, no. 2, pp. 183–187, 2011.
- [70] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, *et al.*, “Fiji: an open-source platform for biological-image analysis,” *Nature Methods*, vol. 9, no. 7, pp. 676–682, 2012.
- [71] “Cython.” www.cython.org/. Accessed: 2016-12-14.
- [72] M. Linkert, C. T. Rueden, C. Allan, J.-M. Burel, W. Moore, A. Patterson, B. Lorange, J. Moore, C. Neves, D. MacDonald, *et al.*, “Metadata matters: access to image data in the real world,” *The Journal of Cell Biology*, vol. 189, no. 5, pp. 777–782, 2010.
- [73] I. G. Goldberg, C. Allan, J.-M. Burel, D. Creager, A. Falconi, H. Hochheiser, J. Johnston, J. Mellen, P. K. Sorger, and J. R. Swedlow, “The Open Microscopy Environment (OME) Data Model and XML file: open tools for informatics and quantitative analysis in biological imaging,” *Genome Biology*, vol. 6, no. 5, p. R47, 2005.
- [74] C. A. Schneider, W. S. Rasband, and K. W. Eliceiri, “NIH Image to ImageJ: 25 years of image analysis,” *Nature Methods*, vol. 9, no. 7, p. 671, 2012.

- [75] “CellProfiler Documentation.” <http://d1zypm9ayga15t.cloudfront.net/CPmanual/index.html>. Accessed: 2016-09-12.
- [76] “Gml: A portable graph file format.” <https://www.fim.uni-passau.de/fileadmin/files/lehrstuhl/brandenburg/projekte/gml/gml-technical-report.pdf>. Accessed: 2016-09-26.
- [77] R. Gandhi, A. Mishra, and S. Parthasarathy, “Minimizing broadcast latency and redundancy in ad hoc networks,” *IEEE/ACM Transactions on Networking*, vol. 16, no. 4, pp. 840–851, 2008.
- [78] B. Adamcsek, G. Palla, I. J. Farkas, I. Derényi, and T. Vicsek, “CFinder: locating cliques and overlapping modules in biological networks,” *Bioinformatics*, vol. 22, no. 8, pp. 1021–1023, 2006.
- [79] “Bio-Formats.” www.openmicroscopy.org/. Accessed: 2017-03-14.
- [80] J. Moore, M. Linkert, C. Blackburn, M. Carroll, R. K. Ferguson, H. Flynn, K. Gillen, R. Leigh, S. Li, D. Lindner, *et al.*, “OMERO and Bio-Formats 5: flexible access to large bioimaging datasets at scale,” in *SPIE Medical Imaging*, pp. 941307–941307, International Society for Optics and Photonics, 2015.
- [81] J. Monaco, J. Hipp, D. Lucas, S. Smith, U. Balis, and A. Madabhushi, “Image segmentation with implicit color standardization using spatially constrained expectation maximization: Detection of nuclei,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 365–372, Springer, 2012.
- [82] C. L. Willard-Mack, “Normal structure, function, and histology of lymph nodes,” *Toxicologic Pathology*, vol. 34, no. 5, pp. 409–424, 2006.
- [83] H. G. Drexler, G. Gaedicke, M. S. Lok, V. Diehl, and J. Minowada, “Hodgkin’s disease derived cell lines HDLM-2 and L-428: comparison of morphology, immunological and isoenzyme profiles,” *Leukemia Research*, vol. 10, no. 5, pp. 487–500, 1986.

- [84] B. Rengstl, S. Newrzela, T. Heinrich, C. Weiser, F. B. Thalheimer, F. Schmid, K. Warner, S. Hartmann, T. Schroeder, R. Küppers, *et al.*, “Incomplete cytokinesis and re-fusion of small mononucleated Hodgkin cells lead to giant multinucleated Reed–Sternberg cells,” *Proceedings of the National Academy of Sciences*, vol. 110, no. 51, pp. 20729–20734, 2013.
- [85] A. Schmitz, S. C. Fischer, C. Mattheyer, F. Pampaloni, and E. H. Stelzer, “Multi-scale image analysis reveals structural heterogeneity of the cell microenvironment in homotypic spheroids,” *Scientific Reports*, vol. 7, no. 43693, 2017.

Lebenslauf

Persönliche Daten

Name Hendrik Schäfer

Begutachtete Publikationen

H. Schäfer*, T. Schäfer*, J. Ackermann, N. Dichter, C. Döring, S. Hartmann, M.-L. Hansmann, and I. Koch. CD30 cell graphs of Hodgkin lymphoma are not scale-free – an image analysis approach. *Bioinformatics*, 32(1):122–129, 2016.

T. Schäfer*, **H. Schäfer***, A. Schmitz*, J. Ackermann, N. Dichter, C. Döring, S. Hartmann, M.-L. Hansmann, and I. Koch. Image database analysis of Hodgkin Lymphoma. *Computational Biology and Chemistry*, 46:1–7, 2013.

Konferenzvorträge

H. Schäfer. Automated cell detection in whole slide images of classical Hodgkin lymphoma. Konferenzvortrag auf der *99. Jahrestagung der Deutschen Gesellschaft für Pathologie e. V.*, Frankfurt, 2015.

H. Schäfer. Automated Image Analysis of Hodgkin Lymphoma. Satellite Workshop-Vortrag auf der *German Conference on Bioinformatics*, Jena, 2012.

H. Schäfer. Influence of Alternative Splicing Events to Secondary Structure of Proteins. Study Group Session-Vortrag auf der *Molecular Life Sciences*, Frankfurt, 2011.

*Geteilte Erstautorenschaft.

*Geteilte Erstautorenschaft.

Poster

H. Schäfer, T. Schäfer, J. Ackermann, C. Döring, S. Hartmann, M.-L. Hansmann, and I. Koch. Graph-based analysis of CD30⁺ cell distributions in Hodgkin lymphoma. *German Conference on Bioinformatics*, Dortmund, 2015.

H. Schäfer, T. Schäfer, J. Ackermann, C. Döring, S. Hartmann, M.-L. Hansmann and I. Koch. Hodgkin Lymphoma – From Image Analysis to Cell Graphs. *German Conference on Bioinformatics*, Bielefeld, 2014.

H. Schäfer, T. Schäfer, J. Ackermann, N. Dichter, C. Döring, S. Hartmann, M.-L. Hansmann, and I. Koch. Statistical Analysis of Cell Properties and Distribution in Hodgkin Lymphoma Whole Slide Images. *Molecular Life Science*, Frankfurt, 2013.

H. Schäfer, T. Schäfer, J. Ackermann, C. Döring, S. Hartmann, M.-L. Hansmann, and I. Koch. Spatial distribution of cells in Hodgkin lymphoma. *German Conference on Bioinformatics*, Göttingen, 2013.

A. Schmitz, T. Schäfer, H. Schäfer, C. Döring, J. Ackermann, N. Dichter, S. Hartmann, M.-L. Hansmann, and I. Koch. Automated image analysis of Hodgkin lymphoma. *German Conference on Bioinformatics*, Jena, 2012.

H. Schäfer, and I. Koch. A pipeline to explore alternative splicing events. *German Conference on Bioinformatics*, Weihenstephan, 2011