New Computational Tools for Sample Purification and Early-Stage Data Processing in High-Resolution Cryo-Electron Microscopy

Dissertation
for the award of the degree
"Doctor rerum naturalium"
of the Georg-August-Universität Göttingen

within the doctoral degree programme
"Biomolecules: Structure - Function - Dynamics"
of the Georg-August University School of Science (GAUSS)

submitted by

Lukas Schulte

from Eckernförde, Germany

Göttingen, 2018

Thesis Committee

Prof. Dr. Holger Stark
Department of Structural Dynamics, Max-Planck-Institute for biophysical Chemistry

Prof. Dr. Ralf Ficner
Department of Molecular Structural Biology, Institute for Microbiology and Genetics, Georg-August-University Göttingen

Prof. Dr. Jörg Enderlein
III. Physical Institute: Biophysics/Complex Systems, Georg-August-University Göttingen

Members of the Examination Board

Referee: Prof. Dr. Holger Stark
Department of Structural Dynamics, Max-Planck-Institute for biophysical Chemistry

Second Referee: Prof. Dr. Ralf Ficner
Department of Molecular Structural Biology, Institute for Microbiology and Genetics, Georg-August-University Göttingen

Further Members of the Examination Board

Prof. Dr. Jörg Enderlein
III. Physical Institute: Biophysics/Complex Systems, Georg-August-University Göttingen

Prof. Dr. Kai Tittmann
Bioanalytics, Albrecht-von-Haller Institute, Georg-August-University Göttingen, and Molecular Enzymology, Max-Planck-Institute for biophysical Chemistry

Prof. Dr. Markus Zweckstetter
Structure Determination of Proteins using NMR, Max-Planck-Institute for biophysical Chemistry and Translational Structural Biology in Dementia, German Center for Neurodegenerative Diseases

Dr. Michael Habeck
Statistical Inverse Problems in Biophysics, Max-Planck-Institute for biophysical Chemistry, and Computational Structural Biology, Institute for Mathematical Stochastics, Georg-August-University Göttingen

Date of the oral Examination: September 14th, 2018

Affidavit

I hereby declare that this PhD thesis “New Computational Tools for Sample Purification and Early-Stage Data Processing in High-Resolution Cryo-Electron Microscopy” has been written independently with no other aids or sources than quoted. This thesis (wholly or in part) has not been submitted elsewhere for any academic award or qualification.

Lukas Schulte

"In God we trust, all others bring data."

William Edwards Deming

Contents

Affidavit **i**

Acknowledgements **xiii**

Abstract **xv**

1 Introduction **1**

1.1 High-resolution 3D Structure Determination of Macromolecular Machines . . . 1
  1.1.1 Methods for Conformational Dynamics Analysis . . . 2
  1.1.2 Structural Methods . . . 2
    1.1.2.1 NMR . . . 2
    1.1.2.2 X-ray Crystallography . . . 3
    1.1.2.3 Cryo-Electron Microscopy . . . 3
1.2 Basics of Single particle Cryo-EM . . . 3
  1.2.1 Overview . . . 3
  1.2.2 Instrumentation . . . 4
  1.2.3 Image Formation . . . 8
  1.2.4 Sample Preparation . . . 11
    1.2.4.1 Negative Staining Preparation . . . 11
    1.2.4.2 Cryo Preparation . . . 12
  1.2.5 TEM Images . . . 13
    1.2.5.1 Pixel Defects . . . 13
    1.2.5.2 Image Contaminations and Defects . . . 13
  1.2.6 Computational Image Processing . . . 14
    1.2.6.1 Frame Alignment . . . 14
    1.2.6.2 CTF Correction . . . 15
    1.2.6.3 Particle Picking . . . 15
    1.2.6.4 Preprocessing . . . 16
    1.2.6.5 Alignment . . . 16
    1.2.6.6 Classification . . . 19
    1.2.6.7 Angular Assignment . . . 21
    1.2.6.8 3D Reconstruction . . . 22
    1.2.6.9 3D Refinement . . . 24
    1.2.6.10 Resolution Determination . . . 24
    1.2.6.11 Accessing Conformational Information . . . 27
    1.2.6.12 Postprocessing . . . 28
1.3 Sample Optimization in Structural Biology . . . 28
  1.3.1 Basics of Density Gradient Centrifugation . . . 29
  1.3.2 Theory of Rate-Zonal Separation . . . 30
  1.3.3 Particle Distribution around the Peak . . . 31
  1.3.4 Challenges in Obtaining Run Parameters . . . 31
1.4 Scientific Software Development . . . 33

1.4.1 Parallelization 33
1.4.1.1 Requirements for Parallelization 33
1.4.1.2 Farming 34
1.4.1.3 GPU Parallelization Programming 34
1.5 Current Challenges in Cryo-EM Data Processing 34
1.5.1 Imaging Automation and Amount of Data 34
1.5.2 Computation Speed and Live Processing 35
1.5.3 Self-Referential Processing- the Need for Data Cleaning 35
1.5.4 Processing Software landscape 36
1.6 Aim of the Work 36

2 Materials and Methods **39**
2.1 Mathematical Tools for Image Processing 39
2.1.1 Image Statistics and Normalization 39
2.1.1.1 Mean 39
2.1.1.2 Variance and Standard Deviation 40
2.1.1.3 Normal Distribution 40
2.1.1.4 Image Normalization 41
2.1.2 Image Contrast Enhancement 42
2.1.2.1 Pixel Value Histogram Equalization 42
2.1.2.2 Coarsing 43
2.1.3 Fourier Transformations 44
2.1.3.1 Non-Square Images in Fourier Space 45
2.1.3.2 Convolution 45
2.1.3.3 Correlation 46
2.1.3.4 Power Spectra 46
2.1.4 Real Space Image Filters 46
2.1.4.1 Boxcar Filter 46
2.1.4.2 Difference Of Gaussians Filter 47
2.1.5 Fourier Space Image Filters 48
2.1.5.1 Lowpass Filter 48
2.1.5.2 Highpass Filter 48
2.1.5.3 Bandpass Filter 49
2.2 Machine Learning: Support Vector Machines 49
2.2.1 Pattern Classification 50
2.2.2 Separating Hyperplanes 50
2.2.3 Soft Margin Classification 52
2.2.3.1 Feature Selection for SVM Classification 53
2.2.3.2 Hyperparameter Tuning and Evaluating SVM Model Performance 54
2.3 Mathematical Tools for Density Gradient Centrifugation Simulation . . 57
2.3.1 Calculation of Medium Density and Viscosity 57
2.3.1.1 Sucrose 57
2.3.1.2 Glycerol 58
2.3.2 Calculation of Sedimentation Value Distributions 58
2.3.2.1 Prediction of Sedimentation Coefficients based on Molecular Weight 59
2.3.3 Extracting Intensity Distributions from SDS Gels 59
2.3.4 Fitting Intensity Distributions from SDS Gels 59
2.4 Programming Languages and Frameworks 60
2.4.1 C++ 60

2.4.1.1 Used C++ Libraries 60
2.4.2 R 61
2.5 Hard- and Software 62
2.5.1 UML 62
2.5.1.1 Static Class Diagrams 62
2.5.1.2 Activity Diagram 63
2.6 The COW 63
2.6.1 History and Concept 63
2.6.2 CowLib 64
2.6.2.1 Parameter Objects 65
2.6.2.2 Data Objects 65
2.6.2.3 Logics 65
2.6.2.4 Processors 65
2.6.2.5 IO Objects 65
2.6.3 CowPicker 66
2.6.4 CowEyes 66
2.6.4.1 Project Management 67
2.6.4.2 Viewer 68
2.6.4.3 Visual Programming 68
2.6.4.4 CowPlug Ecosystem 69
2.6.5 CowServer 69
2.7 Preparation of Test Data 69
2.7.1 Test Data for CowGraCE 69
2.7.1.1 Considered Rotors and Gradients 70
2.7.2 Test Data for the MQC 70

3 Results 71
3.1 CowGraCE 71
3.1.1 Mean Svedberg Value Prediction Algorithm 72
3.1.1.1 User Inputs 72
3.1.1.2 Initialization 73
3.1.1.3 Calculation Loop 73
3.1.2 Distribution Profile Prediction 74
3.1.3 Interpolation 74
3.1.4 Computational Complexity of the Simulation Algorithm 75
3.1.5 Simulation Result Comparisons 75
3.1.6 Prediction Tests 77
3.1.6.1 Tests on Purified Human Proteasomes 78
3.1.6.2 Tests on Realistic Purification Scenarios 80
3.1.6.3 Statistical Evaluation of Prediction Results 82
3.1.7 Run Condition Optimization 85
3.1.7.1 Invariable User Inputs 85
3.1.7.2 Parameters to Optimize Based on User Constraints . . 86
3.1.7.3 Precalculations and Initialization 86
3.1.7.4 Positional Interpolation and Pre-Check 86
3.1.7.5 Optimization Score Calculation 87
3.1.7.6 Looping and Abortion Criteria 87
3.1.7.7 Computational Complexity 88
3.1.7.8 Simulation Optimization Results 88
3.1.8 CowGrace Frontend 90
3.1.8.1 Rotor Management 90

3.1.8.2 Result Loading . . . 90
3.1.8.3 Simulation Mode . . . 92
3.1.8.4 Optimization Mode . . . 92
3.1.8.5 Result Display . . . 93
3.1.9 CowGraCE Backend . . . 93
3.2 Micrograph Quality Checker . . . 96
3.2.1 MQC Backend . . . 96
3.2.1.1 Main Window . . . 96
3.2.1.2 Live Processing . . . 98
3.2.1.3 Calculation Jobs . . . 99
3.2.1.4 User Interaction . . . 100
3.2.2 MQC Frontend . . . 100
3.2.3 Image Metrics . . . 101
3.2.4 SVM Data Analysis . . . 102
3.2.4.1 Datasets Chosen for Modeling Analysis . . . 104
3.2.4.2 Data Preprocessing . . . 105
3.2.4.3 Hyperparameter Tuning and Performance Evaluation 105
3.2.4.4 Learning Curves . . . 108
3.2.4.5 Computational Complexity of the MQC Workflow . . 108
3.2.4.6 Generalization Capabilities . . . 110

4 Discussion **113**
4.1 CowGraCE . . . 113
4.1.1 Implementation and Simulation Speed . . . 113
4.1.2 Simulation Results and Optimization Capatibilities . . . 113
4.2 Micrograph Quality Checker and COW . . . 114
4.2.1 Implementation, Processing Workflow and Speed . . . 114
4.2.2 SVM Model Performance on Chosen Datasets . . . 115
4.2.3 Generalization of Models . . . 115
4.2.4 Proposed Live Training Implementation . . . 116
4.2.5 COW Framework and Integration of MQC . . . 117
4.3 Outlook . . . 118

A Additional Data for CowGraCE **121**

B Additional Data for the MQC **123**

Bibliography **127**

Curriculum Vitae **143**

List of Figures

1.1 Maps deposited in the EMDB by Year since 2000 4
1.2 Simplied Cryo-EM Workflow 5
1.3 Simplified Architecture of a Modern Electron Microscope 7
1.4 Possible Electron Interactions with a thin Specimen 8
1.5 1D CTF Functions at Different Defoci 11
1.6 Difference between Negative Stain and Cryo Preparation 12
1.7 Classes of Micrograph Contaminations and Defects 14
1.8 Example of Particle Picking 16
1.9 Effect of Preprocessing on Extracted Particles 17
1.10 Illustration of Image Alignment 18
1.11 Illustration of Reference-Free Alignment 19
1.12 Example of Eigenimages 20
1.13 Classification and Class Sums 22
1.14 Optimal Euler Angle Distribution 23
1.15 Comparison of a Structure at different Resolutions 25
1.16 Detail Comparison of a Structure at different Resolutions 26
1.17 Global FSC Curve and Local Resolution Map 27
1.18 Effects of Sharpening on a Refined Structure 29
1.19 Principle of rate-zonal Separation 32
1.20 Maps deposited in the EMDB since 2000 and their resolutions 35

2.1 Gaussian Distributions with a Mean of 10 and different Standard Deviations 41
2.2 Histogram Equalization Effects 43
2.3 Effects of Coarsing 44
2.4 Power Spectrum of a Sample Image 47
2.5 Effects of Real Space Filters on a Sample Image 48
2.6 Illustration of Binary Fourier Space Filters 49
2.7 Binary Classification Problem with possible Hyperplanes 51
2.8 Soft Margin SVM Classification of a non linearly separable Data Set . . 53
2.9 Linear Separation in Feature Space 54
2.10 Typical Learning Curve 56
2.11 Intensity Distribution Extraction from a SDS Gel 60
2.12 Modules of UML 2 Diagrams 63
2.13 Static UML2 Class Diagram of the CowLib 64
2.14 User Interface of CowPicker 66
2.15 User Interface of CowEyes 67
2.16 CowEyes Viewers 68

3.1 Logos of the Softwares developed in this Thesis 71
3.2 Activity Diagram for the Peak Svedberg Value Prediction Algorithm . 72
3.3 Dependence of CowGraCE Calculation Speed on Number of Fractions 76
3.4 Competing Softwares 76

3.5 Comparison between CowGraCE and COMPASS for Peak Svedberg Values at Varying Gradients for three Rotors 77
3.6 Difference between CowGraCE and COMPASS for Peak Svedberg Values at Varying Gradients and Rotors at the last Fraction 78
3.7 Purified h20S Proteasome Density Gradient Centrifugation Results for three Different Conditions 79
3.8 Human and Drosophila 20S Proteasome Density Gradient Centrifugation Results at Different Purification Steps 81
3.9 Unknown Protein Complex Density Gradient Centrifugation Results . 82
3.10 Comparison of Simulated Gaussian Distributions to the Experimental Data 83
3.11 Boxplots of CowGraCE and COMPASS Prediction Performance Metrics 84
3.12 Activity Diagram of the Run Condition Optimization Algorithm 85
3.13 Runtime of the Optimization Algorithm depending on Input Size . . . 89
3.14 Found Good Conditions for single S Values depending on Number of Conditions 89
3.15 Maximum Peak Distance and Optimization Score for two Optimization Simulations with Different Number of Conditions 90
3.16 CowGraCE Main Window 91
3.17 Simple Algorithm for Finding the closest Runtime 93
3.18 Result Display for a simulation with set Parameters 94
3.19 UML 2 Diagram of CowGraCE Architecture 95
3.20 Overview of the Micrograph Quality Checker 97
3.21 UML2 Class Diagram of the MQC Backend. 97
3.22 Comparison of original Image to contrast-enhanced Image 99
3.23 MQC GUI 102
3.24 Statistical Image Metric Acquisition Procedure 103
3.25 Effects of the Boxcar and DoG Filter Procedure 104
3.26 Illustration of the SVM Classification Workflow 106
3.27 Performance Metric Comparison for SVM Models Trained on the Three Datasets 107
3.28 Learning Curves of the SVM Models 109
3.29 Computational Speed of the Two Parts of the MQC Data Processing . . 110
3.30 Performance Metric Comparison for the Differently Trained SVM Models on the two Datasets. 112

4.1 Proposed Live SVM Model Training Algorithm 116
4.2 Structure of the COW 118

A.1 Determination of Mean and Standard Deviation from COMPASS . . . 121

B.1 Exemplary Good and Bad Micrographs and their Power Spectrum with Ctf Fit from Dataset A 123
B.2 Exemplary Good and Bad Micrographs and their Power Spectrum with Ctf Fit from Dataset B 124
B.3 Exemplary Results of Hyperparameter Tuning 126

List of Tables

2.1 Confusion Matrix for Two Classes 55
2.2 Used C++ Libraries 61
2.3 Used C++ Tool Chains 62
2.4 Available Ultracentrifuge Swing-Out Rotors in the Department of Structural Dynamics 70

3.1 Comparison of Distribution Parameters for the Tests on Isolated Proteasomes 79
3.2 Comparison of Distribution Parameters for the Tests on Purification Gradients 80
3.3 Calculated Prediction Metric Results 84
3.4 Overview of Datasets Selected for Machine Learning Analysis 104
3.5 Confusion Matrix of Intra-Rater Consistency 104
3.6 Removed Metrics from the Datasets based on Correlation Values . . . 105
3.7 Prediction Performance of the Hypertuned Models on the Testing Data of each Dataset 107
3.8 Average Performance Metrics of the Hypertuned Models of the Three Datasets 107
3.9 Prediction Performance of SVM Models Trained on one Dataset on the other Datasets 110
3.10 Prediction Performance of the Models Trained on the Joined Dataset on the Test Data Split by Dataset Origin 111
3.11 Prediction Performance of the Models Trained on the Equal Number Joined Dataset on the Test Data Split by Dataset Origin 111
3.12 Prediction Performance Metrics of the Hypertuned Models of the Joined Datasets on the Datasets split by Origin 111

A.1 Supported Sucrose Gradients in the BioComp GradientMaker. 121
A.2 Supported Glycerol Gradients in the BioComp GradientMaker 121
A.3 Predicted and experimentally determined S Values 122

B.1 Correlation Matrix 1 from Dataset A 124
B.2 Correlation Matrix 2 from Dataset A 124
B.3 Correlation Matrix 1 from Dataset B 125
B.4 Correlation Matrix 2 from Dataset B 125
B.5 Correlation Matrix 1 from Dataset C 125
B.6 Correlation Matrix 2 from Dataset C 125
B.7 Determined Optimal Hyperparameters for the Datasets in the Different Independent Tuning Runs 125

Acknowledgements

First and foremost, I would like to thank Prof. Dr. Holger Stark for his supervision, guidance, enthusiasm and for giving me the opportunity to conduct my PhD research in his group. Always having an open door (literally) and always taking the time to assist with any problems one might have, he always showed his trust in his doctoral students by making sure that we had the best equipment, the best working situation and that we could develop our research interests. I couldn't have imagined a better PI when I moved to Göttingen for my PhD and it cannot be understated how much I grew as a scientist and as a person through the 4 years in the Department of Structural Dynamics.

I would like to thank Prof. Dr. Ralf Ficner and Prof. Dr. Jörg Enderlein for serving on my thesis committee, for giving valuable suggestions and input and for making the TAC meetings very enjoyable.

For all the help, guidance and discussions, I would like to express my sincere gratitude to Dr. Mario Lüttich, my direct supervisor. Through his patience, more than impressive all-around knowledge and know-how, he was and is a main factor in the method development success of this group. No matter what he was doing, he would always take the time to discuss anything and everything. His calm demeanor in the face of crashing servers, persistent compile errors, heated discussions and all other everyday dramas in the world of computers inspired me majorly.

Before he unfortunately left to start his own group, Dr. David Haselbach was a big part in making the work in the Department so enjoyable. His enthusiasm, curiosity, willingness to teach everyone the basics of EM and never-ending flow of ideas for new and improved methods was a big help, as well as the cakes he brought on a weekly basis. For valuable discussions, suggestions and beta-testing, I would like to thank Dr. Ashwin Chari, Dr. Niels Fischer, Dr. Dietmar Riedel, Dr. Boris Busche who taught me the ropes at the start of my thesis, Dr. Jan-Martin Kirves, Georg Bunzel, Sabrina Fiedler, Jan-Erik Schliep and Dr. Wen-Ti Liu. I would like to give a special thanks to Uma Lakshmi Dakshinamoorthy, who started the CowGraCE project by applying as a student assistant and being foolish enough to admit that she could program. Her enthusiasm and persistence in digging through libraries to find old centrifugation books and her contributions to the project are very much appreciated.

For all their support, love and for raising me to be a hopefully adequate scientist, I would like to thank my parents, Monika Krah-Schulte and Rolf Schulte, from the bottom of my heart. Also, I would like to express my gratitude to Leonie Herde for having accompanied me on my path in science and in life for a long time, for having dealt with my non-existent work-life balance and for always having my back. I would like to thank my wonderful girlfriend Charlotte Karnasch for her support, love and for being the complementary half-sphere to the author of this thesis in all aspects imaginable.

Last but definitely not least, Kashish Singh, Karl Bertram, Fabian Henneberg (and joining later, Cole Townsend + Alex Mehr), the people that I spent the most time in Göttingen with, were a big part of my PhD. I couldn't have done it without you guys, on a scientific as well as on a personal level. Thank you for the best office imaginable and for all the memories- office113productions will roll on.

Abstract

Large macromolecular protein complexes play an important role in the quest of understanding life at a molecular level. For this endeavor, the highly-resolved 3D structure of a large complex and its changes, which are called conformational dynamics, are of high interest due to their close relation to biological function. A popular method to gain access to this information is cryo-electron microscopy.

Here, biological samples are vitrified in their native buffer conditions and then imaged in a transmission electron microscope. The large amount of recorded noisy images, called micrographs, are then subjected to a series of sophisticated, computationally expensive algorithmic steps that include particle extraction, averaging of similar particle images, angular assignment and 3D reconstruction. While a growing amount of 3D structures determined by cryo EM reach resolutions that allow atomic model building, a large share of structures never reach that level of possible biological detail.

In this thesis, improvements to two of the problems that can limit the resolution of structures are presented: Sample quality and elimination of suboptimal micrographs. First, a new software for simulation of density gradient centrifugation experiments, which is a popular method for sample purification, is presented that can aid users in finding the right conditions to optimize their purification protocols through new algorithmic approaches. This can lead to purer and more stable samples that improve the recorded images.

Then, a new software for live processing and machine learning based classification of TEM images whose quality is judged live by a user is introduced. This software, called the Micrograph Quality Checker, is part of the COW, which is a novel single particle cryo EM image processing framework that covers the whole data analysis workflow. Elimination of unwanted data and collection of metadata in parallel to image acquisition can reduce the data overhead, improve computational speed and can also lead to higher resolution in the obtained structures due to the self-referential image processing.

Keywords: cryo-electron microscopy, single particle imaging framework, machine learning, density gradient centrifugation, micrograph quality criteria, software

List of Abbreviations

1D	One-**d**imensional
2D	Two-**d**imensional
3D	Three-**d**imensional
Acc	**Acc**uracy
API	**A**pplication **P**rogramming **I**nterface
CCC	**C**ross **C**orrelation **C**oefficient
CCD	**C**harge **C**oupled **D**evice
CM	**C**orrelation **M**atrix
CTF	**C**ontrast **T**ransfer **F**unction
CPU	**C**entral **P**rocessing **U**nit
DFT	**D**iscrete **F**ourier **T**ransform
DoG	**D**ifference **o**f **G**aussians
DQE	**D**etective **Q**uantum **E**fficiency
EM	**E**lectron **M**icroscopy
EMDB	**E**lectron **M**icroscopy **D**ata **B**ank
FNR	**F**alse **N**egative **R**atio
FPR	**F**alse **P**ositive **R**atio
FRET	**F**örster **R**esonance **E**nergy **T**ransfer
FT	**F**ourier **T**ransform
GNU	**G**NU's **N**ot **U**nix!
GPU	**G**raphical **P**rocessing **U**nit
IDE	**I**ntegrated **D**evelopment **E**nvironment
IO	**I**nput/ **O**utput
kDa	**k**ilo **Da**lton
mRNA	**m**essenger **R**ibo**N**ucleic **A**cid
MD	**M**olecular **D**ynamics
MQC	**M**icrograph **Q**uality **C**hecker
MTF	**M**odulation **T**ransfer **F**unction
MW	**M**olecular **W**eight
NMR	**N**uclear **M**agnetic **R**esonance
NPV	**Negative** **P**redictive **V**alue
OS	**O**perating **S**ystem
PCC	**P**earson **C**orrelation **C**oefficient
PCA	**P**rincipal **C**omponent **A**nalysis
PDB	**P**rotein **D**ata **B**ase
PPV	**P**ositive **P**redictive **V**alue
PSD	**P**ower **S**pectral **D**ensity
SNR	**S**ignal to **N**oise **R**atio
SVM	**S**upport **V**ector **M**achine
UML	**U**nified **M**odelling **L**anguage

Chapter 1

Introduction

1.1 High-resolution 3D Structure Determination of Macromolecular Machines

The central dogma of molecular biology [39] states that sequential genetic information is transcribed to mRNA, which is then transported out of a cell's nucleus to be translated by the ribosome to a peptide chain consisting of a combination of 21 possible amino acids in the order that is specified by the codons of the mRNA. Based on the free energy determined by the amino acid sequence[8], these chains then fold into a three dimensional structure, unassisted or assisted by chaperones[45], with four different organizational levels:

1. **Primary structure:** The linear amino acid sequence that defines a protein, first described in 1951 [155].
2. **Secondary structure:** Local structures based on stabilization by hydrogen bonds, e.g. α-helices, β-sheets[126].
3. **Tertiary structure:** The global 3D shape of a protein that is commonly stabilized by a multitude of chemical processes, e.g. separation of hydrophobic and hydrophilic domains, Cystein-Cystein disulfide bonds and salt bridges [111].
4. **Quaternary structure:** The formation of complexes by either multiple identical peptides or different ones. Also included may be cofactors, nucleid acids, metal ions and such[111].

For architecture, Louis Sullivan coined the phrase "Form follows function"[181] in 1896. For proteins, usually the reverse is claimed[169]. Undisputedly, understanding protein structure is essential to reaching a better understanding of nature's inner workings on a molecular level. The findings of structural biology have impactful implications for biotechnology, drug discovery, disease mechanism research and much more.

While it has been known for a long time that proteins were able to adopt different conformations, for a long time these structural changes were explained by the lock-and-key model[51], according to which only changes in the chemical environment would lead to transitions between different substates. Later, it became apparent that all possible conformations are in an equilibrium with one another based on the free energy landscape[61]. Conformational dynamics were shown to have important implications, including for enzyme catalysis[60] and allostery in cell signaling[25]. Elucidating the 3D structures and the dynamic nature of proteins and protein complexes can be realized with different biophysical methods that are introduced in the following.

1.1.1 Methods for Conformational Dynamics Analysis

Spectroscopical approaches such as Circular Dichroism (secondary structures), Infrared (atomic vibrational movement) and UV-Vis Spectroscopy (light-absorbing regions) rely on interaction of light with molecules to gain information about specific aspects of protein structure. However, the clear limitation is the signal integration over a large number of non-synchronized molecules which leads to averaging effects, thus resulting in only approximations of the overall dynamics. In some cases, synchronization effects can be introduced through temperature and substrate binding.

Single molecule Förster Resonance Energy Transfer (FRET) spectroscopy enables live measurements of changes in distance between a donor and an acceptor fluorophore attached to suitable amino acid residues in a protein molecule[78]. Through this, movement of protein domains can be monitored[88]. Suitable positions for probe attachment rely on preexisting structures in different conformations. Accessible through this method are therefore conformational rates.

A computational method to gain access to conformational information on very small timescales are Molecular Dynamics (MD) simulations[4]. Based on a preexisting structure, atomic movements are simulated through numerical solution of Newtonian mechanics equations for the atoms, utilizing for example force fields to calculate potential energies between them. The method was originally established in the biological sciences for simulations of protein folding[114], but is now routinely used for conformational dynamics. The drawback of this method is that specially for larger protein complexes, larger timescales are computationally expensive to simulate.

1.1.2 Structural Methods

The methods mentioned in the previous section either rely on extensive averaging or preexisting structural models to get information about a protein's conformational landscape with a good temporal resolution. Structural methods on the other hand offer a high spatial and usually a low time resolution.

1.1.2.1 NMR

In Nuclear Magnetic Resonance (NMR) spectroscopy, a sample solution is introduced to a strong magnetic field of several Tesla. The spins of isotopes with a magnetic moment resulting from an odd number of protons are then aligned with the field's direction. Application of pulsed radio waves results in a spin oscillation with a specific Larmor frequency that is dependent on a nuclei's chemical environment. A relaxation back to the aligned state and emission of the previously absorbed energy enables detection of an electric current and extraction of the Larmor frequencies to create a spectrum[103]. This, in addition to measured relaxation times, can be utilized to calculate the distance between atoms, thus accessing structural and conformational information on low time scales and high resolution[102].

Due to the combinatorial nature of spectrum interpretation, the method is mostly limited to small proteins where peak broadening and overlapping signals are limited. Usually, the upper molecular weight (MW) limit is listed at around 100 kilo Dalton (kDa). In the Protein Data Bank (PDB), 10946 atomic models determined by NMR spectroscopy were deposited as of April 2018.

1.1.2.2 X-ray Crystallography

The oldest of the structural methods, X-ray crystallography is based on protein crystals, regular crystal lattices of protein molecules, which also represents a bottleneck in the method due to the need to empirically optimize a multitude of individual crystallization parameters. A crystal is introduced into an X-ray beam that is diffracted by the outer shell electrons of the protein's atoms[46]. Structural information is then available through the characteristic diffraction patterns, which however contain no phase information. This information has to be determined separately through approaches such as molecular replacement, heavy atom substitution or usage of external phase information, for example from homologous structures.

X-ray crystallography can yield atomic resolution of small proteins, with larger protein complexes being more of a challenge, resulting in usual resolutions of 2-3Å. Different conformations can sometimes be crystallized. Time-resolved crystallography is available in case of still functional proteins in the crystal. Through crystal packing effects, differences to solution structures can occur.

As of April 2018, 122988 atomic models determined by X-ray crystallography were deposited in the PDB, by far the most out of all structural methods.

1.1.2.3 Cryo-Electron Microscopy

Electron microscopy(EM) of biological macromolecules is the youngest structural method and involves recording 2D projections of individual molecules, usually embedded in ice, which are ultimately inverse projected to yield a reconstruction of the underlying 3D structure.

Single particle cryo-EM has gained a lot of traction in recent years [10], including the Nobel Prize in Chemistry being awarded to Richard Henderson, Jacques Dubouchet and Joachim Frank for their contributions to the method[149], due to being able to reach near atomic resolution for large macromolecular complexes and has become the method of choice for a large portion of structural biologists. 2164 atomic models determined by EM have been deposited in the PDB by April 2018, with more maps being available in the Electron Microscopy Database(EMDB). Figure 1.1 shows the growing number of annually deposited maps in the EMDB over the past 18 years. An advantage over the other structural methods is the usually 1000-fold smaller sample amount requirement and the native buffer environment during imaging.

Due to the vitrification requirement to withstand radiation damage, time resolution of cryo-EM is usually not existent. A retrospective approach to conformational dynamics is made possible, though, through sorting the recorded particles into their respective structural state.

1.2 Basics of Single particle Cryo-EM

1.2.1 Overview

Single particle cryo-electron microscopy derives its name from the technique of vitrifying samples of protein complexes in low concentration in native buffer conditions

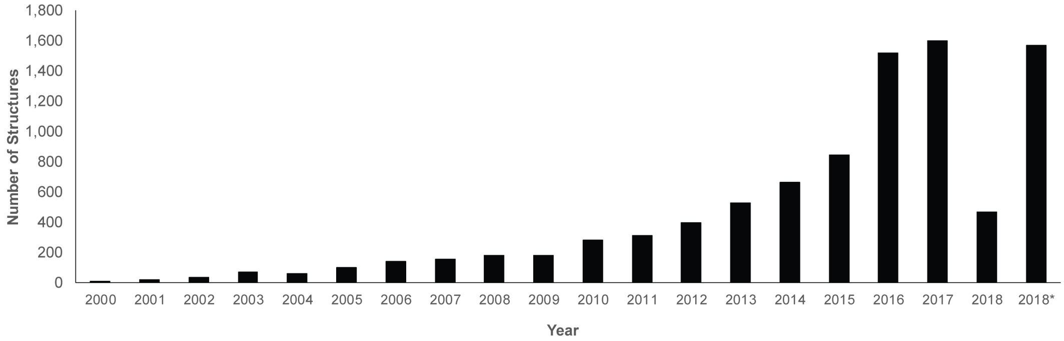


FIGURE 1.1: Maps deposited in the EMDB by year since 2000. The growing number of released structures shows the increased popularity of the method. Denoted with 2018* is the linearly interpolated value for 2018 based on the amount of maps deposited until April 2018.

to then expose them to an electron beam in an electron microscope. Ethane vitrification is essential to reduce radiation damage to the sample, but also introduces additional noise during the imaging process. Combined with the low dose electron beam, this results in digital images with a very low signal-to-noise ratio (SNR). As the image processing is based on several steps of averaging similar images, the amount of data needed is immense, resulting in a high computational cost. An alternative to vitrification is negative stain treatment of samples, which enhances contrast and therefore reduces the data demand, but limits the achievable resolution and is therefore mostly used for screening purposes only.

Figure 1.2 gives an overview about the simplified workflow of the technique. In a nutshell, 2D projections of the underlying 3D structures are recorded in the imaging process. Due to beam-induced charging effects[24] that result in slight sample movements, multiple frames with low SNR each are recorded and aligned to produce a micrograph, containing up to hundreds of particles, whose contrast transfer function (CTF) parameters are then determined. Afterwards, the areas containing particles are extracted and, after iterative averaging steps of projections with similar angles, a 3D structure is calculated through real space backprojection or Fourier reconstruction. This happens in a self-referential process that is usually repeated several times, producing structures with better resolution in each iteration whose projections can then be used as references to sort the dataset images to their closest projection angles. The higher the resolution is, the bigger the amount of different projections. After convergence of the structure(s), the resolution can be determined and if high enough resolution was reached, atomic models can then be constructed to enable interpretation of biological questions. In the following sections, the mentioned experimental and computational procedures are described in further detail.

1.2.2 Instrumentation

The concept of resolution *d* is the most important parameter when discussing structural biology techniques. It is defined as the smallest distance where structural features can be distinguished from one another. In the 19th century, visible light was

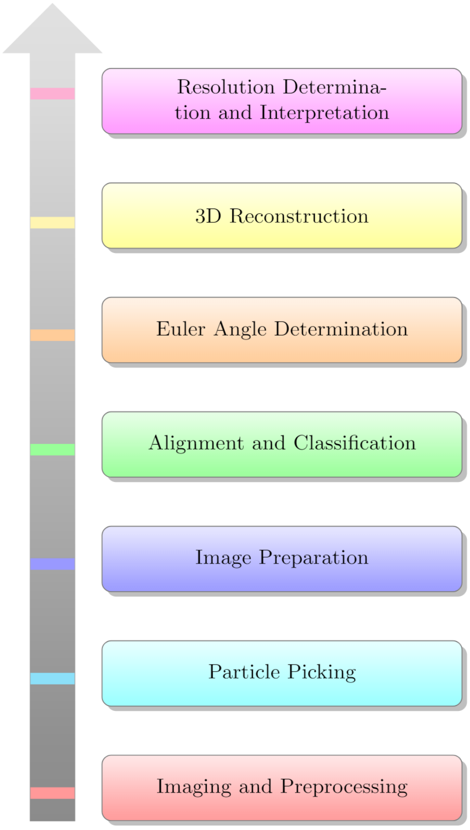

Resolution Determination and Interpretation
3D Reconstruction
Euler Angle Determination
Alignment and Classification
Image Preparation
Particle Picking
Imaging and Preprocessing

FIGURE 1.2: Simplified Cryo-EM workflow

first used to magnify biological structures. Ernst Abbe then discovered the diffraction limit that holds true for microscopes in general[1]:

$$d = \frac{\lambda}{2\, nsin(\alpha)}\,, \tag{1.1}$$

where λ is the wavelength of the particle that is used for imaging, n is the refractive index of the surrounding medium and α is the microscope objective's opening angle. This diffraction limit puts the maximum achievable resolution at $\frac{\lambda}{2}$. For visible light with wavelengths from 200-800 nm, the limit is therefore between 100-400nm, enabling imaging of a cell's inner workings up to the organelle level. In the late 20th century, the STED method[85] found a way to circumvent the diffraction limit, reaching resolutions of around 20 Å with visible light.

In the advent of quantum mechanics and the discovery of the wave-particle duality, Louis De Broglie's research in 1924 found that a wavelength can be assigned to every particle[43]:

$$\lambda = \frac{\hbar}{mv}\,, \tag{1.2}$$

where $\hbar$ is the Planck Constant, m the mass and v the velocity. This implicated that a higher resolution could be achieved in a microscope utilizing heavy particles with high velocity that would therefore have a very small wavelength. A prime candidate for this was the electron.

Shortly after an electromagnetic lens capable of focussing electrons was invented by Max Knoll, Ernst Ruska and him built the first electron microscope in 1931. First only showcasing a magnification factor of 31, the improved prototype two years later already surpassed the resolution limit of visible light. While a lot of technical advances have since been made, the main architecture of a transmission electron microscope is still comprised of an electron source, condensor lenses, a sample holder, an objective lens, projector lenses and a camera that records an image. Figure 1.3 shows the architecture of a modern TEM as used in the Department of Structural Dynamics.

The electron source used in high-end electron microscopes is a Schottky field emission gun (FEG) that produces a very coherent beam. In it, a tungsten crystal coated with zirconium dioxide is heated to 3000 °C , thus emitting electrons. Using an acceleration voltage of usually 300kV, the electrons are accelerated in an electrostatic field and are then emitted into the vacuum. Taking into account relativistic effects, the resulting wavelength of the electrons can be approximated as[12]

$$\lambda = \frac{12.3}{\sqrt{\phi + 9.78 \cdot 10^{-7} \phi^2}}, \tag{1.3}$$

where ϕ is the applied electric field. For 300 kV, the electron wavelength is thus determined as 1.96 pm. The maximum reachable resolution in modern microscopes could thus theoretically be below the 1 Å level, which leads to the conclusion that the technique is not diffraction limited.

The electron beam then passes through a system of lenses. These are matterless magnetic fields, which limits the achievable shapes and the achievable resolution due to various different aberrations that result in image blurring:

1. **2-fold Astigmatism** occurs when perpendicular propagating waves hitting a lens have different focal lengths.
2. **Spherical aberration**, also called Cs, occurs in spherical lenses that focus waves arriving closer to the optical axis differently than waves with more distance to the optical axis.
3. **Comatic aberration**, also called coma, describes a resulting cone of rays in the image plane that passed through an off-axis specimen point[144]. Off-axis alignments of the electron beam increase the comatic abberation further.

There are several ways to minimize these aberrations. Astigmatism and coma can be minimized through coma-free alignment [206]. In two of the four available microscopes in the department, a Cs corrector is used[62], which all but eliminates spherical aberration through introduction of a symmetric hexapole doublet into the electron optics.

The first lens system in a TEM are the condenser lenses, consisting of a variable amount of lenses and apertures. Here, the illumination of the specimen is controlled through adjustment of beam size and therefore the exposure electron dosage.

The specimen, on a carbon grid as described in the previous section, is positioned in a holder, called the compustage, which is inserted between the two objective lenses.

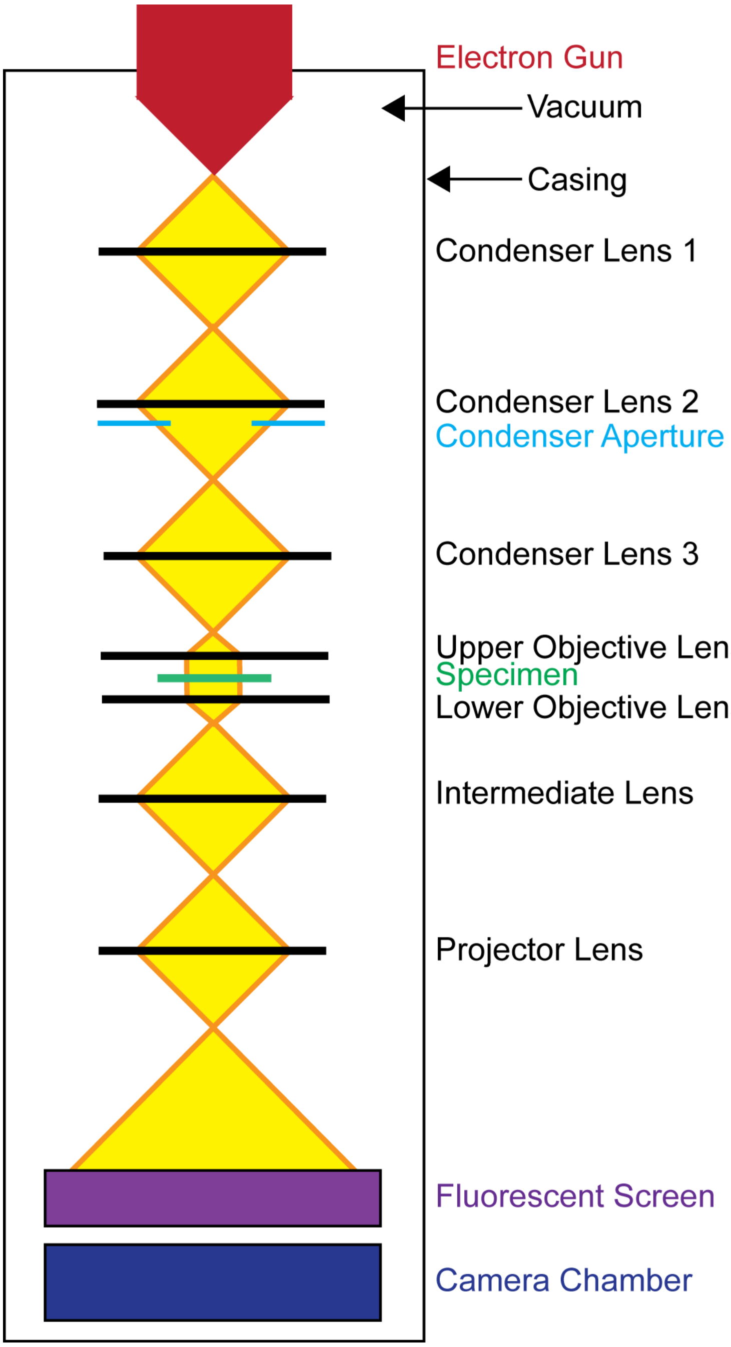


FIGURE 1.3: Simplified architecture of a modern electron Microscope. Shown in yellow is the electron beam. The number of condenser and intermediate lenses can vary between microscopes, also an additional condenser aperture and other optional parts can be used.

Through the compustage, the grid can be moved and tilted to illuminate specific areas. The incoming electrons interact with the specimen in different ways, which is described in detail in the following section.

After passing through the sample as a parallel beam, the electrons are refocussed in the intermediate lens system. The projector lens then magnifies the image.

On the removable fluorescent screen, the images can be viewed in real time. When a digital image is to be recorded, the screen is removed and the electrons are recorded on a camera. In the past, charge coupled device (CCD) cameras that are based on the inner photo effect were the instrument of choice. In recent years, the development of direct electron detectors with vastly improved detective quantum efficiency (DQE) and modulation transfer function (MTF) [150] such as the Gatan K2 and the FEI Falcon II + Falcon III improved the obtainable image quality majorly. Each of those detectors is based on backthinned CMOS sensors, but shows different DQE values at different spatial frequencies compared to one another[120].

1.2.3 Image Formation

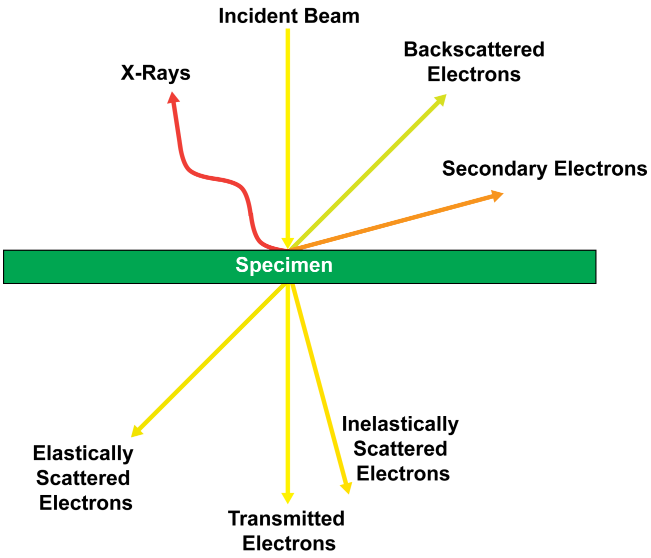


FIGURE 1.4: Possible electron interactions with a thin specimen.

Image formation in TEM is based on the elastic interaction of electrons with the nucleus of the specimen. All possible interactions [144] are shown in figure 1.4.

An electron penetrating an atom interacts with the positively charged nucleus through the Coulomb Force:

$$F = \frac{1}{4\pi\epsilon_0}\frac{e_0^2 \cdot Z}{r^2}, \tag{1.4}$$

where $\epsilon_0 = 8.854 \cdot 10^{-12}\frac{F}{m}$ is the vacuum permittivity, $e = 1.602 \cdot 10^{-19}$C is the elementary charge constant, r is the distance between electron and nucleus and Z is the number of protons in the nucleus. The higher the Couloumb force, the higher the deflection. This leads to negative contrast in the image. Due to the big mass difference between nucleus and an incident electron, no energy is transferred. This phenomenon is called elastic scattering. Heavy atoms containing more protons result in more amplitude contrast, but as biological samples mostly consist of light elements, these strong scatter effects occur very rarely. More likely, the electron is only lightly scattered by the Coulomb Force.

Inelastic scattering on the other hand occurs through interaction of an incident electron with a shell electron of the sample. Here, energy is deposited in the specimen, leading to radiation damage involving electron displacement, bond damage, ionization and other effects all contributing to noise in the image. Inelastic scattering effects are the reason for the necessary low electron dosage. Energy filters and objective apertures can filter out the high angle inelastic scattering, but due to occurring up to three times more frequently than elastic scattering[86] and providing low resolution information about the sample, these electrons are usually tolerated and give rise to amplitude contrast in the image, although this only contributes around 10-15% to the overall image contrast.

The aforementioned elastic scattering can be used to create phase contrast. The incoming electron wave traveling in z direction before scattering is defined as

$$\psi_0 = e^{ikz} . \tag{1.5}$$

A phase shift $\phi(p)$ is induced through interactions with the sample:

$$\phi(p) = \int C(p,z)dz . \tag{1.6}$$

Here, p is a 2D column vector describing interaction positions and $C(p,z)$ is the 3D Coulomb potential distribution of the specimen with thickness z [59]. The outgoing wave is therefore

$$\psi_p = \psi_0 \cdot e^{i\phi(p)} . \tag{1.7}$$

Equation 1.7 can be rewritten through use of the Taylor series $e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$ into

$$\begin{aligned} \psi_p &= \psi_0 \cdot \sum_{n=0}^{\infty} \frac{i \cdot \phi(p)^n}{n!} , \qquad (1.8) \\ \psi_p &= \psi_0 \cdot \left(1 + i \cdot \phi(p) - \frac{1}{2}\phi(p)^2 + \frac{1}{6}i \cdot \phi(p)^3 - ...\right) . \end{aligned}$$

For thin biological samples, the weak-phase approximation $\phi(p) << 1$ is assumed to hold true, enabling cutting off the expansion after the second term. It should be noted that in this form, the resulting wave is the sum of the scattered and the unscattered wave, with the scattered wave being $\frac{\pi}{2}$ out of phase with the incoming one [59]. Then, the intensity distribution I_p of the outgoing wave can be approximated as

$$I_p = \psi_p \cdot \psi_p^* \approx 1 + \phi(p)^2 , \tag{1.9}$$

where ψ_p^* is the complex conjugate of the outgoing wave. Using the weak phase approximation again, it can be deduced that the amplitude change through the phase shift is negligible and no image contrast is generated. Thus, an additional phase shift of $\frac{\pi}{2}$ needs to be introduced, resulting in a linear dependence of the intensity on the phase shift:

$$\begin{aligned} \psi_p &\approx \psi_0 \cdot (1 - \phi(p)) , \qquad (1.10) \\ I_p &= \psi_p \cdot \psi_p^* \approx 1 - 2\phi(p) . \end{aligned}$$

While there has been a lot of research concerning phase plates for TEM, e.g. the Volta phase plate [42] with some recent success for small proteins [106], their handling currently requires a lot of manual input and cannot be automated apart from the small amount of structures determined by it, so it remains to be seen whether phase plates can be applied to normal cryo-EM workflows. Therefore, commonly the more traditional method of generating the additional phase shift through an applied underfocus is used.

Through the previously described imaging imperfections, the actually observed wave ψ_o is generated through a convolution (see 2.1.3.2) of ψ_p with a point spread function (PSF), resulting in a smearing out of a scattering point source:

$$\psi_o = \psi_p * PSF\,. \tag{1.11}$$

Taking advantage of convolutions being multiplications of Fourier Transforms (see 2.1.3.2), this can be written as

$$F(\psi_o) = F(\psi_p) \cdot PhCTF \cdot E\,, \tag{1.12}$$

where the transfer function $PhCTF$ is defined as the Fourier transform of the PSF that is dependent on spatial frequency and E is an envelope decay function to account for signal loss at high frequencies[50].

For approximating the $PhCTF$, the Scherzer formula for wave aberrations $W(\ominus)$ can be used:

$$W(\ominus) = \frac{\pi}{2\lambda}\left(C_s \ominus^4 - 2\triangle d(\ominus)\lambda\ominus^2\right)\,, \tag{1.13}$$

where $\ominus$ is the scattering angle, C_s is the spherical aberration constant , λ the electron wavelength and $\triangle d(\ominus)$ the defocus in the direction of the varying scattering angle $\ominus$. The effect of amplitude contrast needs to be taken into account, which is expressed as the amplitude contrast ratio A that is commonly assumed to be 0.07. The CTF at a spatial frequency $\vec{s}$ is defined as[12]:

$$CTF(\vec{s}) = -\sqrt{1-A^2} \cdot sin(\gamma(\vec{s})) - A \cdot cos(\vec{s})\,, \tag{1.14}$$

where the phase difference $\gamma(\vec{s})$ is given by

$$\gamma(\vec{s}) = \gamma(s,\ominus) = \frac{\pi}{2\lambda}\left(C_s\lambda^2 \ominus^4 - 2\triangle d(\ominus)\lambda\ominus^2\right)\,, \tag{1.15}$$

where s is the modulus $|\vec{s}|$. The defocus $\triangle d(\ominus)$ can be calculated through[208]:

$$\triangle d(\ominus) = \triangle d_u cos^2(\ominus - \ominus_{ast}) + \triangle d_v sin^2(\ominus - \ominus_{ast})\,. \tag{1.16}$$

Here, $\triangle d_u$ and $\triangle d_v$ represent the maximum and minimum defocus and $\ominus_{ast}$ is the fixed anti-clockwise angle from the x-axis to $\triangle d_u$. Examples for 1D CTF functions at different defoci are shown in figure 1.5.

The envelope function E accounts for less signal at high spatial frequencies due to beam incoherence, MTF of the imaging system, optical abberrations of the microscope and other contributions [50]. It is commonly expressed as

$$E(\ominus) = e^{-2B\ominus^2}\,, \tag{1.17}$$

where B is an experimental factor to account for the specific decay.

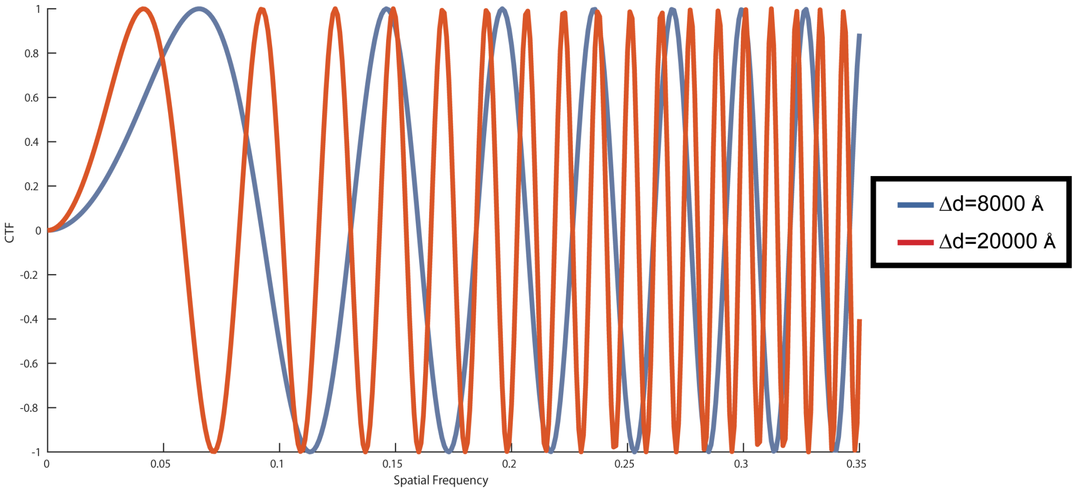


FIGURE 1.5: 1D CTF functions at different defoci. Shown are two simulated CTF curves at two different underfocus values plotted against spatial frequency. The simulations used an acceleration voltage of 300kV, a spherical abberation of 0.01 and a pixel size of 1.16. Dampening and possible astigmatism were neglected in the simulation.

1.2.4 Sample Preparation

As the imaging process is performed in vacuum conditions, specimen preparation methods are tasked with stabilizing the samples that are initially in a buffer solution and allow imaging despite the hefty radiation damage a biological sample experiences. For this, several approaches were developed over the years. Negative staining increases the amplitude contrast but only allows access to low resolution information, while cryo preparation relies on phase contrast information. Figure 1.6 shows the contrast differences of these preparation methods.

The preparative methods assume the sample of interest to be highly pure and to consist of stable complexes in a suitable concentration. Methods to reach these prerequisites are detailed in section 1.3. Both methods use a mostly copper carrier grid with a diameter of 3 mm and a mesh size of 200-300 μm.

1.2.4.1 Negative Staining Preparation

Negative staining was introduced in 1959 and has since been used to produce high contrast images. First, a continuous carbon film is applied on to the carrier grid. The liquid sample solution is then added to the grid, mixed with a 1-2% uranyl formate (or other heavy metal salt) suspension. The sample adheres to the grid through electrostatic interactions. Blotting then removes excess liquid and the grid can be inserted into the microscope[59].

The stain adheres to the outside of the protein complexes, rarely entering crevices. The heavy atoms from the stain interact with electrons more than the protein complexes do. Therefore the amplitude contrast is increased, leading to information about the outer shape of the molecules, e.g. low resolution information. This can be used to generate starting structures and to screen samples for biochemical quality, concentration and further parameters.

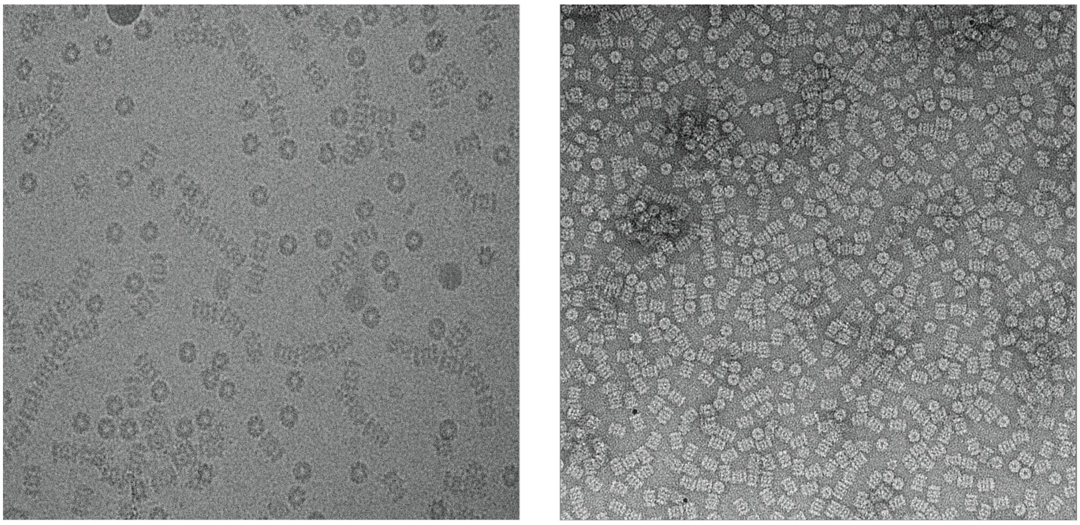


FIGURE 1.6: Difference between negative stain and cryo preparation. On the left a micrograph after cryo preparation and on the right a micrograph after negative stain preparation is shown. By convention, cryo micrographs as shown in this picture are contrast inverted before further processing, resulting in particles as seen on the right. The protein complex in both pictures is the 20S Proteasome.

There are important drawbacks to consider, though. Large complexes being incompletely stained in addition to stain thickness leads to high particle appearance variability and only parts of the shape correctly contributing to the imaged projection. This can partially be overcome with a double-carbon layer approach [59], which has the disadvantage of potentially increasing the shape distortion and flattening[58] that is introduced by the air drying and staining in general[104], leading to variations in the diameter of reconstructed structures.

Different staining agents and methods have been tested out[72] and recently, cryo-negative staining[2] has gained some popularity, but is hindered by the resolution limitation of around 10 Å.

1.2.4.2 Cryo Preparation

Non-staining sample preparation does not introduce any contrast enhancements. Multiple methods were developed for the purpose of keeping the sample in a state close to its native aqueous environment and reducing radiation damage, i.e. embedment in tannic acid[3], glucose[186] or vitrous ice[183]. The latter was found to be the preferable method due to quality of the reconstructed structures.

On a carrier grid, a very thin carbon film with μm-sized holes is added. The sample solution is applied onto the grid as a thin film and vitrified by plunging the grid into liquid ethane. The vitrification results in amorphous ice due to the suppression of ice crystal growth through the speed of the freezing process. Nevertheless, the ice layer is not guaranteed to be of uniform thickness.

This preparation method has the big advantage of reduced specimen damage at low temperatures[36], apart from not relying on external contrast agents. Thus, despite the low contrast of cryo images and noise introduced through the amorphous ice layer, resolutions at atomic detail can be achieved.

1.2.5 TEM Images

A grid contains foil holes from which an amount of images, depending on the applied magnification, can be taken. These images recorded on a modern electron microscope are very large. The most commonly used cameras, Falcon (II/III) and Gatan K2, differ in the resulting image size. The Falcon II writes out 4096 pixel x 4096 pixel images. The K2 camera can be operated in two different modes, counting mode resulting in 3838 pixel x 3710 pixel images, and the integration mode writing out 7676 pixel x 7420 pixel images. These modes are available in the Falcon III camera as well, but in this case do not change the image size. These images with overall pixel counts of over 16 million each contain gray scale values in the floating point range after gain correction. Every detector has a pixel size s_p, which is defined as the physical distance information that is contained in one pixel, commonly expressed in $\frac{\text{Å}}{pixel}$.

1.2.5.1 Pixel Defects

In most cameras, hot and cold pixels exist. These are pixels where either an unrealistic high or low intensity is found and that represent outliers that can disturb statistical approaches to working with the image. For invariable defective pixel positions, trivial corrections can be applied. As described in the literature [210], in modern cameras pixel defects can result in a cluster of pixels with distorted intensity distributions and not only vary in location, but also grow in numbers with longer exposure times. These pixel defects can be detected and corrected through statistical approaches [210].

1.2.5.2 Image Contaminations and Defects

Apart from defective pixels, images can have several defects that make them undesired for image processing. These can be described as several classes of defects based on their origin:

1. **Contrast deficiencies**: Here, an image doesn't have enough contrast to yield high-resolution information about the sample. This can commonly be detected in real space as well as in the power spectrum and can have various reasons, including imaging problems or a suboptimal vitrous ice layer.

2. **Sample defects**: When the sample is unstable, the complex can visibly fall apart. On the other end of the spectrum, sample aggregations also render the image unwanted.

3. **Foil and grid defects**: When either the foil has creases or parts of the foil hole perimeter are imaged, edges can be seen in the image. Also a defect called leopard skin has unclear origins, but is assumed to be due to foil defects.

4. **Vitrous ice defects**: When the ice layer is too thin, so called dry spots can be seen on the micrograph. Also, ice crystals can be encountered that result in characteristic signals in Fourier space.

5. **Artefacts**: In negative stain preparation, artefacts are usually due to the stain itself. In cryo micrographs, artefacts can be due to foil defects or the vitrous ice layer.

6. **Astigmatism**: In case of strong astigmatism as described by the defocus values $\triangle d_u$, $\triangle d_v$ and the astigmatism angle $\ominus_{ast}$ in a micrograph's power spectrum, resulting phase errors can turn a micrograph unusable.

Figure 1.7 shows examples of some of the described contaminations of recorded images.

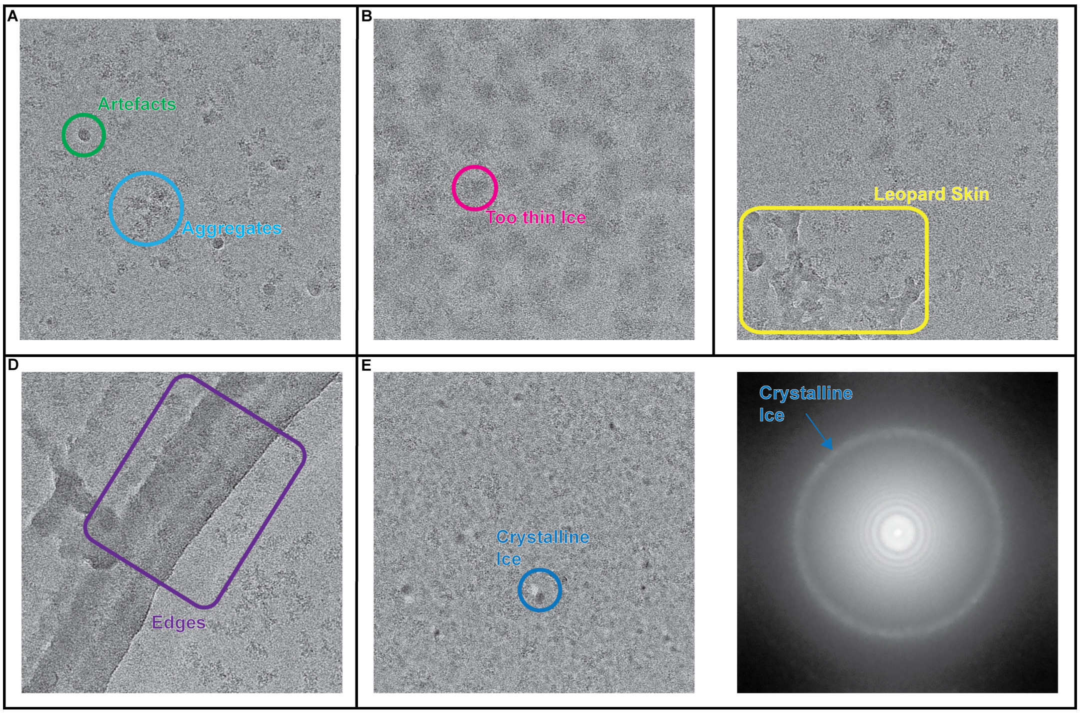


FIGURE 1.7: Classes of micrograph contaminations and defects. Shown are micrographs containing a Spliceosomal complex each showcasing one or more of the commonly encountered image deficiencies. For crystalline ice, the micrograph's power spectrum is also shown due to the strong ice ring resulting from crystalline ice in the micrograph.

1.2.6 Computational Image Processing

In image processing, the low SNR of every extracted particle from a preprocessed micrograph is slowly combined with similar images to increase the signal and to ultimately be able to reconstruct 3D information of the particle of interest. This process is performed in an iterative and self-referential manner. For the purpose of illustrating the processing steps, a dataset of the 20S Proteasome that was recorded on Titan Krios with a Gatan K2 camera is used that was kindly provided to the author by Karl Bertram.

1.2.6.1 Frame Alignment

Despite vitrification of the samples, the electron beam of a TEM induces changes in the sample that cause the molecules to slightly change their position on the grid due to charging effects[24], which limits the achievable resolution. Also, the applied electron dose influences the SNR in a spatial frequency dependent manner, which can be taken into account through dose weighting[11, 68].

Modern direct electron detector cameras can detect up to 400 frames per second. To counteract the blurring caused by the induced motions, these frames are then aligned on one another[28] to produce a frame sum that can be used for the subsequent CTF correction. For particle picking and beyond, a dose-weighted sum is calculated.

A popular software for frame alignment is MotionCorr2[210]. In the underlying algorithm, systematic motions that concern the whole frame are first corrected through Fourier space alignment of the whole frames. Afterwards, localized particle motions are corrected. This is done through a model of the local motions as projections of 3D motions that can be described with time-dependent 2D polynomial functions. For this, the frame stack is divided into a set of patches that are aligned on the patch sum, the determined shifts are fitted as part of a 3D motion and every pixel is subsequently interpolated based on the fit results.

1.2.6.2 CTF Correction

The sinusoid shape of the CTF with oscillating phase contrast between positive and negative results in no phase information being transferred at frequencies where zero crossings take place. This information is not accessible unless multiple defocus values with shifted zero crossings are used to record a dataset to later average the images. Therefore , the CTF parameters need to be determined for every micrograph to be able to flip the negative phase information later on. This is usually done through comparing the power spectra of the micrographs to ones simulated with known parameters. Due to foil defects, the CTF parameters on different areas of a micrograph are not necessarily equal within certain limits. The resulting parameter inaccuracy can be prevented by performing CTF parameter determination at a later stage in image processing by classifying and fitting the power spectra of particles to increase the SNR for accurate results[152].

For micrograph 2D CTF fitting, several algorithms and softwares exist that showcase different qualities in terms of accuracy and speed, as recently studied in the CTF challenge[118]. Two of the most widely used softwares are CTFFIND4[147] and GCTF[208], the latter employing a new approach called equiphase averaging(EPA) which takes astigmatism into account for rotational averaging. GCTF also calculates a resolution limit parameter that specifies until which spatial frequency CTF signal could be detected. This can be used to judge a micrographs usability.

1.2.6.3 Particle Picking

After having obtained an aligned frame sum and having determined the CTF parameters, the transition from working with micrographs to working with particle stacks can be performed. The goal of this step is to extract as many good particles as possible that represent all projection angles found in the dataset. Furthermore, the extracted particles should be centered fairly well, as this decreases the computational complexity of successive processing steps. For many years, this task was performed manually by some scientists. Due to the ever increasing need for bigger datasets that lead to higher resolution structures, manual particle picking has become an overbearing task. Therefore, softwares that operate in a semi-automatic or fully automatic manner are commonly employed, using template matching algorithms[207, 209] based on supplying a reference structure of the particles to be picked, machine learning[197, 203] or image statistics approaches as the CowPicker[26] (see 2.6.3) that was developed in the Department of Structural Dynamics.

Every approach has its advantages and disadvantages concerning computational speed, false negative and false positive picking, usability considerations, robustness to image quality variations and bias. Figure 1.8 shows a sample micrograph with exemplary particle locations.

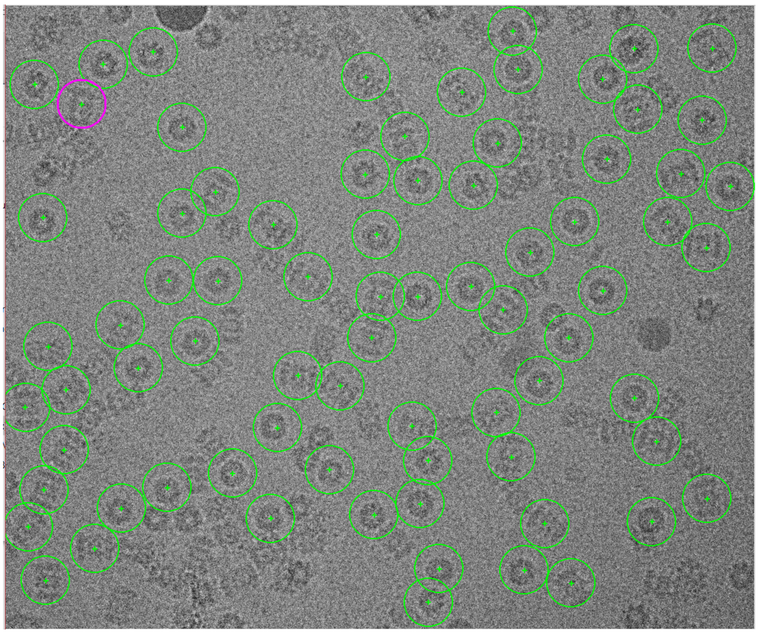


FIGURE 1.8: Example of particle picking. A non-inverted cryo micrograph containing 20S Proteasomes is shown with circled exemplary particle locations and their centers.

1.2.6.4 Preprocessing

After obtaining the particle stack, the first step is usually to normalize the images (see 2.1.1) and to apply a wide circular mask on them to simplify rotational operations during later processing steps like alignment. Furthermore, a band pass filter (see 2.1.5.3) is used to increase the SNR by removing low frequency gradients and high-frequency noise. The effects of the preprocessing steps are shown in figure 1.9.

1.2.6.5 Alignment

As described previously, the 3D Volume is imaged on a TEM as 2D projections that are based on the random orientation of the sample molecules on the grid. A mathematical description of this orientation consists of two components, translation and rotation. The former has three free parameters- the x, y, z components. Assuming the samples to be thin and monolayered, the z component, the defocus applied during recording, is commonly neglected as a free parameter for the alignment step. The

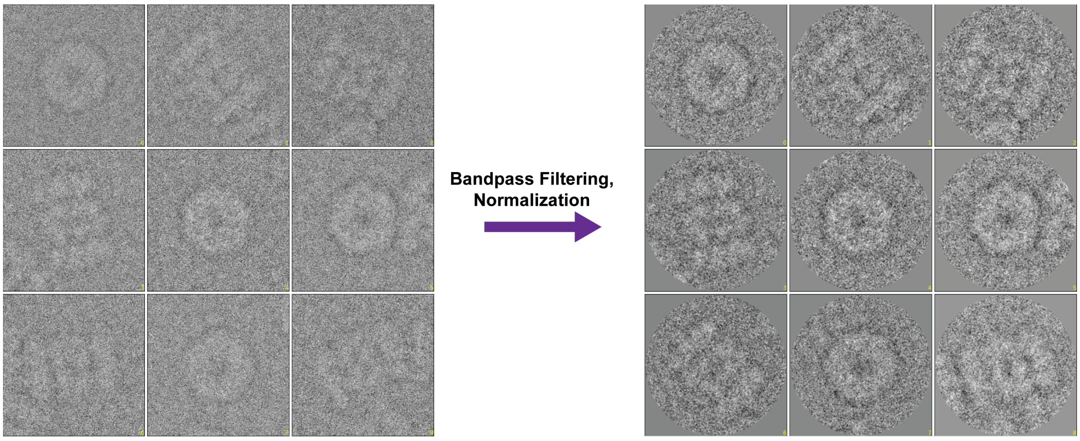


FIGURE 1.9: Effect of preprocessing on extracted particles. The particles were normalized, masked using a relative circle radius of 0.99 and then band pass filtered with an upper value of 0.7 and a lower value of 0.05

orientation can be described with three free parameters, the Euler Angles[91] α, β, γ. During alignment, only the in-plane rotation α is considered a free parameter.

2D alignment aims to identify similar projections of the input image set S by comparison operations. Due do the low SNR of the images and computational cost that would be involved, comparing all images to one another would be suboptimal. Therefore, all images from S are compared to a smaller set of images R with a higher SNR, called the references, and the described free parameters are optimized. For this, the transformation matrix M is used:

$$\begin{bmatrix} cos(\alpha) & -sin(\alpha) & x \\ sin(\alpha) & cos(\alpha) & y \\ 0 & 0 & 1 \end{bmatrix}$$

The goal, if using the Euclidean distance as the distance metric, is to find the reference r_i for the images_j that satisfies the minimization problem

$$r_i = min_{r \in R} \sum_{p \in s_j} |s_j(T(p)) - r(p)|, \tag{1.18}$$

where p are the pixel values of the input image s_j and the reference r is it compared to. Both images need to have the same dimensions. Alternatively, the Cross Correlation Coefficient (CCC) can be used (see 2.1.3.3) as the distance metric.

Computationally, this is a costly operation. Therefore, apart from algorithms employing iterative exhaustive search in real space or after transformation to polar coordinates[100], techniques to reduce the search space [153] were developed that employ invariant transformations or separate free parameter optimization. Recently, a maximum likelihood approach became popular where an image is aligned based on a weighted average of its determined orientation probabilities[167]. Figure 1.10 illustrates the general alignment procedure.

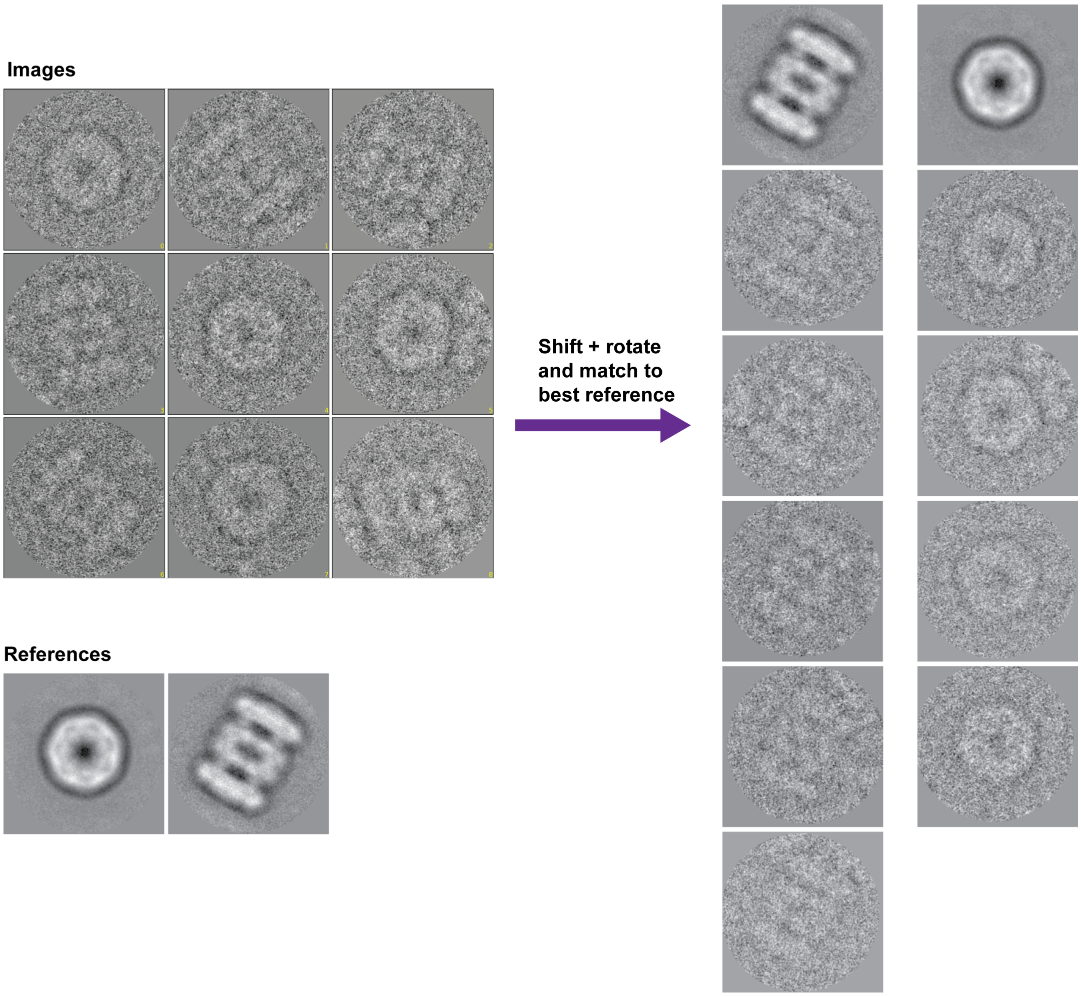


FIGURE 1.10: Illustration of image alignment. The input images are compared to the provided references, matched to the best one and have their free parameters in-plane rotation and 2D shifts optimized and applied

A big challenge for alignment is to find a suitable set of references. As demonstrated in the literature, model bias is an effect that cannot be understated due to the prevalent noise signal that is readily aligned given the degrees of freedom. If a previously determined structural model is available, its projections are commonly used as references. Apart from the orientation effect, alignment on multiple references could also be used for a first sorting of the input images, although the effect of model bias reduces the interpretability of the sorting results, as no unknown shared information can be contained in images matched to references.

In the beginning of image processing, so called reference-free alignment can be used, where all images from S are summed up and then rotationally averaged in one degree steps. Aligning to this reference has the effect of centering the input images, pre-optimizing x and y. This is illustrated in figure 1.11.
Afterwards, the centered images can be classified as described in the next section to extract similar views with a higher SNR to use as references for alignment. Through this procedure, the references in figure 1.10 were created.

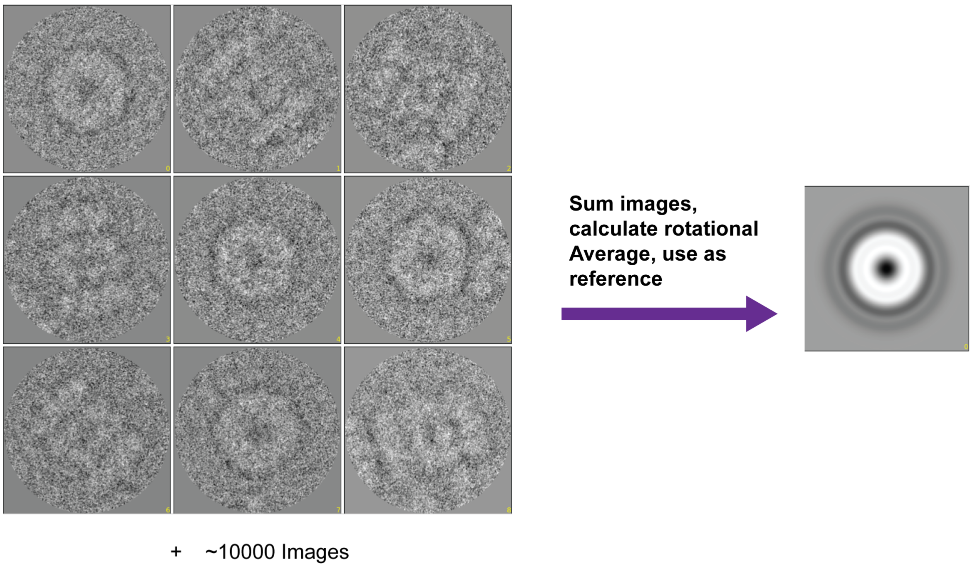


FIGURE 1.11: Illustration of reference-free alignment. Shown are the input images and the rotational average of the image's sum created from 10000 images from the dataset that is used as a reference for centering the images correctly. Due to the similarity of the top view of the 20S Proteasome to the rotational average, the centering works better for these views than for the side views.

1.2.6.6 Classification

Classification procedures aim to separate the input image set into a usually predefined number of classes that in an ideal case represent different projection angles of the underlying 3D structure. Afterwards, the images belonging to a class are summed up to create class sums with a higher SNR. In this step, external input like in alignment that leads to bias is avoided in favor of an approach called Multivariate Statistical Analysis(MSA) , using the image pixel information as its basis[22]. Due to the large amount of images and the large amount of pixels contained, a dimension reduction is employed due to the otherwise very high computational costs. As the best solution for this, the so-called Principal Component Analysis (PCA) was established in the field[128].

In the PCA, a set of linearly uncorrelated variables, called the principal components, is determined from a set of possibly correlated variables through an orthogonal transformation[127]. The transformation is defined as finding a new coordinate axis in the n-dimensional space for every principal component that describes the biggest variance in the dataset with the restriction that every principal component's axis (apart from the first ones) is required to be orthogonal to the previous ones. The data can then be described through a linear combination of those principal components that together account for the data variability.

Mathematically, an input image of dimensions n x n is treated as n^2-dimensional vectors in hyperspace that contain the image's pixel values. The data set of N input images each containing n x $n = M$ pixels can be described with a matrix X where every row represents the pixel values of an image:

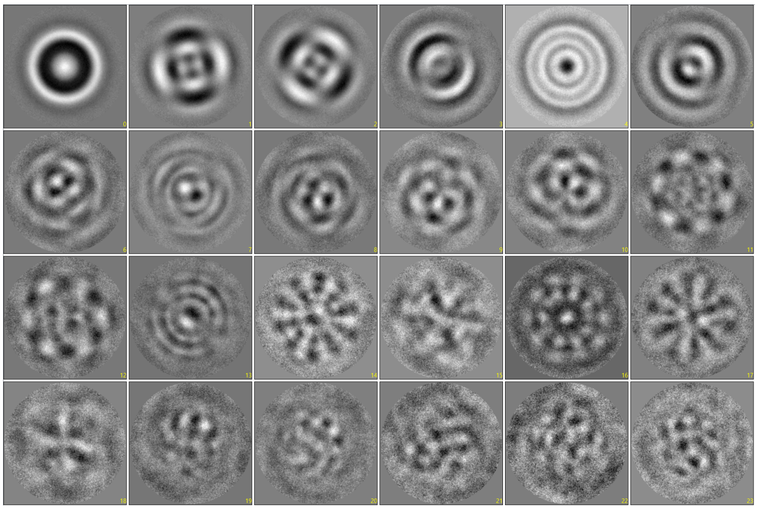


FIGURE 1.12: Example of Eigenimages. Shown are the first 24 Eigenimages calculated from the sample data set.

$$\begin{bmatrix} x_{11} & x_{11} & \dots & x_{1M} \\ x_{21} & x_{22} & \dots & x_{2M} \\ . & . & & . \\ . & . & & . \\ . & . & & . \\ x_{N1} & x_{N2} & \dots & x_{NM} \end{bmatrix}$$

The covariance matrix D of X can then be computed as

$$D = (X - \overline{X})^T (X - \overline{X}) \,, \tag{1.19}$$

where $\overline{X}$ is the matrix consisting of the average of the input images N in each row. The PCA can be formulated as the Eigenvector-Eigenvalue problem

$$D\vec{u} = \lambda\vec{u} \,, \tag{1.20}$$

where $\vec{u}$ are the eigenvectors that are the principal components and λ is an Eigenvalue. Solutions to this equation can be found by diagonalizing D, leading to at most $p = min(N, M)$ solutions $\vec{u}_1, \vec{u}_2, ..., \vec{u}_p$. These basis vectors that form the Eigenspace have an associated set of λ values. Thus, every image from the dataset can be described as

$$x_i = a_{i1}\vec{u}_1 + a_{i2}\vec{u}_2 + ... + a_{ip}\vec{u}_p \,, \tag{1.21}$$

where $a_1, a_2, ..., a_p$ are the scalar linear factors. Due to the lower order Eigenvectors describing the biggest variances in the dataset, not all mathematically possible Eigenvectors are used, as only the undesired noise information is described through the higher order Eigenvectors.

Eigenvectors hold the same amount of pixels as the input images and can thus be visualized in 2D, called Eigenimages, as shown in figure 1.12. Visual examination is commonly used to select the first k Eigenvectors that contain useful information. Thus, the PCA results in a massive data reduction of the dataset to k Eigenimages and $k \cdot N$ scalar Eigenvalues. The compressed data can then be classified.

The classification problem consists of splitting the input images into subsets of similar images in such a way that the inter-class similarities are minimal while the intra-class similarities are maximal. This problem is NP-complete, so no algorithm that can find a global solution to the requirements in polynomial time exists. Thus, fast and approximative algorithms are commonly used that usually find passable solutions, i.e. k-means clustering[44], hierarchical clustering and hybrid methods[188].

K-means is a non-deterministic clustering algorithm that is based on random selection of k images that serve as class centers. The value k is chosen by the user. Then, based on a distance criterion that is usually the euclidean distance, all other images are assigned to their closest seed. The center of these classes is then calculated from the images that were assigned to it. Afterwards, the assignment is repeated with the new centers. The last two steps are iterated until the class center position converges.

Hierarchical clustering is a deterministic algorithm that splits the input data set into a user defined amount of classes, starting from either one class that contains all images or from each image representing a single class. In the latter case, in every iteration until the desired class amount is reached, the two classes with the smallest distance are merged while minimizing the intra-class variance[198]. Here, the advantage lies in being able to isolate outlier images. In the other approach, the big class is successively split up through the same criteria until the desired number of classes is reached. This results in a more even class size distribution. Another option for classification is the computationally expensive maximum likelihood classification[161] that doesn't involve data reduction but yields good results.

The obtained class sums, the averages of all images that were grouped together, have a higher SNR than the input images and can be used as references for an alignment. Figure 1.13 shows an example of calculated class sums. These steps are usually iterated with less and less images per class until a stable solution is reached. Usually, not all class averages will represent unique views of the 3D structure, so a user selection is performed where undesired classes are discarded. When the results converge, the projection angles of the selected class averages can be determined.

1.2.6.7 Angular Assignment

In order to reconstruct a 3D volume, the projection angles of every class sum need to be determined. The in-plane rotation α is fixed through the alignment process, therefore only the two remaining angles β and γ need to be assigned.

If no 3D information in form of a previously determined structure is available, a method called angular reconstitution[189] can be used. Using the Fourier Slice Theorem[63], the common lines method[41] is based on the assumption that 2D projections of a 3D volume share at least one common section in Fourier space (see 2.1.3). With at least 3 projections, angular relationships can be determined. As the common line in Fourier space represents a 1D projection of the original 2D projection after being inverse Fourier transformed, this property can be used to determine the 2D projection angles in real space. In the common lines approach, this is realized through Radon transforms, or sinograms, where the input image is rotated in one degree steps and summed up along one axis until a full rotation is reached, generating an image with the x dimension of the original image and a height of 360 pixel. All

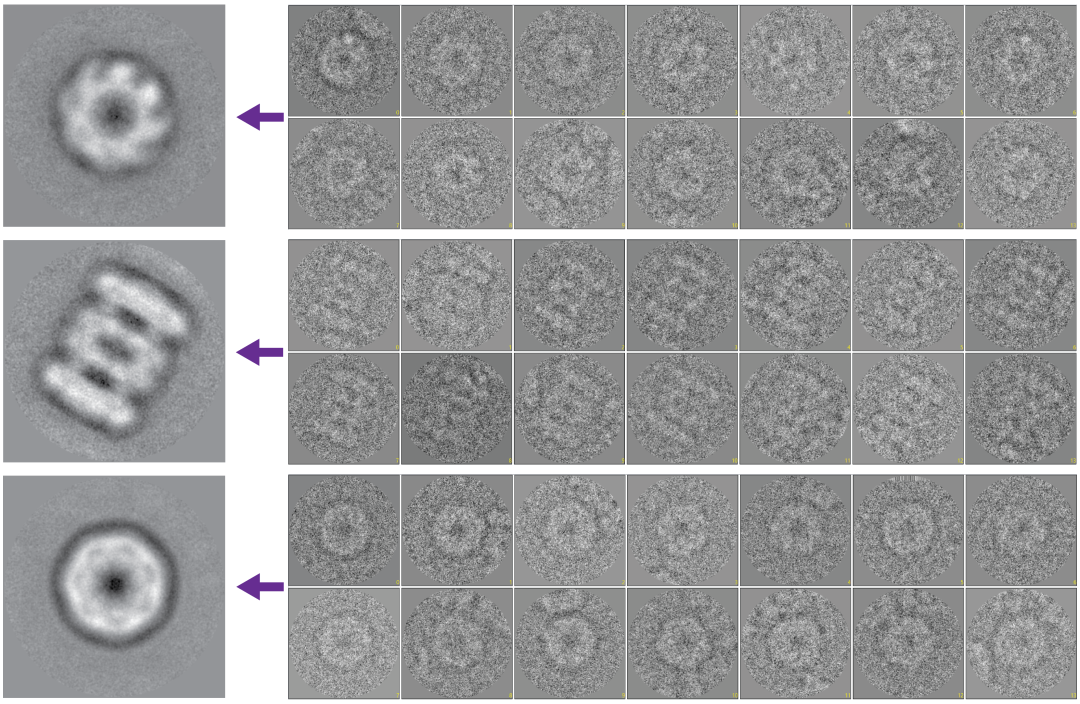


FIGURE 1.13: Classification and class sums. On the left, the class sums resulting from the images on the right are shown. The class member images were coarsed for better visibility, the actual dimensions of class members and class sums were the same.

sinograms can then be cross-correlated (see 2.1.3.3) and the angles towards one another can be assigned based on the correlation peaks. Initially limited to symmetric complexes[47] due to the larger amount of correlation peaks overcoming the noise limitations, further algorithmic improvements allowed the method to routinely be used for asymmetric structures[174, 173]. Recently, further improvements were suggested that include a consistency check of found common lines[168].

If a validated 3D volume is available, less effort has to be put into angular assignment. Generated equidistant projections of the volume can be used as references for an alignment. Thus, every image gets then assigned to a reference of known Euler angles. This method is usually employed in later stages of image processing in iterative steps. In this step, a good angular coverage is desired. An example of such a coverage is shown in figure 1.14.

1.2.6.8 3D Reconstruction

The general concept behind reconstructing a 3D volume from its 2D projections is Radon's theorem[142, 141] which states that in an n-dimensional Euclidean space, it is possible to express a real function from that space by the integrals over all hyperplanes with $(n-1)$ dimensions. Therefore, given sufficient 2D projections with known Euler angles, the original 3D volume can be retrieved unambiguously. The available algorithms for this step can be divided into two categories - backprojection methods and algebraic methods[129].

Backprojection consists of smearing out every projection into 3D space along its normal vector. In real space, every voxel's (the 3D equivalent of a pixel) gray value is

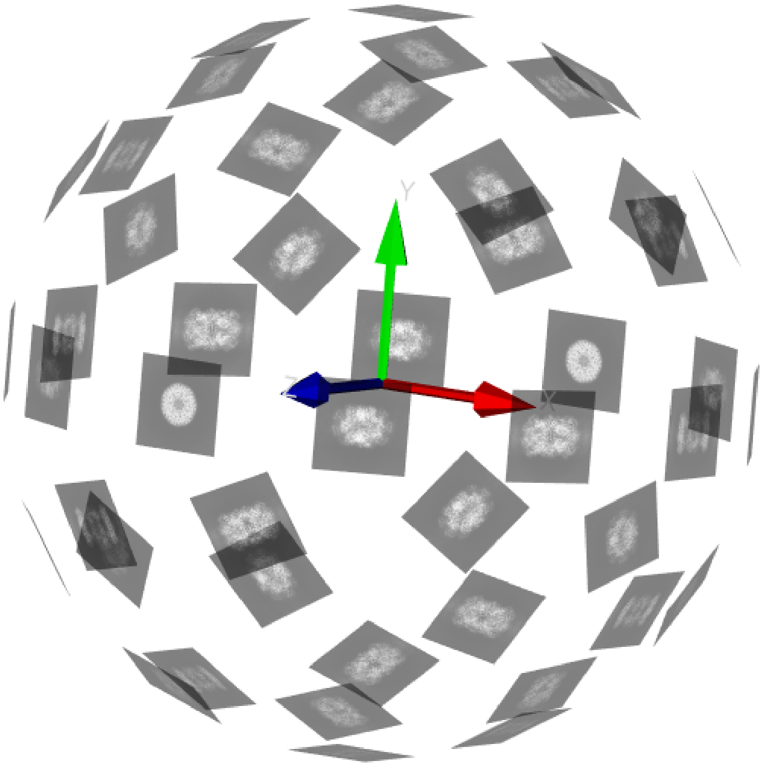


FIGURE 1.14: Optimal Euler angle distribution. Shown are possible class sums after angular reconstitution at the determined Euler angles. This example shows an evenly distributed angular coverage which represents the optimum result.

the average of all the projection's pixel values whose normal vector intersect with the voxel. This procedure involves interpolation and introduces blurriness to the volume. Thus, a filter is applied afterwards that smoothes the result. Commonly a ramp filter is used, but effects i.e. noise level, angular coverage, volume size should be taken into account when deciding on a filter[188].

This approach can also be used in Fourier space, again making use of the Fourier slice theorem[63]. Here, the 3D Fourier transform is resampled from the Fourier transforms of the projection images, with the real space structure being obtained after an inverse Fourier transform. This involves non-trivial interpolation of complex numbers[134].

While backprojection methods are very fast to calculate, algebraic reconstruction approaches are much slower. Due to their sophisticated approaches that include weighting, constraints and statistical considerations[129], their results can exceed backprojection techniques', though. For M available projections with N x N pixels each, the 3D volume to be reconstructed consists of N x N x N unknown voxels. For every pixel of every projection, a linear equation can then be constructed that expresses the weighted sum of every voxel that intersects with the normal vector of the pixel. This leads to M x N x N equations that need to be solved in dependency of one another, which is at this point computationally impossible to solve sufficiently fast. Additionally, the presence of noise complicates the posed problem. Therefore, several algorithms for iterative approximation have been developed: The algebraic reconstruction technique(ART)[67], the simultaneous iterative reconstruction technique(SIRT)[65] and the extension of ART, the simultaneous algebraic reconstruction technique(SART)[6]. These algorithms all iteratively minimize the error between input projections and projections from the calculated 3D volume, omitting direct solving of the equation system.

1.2.6.9 3D Refinement

The first 3D volume obtained through the processing steps described in the previous sections yields a starting structure with comparably low resolution that serves as a rough estimate of the free parameters. Through 3D refinement, the images of the dataset are iteratively aligned to projections with smaller angular spacing in a projection matching approach (see 1.2.6.5). In every iteration, projections from the current structure are generated, the images are aligned to these projections and summed up according to their assignment, with a new structure being reconstructed afterwards. Mostly, the RELION algorithm[157] is used that is based on maximum-likelihood considerations together with a Bayesian approach where the input images are weighted according to the probabilities assigned to their orientation parameters.

1.2.6.10 Resolution Determination

To be able to interpret the structure(s) at the end of image processing, the resolution of the 3D volume needs to be determined. Figure 1.16 shows the difference in determined maps at different resolutions. As different levels of structural details are only visible at specific resolutions as shown in figure 1.15, one could approximate the resolution through this visibility, but this approach only works for resolutions surpassing 10 Å and is too inexact to be used in practice. Also, the resolution could be determined in a comparative manner through known structures determined through other methods, but apart from the introduced bias, this would rely too much on the structure availability.

A bias-free mathematical solution to resolution assessment that uses the SNR influence on resolution is the Fourier shell correlation(FSC)[71], which is the 3D equivalent of the Fourier ring correlation(FRC)[80]. Here, the dataset is split in half and the normalized cross-correlation of the Fourier shells of the two independently refined 3D reconstructions is calculated. Fourier shells contain the signal at a specific radius corresponding to a spatial frequency, which is the independent variable of the equation:

$$FSC(r) = \frac{\sum_{r_i \in r} F_1(r_i) \cdot F_2^*(r_i)}{\sqrt{\sum_{r_i \in r} |F_1(r_i)|^2 \cdot \sum_{r_i \in r} |F_2(r_i)|^2}} \,. \tag{1.22}$$

Here, $F_1(r_i)$ is the 3D Fourier Transform of the first volume and $F_2^*(r_i)$ the complex conjugate of the second volume at the radius r_i. Due to the abundance of low frequency signal in a structure, the correlation is very high at low spatial frequencies and drops off when going towards higher resolution. Through the FSC, the spectral SNR can be determined[131] :

$$SSNR(r_i) = \frac{FSC(r_i)}{1 - FSC(r_i)} \,. \tag{1.23}$$

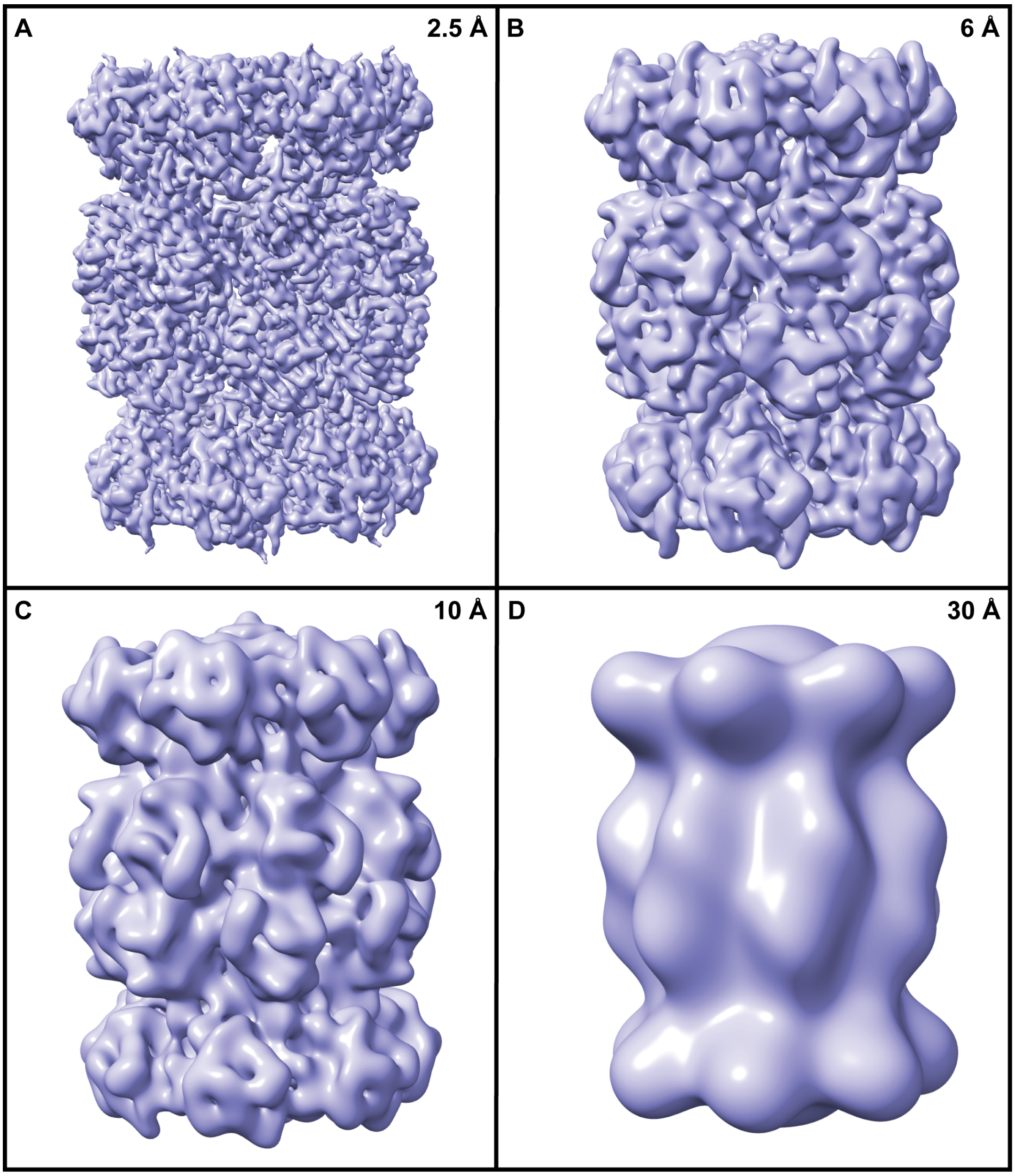


FIGURE 1.15: Comparison of a structure at different resolutions. Shown in A is a refined structure of the T20S Proteasome at 2.5Å. Structures in B, C and D were lowpass filtered to the specified resolutions. Clearly visible is the difference in recognizable detail.

The so called gold standard refinement[87] requires the dataset to be split in half at the very beginning of data processing, as otherwise no independence of the structures can be assumed and overfitting of noise would lead to falsification of results[159].

To judge the resolution of a structure from the FSC curve fall off, a threshold needs to be defined that signifies the last spatial frequency that still contains measurable signal above the noise level. The FSC threshold selection has sparked lively debate in the community[83, 171], with multiple values being used. 0.5 for example was chosen due to the SSNR dipping below 1 after this point[130]. The commonly used 0.143 threshold was developed in similarity to the figure-of-merit in crystallography

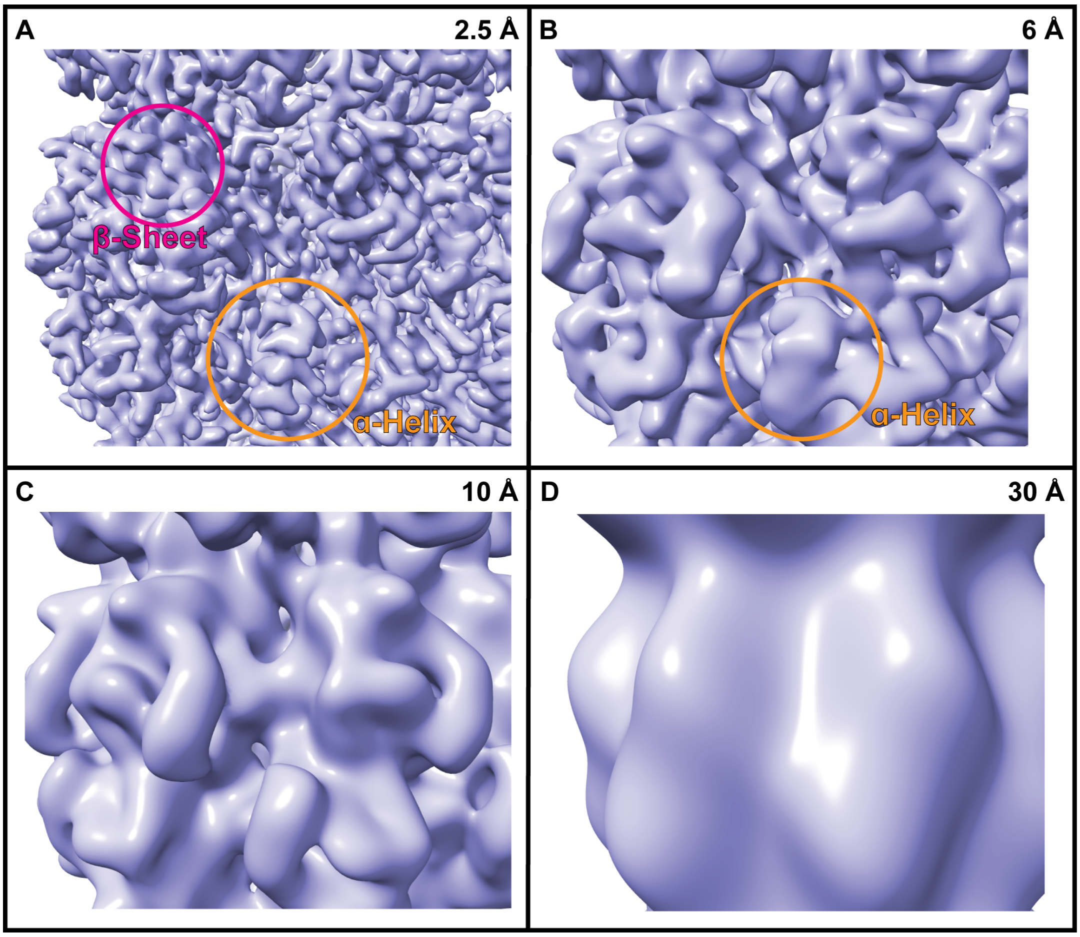


FIGURE 1.16: Detail comparison of a structure at different resolutions. Shown in A is a refined structure of the T20S Proteasome at 2.5Å. Structures in B, C and D were low pass filtered to the specified resolutions. Examples of recognizable secondary structures are circled. Clearly visible is the difference in recognizable details. At 30Å, only the general shape of the Proteasome can be seen, at 10 Å finer details emerge, at 6Å the first α-helices are visible and at 2.5Å α-helices, β-sheets and amino acid side chains can be seen.

and signifies the correlation between a perfect reference and an experimentally determined map[148].

The global FSC is based on the assumption of a homogeneous structure. In most cases, this doesn't hold true, with the dynamic regions showcasing lower resolution. Therefore, the validity of describing a 3D volume with a singular resolution parameter can be challenged. To get around this problem, a local FSC approach can be used. Here, the FSC is calculated by a moving kernel of a user specified size, with the central voxel of the kernel being assigned the calculated resolution value[90]. Problematic about this approach are the anticorrelated effects of averaging and sample size that are both tied to the kernel size, which influences results majorly. Recently, an alternative was proposed[109] where a local resolution map is calculated without data set splitting by fitting sinusoid functions of different wavelength to points in the structure and including corrections for false discovery rates. A comparison of global and local FSC is shown in figure 1.17.

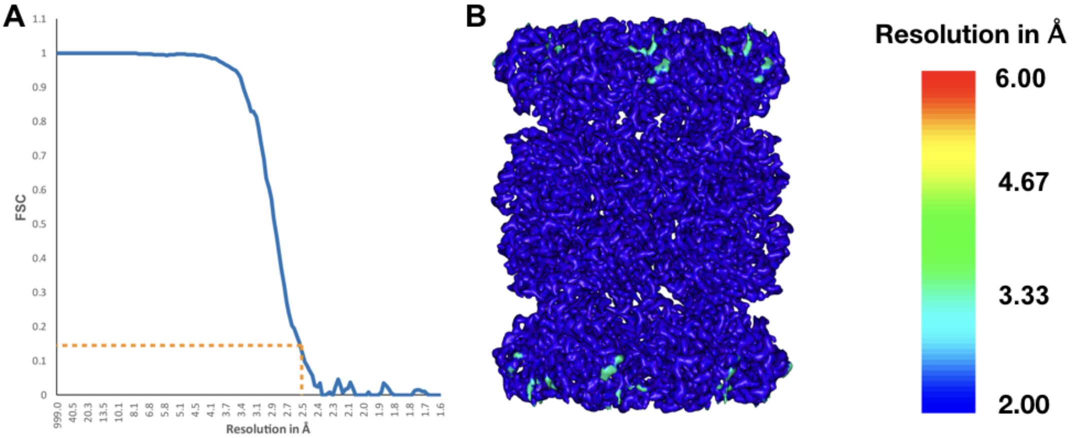


FIGURE 1.17: Global FSC curve and local resolution map. On the left, the gold standard FSC curve is shown with the 0.143 cutoff which yields a 2.5 Å resolution for the structure. On the right, a local resolution map determined with a kernel size of 13 is shown, with a lower resolution cutoff at 2 Å.

Limited by the Nyquist frequency $f_N = \frac{1}{2s_p}$, where s_p is the pixel size, the maximum achievable resolution for any part of the structure is $d = \frac{1}{f_N}$. Also, the angular distance $\triangle\ominus$ between n evenly angularly spaced class sums with diameter D used for reconstruction limits the maximum resolution d [81]:

$$d = D\,sin^2\left(\frac{\triangle\ominus}{2}\right) = \frac{2D}{n}\,. \tag{1.24}$$

1.2.6.11 Accessing Conformational Information

While structural heterogeneity increases the difficulty of determining high resolution structures, it also holds essential information about the relationship between structure and function of the macromolecular machine of interest. Accurately determining structures in different conformational states is a subject of great research interest.

If the heterogeneity is localized to specific regions of the protein complex, a local resolution map as described in 1.2.6.10 can help identify those. A developed bootstrapping approach [132, 133] where random subsets of the dataset are reconstructed independently can additionally assist by enabling creation of a variance map of the structure.

In case of non-localized heterogeneity, the image processing may fail to generate even a stable starting structure. Here, random conical tilt (RCT) approaches combined with 3D classification [154] can be employed. Alternatively, approaches based on common line sorting[89] or probabilistic model building[49] can be used.

If a starting model can be generated, there are several approaches to generate references based on it to split the dataset into different structures. Normal mode analysis (NMA) can generate theoretical conformations on which the dataset can then be aligned. Also, maximum-likelihood approaches once again yield good results combined with a supervised classification[160].

On the 2D level, there has been some success with sorting on the PCA level[48]. Recently, a project from the Department of Structural Dynamics showed that a 3D PCA analysis can be utilized to analyze the conformational landscape and its changes upon inhibitor binding, as shown on the 26S Proteasome[74]. A comprehensive discussion of conformational analysis procedures can be found in [73].

1.2.6.12 Postprocessing

As mentioned in 1.2.3, the imaging process introduces a contrast fall-off, specially at higher resolutions that can be approximated with an exponential decay. The image processing results in another contrast loss through inaccurate determination of the orientation parameters, which results in blurring out of finer details. These effects can be combined into a global B factor that describes the contrast loss and reduces the visible quality of the structure. The global B factor can be interpolated from measured low resolution structure factors that should not be majorly influenced by the contrast loss. Applying a negative B factor to the spherically averaged structure factors can restore the contrast[79]. This factor can be determined through comparisons of the experimental B-factor to the B-factor of perfect scattering curves acquired from small angle X-ray scattering or from atomic maps determined through X-ray crystallography[148]. Both of these should obviously be determined for the same protein complex. An example for the effects of this so-called sharpening procedure are shown in figure 1.18. Alternatively, heuristic multiples of the measured experimental B-Factor can be used for restoration. As the background noise is also amplified, the resulting structure is commonly low pass filtered to remove the unwanted signal.

The availability of reliable theoretical scattering curves and the spherical averaging in this procedure limits its applicability in some use cases. With structural homogeneity in play, applying a singular B-factor is debatable. Recently, an approach was developed that, while still relying on an available model, introduces windowed contrast restoration that better accounts for local differences in the structure[97].

1.3 Sample Optimization in Structural Biology

As mentioned in section 1.2.4, a stable and pure sample is necessary to be able to acquire high quality structural information, whether through X-ray crystallography or cryo-EM. The first step in this endeavor is the cloning and expression system. Through technical advances, large-scale expression and subsequent purification of many recombinant multiproteins is now routinely possible[146], also in a high-throughput manner[196]. Nevertheless, new and largely unknown protein complexes are challenging to control biochemically due to conformational and compositional instability, aggregation and buffer incompatibility. To tackle this problem, an automated approach based on differential scanning fluorimetry(DSF) to optimize buffer conditions to increase stability called ProteoPlex[33] has recently been developed in the Department.

Recent successes with chromatography-free purification approaches[163, 74] indicate improved protein yield and complex stability when avoiding tag-based purifications. Involved methods in these approaches include polyethylene glycol (PEG)

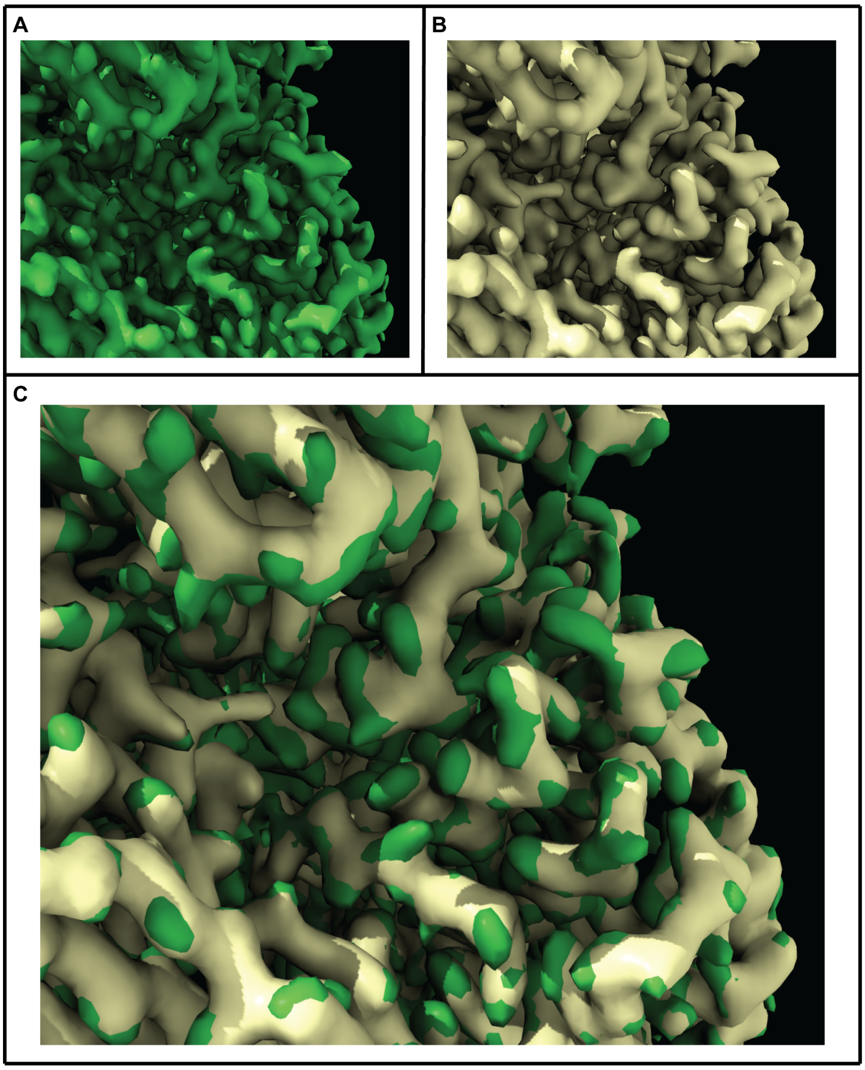


FIGURE 1.18: Effects of sharpening on a refined structure. Shown is an exemplary region of the refined T20S Structure in A and the the same region after sharpening with a B-Factor of 60 in B. C shows an overlay of both maps. In B and C, the visual improvement of high resolution information can be seen through better recognizable protein backbones.

precipitations and density gradient centrifugation, which has also found use in preparation of membrane proteins in a method called GraDeR[77]. The most prominent use of density gradient centrifugation is through the GraFix protocol[101, 172]. It consists of zonal ultracentrifugation with a cross-linking reagent that leads to intramolecular crosslinks and promotes complex stability.

1.3.1 Basics of Density Gradient Centrifugation

This technique, invented by Brakke in 1951[23], is based on the different sedimentation speeds of particles through a tube containing a gradient of an osmolyte under application of a high centrifugal force in ultracentrifuges.

The technique can be divided into two main approaches. In isopycnic centrifugation, particles are separated based on their density and sediment to the position in a tube with the same density. Here, particle size affects the time until the equilibrium is reached. All the previously mentioned biochemical techniques in structural biology use rate-zonal centrifugation, though. In it, a sample is loaded onto a preformed, continuous gradient as a narrow zone with the particles sedimenting primarily based on their mass and size. If centrifuged long enough, the particles will eventually pellet.

As osmolytes for rate-zonal centrifugation in protein complex purification, commonly used chemicals are sucrose, glycerol and to a lesser extent trehalose, arabinose and CsCl. A continous gradient can be formed through different manual approaches, some even involving Play-Doh[15], but commercially available solutions are usually preferable due to reproducibility and speed considerations. In the Department of Structural Dynamics, the Gradient Maker from Biocomp is used to generate gradients, with a list of predefined density combinations at the bottom and top of the tube being available for combinations of tube and osmolyte (see A.1,A.2).

Rate-zonal centrifugation uses almost exclusively swing-out rotors. In these rotors, the tubes swing out so that they are orientated in a parallel manner to the centrifugal force. A swing-out rotor is characterized by its tube volume, its maximum speed, its acceleration and deceleration time and the minimum and maximum radius r_{min} and r_{max} from the center of the centrifuge at which the contents of the tube are positioned. After centrifugation at a specific temperature (for proteins usually 4°C due to aggregation considerations) and speed for a specific amount of time, the tube contents are fractionated. This can be done in a top-to-bottom or in the more complicated bottom-to-top manner. The protein contents per fraction can be analyzed on a SDS-Gel or through other protein concentration determination protocols. A summary of the method is shown in figure 1.19.

1.3.2 Theory of Rate-Zonal Separation

Particles of volume V and density ρ_p layered on top of a linear gradient experience a centrifugal force and based on the Archimedes principle a force in the opposite direction depending on the density ρ_m of the medium. The net force F can then be expressed as

$$F = V(\rho_p - \rho_m)r\omega^2\,, \tag{1.25}$$

where ω is the angular velocity of centrifugation in $\frac{radians}{second}$. For a spherical particle with radius R moving with velocity v, Stoke's law indicates the frictional force in the opposite direction to be

$$S = 6\pi R\eta v\,, \tag{1.26}$$

where η is the viscosity of the medium. Under the assumption that the comparatively light particles reach their terminal velocities instantly, the two forces acting on it are the same, so the sedimentation velocity can be written as

$$v = \frac{V}{6\pi R} \cdot \frac{(\rho_p - \rho_m)}{\eta} \cdot r\omega^2\,. \tag{1.27}$$

This shows the proportionality of v to the centrifugal acceleration. The proportionality factor s for a specific medium is therefore

$$s = \frac{V}{6\pi R} \cdot \frac{(\rho_p - \rho_m)}{\eta}\,. \tag{1.28}$$

Assuming size and shape of the particles to be medium composition independent, the factors s_1 and s_2 corresponding to two media with densities (ρ_1, ρ_2 and viscosities η_1, η_2 can be expressed as

$$s_2 = s_1 \frac{(\rho_p - \rho_2)}{(\rho_p - \rho_1)} \cdot \frac{\eta_1}{\eta_2} \, . \tag{1.29}$$

Based on this, s, the sedimentation coefficient, can be defined for a particle under standard temperature T, usually 20 °C, in a medium M, commonly water, as $s_{T,M}$ or $s_{20,w}$. Taking this into account, the sedimentation velocity is therefore

$$v = s_{20,w} \frac{\rho_p - \rho_{T,M}}{\rho_p - \rho_{20,w}} \cdot \frac{\eta_{20,w}}{\eta_{T,M}} \cdot r\omega^2 \, . \tag{1.30}$$

From these equations, an integratable expression can be formulated:

$$s_{20,w} \int_0^t \omega^2 dt = \frac{\rho_p - \rho_{20,w}}{\eta_{20,w}} \int_{r_{min}}^{r_{max}} \frac{\eta_{T,M}}{\rho_p - \rho_{T,M}} \frac{dr}{r} \, . \tag{1.31}$$

As $s_{20,w}$ is usually very small for biological particles, it is commonly multiplied by 10^{13} and written as $S_{20,w}$ or just S and called a Svedberg unit or S value. Its unit is 10^{-13} seconds.

Based on these considerations, it can be seen that in order for a particle to sediment into a medium, the particle density must be higher then the medium density. A gradient solution whose viscosity is dependent on its concentration will therefore decrease the sedimentation velocity dependent on the position of a particle along the tube. This effect is opposed to the increased centrifugal force that particles experience upon sedimentation away from the center of the ultracentrifuge.

1.3.3 Particle Distribution around the Peak

Apart from the sedimentation processes, diffusional processes of the osmolyte and the sample lead to a broadening of the particle zone in the leading and trailing end[178]. In practice, in a centrifugation with a large amount of particles sedimenting at the same time, more effects come into play. Diffusion, hydrodynamic instability, droplet sedimentation and, in an unknown sample, biochemical instability, protein-protein interactions, aggregation and more can lead to broadening of the end position distribution of particles that is hard to predict.

1.3.4 Challenges in Obtaining Run Parameters

For a successful purification of a protein complex of interest from other components contained in the sample, the complex of interest should have a distribution peak position at $\frac{1}{2}\, r_{max}$ if there are known contaminants with MW in the same order of magnitude in the sample and it should be at $\frac{2}{3}\, r_{max}$ if a purification from unspecific contaminations is to be performed. Both these values were empirically determined to maximize separation. To estimate whether a set of parameters will lead to the desired results, simulation of the run conditions need to be performed to avoid wasting sample material and time.

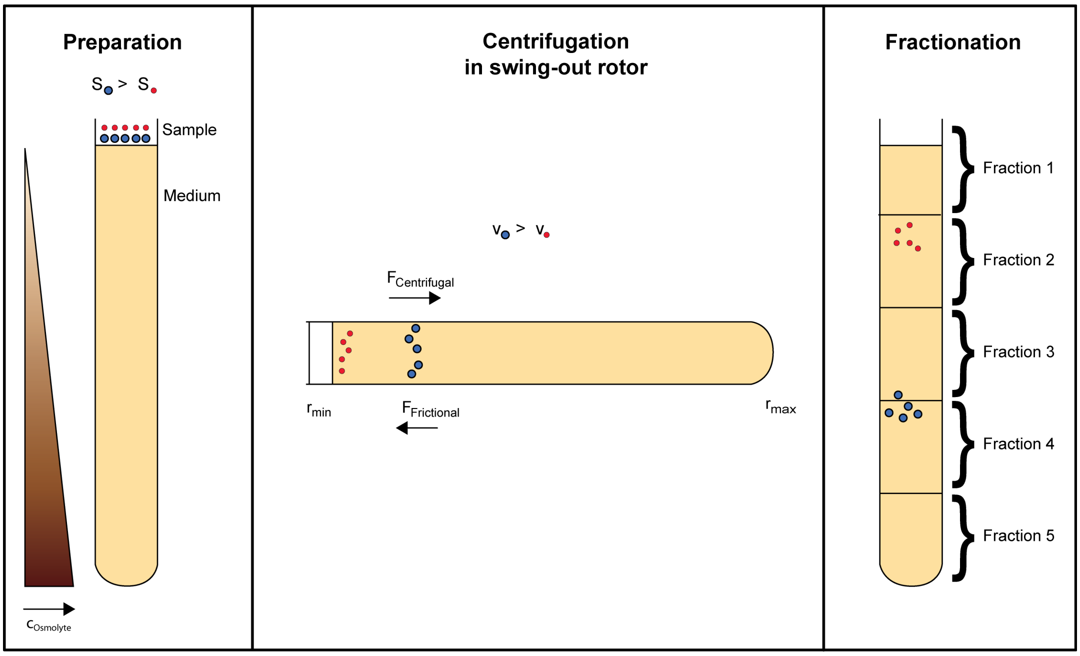


FIGURE 1.19: Principle of rate-zonal separation. The steps of a rate-zonal separation experiment are illustrated with two exemplary molecule classes. It should be noted that the number of fractions is usually way higher than 5.

When trying to predict the sedimentation results of a gradient based on equation 1.30, the varying densities and viscosities of the used medium need to be calculated and the centrifuge run conditions, e.g. speed, run time, temperature, r_{min} and r_{max} of the rotor need to be taken into account. Furthermore, the particles are assumed to be spherical with a known sedimentation coefficient. In practice, having a S value at hand is hardly ever the case and usually only the molecular weight of a new complex to be purified is known approximately. Large complexes are usually non-spherical and showcase deviations from the expected sedimentation behavior, complicating the simulation task. Determination of S values is commonly achieved through complicated fits of sedimentation velocity data from time-resolved analytical ultracentrifugation experiments[137, 135, 209, 136] that get even more complex for assumed heterogeneous systems [164].

For established purification protocols of well-characterized protein complexes such as the ribosome, empirically determined run conditions are used. For less known samples, simulation softwares are employed, in which the run conditions are entered and a prediction of the distribution of a protein complex with a chosen S value is then displayed. Over the years, several algorithms have been developed to predict density gradient centrifugation results[178, 92, 156].

The only available softwares for simulations are COMPASS(1987)[177], which was available to the author, but is not on the market anymore, and its updated version, both of which are very likely based on the theoretical method developed by one of its developers[178]. The COMPASS software is proprietary and sold by Thermo-Fisher Scientific. COMPASS only allows choosing from a predefined set of rotors and needs a complete parameter set for a simulation, which is insufficient when trying to determine optimal experimental conditions.

Therefore, there is need for a new software that aids users in finding the optimal conditions to maximize the purification success in density gradient centrifugation in a qualitative manner assuming that very little about the protein complex of interest is known.

1.4 Scientific Software Development

Software has become an essential part of modern science, supporting both experimentation and theory[193]. A rift in philosophy between scientific computing and software engineering has developed[195], with the software engineering community calling for domain-independent methods and the scientific community that consists of primarily self-taught professional end-user developers[165] rejecting this approach[105]. No universally accepted standards for scientific software development have been agreed on at this point, but voluntary guidelines to follow[202, 14] exist.

In general, scientific software should enable reproduction of results but also answering of new scientific questions[99]. For end-user developers, usability concerns and good software engineering practice tend to be lower on the priority list[166].

Traditionally, adaption of object-oriented languages employing software patterns [20] to make software more reliable, scalable, extensible and maintainable has been slower in the scientific community. These languages and pattern enable a separation of algorithmic and architectural code. One can argument that this makes open-sourcing a complete software, which is recommended by many [138], unnecessary if only the algorithmic parts can be exposed to potential contributors. In the work of this thesis, recommended software engineering practices were adopted to also accommodate inexperienced users in the field.

Apart from architectural considerations and the validity of the underlying algorithms, processing speed is a big aspect that drives software development, specially in a data-heavy field such as cryo-electron microscopy, as further explained in section 1.5. Improving algorithms in either theory or implementation or optimizing the workflow is one way of tackling that problem. The second one is processing data in a parallel manner, which has become more and more relevant due to an expected slowdown of the processing power increase.

1.4.1 Parallelization

Moore's law about the processing power of central processing units (CPUs) doubling every 18 months[122] held true for a long time, but the increase has slowed down recently[38] due to physical limitations. Therefore, parallelization has become more and more important, which involves performing calculations on multiple CPUs on a local computer or on a connected set of computers, a computer cluster. Furthermore, graphical processing units (GPU) with inherent parallalization-heavy design have emerged as invaluable tools to increase processing speed[162].

1.4.1.1 Requirements for Parallelization

To be able to split a problem up into smaller parts, creation of independent packages needs to be possible, as every communication between the packages is a costly endeavor and introduces the risk for errors like race conditions. Therefore, a problem where a big dataset requires repetition of a calculation on every small subset of the data is suited for this. In TEM image processing, where a lot of operations are performed on every pixel of an image, these requirements are given.

1.4.1.2 Farming

A computation can be divided onto server nodes on a computer cluster, which is called farming. Usually, the nodes possess the exact same hardware specifications and a master node distributes the computation through communication standards like the message passing interface[55]. In case of heterogeneous hardware, the BOINC framework can be used [7].

1.4.1.3 GPU Parallelization Programming

On a given node that has received a part of the computational problem, further parallelization can be achieved through calculation division on CPU or GPU threads. General purpose GPUs (GPGPU) became available in the early 21st century, changing the previously specialized shader computing units into unified shaders. Through this, a huge amount of arithmetic logical units can be used to run threads in parallel, speeding up computations despite the lower frequency of a GPU unit.The bottle neck in GPU parallelization is the data transfer from CPU to GPU.

For NVIDIA graphical boards, as used in the Department for Structural Dynamics, the company provides the CUDA framework[124] that enables a user to write code that executes on a GPU. It is an extension of the C programming language and includes a special compiler for files that contain CUDA code, the Nvidia C-Compiler (NVCC). Another option is OpenCL, a GPU computing framework that is hardware independent.

1.5 Current Challenges in Cryo-EM Data Processing

While the amount of published EM structures has steadily increased over the years, as shown in figure 1.1, a closer look reveals that the resolution distribution of the structures has naturally shifted to higher resolutions, but still, very few structures reach resolutions above 3 Å that enable atomic model building. This resolution distribution is shown in 1.20. A further look reveals that most of the highly resolved structures are either symmetrical virus particles or well-characterized complexes such as the ribosome where atomic models are available. The reasons for the troubles in reaching atomic detail are partially found on the biochemical side as explained in section 1.3, but also in image processing. The reasons for this are explained in the following sections.

1.5.1 Imaging Automation and Amount of Data

Modern microscopes are able to record images in an automated fashion after minimal input from the user while also being able to switch between up to 12 grids at a time. Furthermore, DED devices routinely record up to 60 frames per micrograph. This results in a huge amount of data being produced that needs to be stored for

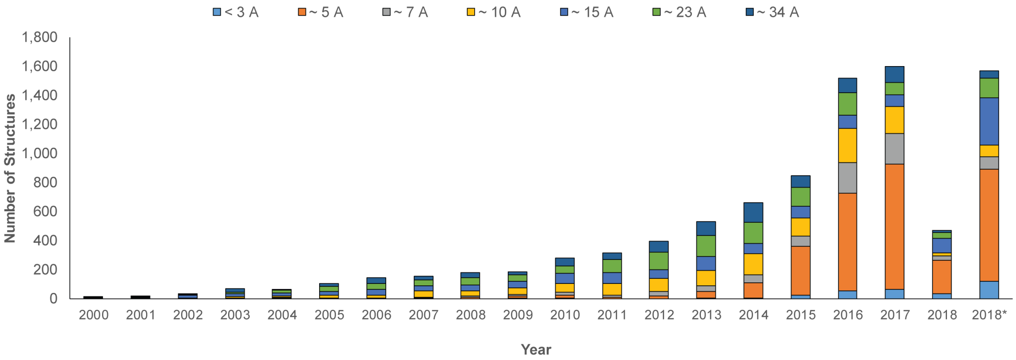


FIGURE 1.20: Maps deposited in the EMDB since 2000 and their resolutions. Denoted with 2018* is the linearly interpolated value for 2018 based on the amount of maps deposited until April 2018. The data shows that a lot of structures don't reach high enough resolution to enable atomic detail interpretation.

further processing. For example, if recording data the whole day, a Titan Krios microscope at the Department of Structural Dynamics produces >10 TB of images.

Due to the varying image quality that is partially due to automation and partially due to metrics that only surface during the first processing steps, a lot of the recorded data should be discarded as early as possible, a step for which good methods are lacking as of now.

1.5.2 Computation Speed and Live Processing

Due to the large amount of statistics that are needed to achieve high resolution structures and the inherently computationally expensive image processing steps, computational speed is always a concern. Unless calculations take place on a modern computer cluster, the bottlenecks such as alignment and refinement can take days to even weeks on an older desktop computer. Intelligent GPU computing has reduced the computational time, but is very hardware dependent due to the GPU architecture which is in constant change. Therefore, an early reduction in amount of data can again be beneficial.

Currently, common image processing softwares work in a static manner, with every algorithmic step being performed one after the other. The only exception to this is Focus[19], which includes live monitoring capabilities. As of now, the achievable resolution and quality of a dataset can sometimes only be estimated after a 3D reconstruction or even refinement, so it would be beneficial to enable some sort of live processing where a user can process the recorded images during acquisition so that unsuitable micrographs, biochemical instability, faulty TEM alignment and such can be detected without having to record a whole dataset.

1.5.3 Self-Referential Processing- the Need for Data Cleaning

One of the main strengths of single particle cryo-EM is the lack of introduced external bias. This necessitates a carefully chosen data set, though. Empty, faulty, noise or in other ways unsuitable images will introduce larger errors in determination of the orientation parameters. Therefore, in addition to the implicit cleaning steps in

2D and 3D classification where bad images are combined into visually identifiable bad class sums that are then removed from the data set, it is important to reduce the amount of undesired images as early as possible. As the expected outcome for datasets can vary considerably - for example between a well characterized complex like the E.Coli ribosome and a previously structurally uncharacterized complex- the generalization capabilities are assumed to be limited, necessitating some sort of user input.

1.5.4 Processing Software Landscape

With the increasing popularity of cryo-EM, the amount of available softwares has also increased[158, 19, 82, 190, 123, 69]. Usually, an algorithmic approach is turned into a stand-alone software that is used for that specific processing step. While some software packages cover most of the workflow, extensibility is at most implemented through execution of external command line softwares in scripts.

Due to the reliance on several softwares, each with varying parameter conventions, file formats and such, data management and chaining different softwares together for a series of isolated algorithmic steps represents a challenge, specially for novice users of the method.

1.6 Aim of the Work

In the fast progressing field of single particle cryo-electron microscopy, more and more biological information is made accessible through high-resolution structures and the access to conformational sorting. For this, a large amount of high-quality images of stable protein complexes are necessary to get the best possible results out of the self-referential statistical image processing workflow. Also highly relevant is the computational demand and the high amount of data storage required for routinely recorded large datasets, both of which should be reduced as much as possible.

As described in section 1.3, the important purification method density gradient centrifugation needs to be supported by theoretical predictions about the end position of a protein complex, about which little is known apart from the molecular weight, on the gradient. Furthermore, a parameter optimization algorithm is needed to enable better purifications. Both of these problems are tackled in a software called CowGraCE, which stands for **Cow Gra**dient **C**entrifugation **E**stimation, that was developed as part of this thesis. Apart from the algorithmic and computational considerations, usability aspect were taken into account to support users in finding the optimal conditions for their purification approaches to acquire an optimal, highly pure sample to image on a TEM or for use in crystallization.

The second part of the work of this thesis is the COW, a software suite that covers the whole workflow of TEM image processing from micrographs to 3D structures. Work on the COW has been ongoing since 2008[26, 107, 84, 117]. It offers high degrees of automation but also a highly flexible graphical programming interface with a large variety of algorithmic modules that can be chained together to customize the workflow and solve problems specific to the sample of interest. The COW includes high degrees of parallelization across CPU and GPU architecture and also offers a server version that takes full advantage of the speed-up capabilities of modern high performance computer clusters.

Both COW and CowGrace follow the software development principles outlined in section 1.4 and are multi platform compatible with a focus on user-friendliness and stability.

Within the COW, a new front end called the Micrograph Quality Checker (MQC) was developed that serves as the entry point to the COW ecosystem and represents the first step towards live processing. The MQC continuously grabs newly recorded data and presents it to the user that makes binary decisions on the quality. From this acquired data, a machine learning model is trained to correctly classify the images.

The MQC is part of a larger philosophy of the COW project which is based on acquiring and judging meta-information about the dataset. As the image processing workflow boils down to finding the best images and optimizing their unknown orientation parameters, metadata that is collected at all steps from image acquisition up to the final refinement can prove invaluable in getting closer to true atomic resolution. Removing unwanted images at the very beginning reduces the computational overhead and starting to collect data about the good images implements the COW philosophy in the MQC.

Chapter 2

Materials and Methods

"Explanations exist: they have existed for all times, for there is always an easy solution to every problem – neat, plausible and wrong."

H.L. Mencken

In this chapter, the theoretical and practical basis is described. For the Micrograph Quality Checker, the methods used to process images and to extract statistical metrics about them are introduced as well as the machine learning framework employed for the classification approach. For CowGraCE, the established mathematical methods that the density gradient simulation is based on are presented. Afterwards, an overview over the used programming languages, hardware and software is given, as well as an overview over the COW framework. As a last part, considerations in preparation of test data are presented.

2.1 Mathematical Tools for Image Processing

For cryo-EM image processing in general and in the quality judgment framework that is the MQC specifically, a lot of well-known mathematical operations are used and combined. In the following section, image statistics and associated operations, Fourier transformations and used image filters are described.

2.1.1 Image Statistics and Normalization

2.1.1.1 Mean

The first step to image statistics is localizing the image mean, which is a metric to describe the distribution of pixel values in an image or a stack of images. For n pixel values $p_1, ..., p_n$, the mean will be a number that lies between the minimum and maximum values of p. The simplest way to calculate the pixel value mean of an image is the Arithmetic mean:

$$\overline{p}_{arithmetic} = \frac{1}{n} \sum_{i=1}^{n} p_i \,. \tag{2.1}$$

While being a decent measure of distribution tendency, the disadvantage of the arithmetic mean is that it's prone to distortion by statistical outliers, therefore not making it a robust statistic[94]. This is often dealt with by adjusting pixel value outliers before relying on statistical measures.

Another way of approximating the mean is the Median, which is defined as the value from the dataset that has the property that a given pixel value is equally likely to be above or below it. Given $p_1, ..., p_n$ to be ordered ascendingly, it is defined as

$$\overline{p}_{median} = p_{\frac{n+1}{2}} , \tag{2.2}$$

for when n is odd, and is defined for an even n as

$$\overline{p}_{median} = 0.5 \left(p_{\frac{n}{2}} + p_{\frac{n+1}{2}} \right) . \tag{2.3}$$

2.1.1.2 Variance and Standard Deviation

While knowing the mean can give an idea about the approximate expectation value of a sample pixel value of an image, it is equally important to look at the spread of values. The standard deviation is a dispersion parameter that describes the average deviation of the dataset from its mean:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} (p_i - \overline{p})^2}{n}} . \tag{2.4}$$

The variance is defined as σ^2. The squaring process in the calculation is necessary to eliminate negative values, as only the distance from the mean is relevant. Computationally it is advantageous to calculate the variance in a single iteration, which can be achieved by using the alternative formula

$$\sigma^2 = \left(\frac{\sum_{i=1}^{n} p_i^2}{n} \right) - \overline{p}^2 . \tag{2.5}$$

Due to the large images routinely having more than 16 million pixels, variance calculation can easily reach numerical limits of floating point numbers, therefore different computational strategies are commonly used to avoid the problem [30].

2.1.1.3 Normal Distribution

A normal - also called Gaussian - distribution is the most common continuous probability distribution. Random variables that are perturbed from the mean value by a quantity of independent, small influences, are said to be normal distributed as long as the number of samples is high enough. This can be said for pixel values in electron microscopy after extreme outliers caused by imaging defects are eliminated. The normal distribution's probability density is defined as

$$f(p|\overline{p}, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(p-\overline{p})^2}{2\sigma^2}} . \tag{2.6}$$

The fractional term regularizes the function so that the area under the curve is always 1. Applied to problems that are non-probabilistic in nature, the fractional term can also be replaced by a free parameter A to regulate the peak height. Different normal distributions sharing the same mean are shown in figure 2.1.

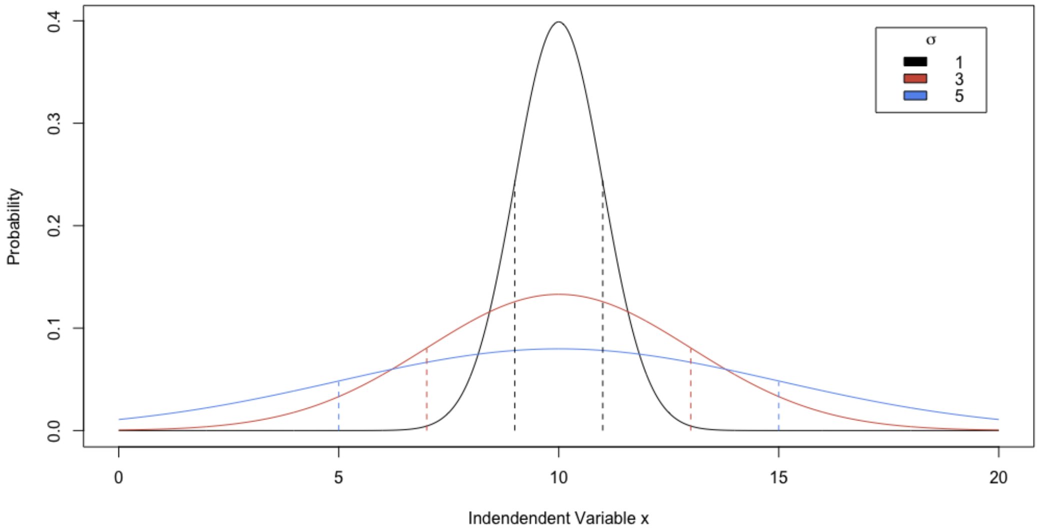


FIGURE 2.1: Gaussian distributions with a mean of 10 and different standard deviations. The dotted lines mark the area between $\overline{p}-\sigma$ and $\overline{p}+\sigma$ where 68.3% of values for the distribution are expected to be found.

The so called 68.3–95.5–99.7 rule[139] describes the probability Pr of values to fall within a certain interval around the mean:

$$\begin{aligned} Pr(\overline{p}-1\sigma \leq p \leq \overline{p}+1\sigma) &\approx 0.6827\,, \\ Pr(\overline{p}-2\sigma \leq p \leq \overline{p}+2\sigma) &\approx 0.9545\,, \\ Pr(\overline{p}-3\sigma \leq p \leq \overline{p}+3\sigma) &\approx 0.9973\,. \end{aligned} \tag{2.7}$$

2.1.1.4 Image Normalization

Due to differing conditions during the imaging process (for example electron dose, ice thickness, defocus, microscope model) and also due to differences in the sample being imaged, a recorded dataset can be vastly different from another one and also show considerable variations in itself. This can be problematic in the self-referential image processing workflow where images are compared and added to one another. Therefore, image normalization is a steadfast at the beginning of data processing and also needed during intermediate steps. Also, it is a computational requirement to keep the pixel value range within certain intervals to prevent calculations failing due to reaching numerical data type limits.

The normalization process involves adjusting mean as well as variance through the following formula:

$$I_{norm}(x,y) = \frac{\sigma^2_{new}}{\sigma^2_{old}}\left(I(x,y) - \overline{p}_{old} + \overline{p}_{new}\right)\,, \tag{2.8}$$

where I_{norm} is the normalized image, $\sigma^2_{new}, \sigma^2_{old}$ are the desired new and the calculated old variance and $\overline{p}_{new}, \overline{p}_{old}$ are the desired new mean and the calculated old mean.

Due to floating point representation of pixel values having limits in terms of accurate value distance representation without information loss, heuristics are usually used to decide on optimal normalization parameters for a newly acquired dataset. Commonly, $\overline{p} = 0$ and $\sigma^2_{new} = 10$ are used.

Another possibility is to normalize to a certain pixel value range instead of selection of $\overline{p}$ and σ^2_{new}. Here, the normalized image is calculated in the following way:

$$I_{normRange}(x,y) = \frac{I(x,y) - min_{old}}{max_{old} - min_{old}}(max_{new} - min_{new}) + min_{new}\,, \tag{2.9}$$

where min_{old}, max_{old} are the calculated minimum and maximum pixel values of the input image and min_{new}, max_{new} are the selected new minimum and maximum pixel values.

2.1.2 Image Contrast Enhancement

2.1.2.1 Pixel Value Histogram Equalization

Due to the set number of gray values, how pixel values are distributed between the extreme values in a histogram of the 0-256 range affects how much contrast is visible to the human eye: values that are close together are shown in similar colors and are therefore hard to distinguish.

A gray scale histogram is defined as

$$H(g) = \sum_{y=0}^{y_{max}} \sum_{x=0}^{x_{max}} u(x,y), \tag{2.10}$$

$$u(x,y) = \begin{cases} 1, & \text{if } round(I(x,y)) = g \\ 0, & \text{else} \end{cases}\,.$$

where H is the one dimensional histogram of the pixel values, g is the pixel value bin, x, y are the image coordinates and u is the binary scoring function that rounds pixel values to the next integer.

Based on the histogram, the cumulative distribution function sums up the member count of all histogram bins up to the current value:

$$cd(g) = \sum_{i=0}^{g} H(i)\,. \tag{2.11}$$

Here, cd is the aforementioned cumulative distribution function at the gray scale bin g. A perfectly evenly distributed image would have a linear cd. To increase contrast, the distribution of pixel values can be linearized to maximize optical differences between values[175]:

$$H_{equal}(g) = round\left(\frac{cd(g) - cd_{min}}{n - cd_{min}}(M-1)\right)\,. \tag{2.12}$$

Here, H_{equal} is the equalized histogram at gray scale position g, cd_{min} the amount of values in the very first bin of the cumulative distribution, n is the number of pixels the image has and M is the number of histogram bins used.

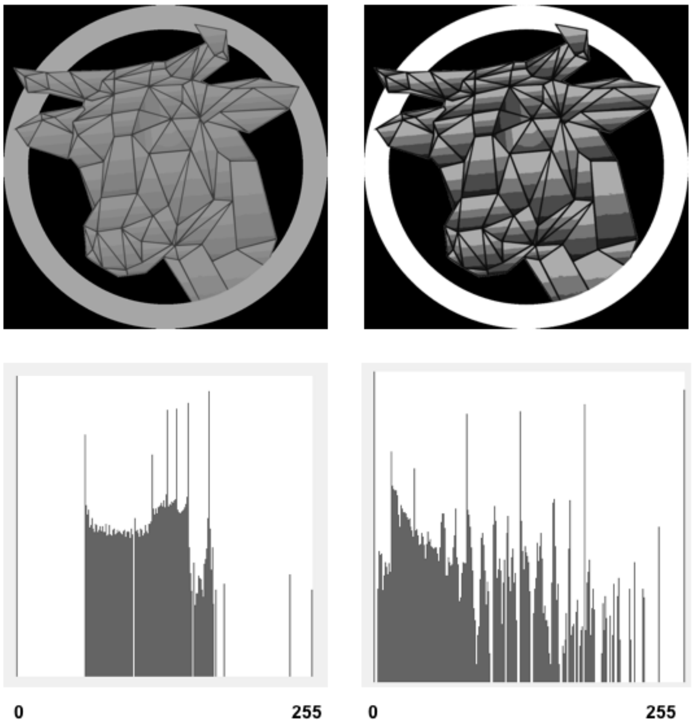


FIGURE 2.2: Histogram equalization effects. Shown in the left column is the COW logo and its original pixel value histogram. Shown on the right is the same image and its pixel value histogram after application of the histogram equalization procedure.

Figure 2.2 shows the contrast enhancement of an image through histogram equalization. While suitable for data processing steps where visual user input is necessary, histogram equalization is not recommended in combination with algorithmic image operations, as this process irreversibly manipulates the image information, rendering them unsuitable for several processing steps.

2.1.2.2 Coarsing

Coarsing of an image is a method that consists of compressing the pixel information by averaging neighboring pixels in a square kernel with sidelength k and therefore reducing the image size by k^2:

$$I_{coarsed}(x,y) = \frac{1}{k}\sum_{i=0}^{k}\sum_{j=0}^{k} I(x+i,y+j)\,. \tag{2.13}$$

This operation results in a visible noise reduction and contrast increase. As a side effect, while the highest resolution of image information is also decreased by the factor k, calculations involving the image are sped up, which can be advantageous for computationally expensive and low frequency focused operations like image alignment and classification. Figure 2.3 illustrates the effect of coarsing.

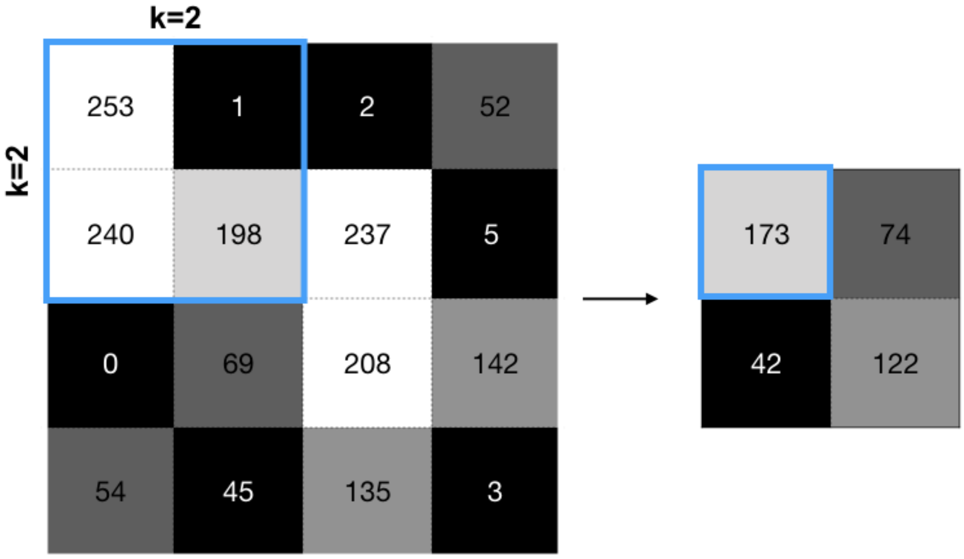


FIGURE 2.3: Effects of coarsing. Shown on the left is a 4 x 4 pixel image with a simplified gray scale of 5 possible values between white and black. The pixel values between 0 and 255 are noted for each pixel. A sample coarsing kernel of 2x2 pixel is drawn in blue. On the right the coarsed image is shown, including the resulting pixel from the area inside the blue kernel.

2.1.3 Fourier Transformations

Fourier transformation (FT) is a linear decomposition operation that approximates a function through a combination of sine waves with different amplitudes, frequencies and phases in its reciprocal domain. In its most well known application, a 1D time dependent function is mapped to the frequency domain. First developed in the 19th century[56, 64], it is defined as

$$\begin{aligned} F_{cont}(k) &= \int_{-\infty}^{\infty} f(x) e^{-2\pi i k x} dx\,, \\ f(x) &= \int_{-\infty}^{\infty} F_{cont}(k) e^{2\pi i k x} dk\,. \end{aligned} \tag{2.14}$$

where $f(x)$ is an integratable function of the independent variable x, $F_{cont}(k)$ is the continuous Fourier transform with the transform variable k in inverse units of x.

To calculate a FT of an image, the non-continuous information of the pixel values must be considered. Therefore, 2.1.3 must be rewritten for the discrete case as

$$\begin{aligned} F_{dis}(k) &= \sum_{k=0}^{N-1} f(x) e^{\frac{-2\pi i k x}{N}}\,, \\ f(x) &= \sum_{k=0}^{N-1} F_{dis}(k) e^{\frac{2\pi i k x}{N}}\,. \end{aligned} \tag{2.15}$$

where $F_{dis}(k)$ is the 1D discrete Fourier transform (DFT) and N is the number of x values of the input 1D function $f(x)$. For a 2D image, the DFT is calculated for each row of pixels and each column of pixels:

$$F_{disc,2D}(k,l) = \sum_{x=0}^{S-1}\sum_{y=0}^{S-1} f(x,y) e^{\frac{-2\pi ikx}{S}} e^{\frac{-2\pi ily}{S}} , \tag{2.16}$$

$$f(x,y) = \frac{1}{S^2}\sum_{x=0}^{S-1}\sum_{y=0}^{S-1} F_{disc,2D}(k,l) e^{\frac{2\pi ikx}{S}} e^{\frac{2\pi ily}{S}} .$$

Here, $F_{disc,2D}(k,l)$ is the DFT of the image $f(x,y)$ with side length S. $\frac{1}{S^2}$ is a correction term to re-normalize the pixel values that could also be applied during forward transformation. A 2D DFT has the symmetric properties

$$F_{disc,2D}(k,l) = {}^{*}F_{disc,2D}(-k,-l) , \tag{2.17}$$

where ${}^{*}F_{disc,2D}(-k,-l)$ is the complex conjugate of $F_{disc,2D}(k,l)$. This property reduces the amount of values to be calculated by the factor 2.

Naive calculation of a 2D DFT results in an algorithmic complexity of $O(N^2)$, where N is the number of pixels in the image. The Fast Fourier Transform(FFT)[192] that is based on factorization of the DFT matrix into sparse factors that are mostly 0 reduces the runtime to $O(N\,log(N))$. The FFT was used in the work of this thesis.

2.1.3.1 Non-Square Images in Fourier Space

The recent popularity of the Gatan K2 camera poses challenges for Fourier transformations that operate on square images. To get around this issue, two strategies are possible:

1. Cropping the image to square dimensions. Due to the irreversibility of the process, the input image needs to be copied and kept in memory to prevent data loss.
2. Padding the image to square dimensions. Here, the added pixels are either filled with static values (usually zero) or with the image's mean pixel value. The latter option is preferable due to the former option introducing edges in the Fourier space image. After transformation, the previously added pixels can be cut off again, restoring the original image.

The second strategy was used in the work of this thesis, where applicable.

2.1.3.2 Convolution

Convolution is an important and often-used mathematical operation in image processing, specially when filtering images. The convolution of two functions f and g, denoted as $f * g$, is defined as the integral of the two functions' product after shifting and reversing either one[170]:

$$(f * g)(x) = \int_{-\infty}^{\infty} f(\tau)g(x-\tau)d\tau \; = \int_{-\infty}^{\infty} f(x-\tau)g(\tau)d\tau . \tag{2.18}$$

The convolution theorem [205] states that a convolution of two functions can be expressed as the inverse FT of the pointwise product of the functions' FT:

$$(f * g)(x) = F^{-1}\left(F(f) \times F(g)\right) . \tag{2.19}$$

Application of this theorem allows efficient implementation of algorithms containing convolution operations by reducing computational complexity.

2.1.3.3 Correlation

The correlation of two functions f and g, denoted $f \star g$, is also a commonly used operation, specially when comparing images to one another. Bearing some resemblance to convolution, it is defined as the integral of the functions' product:

$$(f \star g)(x) = \int_{-\infty}^{\infty} f(\tau)g(x+\tau)d\tau \; = \int_{-\infty}^{\infty} f(x+\tau)g(\tau)d\tau \, . \tag{2.20}$$

The correlation theorem [205] then states that the correlation of two functions can be expressed as the inverse FT of the pointwise product of one function's FT with the other function's complex conjugate of it's FT:

$$(f \star g)(x) = F^{-1}\left(F(f) \times {}^{*}F(g)\right) . \tag{2.21}$$

Again, this theorem allows for a reduction of computational complexity. Correlating two same-sized images with one another results in an image of the same size where every pixel value corresponds to the cross-correlation-coefficient (CCC) for that pixel value. This corresponds to the similarity of the images to one another when shifted by the coordinates of the pixel. The CCC is one of the most important metrics in cryo-EM image processing, specifically during CTF correction, alignment and refinement.

2.1.3.4 Power Spectra

For ease of visualization and for specific operations where the phase information isn't needed, a DFT of an image can be transformed into a power spectrum, or power spectral density (PSD). This operation consists of squaring every complex frequency component:

$$PSD(u,v) = |F(u,v)|^2 \, , \tag{2.22}$$

where u and v are the spatial coordinates.

To increase contrast by attenuating the power differences and to reduce noise influence[187], a power spectrum is commonly normalized through logarithmation:

$$PSD_{norm}(u,v) = \ln(|F(u,v)|^2) \, . \tag{2.23}$$

Due to the irreversible squaring operation, the PSD is a real space image where the phase information is lost. Figure 2.4 shows an example power spectrum.

2.1.4 Real Space Image Filters

2.1.4.1 Boxcar Filter

A boxcar filter is a moving average filter, transversing the input image with a sliding window in which all pixels in it are replaced by the average of the window. This

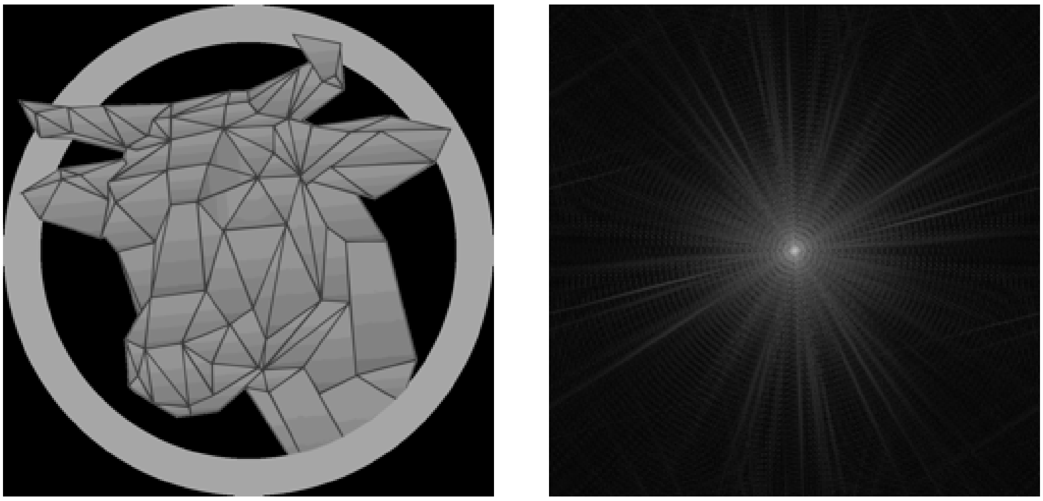


FIGURE 2.4: Power spectrum of a sample image. Shown on the left is the original image, shown on the right is its power spectrum, displaying the symmetric nature of Fourier transforms.

results in a blurring effect where high resolution information is lost, which is useful for noise reduction.

Mathematically, it is a convolution of an image with a mask:

$$I^* = I * M \,. \tag{2.24}$$

Here, I^* is the resulting filtered image, I is the input image and M is the Boxcar mask. At a specific point in the image, it is defined as

$$\begin{gathered} I^*_{x,y} = \sum_{i=-m}^{m} \sum_{j=-n}^{n} M_{i,j} \, I_{x-i,y-j} \,, \\ with \, i \in \{-m, ..., m\}, j \in \{-n, ..., n\} \,. \end{gathered} \tag{2.25}$$

where x and y are the image coordinates, i and i are the kernel coordinates and m and n are the kernel dimensions. The effect of a boxcar filter on a sample image is shown in figure 2.5.

2.1.4.2 Difference Of Gaussians Filter

A Difference of Gaussians filter (DoG) is commonly used for low frequency feature enhancement. The original image is blurred by Gaussian kernels with different standard deviations, the results of which are subtracted from one another [115]. Mathematically, it's a subtraction of two convolutions:

$$I^*_{x,y} = I * \left(\frac{1}{2\pi\sigma_1^2} e^{-(x^2+y^2)/(2\sigma_1^2)} - \frac{1}{2\pi\sigma_2^2} e^{-(x^2+y^2)/(2\sigma_2^2)} \right) \,. \tag{2.26}$$

The effect of a DoG filter on a sample image is shown in Figure 2.5.

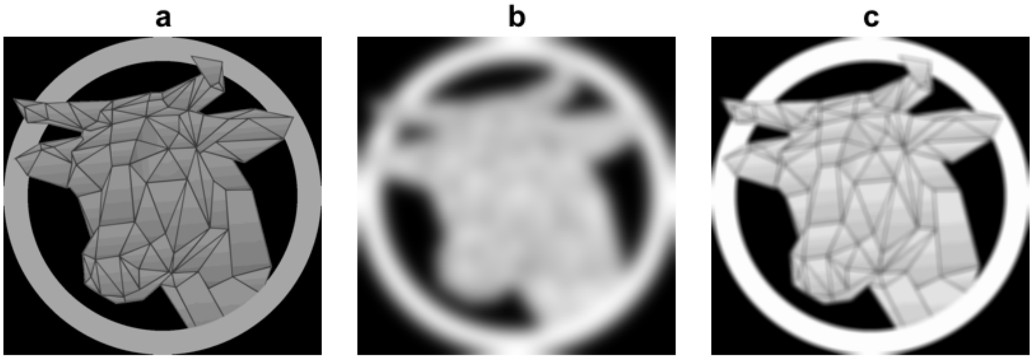


FIGURE 2.5: Effects of real space filters on a sample image. a: COW Logo, b: DoG filtered image with narrow standard deviation 1 and wide standard deviation 6, c: Boxcar filtered image with a 20 x 20 pixel kernel. The blurring effect of both filters is visible in the filtered images.

2.1.5 Fourier Space Image Filters

Similar to real space filters, Fourier space filters consist of convoluting an image with a filter mask. In this case, the filter mask additionally contains an imaginary part that only holds zeros, though, so the mask can be seen as only half-complex. Fourier space filters can be utilized to eliminate undesired frequency information from the input image.

Figure 2.6 illustrates the effect of the three different types of commonly used Fourier space image filters. In general, as filter masks, either binary masks or Gaussian gradients can be used. In the first case, zeros mark frequencies that are eliminated, while ones mark frequencies to pass the filter. This harsh cut-off can result in artifacts, so the latter option is often used, where the a transition between passing and non-passing frequencies are smoothed out through use of a gradient.

Filter values as used in this section are relative values between 0 and 1, with 1 denoting the highest frequency information, the Nyquist frequency [125], and 0 the lowest frequency.

2.1.5.1 Lowpass Filter

Lowpass filtering an image results in elimination of frequencies higher than the cut-off value. High frequencies correspond to fine structural details, but also to noise, resulting in an information tradeoff. Overall, a blurring effect is introduced to the image, with larger details being kept intact, i.e. the shape of a protein complex.

2.1.5.2 Highpass Filter

Highpass filtering an image eliminates frequencies lower than the cut-off value. Therefore, it is an inverse lowpass filter operation. Removing low frequencies corresponding to large structural details results in a visual sharpening effect.

2.1.5.3 Bandpass Filter

A bandpass filter combines a lowpass and a highpass filter. This is the most commonly used filter at the start of image processing, through which undesirable noise and irrelevant low frequency information are eliminated from the image.

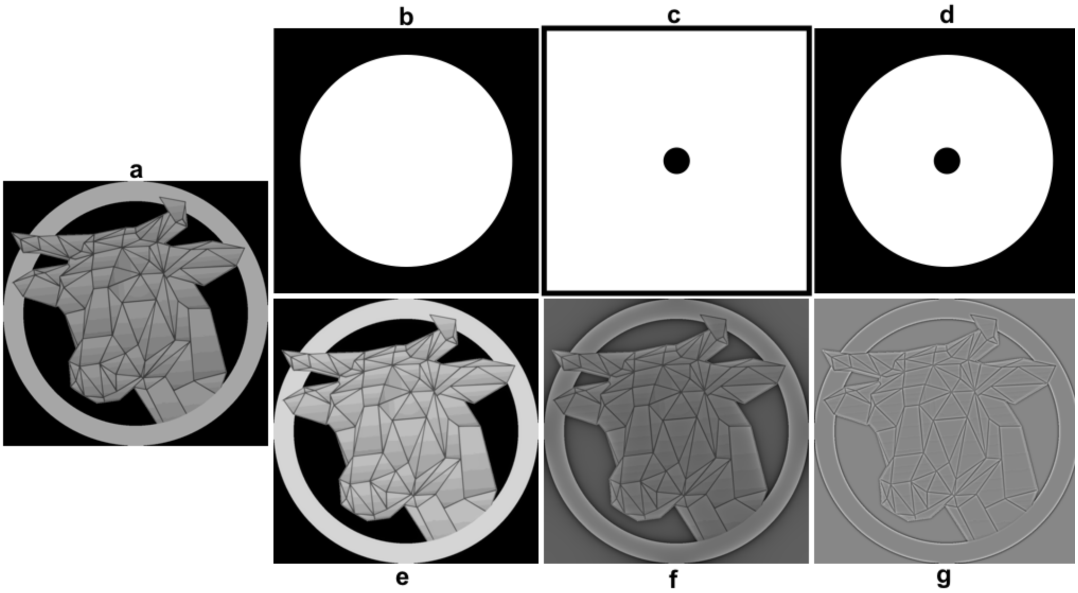


FIGURE 2.6: Illustration of binary Fourier filter masks and their effects on a sample image. In b-d, white areas indicate transmission by the filter and black areas indicate non-transmission, in e-g, effects of the filters on an image are shown. a: Original COW logo, b: Lowpass filter mask, threshold 0.8, c: Highpass filter mask, threshold 0.1, d: Band pass filter mask, highpass threshold 0.1 and lowpass threshold 0.8, e: Image from a after applying filter mask from b, f: Image from a after applying filter mask from c, f: Image from a after applying filter mask from d

2.2 Machine Learning: Support Vector Machines

Machine learning, first called that way in 1959[151], is a flexible collection of techniques to enable a computer to perform a specific task based on provided data. The two big categories within the field are:

1. **Unsupervised learning**: Here, the task of the computer is to find some sort of structure in the input data[5]. An example for methods from this category in cryo-EM is PCA[98, 74].

2. **Supervised learning**: In this case, the computer is provided with examples of the desired outputs of a set of inputs, which changes the task to finding the function that maps the input to the output[5]. In this thesis, only techniques from this category are used.

From the broad range of applications, the work of this thesis only includes classification problems, where a multidimensional input is assigned to 2 or more classes[5]. Specifically, here, users make decisions about the quality of recorded TEM images and through a supervised learning classification of a subset of the data the rest of the data is predicted.

The classification method of choice for this thesis are support vector machines (SVM), which is a binary, non-probabilistic classifier that is based on feature metric vectors[37] and uses linear algebra for structure detection in the data. When working with images, this has a lot of advantages due to reduction in computational complexity through omission of working with large micrographs as input data. In the following section, the mathematical basis of this algorithmic approach is described.

2.2.1 Pattern Classification

The given training data has the form

$$S = ((\vec{x}_1, y_1), ..., (\vec{x}_m, y_m))\,, \tag{2.27}$$
$$\vec{x}_i = (x_1, ..., x_n)^t,\; y = \begin{cases} +1 \\ -1 \end{cases} .$$

where S is a training set consisting of m n-dimensional vectors $\vec{x}_i$ and labels y_i that are either 1 or -1.

The objective of the classification approach is to estimate a linear function $f : \Re^n \rightarrow -1, +1$ that has the following form:

$$f(\vec{x}) = \sum_{i=1}^{n} w_i x_i + b\,, \tag{2.28}$$

where $\vec{w} \in R^n$ and $b \in R$ are the function's free parameters that are explained in the following section. Based on this, the decision function $d(\vec{x})$is then defined as

$$d(\vec{x}) = sgn(f(\vec{x})) = \begin{cases} +1 & \text{if } f(\vec{x}) \geq 0 \\ -1 & \text{else} \end{cases} . \tag{2.29}$$

2.2.2 Separating Hyperplanes

Hyperplanes, subspaces with one dimension less than their ambient spaces, can be used as a method of estimating linear decision boundaries that separate classes from one another. Figure 2.7 shows 20 data points belonging to two classes that can be classified through a large number of hyperplanes that are defined like in the Hesse normal form[21] as

$$\vec{w} \cdot \vec{x} - b = 0\,, \tag{2.30}$$

where $\vec{w}$ is the normal vector to the hyperplane and b defines the offset from the origin through $\frac{b}{||\vec{w}||_2}$. The functional distance γ_i of a point $(\vec{x}_i, y_i)$ from the training set to a hyperplane can be calculated as

$$\gamma_i = y_i(\vec{w} \cdot \vec{x}_i + b)\,. \tag{2.31}$$

$\gamma_i \geq 0$ signifies a correct classification, e.g. the point is on the correct side of the hyperplane. For the nearest points on both sides of the hyperplane, denoted $\vec{x}^+$ and

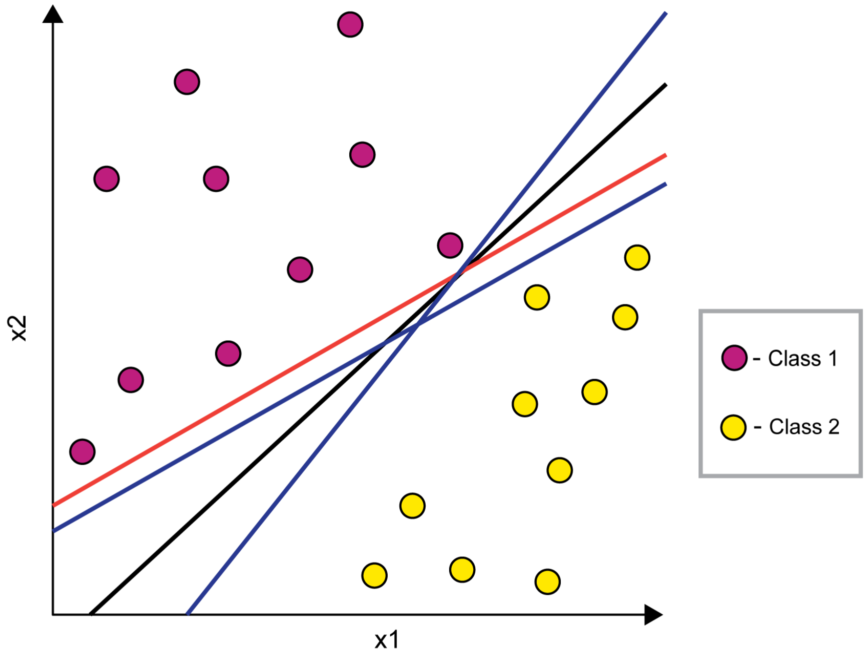


FIGURE 2.7: Binary classification problem with possible hyperplanes. A linearly separable dataset consisting of 20 points in in $\Re^2$ in two classes is shown. The colored lines are examples of the many possible hyperplanes to correctly classify them.

$\vec{x}^-$, the following expressions hold true [40]:

$$\begin{aligned} \vec{w} \cdot \vec{x}^+ + b &= 1\,, \\ \vec{w} \cdot \vec{x}^- + b &= -1\,, \\ \vec{w} \cdot (\vec{x}^+ - \vec{x}^-) &= 2\,, \\ \frac{\vec{w}}{||\vec{w}||_2} \cdot (\vec{x}^+ - \vec{x}^-) &= \frac{2}{||\vec{w}||_2}\,. \end{aligned} \tag{2.32}$$

It should be noted that through this way of approaching the problem of the optimal hyperplane, only the closest points to it are relevant. Those points are denoted the support vectors. As seen in equation 2.2.2, the margin of the data set to the hyperplane is inversely proportional to $||\vec{w}||_2$. The goal of the maximum margin classification is therefore to minimize $\langle w, w \rangle$ given the constraint

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1\,,\; i = 1, ..., m\,. \tag{2.33}$$

This can be solved through a Lagrangian function[40]:

$$L(\vec{w}, b, a) = \frac{1}{2}\langle \vec{w}, \vec{w} \rangle - \sum_i a_i \left(y_i(\langle \vec{w}, \vec{x}_i \rangle + b) - 1\right)\,, \tag{2.34}$$

where $a_i > 0$ is a Lagrange Multiplier[113]. Differentiation of this equation yields the dual form based on imposing the optimal conditions:

$$\frac{\partial L}{\partial \vec{w}} = \vec{w} - \sum_i y_i a_i \vec{x}_i = 0\,, \qquad (2.35)$$
$$\frac{\partial L}{\partial b} = \sum_i y_i a_i = 0\,.$$

Resubstitution into the primal form yields

$$L(\vec{w}, b, a) = \sum_{i=1}^{t} a_i - \frac{1}{2} \sum_{j,k=1}^{t} a_j a_k y_j y_k \langle \vec{x}_j^T, \vec{x}_k \rangle\,. \qquad (2.36)$$

This quadratic programming problem presents a global maximum with linear constraints and can be solved efficiently provided that a separating maximum margin hyperplane exists. A found solution has the property

$$\vec{w} = \sum_i y_i a_i \vec{x}_i\,, \qquad (2.37)$$

where the coefficients a_i are zero unless these points are what was defined as support vectors. Through this sparseness, the algorithmic complexity of the problem is reduced.

2.2.3 Soft Margin Classification

If the classes are not linearly separable, the previous approach cannot find a suitable hyperplane. To enable finding the best possible hyperplane, a slack variable ξ is introduced, which describes how far a wrongly classified point is away from the hyperplane. This modifies the constraint 2.33 to

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 - \xi_i\,,\ i = 1, ..., m\,, \qquad (2.38)$$
$$\xi_i \geq 0, C \geq \sum_i \xi_i\,.$$

The constant C is the cost parameter and can be freely chosen. It sets an upper limit to misclassifications. Nevertheless, the added complexity through allowed misclassifications will not always lead to a fast algorithm conversion in the input feature space. Figure 2.8 shows a soft-margin classification in $\Re^2$.

The method of choice for increasing classification performance in SVMs is the so called kernel trick [37]. Here, a non-linear mapping ϕ from the input vector space $\Re^n$ into a feature space F is used through a kernel function k that takes advantage of the inner product of 2.36 being data-only:

$$\phi : \Re^n \rightarrow F\,, \qquad (2.39)$$
$$k(\vec{x}, \vec{z}) = \phi(\vec{x}) \cdot \phi(\vec{z})\,.$$

Through this, the relative positions of the points are calculated and the maximal margin classifier is applied in the feature space. Figure 2.9 illustrates an ideal example of the described mapping. Ideally, the data will be linearly separable in the feature

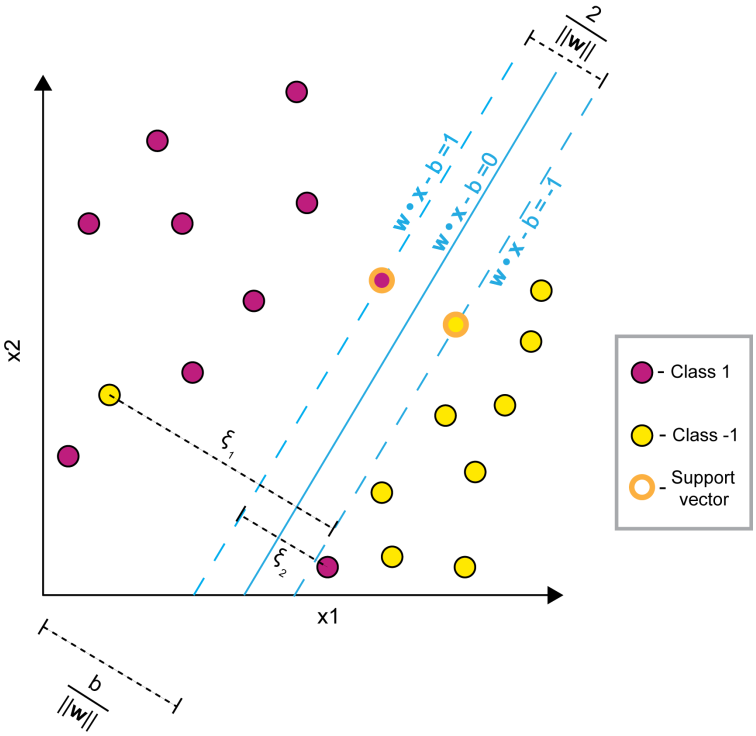


FIGURE 2.8: Soft margin SVM classification of a non linearly separable data set. A dataset consisting of 20 points in in $\Re^2$ in two classes is shown, together with the optimal hyperplanes and support vectors based on the soft margin SVM classification.

space. If this isn't the case, the approach of allowing misclassifications through ξ is used. Examples for commonly used kernels are:

1. Gaussian : $k(\vec{x}, \vec{z}) = e^{-\gamma||\vec{x}-\vec{z}||^2}$ for $\gamma > 0$. γ is a scaling parameter that can be chosen.
2. Homogeneous polynomial: $k(\vec{x}, \vec{z}) = (\vec{x} \cdot \vec{z})^d$. d is a freely choosable scaling parameter that determines the degree of the polynomial.

Ultimately, the decision function 2.28 can be written as:

$$f(\vec{x}) = \sum_{i=1}^{M} a_i y_i k(\vec{x}_i \vec{x}) + b \,. \tag{2.40}$$

In a possibly higher-dimensional feature space where some sort of linear separation is achievable, the cost parameter C regulates the amount of overfitting the input data: A small value of C encourages as little misclassifications as possible at the cost of a small margin to a found hyperplane while a large value will lead to more misclassifications, but a higher margin [76]. Depending on how representative the training data is of the underlying data space, different values for C will be preferential.

2.2.3.1 Feature Selection for SVM Classification

Despite the weighting capabilities of the SVM approach being fairly resilient to uninformative features, feature elimination may still be beneficial to increase the computational efficiency and to remove noisy information that may impair a model's

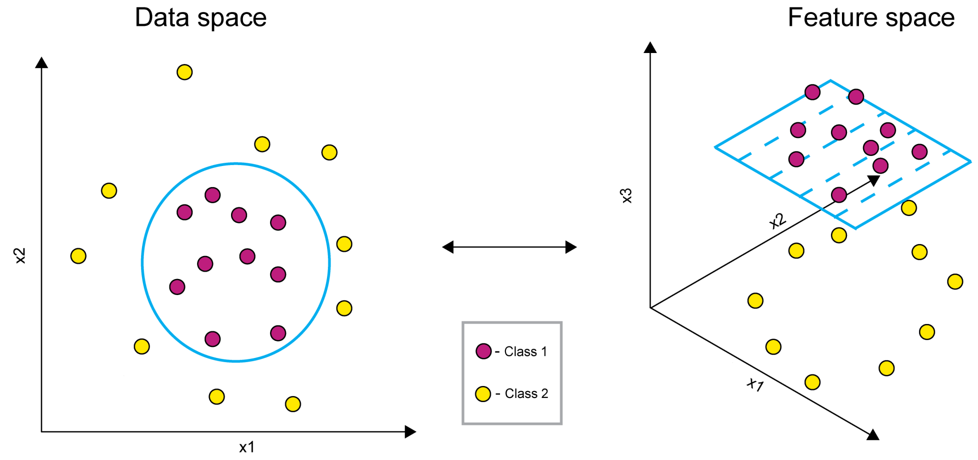


FIGURE 2.9: Linear separation in feature space. A dataset consisting of 20 points in in $\Re^2$ in two classes is shown that is linearly separable in feature space. The left image shows the SVM hyperplane in data space that was determined in feature space as shown in the right image.

performance. Specifically for a large amount of features, algorithms such as the recursive feature elimination are commonly applied[201]. Extensions to this algorithm include relevance ranking based on generalization error bounds sensitivity[143] and correlation-based approaches to remove redundant features[204], which can also be used as a feature selection method on its own. Here, a correlation matrix based on the pearson correlation coefficient PCC is created, which is defined as

$$PCC_{x,y} = \frac{cov(x,y)}{\sigma_x, \sigma_y} . \tag{2.41}$$

where $cov(x,y)$ is the covariance of the two variables and σ_x and σ_y are their standard deviations. Based on a given cut-off value, a feature is iteratively eliminated in every step, first considering the absolute values of pair-wise correlation and then considering the mean correlation to other variables until no pair-wise correlation exceeds the cutoff value, which is commonly chosen to be either 0.5 or 0.75.

2.2.3.2 Hyperparameter Tuning and Evaluating SVM Model Performance

Any sort of model evaluation is based on splitting the data set into a training set and a test set. Based on the predictions of the trained model on the test set, a confusion matrix can be used to assess the performance. The confusion matrix c for two classes, good and bad, is defined as shown in table 2.1.

From this, several quality metrics can be calculated, the most common being the accuracy Acc:

$$Acc = \frac{x_{++} + x_{--}}{x_{++} + x_{+-} + x_{-+} + x_{--}} . \tag{2.42}$$

In case of skewed class ratios, this measure is less informative due to classification of everything as the dominant class leading to a high accuracy. Further important

TABLE 2.1: Confusion matrix for two classes. x_{-+}, for example, is the amount of predictions where the good class $+$ was predicted while the real class was the bad class $-$.

	Real -	**Real +**
Predicted -	x_{--}	x_{+-}
Predicted +	x_{-+}	x_{++}

metrics derived from the confusion matrix are the false positive rate FPR and the false negative rate FNR:

$$\begin{aligned} FPR &= \frac{x_{-+}}{x_{-+} + x_{--}}, \\ FNR &= \frac{x_{+-}}{x_{+-} + x_{++}}. \end{aligned} \tag{2.43}$$

Also, the negative predictive value NPV and the precision or positive predictive value PPV can be calculated that signify how often a prediction is correct:

$$\begin{aligned} NPV &= \frac{x_{--}}{x_{+-} + x_{--}}, \\ PPV &= \frac{x_{++}}{x_{++} + x_{-+}}. \end{aligned} \tag{2.44}$$

In this thesis, FPR as the measure for how well negative images are recognized and PPV as the measure for how much of the images classified as good correspond to images labeled as good are the most relevant apart from general accuracy.

Hyperparameter tuning is commonly done via k-fold cross-validation[108, 199]. Here, the dataset is split into k parts of the same size. $k-1$ parts are used for training the data, while 1 part is used for testing purposes. Through an exhaustive or adaptive grid search[93], the optimal hyperparameters can be determined based on the root mean square error (RMSE) of the results. For every instance k of the cross-validation, the RMSE is calculated as

$$RMSE_k = \sqrt{\frac{\sum_l (x_{lk} - \hat{x_{lk}})^2}{N_k}}, \tag{2.45}$$

where N_k is the size of the prediction set and $\hat{x_{lk}}$ is the prediction for x_{lk}. The overall RMSE for a given hyperparameter set is then given by

$$RMSE_{total} = \sqrt{\frac{RMSE_1^2 + ... + RMSE_k^2}{k}}. \tag{2.46}$$

With a given set of hyperparameters, a model's performance can be assessed through the previously described metrics by training a model with the optimal hyperparameters on a training set and predicting the outcome of the test set. The separation ratio is usually chosen to be 70% training and 30% testing.

To get an idea about the model's behavior based on input size, a learning curve can be calculated [185]. A typical learning curve is shown in figure 2.10. Here, the prediction error within a training set growing in size is plotted together with the prediction error for a fixed validation set. The error within the training set is expected to grow for larger sizes in absence of overfitting while the error in the validation set is expected to be reduced. The error percentage that both curves converge to shows how good the applied model can be at the maximum. How soon this convergence is reached shows how large a training set for the model needs to be. Also shown in the figure is the human-level classification error that is underlying the labeled data.

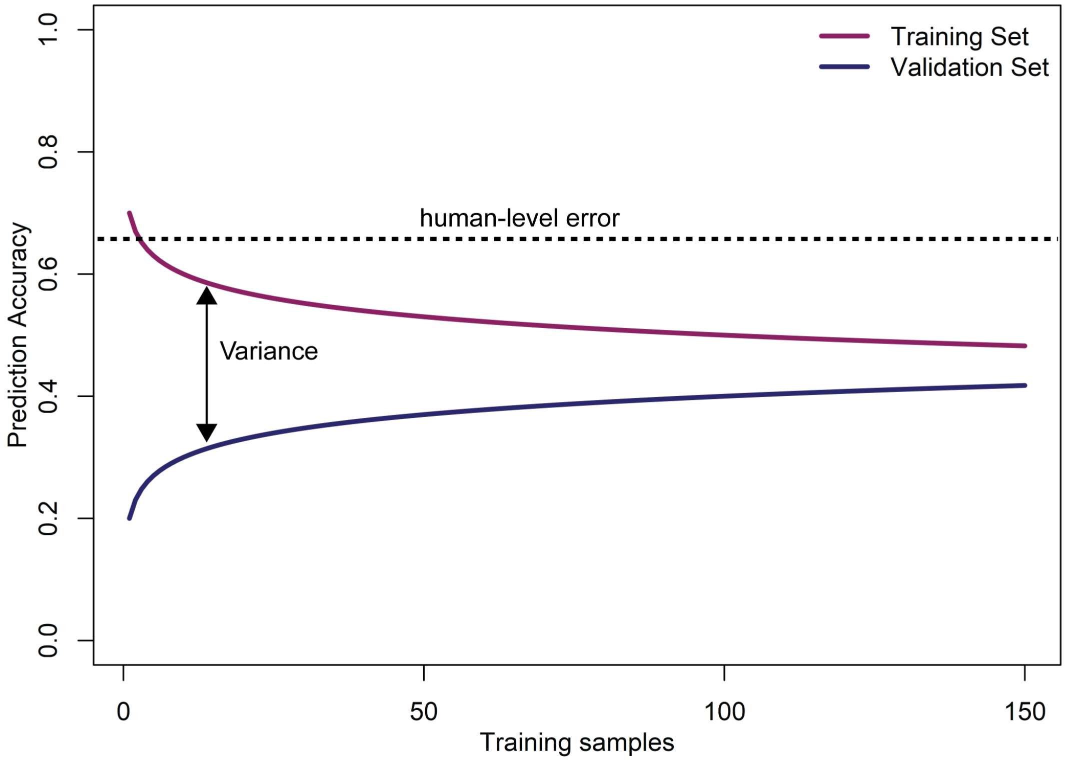


FIGURE 2.10: Typical learning curve. Based on a static test set size, the prediction accuracy within the training and the test set is shown as a function of training set size. The curves are expected to converge against each other, minimizing the variance, with growing training set size if no overfitting is taking place.

The human-level error bar and general reliability of the user-classified data can be assessed through use of Cohen's kappa coefficient κ[119], which measures the agreement between different human raters, but can also be used for the same rater making a repeated judgement, as is applicable in the work of this thesis. It is defined as

$$\kappa = \frac{p_0 - p_c}{1 - p_c}, \tag{2.47}$$

where p_0 is the relative agreement between the repeated judgments and p_c the probability that the agreement occurred by pure chance. Complete agreement would result in $\kappa = 1$, while complete disagreement would result in $\kappa = -1$. Several guidelines for interpretation of κ are found in the literature. Landis and Koch[112] proposed values between 0.41-0.6 as moderate, 0.61-0.8 as substantial and 0.8-1 as excellent agreement, while Fleiss[52] characterized values between 0.4-0.75 as fair to good and 0.75-1 as excellent agreement.

When the reliability of the dataset is confirmed through an acceptably high κ value, either κ or p_0 can be used to determine the human-level error.

2.3 Mathematical Tools for Density Gradient Centrifugation Simulation

In this section, the used methods involved in simulating rate-zonal centrifugation of a protein complex with approximately known molecular weight and unknown physicochemical properties in a linear gradient of either sucrose or glycerol are presented, as well as methods for determining experimental intensity distributions from SDS gels.

2.3.1 Calculation of Medium Density and Viscosity

As the sedimentation behavior of a particle depends on the medium's physicochemical properties, density and viscosity values for every fraction need to be known. Due to the lack of a comprehensive theory for the viscosity of mixed liquids, empirical data for the specific mixture is usually interpolated to yield fairly complex expressions.

2.3.1.1 Sucrose

Calculating the needed values for a sucrose-water solution can be achieved through polynomial expressions as proposed by Barber in 1966 [13]. Here, at a given temperature T, the density of a sucrose-water mixture $\rho_{Sucr,w}$ at a sucrose concentration c_{Sucr} (wt/wt) is calculated as

$$\rho_{Sucr,w} = C_{1,\rho} + \left(\frac{c_{Sucr}}{100} \cdot C_{2,\rho} + C_{3,\rho} \right) \cdot \frac{c_{Sucr}}{100} \,. \tag{2.48}$$

where $C_{1,\rho}$, $C_{2,\rho}$ and $C_{3,\rho}$ are constants that are defined as

$$\begin{aligned} C_{1,\rho} &= (-5.85 \cdot 10^{-6} \cdot T + 3.97 \cdot 10^{-5}) \cdot T + 1 \,, \\ C_{2,\rho} &= (1.24 \cdot 10^{-5} \cdot T - 1.06 \cdot 10^{-3}) \cdot T + 0.39 \,, \\ C_{3,\rho} &= (-8.92 \cdot 10^{-6} \cdot T + 4.75 \cdot 10^{-4}) \cdot T + 0.17 \,. \end{aligned} \tag{2.49}$$

The viscosity η_{Sucr} is calculated as

$$\eta_{Sucr} = 10^{\frac{C_{1,\eta}+C_{2,\eta}}{T+C_{3,\eta}}} \,, \tag{2.50}$$

where $C_{1,\eta}$, $C_{2,\eta}$ and $C_{3,\eta}$ are calculated based on the value Y :

$$\begin{aligned}
Y &= \frac{c_{Sucr}}{c_{Sucr} + 18.99 \cdot (100 - c_{Sucr})}\,, \qquad (2.51)\\
C_{1,\eta} &= ((((((4.59 \cdot 10^9 \cdot Y - 1.1 \cdot 10^9) \cdot Y + 1.03 \cdot 10^8) \cdot Y - 4.69 \cdot 10^6) \cdot Y \\
&\quad +1.05 \cdot 10^5) \cdot Y - 1.14 \cdot 10^3) \cdot Y + 9.41) \cdot Y - 1.5\,,\\
C_{2,eta} &= ((((((-5.5 \cdot 10^{11} \cdot Y - 1.35 \cdot 10^{11}) \cdot Y + 1.3 \cdot 10^{10}) \cdot Y - 6.07 \cdot 10^8) \cdot Y \\
&\quad +1.42 \cdot 10^7) \cdot Y - 1.69 \cdot 10^5) \cdot Y + 1.69 \cdot 10^5) \cdot Y + 2.12 \cdot 10^2\,,\\
C_{3,\eta} &= 146.07 - 25.25 \cdot \sqrt{1 + \left(\frac{Y}{0.07}\right)^2}\,.
\end{aligned}$$

2.3.1.2 Glycerol

In a glycerol-water mixture, the density $\rho_{Gly,w}$ and viscosity $\eta_{Gly,w}$ at a given temperature T of a given concentration c_{Gly} (wt/wt) can be calculated as proposed by Cheng in 2008 [35]. The density is approximated by

$$\rho_{Gly,w} = \rho_{Gly} \cdot c_{Gly} + \rho_w(1 - c_{Gly})\,. \qquad (2.52)$$

where ρ_w and ρ_{Gly} are computed as

$$\begin{aligned}
\rho_{Gly} &= 0.705 - 0.0017 \cdot T\,, \qquad (2.53)\\
\rho_w &= 1000(1 - \left(\frac{T-4}{622}\right)^{1.7})\,.
\end{aligned}$$

The viscosity can be calculated through a power function:

$$\eta_{Gly,w} = \eta_w^{\alpha} \cdot \eta_{Gly}^{1-\alpha}\,, \qquad (2.54)$$

where α is a weighting factor with a value between 0 and 1. It is calculated as

$$\alpha = 1 - c_{Gly} + \frac{a \cdot b \cdot c_{Gly}(1 - c_{Gly})}{a \cdot c_{Gly} + b(1 - c_{Gly})}\,. \qquad (2.55)$$

The two coefficients a and b are approximated by

$$\begin{aligned}
a &= 0.705 - 0.0017 \cdot T\,,\\
b &= (4.9 + 0.036 \cdot T) \cdot a^{2.5}\,. \qquad (2.56)
\end{aligned}$$

The two dynamic viscosities of the two components in the solution are interpolated as

$$\begin{aligned}
\eta_w &= 1.79^{\frac{-1230T - T^2}{36100 + 360T}}\,, \qquad (2.57)\\
\eta_{Gly} &= 12100^{\frac{-1233T + T^2}{9900 + 70T}}\,.
\end{aligned}$$

2.3.2 Calculation of Sedimentation Value Distributions

Multiple approaches for simulating distributions of sedimenting particles in a density gradient centrifugation experiment exist[92, 156, 178]. In this thesis, the indirect approach was used, where peak location and distribution are calculated separately.

The ladder can be approximated through particle flow estimation in predefined segments of the tube[178]. As this approach was avoided in this thesis in favor of a new method, no further elaboration is presented in the following.

Prediction of the peak position of a $s_{20,w}$ value is based on the integratable formula that was previously introduced as eq. 1.31 :

$$s_{20,w} \int_0^t \omega^2 dt = \frac{\rho_P - \rho_{20,w}}{\eta_{20,w}} \int_{r_{min}}^{r_{max}} \frac{\eta_{T,M}}{\rho_P - \rho_{T,M}} \frac{dr}{r} \,. \tag{2.58}$$

As proposed in the literature[18, 178], the equation can be evaluated numerically. The left integral can be solved from the known rotor values, the right integral can be approximated for every fraction. Through this, a $s_{20,w}$ value can be calculated for every fraction corresponding to a sedimented protein complex zone that has its peak concentration at that position.

2.3.2.1 Prediction of Sedimentation Coefficients based on Molecular Weight

In order to simulate a given sample's location in the tube after rate-zonal separation, its sedimentation coefficient needs to be known. With only knowledge of the molecular weight MW, s_{pred} can be approximated as[145]

$$s_{pred} = 2.42 \cdot 10^{-3} \cdot MW^{\frac{2}{3}} \,. \tag{2.59}$$

This assumes a globular protein. While no database of Svedberg values exists, comparisons shown in A.3 indicate that this formula slightly underestimates the sedimentation speed of proteins.

2.3.3 Extracting Intensity Distributions from SDS Gels

The fractions of a density gradient centrifugation run are commonly analyzed by applying them to individual lanes on a SDS gel. After staining with Coomassie Blue, the protein bands are visible in an image of the gel. The signal of a protein complex is usually split up into multiple bands.

To extract information about the intensity distribution of a sample, a semi-automatic approach can be used. Here, the image is converted to gray scale, the lane areas to be quantified are selected, and the area of the band intensity is summed up to the overall area of the protein complex at the specified fraction, called I, as shown in figure 2.11. As this absolute area information is error prone due to the non-normalized images, intensities relative to the highest determined area value are used and denoted as I_{norm}.

2.3.4 Fitting Intensity Distributions from SDS Gels

Under the assumption of a normal distribution, a Gaussian (see section 2.1.1.3) of the form

$$f(x) = A \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.60}$$

can be fitted to the determined relative intensity distribution by an iterative non-linear least squares method[180]. While this requires initial guesses for the three

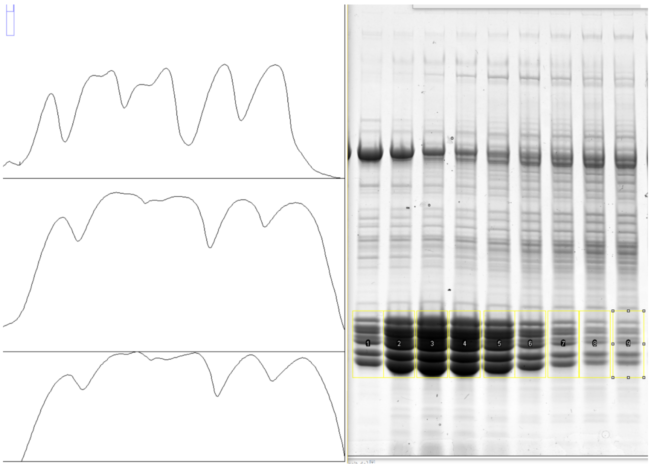


FIGURE 2.11: Intensity distribution extraction from a SDS gel. On the right, selected bands for quantification are shown. On the left, the 1D distributions of the first three bands are shown, where a background cut-off can be defined. Afterwards, the rest of the area is summed up to determine the intensity of the sample at the lane. The image was created from a h20S Proteasome SDS gel in ImageJ.

free parameters, knowledge about the intensity distribution can readily be used to determine an estimate.

2.4 Programming Languages and Frameworks

In this chapter, information about the languages that found application in software development and data analysis in the work of this thesis is given.

2.4.1 C++

C++ is an extension of the C language that is most widely known for its object orientation capabilities coupled with facilities for low-level memory management [116]. Developed by Bjarne Stroustrup and first released in 1985[179], its wide library support, constant development and code execution speed made it the language of choice for the computationally intensive demands of cryo-EM image processing and density gradient simulation.The aforementioned memory management, lack of garbage collection and direct compilation to assembler code make it possible for C++ code to be highly performant[53].

Initially, development of software of this thesis was started using the C++03 standard[95] and was later transferred to the C++11 version[96] that contains several improvements and convenience features over the older implementation.

2.4.1.1 Used C++ Libraries

Several external libraries were used for C++ development that each provided specific functionality.

QT is a popular library for cross-platform GUI development that is based on a signal and slot concept for information flow. Apart from supplying widgets and layouts,

TABLE 2.2: Used C++ libraries

Name	Version	Used In	Further information
Qt	5.10	CowGraCE, COW	www.qt.io
Boost	1.65	CowGraCE, COW	www.boost.org
Qwt	6.1	COW	www.qwt.sourceforge.net
RocksDB	6.1	COW	www.rocksdb.org
MKL	11.3	COW	www.software.intel.com/en-us/mkl
FreeImage	3.17	COW	www.freeimage.sourceforge.net/
OpenSSL	1.0.2	COW	www.openssl.org
VTK	7.0	COW	www.vtk.org

the library also includes a lot of IO, threading and Model-View-Controller(MVC) functionality, all of which was used in the work of this thesis.

Boost is an extension of the C++ standard library (STL) that supplies reference implementations of a wide range of functionality, a lot of which laters gets included in new STL versions. In the presented softwares, Boost libraries were used for system calls, time, date and threading use cases.

Qwt is an extension of the Qt Library that provides data visualization widgets which were used to develop 1D and 2D viewers in the COW(see 2.6.4).

RocksDB is library that provides a speed optimized key-value storage database. In the COW, it was used to generate project files(see 2.6.4.1).

MKL , the Intel Math Kernel Library, provides speed optimized implementations for mathematical operations. In the COW, it was used for Fourier transformations insert ref here and other arithmetic use cases.

FreeImage provides In- and Output functionality for common image formats(e.g png, jpeg, tiff). This was used in the COW for In- and Export of such files.

OpenSSL is a toolkit for TLS and SSL protocols. In the COW, it was used to enable secure connections to a server.

VTK, short for Visualization Toolkit, is a library for 3D Visualization. In the COW, it was used in the 3D Viewer where calculated structures can be displayed.

Further specifications can be found in table 2.2.

2.4.2 R

R can be seen as an extended implementation of the statistical S language[29]. The R environment, containing an integrated IDE called R Studio, is mostly used for data manipulation, analysis, visualization and statistical inference applications[194]. Due to the many built-in features and available packages, R was used in this thesis for rapid prototyping of algorithms, analyzing results and creating figures.

TABLE 2.3: Used C++ tool chains

OS	IDE	Compiler and Linker	CUDA Compiler
macOS	XCode 7.2	Apple LLVM 7.0.2, GNU Linker	-
Windows	Visual Studio 2017	Microsoft C++ Compiler 19.10.25017 and Linker 14.10.25017	Nvidia NVCC 9.0
CentOS 5	-	Gcc 4.1, GNU Linker	-

2.5 Hard- and Software

The two computers used for the work of this thesis were

1. **MacBook Pro Mid 2015** (2.8 Ghz quad-core Intel Core i7 processor, 16 GB 1600 MHz DDR3 Memory and an AMD Radeon R9 M370X 2048 MB GDDR5 Memory GPU) running macOS 10.11.6
2. **Dell Workstation** (3.7 Ghz quad-core Intel Core i7 processor, 32 GB 1600 MHz DDR3 Memory and Nvidia Quadro M4000 8096 MB GDDR5 Memory GPU) running Windows 10

On the MacBook, the native IDE XCode was used to write C++ code. On Windows, Microsoft Visual Studio was used. For Linux bundle building, a CentOS 5 docker image was used to enable backwards compatibility through an old enough libc version. Table 2.3 shows relevant specifications of the used development environments. Generation of solution files was performed through use of the CMake meta language on all operating systems.

Data analysis was mostly performed with R, using R Studio on both computers. Microsoft Excel was also used for specific data analysis. Intensity distribution extraction from SDS gels was done with ImageJ.

For version control and team development, Subversion (SVN) was initially chosen, but was later replaced with the more flexible Git[32] in the GitLab implementation running on a local server, for which GitKraken was the GUI of choice.

2.5.1 UML

The Unified Modeling Language(UML) is a way to visualize different properties of object-oriented software as well as information flow of operations[57, 9]. In this thesis, two types of diagrams in the notation of UML 2 are used, which are explained in the following. Figure 2.12 shows the used symbols for the diagrams.

2.5.1.1 Static Class Diagrams

A static UML 2 class diagram shows the relationship and dependencies between different components of an object-oriented software. Important parts of such a diagram are the classes themselves, which can have a set of attributes that are present in every instantiation of the class. The syntax for attributes is a minus operator at the start, followed by the attribute name and the data type after a colon, for example "- startIndex: Integer".

Also, every class can have a set of operations that can be performed by it. Here, the syntax is a plus operator followed by the operation name with function parameters and their datatypes in brackets. The return value of the function is specified

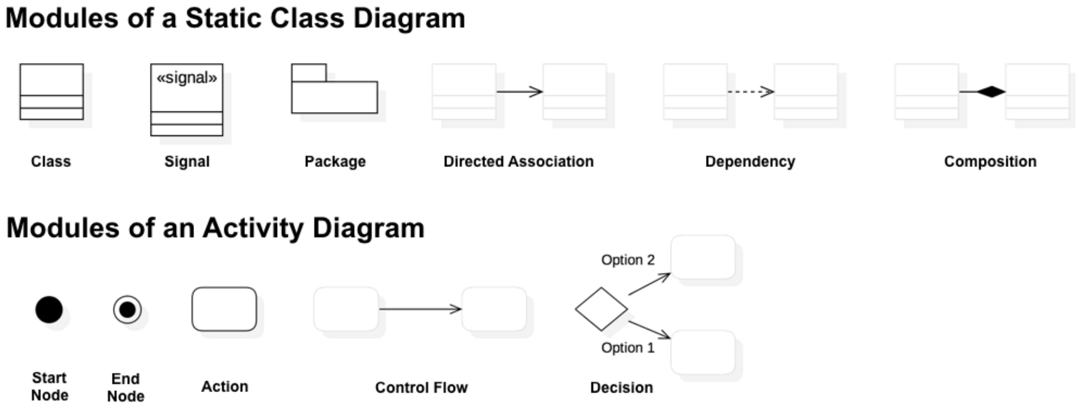


FIGURE 2.12: Modules of UML 2 diagrams. Shown are the used modules for static class diagrams, used to illustrate the relations between parts of object-oriented software architectures, and activity diagrams, used to give an overview over algorithms.

afterwards, again delimited by a colon. An example of a class operation would be "+getValue(index: integer): float".

Relationships among classes are shown through different symbols (see figure 2.12). Arrows indicate a directed association, while dotted arrows show dependencies. Lines with filled diamonds denote a composition, a has-a relationship.

Static UML diagrams as used in this thesis do not claim any sort of completeness but rather show relevant interactions between the most important components.

2.5.1.2 Activity Diagram

This kind of diagram can be used to describe information flow through a software, but also an algorithm. The beginning of an algorithm is shown through a start node and its finish through an end node. In between, two elements describe the operations, with directed arrows as control flow elements connecting them. An action is a general description of any operation that takes place, while a decision element splits up the flow into two or more paths.

2.6 The COW

2.6.1 History and Concept

The name COW is, despite that being surprising to some, not an acronym, but goes back to a german saying that Dr. Mario Lüttich, the main developer from the Stark Department since the project started in 2008, uttered when presented with a list of specifications the software should fulfill: "Das ist so viel Arbeit, das passt doch auf keine Kuhhaut". A literal translation would be "This is too much work to fit on a cow hide". From that point on, the software was called COW.

Since then, the COW has grown to a complex toolbox for all stages of cryo-EM image processing with an increasing amount of automatization capabilities, as described in the following sections. The general philosophy implemented in COW is to give experienced as well as inexperienced users capabilities to process their data in the most optimal way with high performance without having to leave the COW's ecosystem. Extendability(see 2.6.4.4) and scalability (see 2.6.5) are also main features of the COW framework.

All algorithmic and IO operations are made available through the CowLib, a dynamically linked library that is accessed by multiple front ends through its application programming interface (API), the most complex one being CowEyes, where particle stacks are processed to yield 3D structures. In the CowPicker, particles are extracted from micrographs through algorithmic means previously developed in the group[26].During the work of this thesis, another front end, the MicrographQuality-Checker, was added, which is introduced in section 3.2.

2.6.2 CowLib

The CowLib takes advantage of object-oriented development by having abstract base classes for every building block of a data processing step. Through this, method development is facilitated due to the ease of chaining existing algorithmic blocks together in new ways within a controlled development framework.

Figure 2.13 shows an overview of the main architectural pieces of the CowLib and their interactions. In the following sections, they are introduced in more detail.

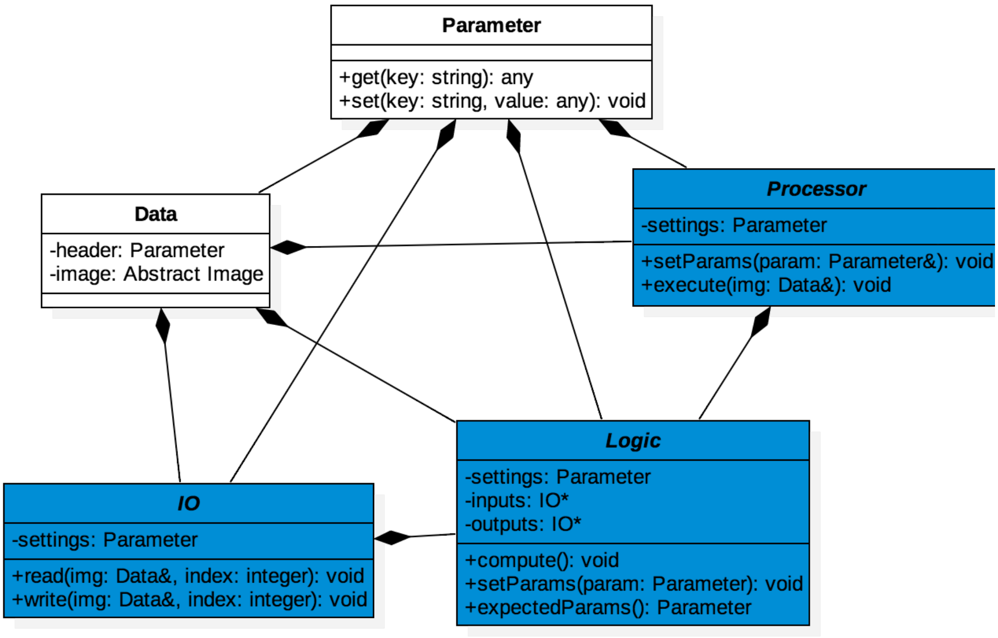

Parameter
+get(key: string): any
+set(key: string, value: any): void
Data
-header: Parameter
-image: Abstract Image
Processor
-settings: Parameter
+setParams(param: Parameter&): void
+execute(img: Data&): void
IO
-settings: Parameter
+read(img: Data&, index: integer): void
+write(img: Data&, index: integer): void
Logic
-settings: Parameter
-inputs: IO*
-outputs: IO*
+compute(): void
+setParams(param: Parameter): void
+expectedParams(): Parameter

FIGURE 2.13: Static UML2 class diagram showing the interaction of the main components of the CowLib. Abstract base classes are colored in blue.

2.6.2.1 Parameter Objects

cow::Parameter objects are arrays of key and value pairs. The key is always a std::string, the value can hold objects of type std::string, standard numerical types, other cow::Parameter objects and also lists of them, enabling tree-like structures. This flexible system enables the structure to hold image metadata and pass settings to logic objects. Furthermore, cow::Parameter objects are used in many internal information transfer operations in the CowLib.

2.6.2.2 Data Objects

At the core of an image processing software are the objects that hold images. A cow::Data object consists of either a CPU or a GPU image and the corresponding metadata in form of a cow::Parameter object. Images are ultimately an array of values that can be data types ranging from the standard numerical types to complex numbers. cow::Data objects also provide a list of operations, i.e. normalization, that can be performed on the contained image.

2.6.2.3 Logics

cow::Logic objects are the building blocks of image processing. Upon being started, a cow::Logic object receives a defined set of in- and outputs and needed variables as a cow::Parameter object. Within this base structure, algorithmic operations are performed on the input image set. In a cow::Logic, cow::Processor objects and cow::IO objects can be executed and other cow::Logic objects can be called.

2.6.2.4 Processors

As opposed to cow::Logic objects that can process a whole image stack (for example, comparing every image to all other images), a cow::Processor object can only execute an operation on one image at once. Therefore, these kind of objects are used for mathematical operations like Fourier transforms.

2.6.2.5 IO Objects

cow::IO objects are the data connecting entities in COW. As a first use case, files from the host system can be imported into the COW framework or exported from it. The most important supported file formats are:

1. MRC2014[34]
2. Imagic[190]
3. Common image formats such as png, jpg, tiff[54]
4. gnuplot[140]
5. star[70]
6. pdb[182]
7. cowIO

The cowIO format holds multiple cow::Data objects(see 2.6.2.2) and is used for internal in- and outputs. External files are read through a specialized IO that implements the abstract IO base class (see figure 2.13) and then subsequently made available as a cowIO.

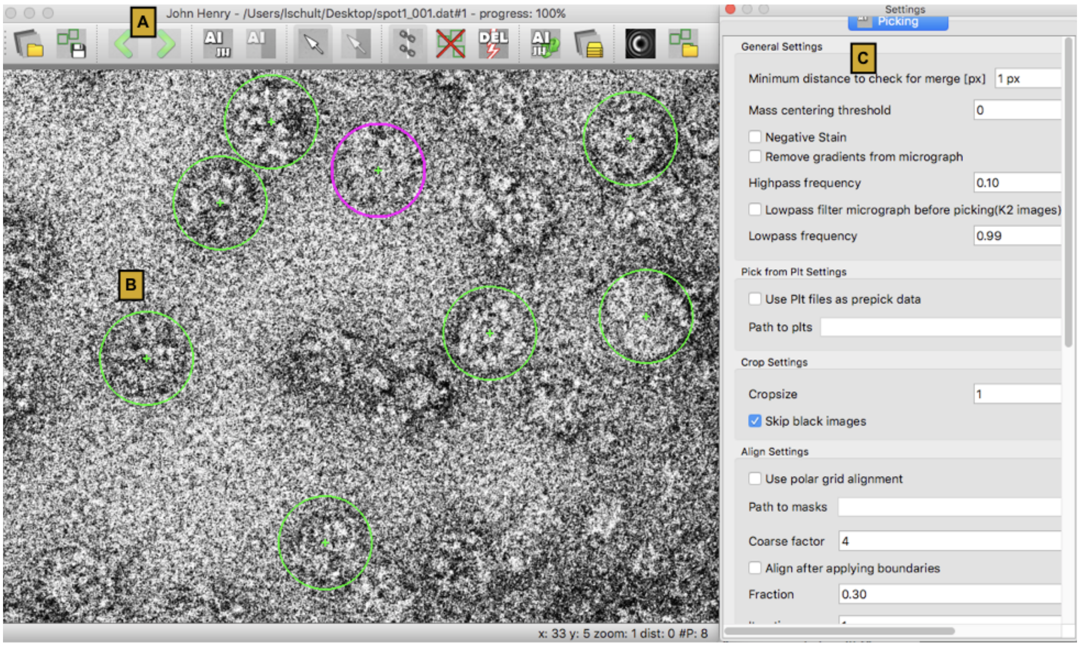
2.6.3 CowPicker

The CowPicker, previously titled John Henry, was developed by Dr. Boris Busche during his thesis work[26]. Maintenance duties were subsequently passed on to the author of this thesis. Until the Micrograph Quality Checker (see section 3.2) was created, the CowPicker was the entry point into the COW ecosystem.

The automatic picking algorithm is based on mass centering combined with a statistical approach. Through pixel value density analysis, places of interest are first defined on a regular grid and subsequently refined, e.g. moved around and merged, based on user inputs like particle size and merge distance. Through statistical thresholding, a subset of these assumed particles are then discarded [26]. Apart from the fully automatic mode, manual (with local mass centering) and semi-automatic modes are also possible. The advantage of this algorithm lies in being completely reference bias free as opposed to projection matching softwares[207].

Figure 2.14 shows the user interface of CowPicker. Apart from the described picking functionality, a lot of convenience functions for working with micrographs are made available.

Picking locations are either stored in a text file format called plt or used to directly crop out the particles and to concatenate them to an image stack that can then be imported into CowEyes, which is described in the next section, to calculate a 3D structure.



John Henry - /Users/lschult/Desktop/spot1_001.dat#1 - progress: 100%

x: 33 y: 5 zoom: 1 dist: 0 #P: 8

General Settings
Minimum distance to check for merge [px] 1 px
Mass centering threshold 0
☐ Negative Stain
☐ Remove gradients from micrograph
Highpass frequency 0.10
☐ Lowpass filter micrograph before picking(K2 images)
Lowpass frequency 0.99

Pick from Plt Settings
☐ Use Plt files as prepick data
Path to plts

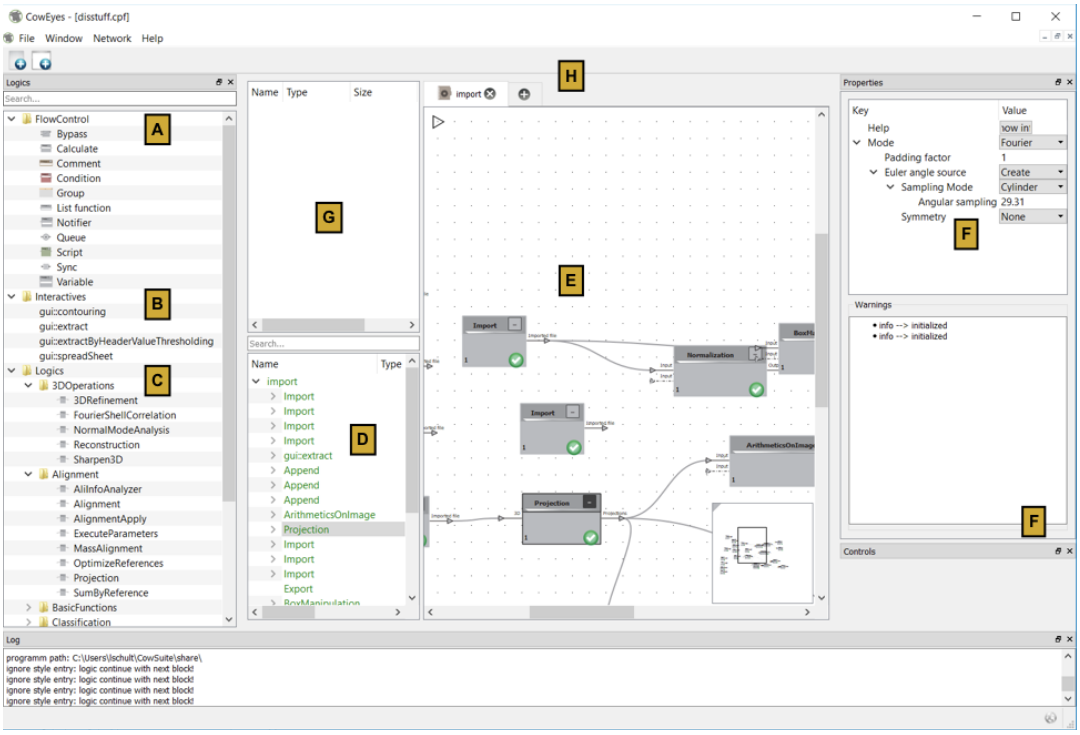
Crop Settings
Cropsize 1
☑ Skip black images

Align Settings
☐ Use polar grid alignment
Path to masks
Coarse factor 4
☐ Align after applying boundaries
Fraction 0.30

FIGURE 2.14: User interface of CowPicker. A: Toolbar, B: Current micrograph and the chosen particles (in green), C: Picking settings window. Shown is a micrograph containing ribosomal complexes.

2.6.4 CowEyes

While the data processing steps up to particle picking can be seen as a linear pipeline, the workflow from particle stacks to 3D structure not only consists of more essential algorithmic steps, but also requires non-linearity and situational tools, all of which are introduced in the following. To enable this, CowEyes provides a big amount of cow::Logic objects to the user that can be connected to one another in a workspace, managed in so called COW Project Files (cpf) (see 2.6.4.1). Results of a cow::Logic can be visualized through different viewers. For more complicated workflows, i.e. when a specific set of connected logics need to be run more than once until the results suffice as indicated by a specific metadata entry, visual programming workflow tools are provided(see 2.6.4.3). While the existing cow::Logic objects contain most commonly used algorithmic approaches in the cryo-EM field, the need for extendability is still given due to non-foreseeable future developments. Therefore, the CowPlug functionality((see 2.6.4.4) was developed, enabling external developers to program their own cow::Logic objects and use them within the workflow of CowEyes. The CowEyes GUI is shown in figure 2.15.



Key	Value
Help	ıow in
Mode	Fourier
Padding factor	1
Euler angle source	Create
Sampling Mode	Cylinder
Angular sampling	29.31
Symmetry	None

- info --> initialized
- info --> initialized

```
programm path: C:\Users\lschult\CowSuite\share\
ignore style entry: logic continue with next block!
ignore style entry: logic continue with next block!
ignore style entry: logic continue with next block!
ignore style entry: logic continue with next block!
```

FIGURE 2.15: User interface of CowEyes. A: FlowControl tools, B: Interactive logics, C: Logics, D: Run history display, E: Main area, F: Parameter field, G: Log area, H: Main area tabs, I: Quick access area.

2.6.4.1 Project Management

Due to the need to save intermediate results, continue data processing from a previous point and to allow portability of processing with the COW, CowEyes works on cow project files, which are based on the tree structure of connected cow::Logic objects that can be compared to a filesystem. Every project consists of a RocksDB database (see 2.4.1.1) that contains the position of the cow::Logic within the tree as a key and a cow::Parameter object as the value that contains all needed metadata. This is saved as a COW Project Files (.cpf), while the data that corresponds to the outputs of logics are saved in the CIO format in a subfolder.

2.6.4.2 Viewer

Viewers are a important tools to enable a user to judge the results of one or more processing steps. Therefore, the COW provides multiple specialized viewers to cover all possible visualization scenarios. Included are the following:

1. Graph viewer that can plot multiple 1D graphs
2. 2D viewer, used as the standard viewer for IOs, where images and their metadata can be displayed. Also included are tools for histogram display manipulation and statistical values.
3. 3D Isosurface viewer that displays one or more reconstructed or imported structures and also offers many tools for display customization.
4. Value viewer that was developed by the author of this thesis that can plot the value distribution of two metadata metrics separately and in dependence of one another in a pseudo heatmap
5. Surface viewer that in addition to the x- and y-information in 2D images plots the pixel values as a third dimension
6. Euler and Halo Viewers that show the spherical distribution of Euler angles of 2D images.

The viewers are shown in figure 2.16.

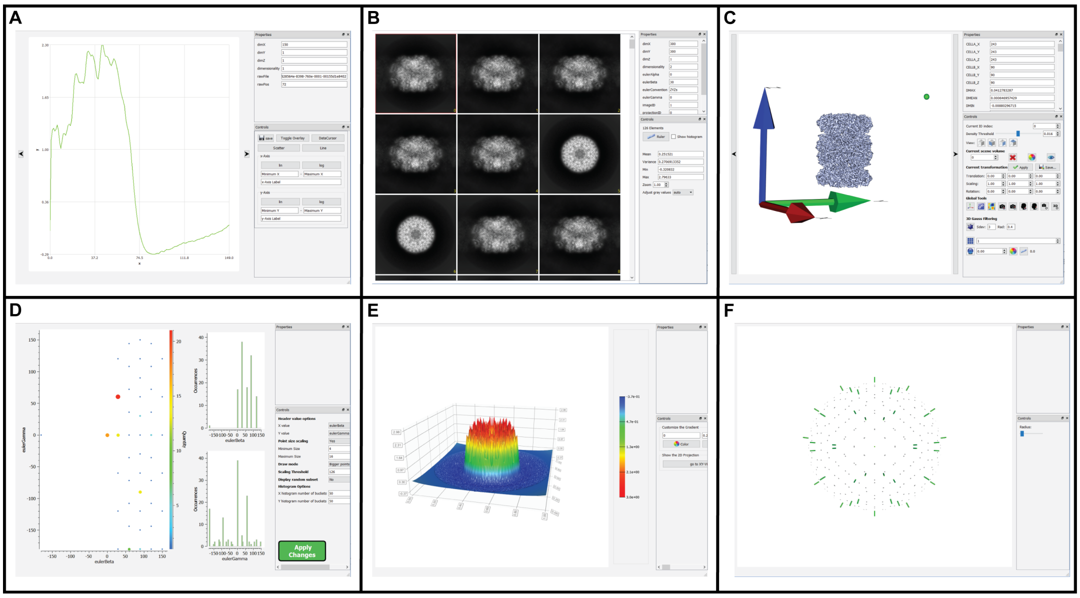


FIGURE 2.16: CowEyes viewers. A: Graph viewer, B: 2D viewer, 3D Isosurface viewer, D: Value viewer, E: Surface viewer, F: Euler viewer. Not shown is the Halo viewer that was used to create figure 1.14.

2.6.4.3 Visual Programming

For advanced users that need additional flow functionality apart from cow::Logic trees, a set of visual programming tools are provided. These include the classical elements from programming languages such as for-loops, if-else conditions, variables and more. Also, convenience tools such as groups, which can hold multiple cow::Logic objects and tools, lists and comments are provided. Through a scripting module that is based on the Lua language, users can further automate their desired workflows.

2.6.4.4 CowPlug Ecosystem

To ensure extendability of the COW, a system was developed by Dr. Mario Lüttich through which external developers can implement their own algorithms for usage in CowEyes. Through a CMake based logic template, a custom cow::Logic with access to the CowLib API can be compiled as a dynamic library that CowEyes can load. This system is called CowPlug due to the plugin nature of the external cow::Logic objects. Due to current stability issues with the dynamic library approach, in the future, this will possibly be changed to compiling separate executables that CowEyes can then launch.

2.6.5 CowServer

Every instance of CowEyes can connect to a CowServer instance that is installed on a host system, where a separate project file needs to either be created or preexist. Due to the prevalence of the OS on server systems, only a Linux version of the CowServer is supported currently.

When a run is started, the CowServer software scans the dependencies of the connected logic modules to determine which can be calculated in parallel. Through the MPI protocol, a logic can provide splitting its calculation into smaller packages that the CowServer can calculate on different server nodes in parallel. This is implemented for the computationally most expensive algorithmic steps such as image alignment. If no packaging is provided, a cow::Logic can be calculated on the CPUs and GPUs of the current node.

2.7 Preparation of Test Data

Evaluation of novel algorithms depends on the careful choice of test data. Ideally, testing is first done on simulated data where the expected outcome is known. However, this was not possible for the two softwares of this thesis, therefore, real data was used.

2.7.1 Test Data for CowGraCE

For the rate-zonal separation, the initial test data consisted of SDS gels of fractions of purified 20S Proteasome due to the known sedimentation coefficient of this complex. Furthermore, as density gradient centrifugation is used in its purification protocol[163], correct prediction of the Proteasome's sedimentation behavior was considered to be of high priority.

Afterwards, realistic samples from purification protocols were used to evaluate prediction of more complex systems where at times only an approximated molecular weight is known about the protein.

TABLE 2.4: Available ultracentrifuge swing-out rotors in the Department of Structural Dynamics

Manufacturer	Model Name	r_{min}[cm]	r_{max}[cm]	Acceleration Time[min]	Deceleration Time[min]	Max rpm	Tube Volume [ml]
Beckman	SW28 TI	7.53	16.1	5	5	28000	38.5
Beckman	SW40 TI	6.67	15.88	7	6	40000	14
Beckman	SW60 TI	6.31	12.03	3.5	2.5	60000	4

2.7.1.1 Considered Rotors and Gradients

Due to the need to compare theoretical predictions with empirical data, all simulations exclusively used the three rotors that were available in the Department of Structural Dynamics and therefore could be used for ultracentrifugation. Table 2.4 shows the relevant information about these rotors. The BioComp GradientMaker that is used in the Department of Structural Dynamics can only produce predefined gradients for the tubes of the mentioned rotors. These are listed in table A.1 and A.2.

2.7.2 Test Data for the MQC

As the goal of the MQC was to classify real data based on user input, usage of synthetic test data would defeat its purpose. Therefore, different datasets of different macromolecular complexes that are currently investigated in the department were used to test the classification capatibilities of the machine learning approach.

Chapter 3

Results

"The most that can be expected from any model is that it can supply a useful approximation to reality: All models are wrong; some models are useful"

George Box, *Statistics for Experimenters*

This thesis introduces two new algorithmic approaches and the softwares that implement them, turning theory into usable tools. Following the order in which they can be applied to a cryo-EM workflow, results of the density gradient centrifugation prediction software CowGraCE are described first, followed by the live processing tool MQC. The two software's logos are shown in figure 3.1.

FIGURE 3.1: Logos of the softwares developed in this thesis. Shown on the left is the CowGraCE logo, kindly created by Karl Bertram, and on the right is the MQC logo, kindly created by Dr. Wen-Ti Liu.

3.1 CowGraCE

As the linear approach of the software and its application area did not fit into the preexisting COW framework, it was designed as a light-weight standalone software written in C++11, using Qt5.9 for the GUI. CowGraCE was developed together with Uma Lakshmi Dakshinamoorthy.

The goal of the CowGraCE project was to develop a software that predicts the end results of rate-zonal separation experiments using swing-out rotors in a qualitatively correct manner in addition to assisting users in finding the optimal run conditions for their sample. In this section, the mean peak S value prediction algorithm and the distribution estimation are introduced, followed by comparisons to the COMPASS software and tests on experimental data. Afterwards, the developed optimization algorithm and its results are presented, closing with an overview of the software's inner workings and GUI.

3.1.1 Mean Svedberg Value Prediction Algorithm

This algorithm was loosely adopted from an algorithm in FORTRAN code written by Steensgaard et al in 1978[176], which was developed for fixed-angle rotors. The concentration dependent steps of the algorithm were changed to reflect the different properties of swing out rotors. An overview of different steps of this algorithm is given in figure 3.2.

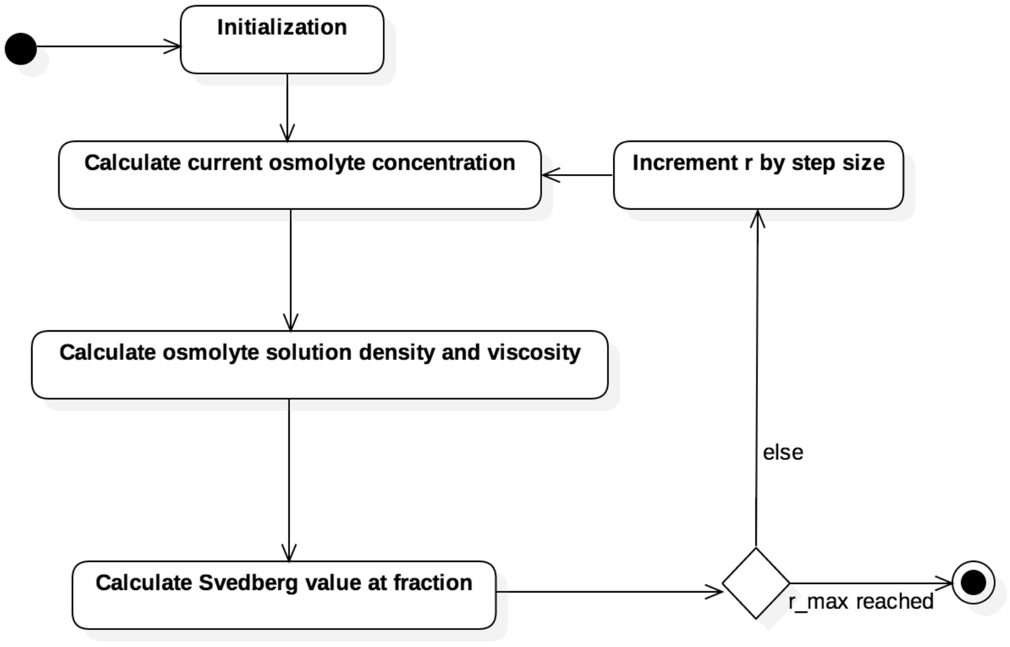

Initialization
Calculate current osmolyte concentration
Increment r by step size
Calculate osmolyte solution density and viscosity
else
Calculate Svedberg value at fraction
r_max reached

FIGURE 3.2: Activity diagram for the peak Svedberg value prediction algorithm. Shown are the different calculation steps that are performed for every fraction radius.

3.1.1.1 User Inputs

To calculate the Svedberg values along the tube after a centrifugation run, several parameters need to be specified by the user:

1. Temperature T
2. Particle density ρ_p
3. Rotor and rotor speed v_{rotor}
4. Gradient volume V_{grad} and sample volume V_{samp}
5. Fraction number n_{frac}, either provided directly or calculated from fraction volume V_{frac}
6. Osmolyte type and concentration at top and bottom of the tube, c_{top} and c_{bottom}
7. Runtime t_{run}

3.1.1.2 Initialization

In the initialization step, multiple constant values necessary for the following steps are calculated. First, the rotor speed ω is converted to angular velocity by the following relationship:

$$\omega = \frac{2\pi \cdot v_{rotor}}{60} \,. \tag{3.1}$$

Then, the constant R that represents the integral over the runtime from equation 2.58 can be calculated :

$$R = (\pi \cdot v_{rotor})^2 \cdot \frac{t_{run}}{15} \,. \tag{3.2}$$

While the effective runtime could theoretically be adjusted to reflect acceleration and deceleration time through interpolation, this was neglected due to modern centrifuges offering different acceleration options, thus adding another layer of uncertainty, and the negligible overall effect due to the long runtimes.

The constant D that represents the constant fraction from equation 2.58 is then calculated:

$$D = \frac{\rho_p - \rho_w}{\eta_w} \,. \tag{3.3}$$

The effective start of the gradient is calculated through determination of the length offset l_{samp} introduced through the sample volume itself:

$$l_{samp} = \frac{V_{samp}}{V_{tube}} \cdot (r_{max} - r_{min}) \,. \tag{3.4}$$

The mass center of the sample is then used to initialize r_{prev}, the starting radius value:

$$r_{prev} = r_{min} + \frac{l_{samp}}{2} \,. \tag{3.5}$$

Furthermore, temperature dependent constants for osmolyte density and viscosity are calculated. Also, the cumulative Svedberg value $S_{20,w,sum}$ is initialized to 0.

3.1.1.3 Calculation Loop

Starting from r_{prev} , the calculation loop is iterated as many times as fractions are selected, with each fraction covering an equal radius length and the radius r_{curr} being incremented.

First, the current osmolyte concentration c_{curr} is calculated through the linear relationship to the radial distance:

$$c_{curr} = c_{top} + \frac{(r_{curr} - (r_{min} + l_{samp})) \cdot (c_{bot} - c_{top})}{r_{max} - (r_{min} + l_{samp})} \,. \tag{3.6}$$

Medium density and viscosity are then calculated as described in equations 2.52 and 2.54 for Glycerol and equations 2.48 and 2.50 for Sucrose.

The value Z for the radius dependent integral from equation 2.58 can then be determined as

$$Z = \frac{r_{curr} - r_{prev}}{\frac{1}{2}(r_{curr} + r_{prev})} \cdot \frac{\eta_{medium}}{\rho_p - \rho_{medium}} \,. \tag{3.7}$$

Combining all of the previous formulas, the peak sedimentation coefficient $S_{20,w,sum}$ at the radius of the fraction can be calculated:

$$S_{20,w,sum} = S_{20,w,sum} + Z \cdot \frac{R}{D} \cdot 10^{13} \,. \tag{3.8}$$

3.1.2 Distribution Profile Prediction

Known methods about predicting a distribution profile of a given species are based on extensive knowledge of the diffusion behavior of a protein complex and its charge and shape dependent physicochemical properties such as partial-specific volume and frictional ratio[164]. Additionally, in a complex sample consisting not only of the protein complex of interest, but also of known and unknown contaminants where interactions between the components can influence the diffusion profile, a numeric evaluation of the described effects is difficult and error-prone. Therefore, an approach independent of specific sample information was chosen. The profile prediction of the COMPASS algorithm is unknown.

Here, based on the assumption of a normal distributed distribution profile, the concept of apparent sedimentation coefficients from the van-Holde Weischet method[191] for analyzing diffusional spreading in analytic ultracentrifugation was used. An apparent sedimentation coefficient s^* is defined as

$$s^* = ln\left(\frac{r_b}{r_{men}}\right)\frac{1}{\omega^2(t_{run} - t_0)} \,. \tag{3.9}$$

Here, r_b is the radial position of the boundary fraction, which is the relative concentration of the sample compared to its plateau along the diffusion profile, r_{men} signifies the radial starting point of the sample and t_0 is the corrected start time of the experiment based on rotor acceleration.

Based on this, a approach for the standard deviation of the normal distribution of a sedimenting species $S_{samp,20,w}$ with its peak at radius r_μ was developed, again neglecting the acceleration and deceleration times. The boundary Svedberg value $S_{boundary,20,w}$ at a specific fraction was defined as:

$$S_{boundary,20,w} = S_{peak,20,w} + \frac{ln\left(\frac{r_\mu}{r_{ini}}\right)}{\omega^2 \cdot t} \cdot 10^{13} \,. \tag{3.10}$$

This value signifies which Svedberg value has its boundary at the specific fraction. The standard deviation was then defined as

$$\sigma = \frac{frac_{sample} - frac_{boundary}}{3} \,. \tag{3.11}$$

From the boundary Svedberg values at every fraction, σ can then be interpolated for every Svedberg value of interest.

3.1.3 Interpolation

From the peak Svedberg and boundary Svedberg values for every fraction, the predicted distribution for a user-specified S value of interest can be determined through linear interpolation. It should be noted that while experimental fractions per definition only exist as real numbers (and also are mixed after removal from the tube), here and in the following, fraction is used interchangeably with length/volume measurements along the tube that are assigned to the fractions they will be split up into after the run.

The interpolation for the peak location is based on determining the two adjacent fractions $frac_{low}$ and $frac_{high}$ where $S_{low,20,w} < S_{sample,20,w} < S_{high,20,w}$. From these values, $frac_{sample}$ can readily be calculated:

$$frac_{sample} = frac_{low} + \frac{S_{sample,20,w} - S_{low,20,w}}{S_{high,20,w} - S_{low,20,w}} \, . \tag{3.12}$$

Obviously, this interpolation isn't necessary if $S_{sample,20,w} = S_{low,20,w}$ or $S_{sample,20,w} = S_{high,20,w}$.

Accordingly, the boundary location is calculated by determining the two fractions $frac_{b,low}$ and $frac_{b,high}$ where $S_{b,low,20,w} < S_{sample,20,w} < S_{b,high,20,w}$ and interpolating $frac_{boundary}$:

$$frac_{boundary} = frac_{b,low} + \frac{(S_{sample,20,w} - S_{b,low,20,w}) \cdot (frac_{b,high} - frac_{b,low})}{S_{b,high,20,w} - S_{b,low,20,w}} \, . \tag{3.13}$$

3.1.4 Computational Complexity of the Simulation Algorithm

Due to the numerically primitive approach and the fast implementation in CowGraCE, a given simulation is calculated in the μs range. Computation speed measurements were plotted against number of calculated fractions in figure 3.3. Clearly visible is the linear dependency of the two variables, as shown through the linear regression trendline. Thus, the simulation algorithm has a running time of $O(n)$, where n is the number of fractions. This fast computation speed made it possible to develop the optimization algorithm detailed in section 3.1.7 that performs a large number of simulations.

3.1.5 Simulation Result Comparisons

While the underlying algorithmic approach of COMPASS[177] for rate-zonal centrifugation simulation is not known due to the source code not being available, it was assumed that the authors utilized their own published methods for peak $S_{20,w}$ value predictions[176], which also form the basis for the algorithm presented in the previous section.

Therefore, a comparison between the calculated values was deemed necessary to assess the validity of the CowGraCE algorithm. While only the old version of the COMPASS software was available to the author, a comparison between two simulations of the old and the new COMPASS software as shown in figure 3.4 led to the conclusion that the underlying algorithm was not changed. Based on this, a comparison with the 1987 version was assumed to be sufficient.

Results from the COMPASS simulations were extracted manually due to the lack of export options provided. Comparisons were performed for three rotors available in

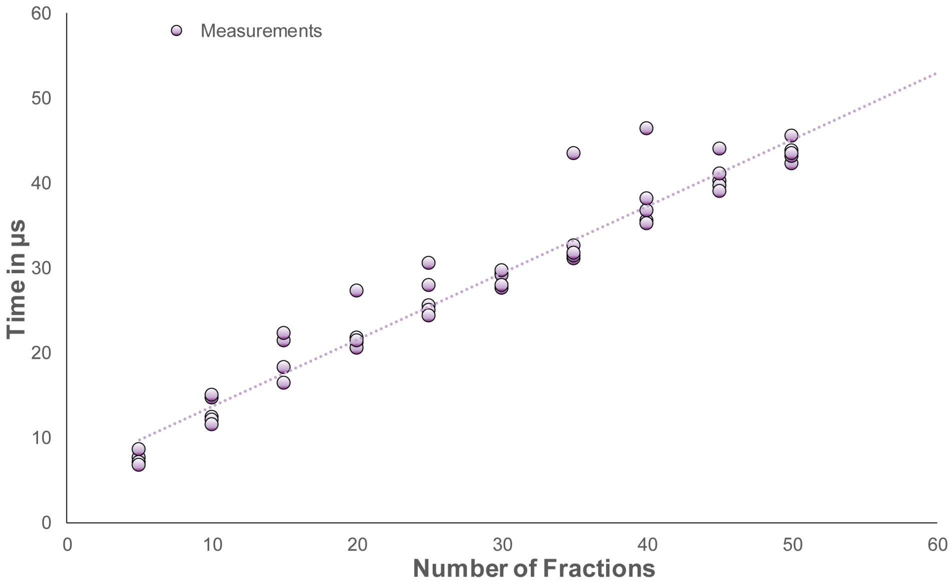
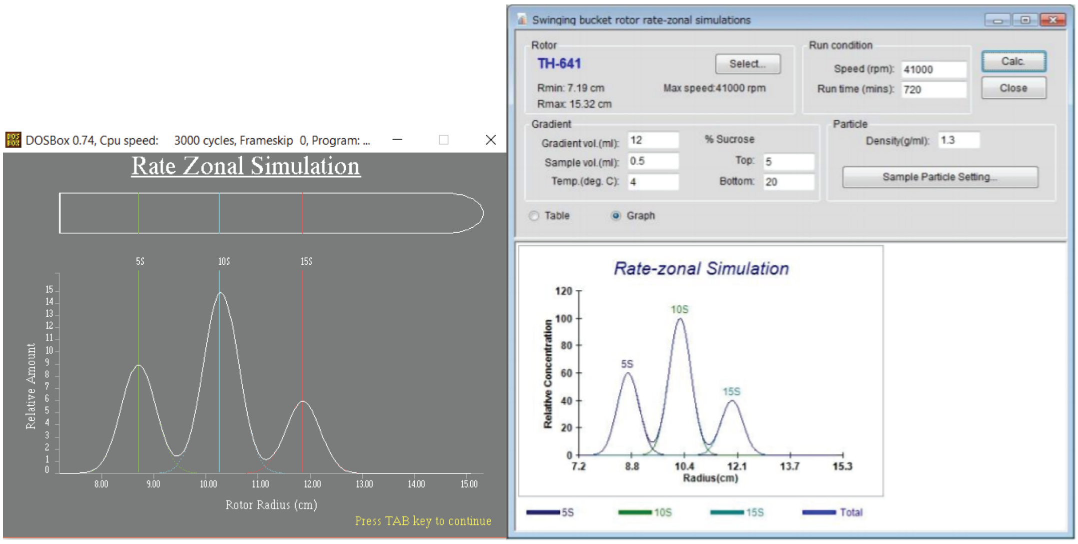


FIGURE 3.3: Dependence of CowGraCE calculation speed on number of fractions. Shown are 5 measurements for every fraction amount and the linear regression of the data points. The speed was measured on the Dell Workstation described in 2.5.


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: ...
Rate Zonal Simulation
5S
10S
15S
Relative Amount
Rotor Radius (cm)
Press TAB key to continue
Swinging bucket rotor rate-zonal simulations
Select...
Calc.
Close
Sample Particle Setting...
Rate-zonal Simulation
Relative Concentration
Radius(cm)
5S
10S
15S
Total

Rotor
TH-641
Rmin: 7.19 cm
Rmax: 15.32 cm
Max speed:41000 rpm

Run condition
Speed (rpm): 41000
Run time (mins): 720

Gradient
Gradient vol.(ml): 12
Sample vol.(ml): 0.5
Temp.(deg. C): 4
% Sucrose
Top: 5
Bottom: 20

Particle
Density(g/ml): 1.3

○ Table ● Graph

Old Compass **New Compass**

FIGURE 3.4: Competing softwares. Shown are screenshots of the result of simulations with the same run parameters for both the old version of COMPASS that was available to the author and the updated version. The right image was taken from [184]. Both simulation results were identical.

the Department of Structural Dynamics, assuming a standard temperature T of 4°C, $t_{run} = 960$ minutes (16 hours), maximum speed v_{rotor}, $\rho_p = 1.4$, $V_{grad} = 0.95 \cdot V_{tube}$, $V_{samp} = 0.05 \cdot V_{tube}$ and $n_{frac} = 25$ due to the lack of other options in COMPASS.

Figure 3.5 shows the comparison between simulations with the same parameters on the SW28, SW40 and SW60 rotor with varying sucrose gradients. Surprisingly, considerable differences could be observed. No systematic offset could be seen, but rather a growing discrepancy at higher fraction numbers.

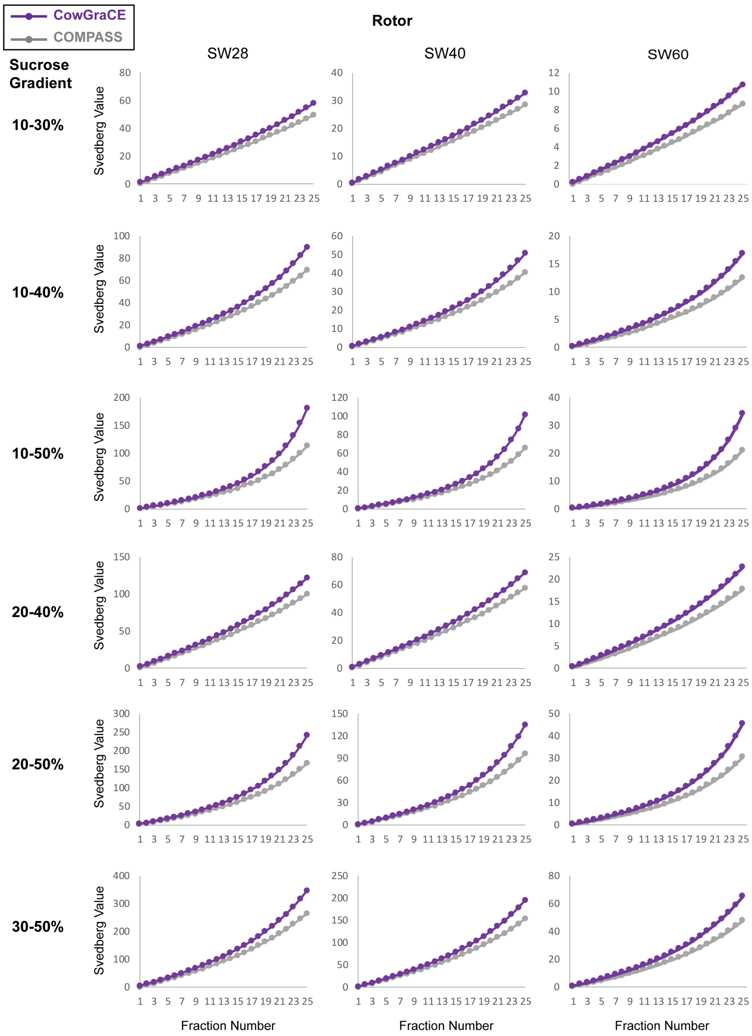


FIGURE 3.5: Comparison of peak Svedberg values for three rotors at varying gradients. Shown values were manually extracted by Uma Lakshmi Dakshinamoorthy.

Comparison of peak $S_{20,w}$ values at the last fraction from the simulations in figure 3.5 , where the biggest difference could be observed, are shown in figure 3.6. Due to the higher absolute sedimentation value resolution, SW28 results show the biggest discrepancy. Similar result discrepancies were observed for other tested parameters (data not shown).

3.1.6 Prediction Tests

To test the peak Svedberg value prediction accuracy and the validity of the new approach for diffusion profile modeling in CowGraCe and to compare it to COMPASS and experimental results, rate-zonal centrifugation experiments were performed under different conditions and the fractions extracted by the top-to-bottom method were applied to individual lanes on SDS gels. The gels were then imaged and converted to a contrast enhanced grey scale. The intensity distributions were extracted

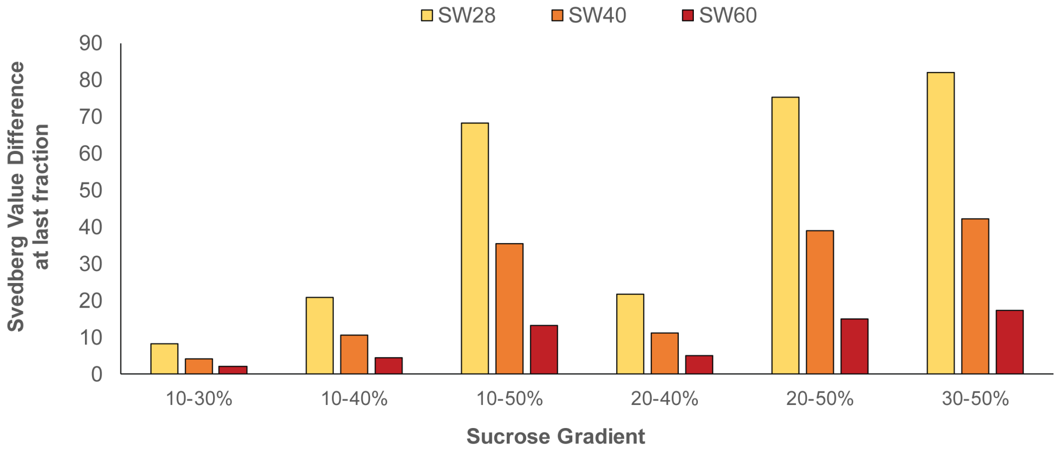


FIGURE 3.6: Difference between CowGraCE and COMPASS for peak Svedberg values at varying gradients and rotors at the last fraction.

in ImageJ, converted to relative intensities based on the highest measured value, fitted with a Gaussian distribution as described in section 2.3 and then plotted with a custom script written in R. Then, simulations under the same conditions were run in CowGraCE and COMPASS. COMPASS peak values were interpolated in the same way as CowGraCe values, while the standard deviation was extracted manually from the program's visual output as shown in figure A.1 in the Appendix. As COMPASS can only calculate 25 fractions, the values were renormalized to the differing number of experimental fractions where necessary.

The mean and distribution predicted of COMPASS and CowGraCE were independent of absolute concentration values. Therefore, the simulated distributions and the fits of relative intensity distribution extracted from SDS gels are displayed in the following as Gaussians of the form

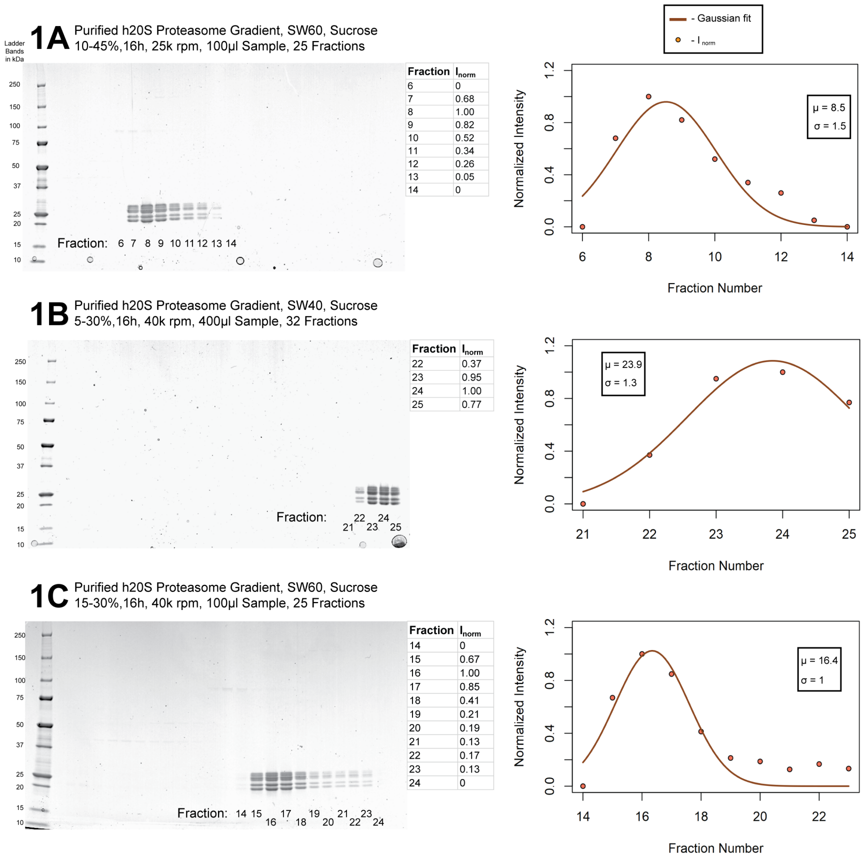
$$f(x) = A \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{3.14}$$

with $A = 1$. Thus, the distributions were not normalized to an area, but rather to the peak value at A. This procedure was chosen because of the assumed unknown absolute protein concentrations within a sample, their exact influence and the other numerous described uncertainties. This approach did not allow for area overlap analysis.

The error of determined COMPASS distribution parameters and of the fits of the relative experimental intensities were hard to quantify, but were estimated to be within 0.1 fractions, therefore all shown values were rounded to one decimal place, which is below the experimentally significant level.

3.1.6.1 Tests on Purified Human Proteasomes

First, tests on a minimal system with no contaminations were performed. Here, purified human 20S Proteasomes were chosen as the test sample due to their well-characterized sedimentation behavior. For all experiments, the standard temperature of 4°C was used. The results of the three centrifugation runs and the intensity distributions under different conditions are shown in figure 3.7 and denoted 1A, 1B and 1C. The run parameters were chosen due to the different position of the simulated peak locations along the tube.


1A Purified h20S Proteasome Gradient, SW60, Sucrose 10-45%,16h, 25k rpm, 100µl Sample, 25 Fractions

Fraction	I_{norm}
6	0
7	0.68
8	1.00
9	0.82
10	0.52
11	0.34
12	0.26
13	0.05
14	0

1B Purified h20S Proteasome Gradient, SW40, Sucrose 5-30%,16h, 40k rpm, 400µl Sample, 32 Fractions

Fraction	I_{norm}
22	0.37
23	0.95
24	1.00
25	0.77

1C Purified h20S Proteasome Gradient, SW60, Sucrose 15-30%,16h, 40k rpm, 100µl Sample, 25 Fractions

Fraction	I_{norm}
14	0
15	0.67
16	1.00
17	0.85
18	0.41
19	0.21
20	0.19
21	0.13
22	0.17
23	0.13
24	0

FIGURE 3.7: Human 20S Proteasome density gradient centrifugation results for three different conditions. For 1A, 1B and 1C, SDS gels where every extracted fractions was loaded as a separate lane in 1A and 1C are shown on the left. In 1B, only fractions 1-25 were loaded. The fractions showing the characteristic Proteasome bands are noted. In the middle, normalized intensities at the fractions and on the right, the Gaussian fit of the distribution and its parameters are shown. The experiments were performed by Uma Lakshmi Dakshinamoorthy.

TABLE 3.1: Comparison of distribution parameters for the tests on purified h20S Proteasomes

Experiment	μ_{Exp}	$\mu_{CowGraCE}$	$\mu_{COMPASS}$	σ_{Exp}	$\sigma_{CowGraCE}$	$\sigma_{COMPASS}$
1A	8.5	8.6	11.7	1.5	0.9	0.3
1B	23.9	24.2	27.2	1.3	2.2	1.6
1C	16.4	18.3	24.6	1	1.4	0.92

It could be seen that due to the top-to-bottom fractionation, the Gaussian fit agreed less with the data in the higher fractions of the protein distribution. Nevertheless, the quality of the fit indicated the assumed normal distribution of the intensity to be valid.

For 1A, 1B and 1C, simulations were performed in CowGraCE and COMPASS with the same parameters. A particle density $\rho_p = 1.4$ was used in both softwares.

A comparison between experimental and the two simulated distributions are shown in figure 3.10. The underlying μ and σ values are detailed in table 3.1. In all three experiments, CowGraCE approximated the experimental data reasonably well in both mean and standard deviation, while COMPASS showed larger deviations from the experiment.

3.1.6.2 Tests on Realistic Purification Scenarios

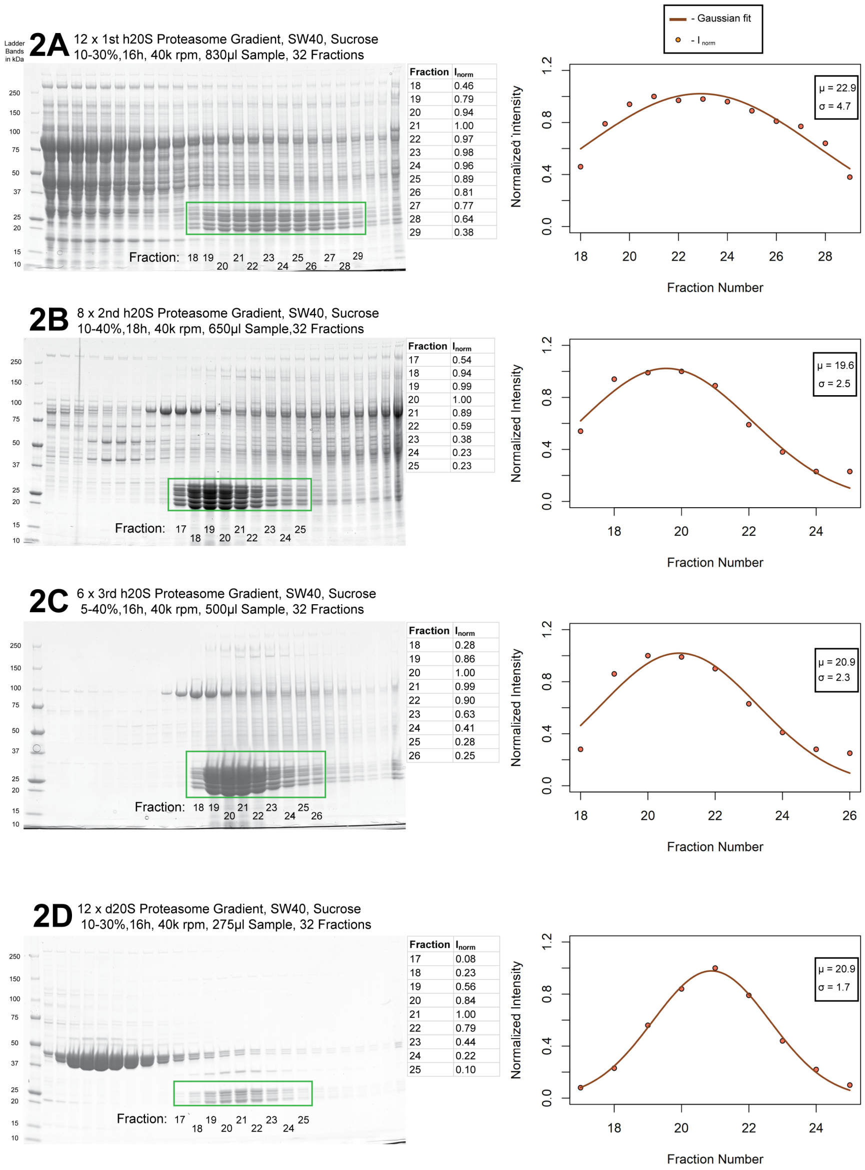
To evaluate the algorithmic performance under realistic experimental scenarios that include large scale contaminations, sucrose gradients at different steps of the purification protocol of the human 20S Proteasome[163] were chosen as test subjects, as well as a gradient from purification of the Drosophila 20S Proteasome. Results from these experiments are shown in figure 3.8, denoted as 2A, 2B, 2C and 2D. Here, multiple tubes were pooled together on the gel as indicated in the figure, resulting in an overall higher protein concentration. For all experiments, the SW40 rotor was used due to practical considerations. Again, the temperature was 4°C.

Clearly visible is the amount of contaminating protein in the applied samples that is reduced during the purification process, as can be seen by comparing 2C to 2A. As expected, a larger smearing out of the Proteasome distribution could be observed that could potentially be related to higher overall protein concentration. Nevertheless, Gaussian distributions could be fitted to the determined intensities. Here, the effect of the top-to-bottom fractionation on the intensity values of the higher fractions of the distribution is less pronounced, presumably due to the larger overall protein concentration. The largest determined standard deviation could be observed in the first gradient of the Proteasome purification.

Again, simulations in both softwares with a particle density $\rho_p = 1.4$ were run. The resulting distributions compared to the experimental one are plotted in figure 3.10 and the underlying distribution parameters are listed in table 3.2. Here, the predicted peak values by CowGraCE were very close to the experimental data and outperformed COMPASS in three out of four runs. The larger standard deviations were underestimated by both softwares, with CowGraCE being slightly closer.

TABLE 3.2: Comparison of distribution parameters for the tests on purification gradients

Experiment	μ_{Exp}	$\mu_{CowGraCE}$	$\mu_{COMPASS}$	σ_{Exp}	$\sigma_{CowGraCE}$	$\sigma_{COMPASS}$
2A	22.9	21.5	23.7	4.7	2	1.1
2B	19.6	19.8	22.2	2.5	1.5	0.8
2C	20.9	20.7	23.3	2.3	1.6	0.8
2D	20.9	20.8	23.3	1.7	1.9	0.8
3A	15	15.5	17.2	3.5	1.7	1
3B	15	16.8	18.8	4.1	1.7	1.2
3C	21.9	19.8	22.4	3.1	2.0	1.3



Fraction	I_{norm}
18	0.46
19	0.79
20	0.94
21	1.00
22	0.97
23	0.98
24	0.96
25	0.89
26	0.81
27	0.77
28	0.64
29	0.38

Fraction	I_{norm}
17	0.54
18	0.94
19	0.99
20	1.00
21	0.89
22	0.59
23	0.38
24	0.23
25	0.23

Fraction	I_{norm}
18	0.28
19	0.86
20	1.00
21	0.99
22	0.90
23	0.63
24	0.41
25	0.28
26	0.25

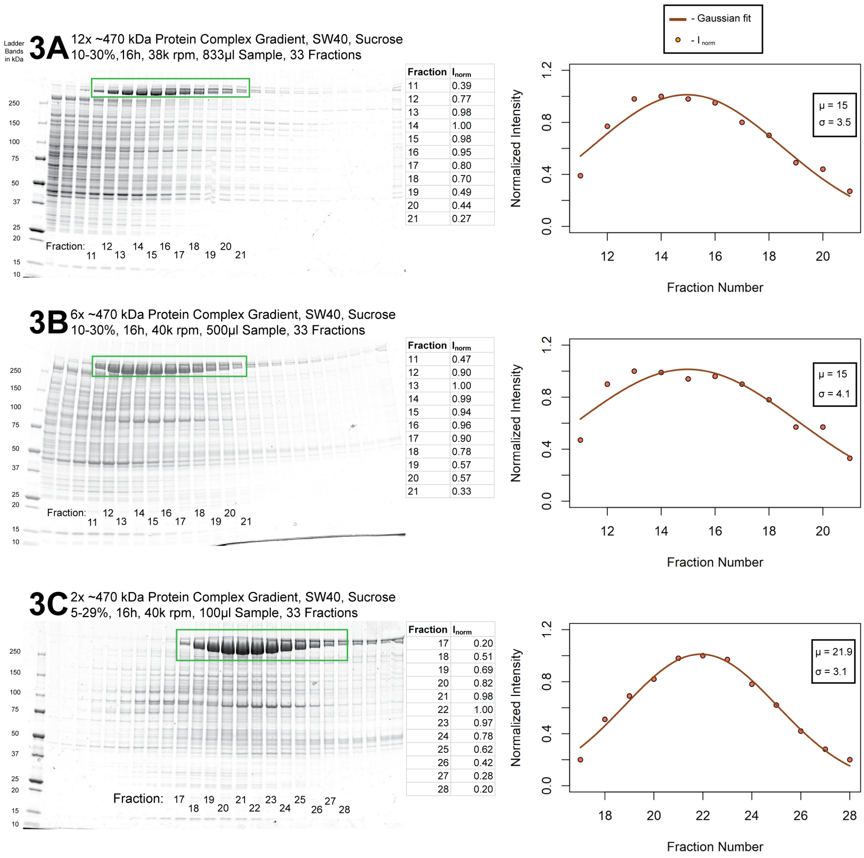
Fraction	I_{norm}
17	0.08
18	0.23
19	0.56
20	0.84
21	1.00
22	0.79
23	0.44
24	0.22
25	0.10

FIGURE 3.8: Human and Drosophila 20S Proteasome density gradient centrifugation results at different purification steps. For 2A, 2B, 2C and 2D, SDS gels where extracted fractions 8-32 of multiple centrifugation runs were pooled and loaded as a separate lane are shown on the left. The fractions showing the characteristic Proteasome bands are noted and marked with a green box for clarity. In the middle, normalized intensities at the fractions and on the right, the Gaussian fit of the distribution and its parameters are shown.The data was kindly provided by Fabian Henneberg.

Next, experimental data of sucrose gradients during purification of an uncharacterized protein complex where only an approximate molecular weight of approximately 470 kDa was known were evaluated. The predicted Svedberg value for this complex was 15. All three experiments again used the SW40 rotor and were performed at 4°C.

The experimental results of this, denoted 3A, 3B and 3C, are shown in figure 3.9.

Contaminations were clearly visible on the gels and the determined normal distributions showed a fairly large standard deviation.



Fraction	I_{norm}
11	0.39
12	0.77
13	0.98
14	1.00
15	0.98
16	0.95
17	0.80
18	0.70
19	0.49
20	0.44
21	0.27

Fraction	I_{norm}
11	0.47
12	0.90
13	1.00
14	0.99
15	0.94
16	0.96
17	0.90
18	0.78
19	0.57
20	0.57
21	0.33

Fraction	I_{norm}
17	0.20
18	0.51
19	0.69
20	0.82
21	0.98
22	1.00
23	0.97
24	0.78
25	0.62
26	0.42
27	0.28
28	0.20

FIGURE 3.9: Unknown protein complex density gradient centrifugation results. For 3A, 3B, and 3C, SDS gels where extracted fractions 8-32 of multiple centrifugation runs were pooled and loaded as a separate lane are shown on the left. The fractions showing the characteristic protein complex bands are noted and marked with a green box for clarity. In the middle, normalized intensities at the fractions and on the right, the Gaussian fit of the distribution and its parameters are shown. The experiments were performed by Uma Lakshmi Dakshinamoorthy.

Once again, a particle density of $\rho_p = 1.4$ was used for the simulations whose results are listed in table 3.2 while the distributions are plotted in figure 3.10. Here, the peak fractions were predicted reasonably well by CowGraCE, with COMPASS being closer to the experimental data in 3C. Both softwares underestimated the large standard deviations.

3.1.6.3 Statistical Evaluation of Prediction Results

While a sample size of $n = 10$ with two protein complexes as test subjects can hardly be used to confidently perform statistical analysis, general trends could be analyzed with the available data.

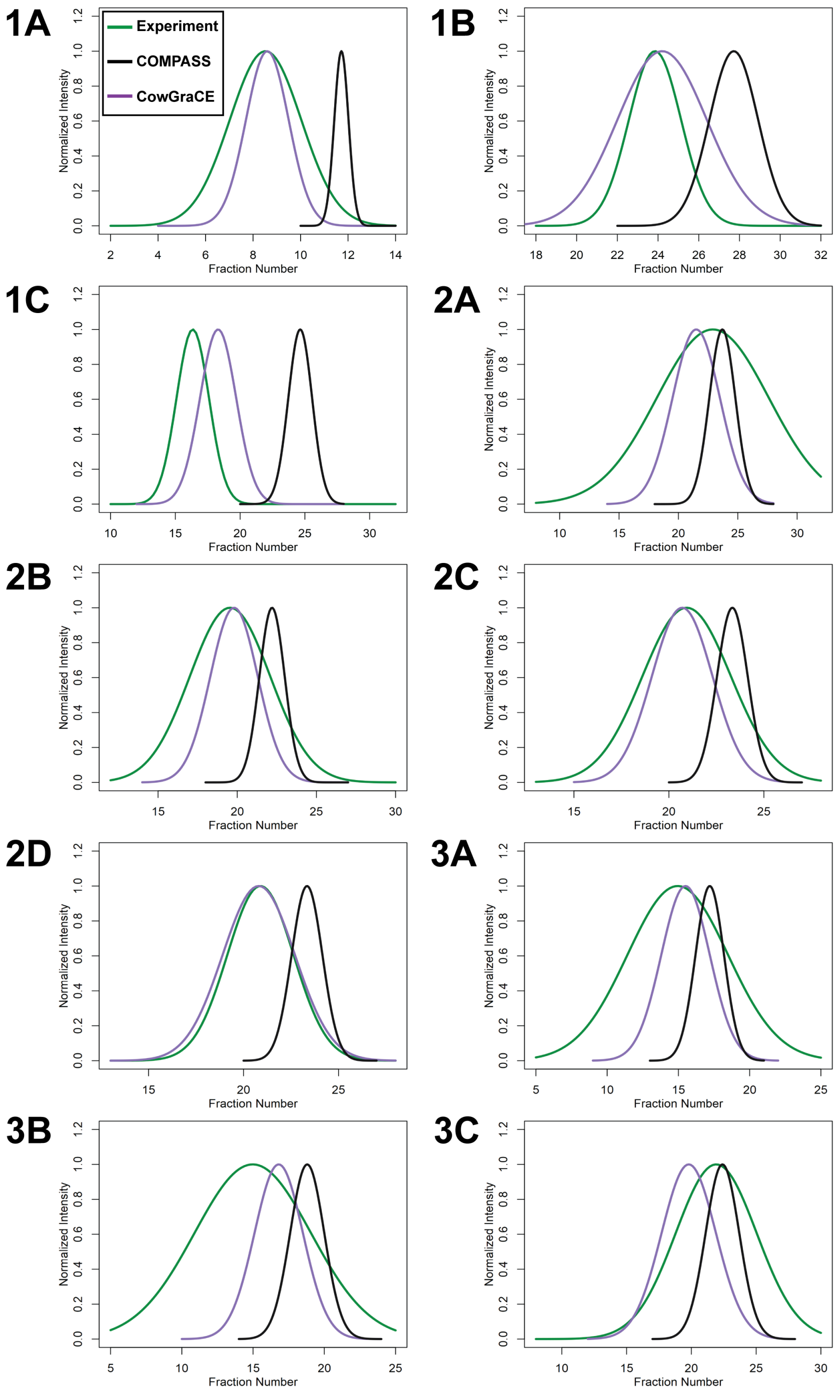


FIGURE 3.10: Comparison of simulated Gaussian distributions to the experimental data. The shown distributions were normalized to a peak value of 1 instead of to a constant area and show the fraction range in which experimental data points were localized compared to the predictions by the two simulation algorithms.

To compare the prediction performance of CowGraCE to COMPASS, several metrics were used. First, the fraction difference for peak fractional position and standard deviation, $\triangle\mu$ and $\triangle\sigma$, were calculated by subtracting the simulated values from the fitted experimental data. From this, the descriptive statistical values median $\tilde{x}$ and average $\overline{x}$ were determined. Due to negative numbers being present in $\triangle\mu$ and $\triangle\sigma$ values, the average of the absolute values, denoted $|\overline{\triangle}|$, was also calculated. These metrics for the test data are shown in table 3.3. For peak fraction prediction, the CowGraCE algorithm predicted the experimental data with an accuracy of within one fraction on average, with deviations spread into both directions equally, while the COMPASS algorithm deviated by a larger value on average into the positive direction. It should also be noted that the peak prediction quality didn't deviate much between the isolated Proteasome experiments and Proteasome purification samples containing more contaminations and larger overall protein concentrations. For the unknown complex in 3A-C, larger differences could be observed for two of the three experiments. This can be explained by the unconfirmed Svedberg value of the sample and the unknown interactions with sample contaminations.

TABLE 3.3: Calculated prediction metric results

Experiment	$\triangle\mu_{CowGraCE}$	$\triangle\mu_{COMPASS}$	$\triangle\sigma_{CowGraCE}$	$\triangle\sigma_{COMPASS}$
1A	0.1	3.2	-0.6	-1.2
1B	0.3	3.3	0.9	-0.1
1C	1.9	8.2	0.4	-0.1
2A	-1.4	0.8	-2.7	-3.6
2B	0.2	2.6	-1	-1.7
2C	-0.2	2.4	-0.7	-1.5
2D	-0.1	2.4	0.2	-0.9
3A	0.5	2.2	-1.8	-2.5
3B	1.8	3.8	-2.4	-2.9
3C	-2.1	0.5	-1	-1.8
$\overline{x}$	0.1, $\lvert\overline{\triangle}\rvert$= 0.86	2.94	-0.87, $\lvert\overline{\triangle}\rvert$= 1.17	-1,63,$\lvert\overline{\triangle}\rvert$=1.63
$\widetilde{x}$	0.15	2.51	-0.85	-1.6

Prediction of a the distribution's standard deviations with the new CowGraCE approach also outperformed COMPASS, but showed more absolute differences to the experimental data's standard deviations than for the peak location prediction. Also, the predicted standard deviations by CowGraCE for isolated Proteasome experiments 1A-1C were more exact than for the purification samples 2A-3C whose determined standard deviations were bigger overall. Large differences could again be observed for the unknown complex. The data trends are visualized as boxplots in figure 3.11.

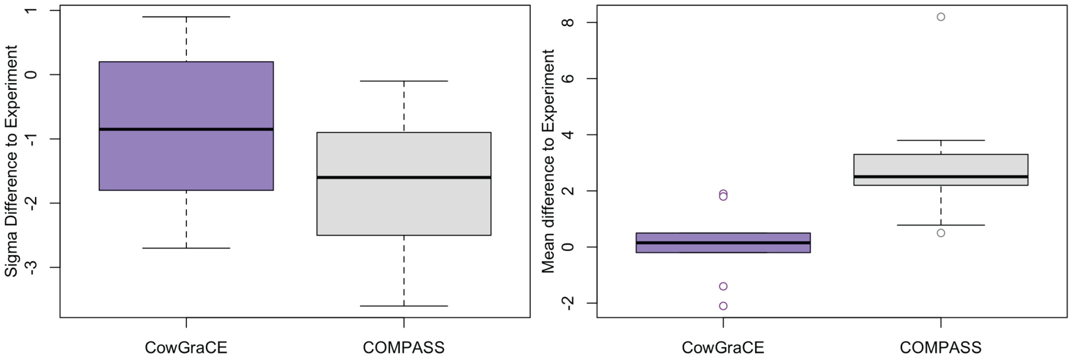


FIGURE 3.11: Boxplots of CowGraCE and COMPASS prediction performance metrics

3.1.7 Run Condition Optimization

While simulating a result from a given set of parameters is useful, what has been missing from density gradient centrifugation softwares is the ability to help the user find the most suitable set of parameters for their experiment. Therefore, a run condition optimization algorithm was developed that is outlined in figure 3.12 and presented in the following. In a nutshell, a brute-force approach is used to find the best possible run conditions under the given constraints through repeated simulation of centrifugation runs with varying parameters and successive result evaluation. To minimize computational cost, the search space was reduced through user input and pragmatic considerations. The algorithm is introduced in the following.

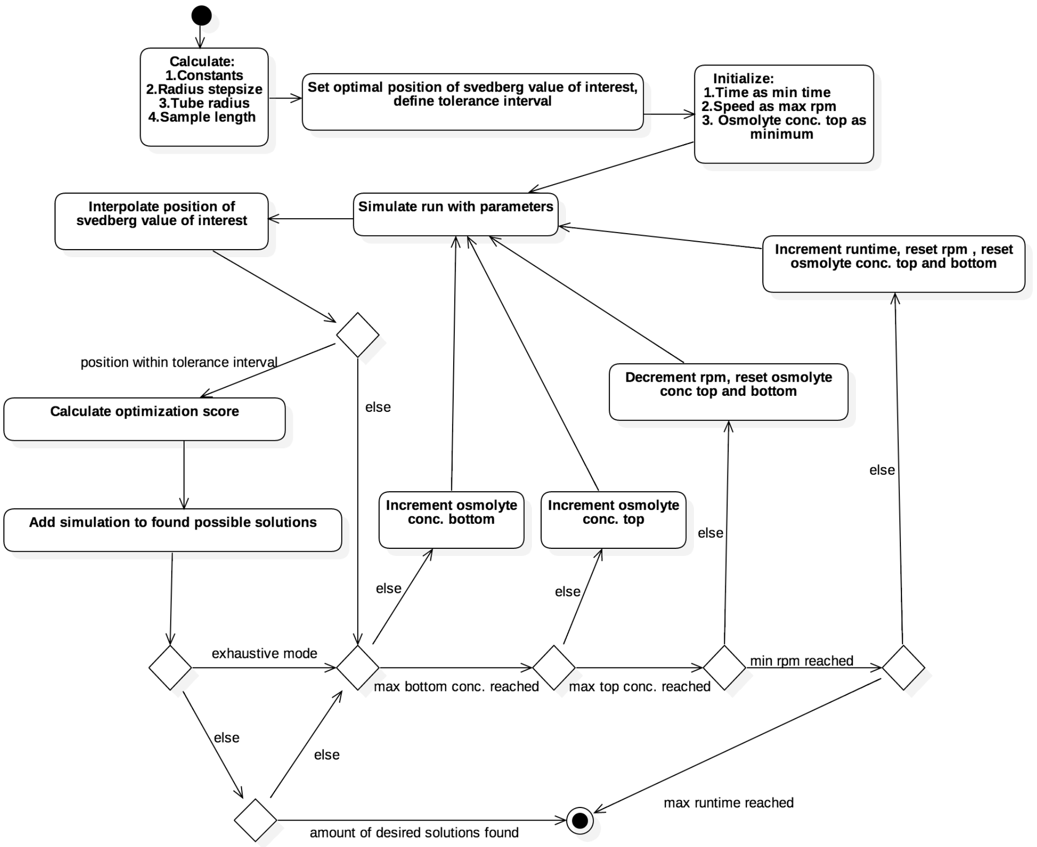

Calculate:
1.Constants
2.Radius stepsize
3.Tube radius
4.Sample length
Set optimal position of svedberg value of interest, define tolerance interval
Initialize:
1.Time as min time
2.Speed as max rpm
3. Osmolyte conc. top as minimum
Interpolate position of svedberg value of interest
Simulate run with parameters
Increment runtime, reset rpm , reset osmolyte conc. top and bottom
position within tolerance interval
else
Calculate optimization score
Decrement rpm, reset osmolyte conc top and bottom
else
Add simulation to found possible solutions
Increment osmolyte conc. bottom
Increment osmolyte conc. top
else
else
else
exhaustive mode
min rpm reached
max bottom conc. reached
max top conc. reached
else
else
max runtime reached
amount of desired solutions found

FIGURE 3.12: Activity diagram of the run condition optimization algorithm. Shown are the nested loops for the brute-force search of the reduced parameter space and the evaluation of parameter combinations.

3.1.7.1 Invariable User Inputs

Practical considerations dictated that a subset of the parameters needed for simulation needed to be held immutable in respect to the user input:

1. T: Proper temperature is essential for ensuring protein stability during the run and therefore needed to be invariable.

2. Rotor, V_{grad} and V_{samp} : As only a limited amount of sample is available to the experimentalist, the suitable rotor needed to be chosen beforehand.
3. ρ_p : Being a property of the protein at hand, the particle density is usually kept at the standard value of $1.40 g/cm^3$.
4. Osmolyte type: Due to biochemical considerations, the kind of osmolyte to be used should not be varied.

Furthermore, the number of fractions or the fraction volume and a Svedberg value of interest needs to be defined with up to four Svedberg values belonging to contaminants being optionally provided.

3.1.7.2 Parameters to Optimize Based on User Constraints

The remaining mutable parameters, whose search space is further reduced through user defined thresholds, are the ones to be optimized and are listed in the following:

1. t_{run} : Here, a lower and an upper threshold were deemed useful due to scheduling considerations of the long centrifugation times of usually 16 hours.
2. c_{top} and c_{bottom} : Again due to biochemical considerations, a maximum threshold for both top and bottom osmolyte concentration values was introduced.
3. v_{rotor} : While no thresholding from the user side was deemed useful, rotor speed is inherently constrained by the rotor's max speed.

3.1.7.3 Precalculations and Initialization

As the used rotor, sample volume and osmolyte type were invariant, the corresponding constants step size, tube radius and sample length are calculated once in the beginning to avoid repetitive calculations at the beginning of every simulation.

Next, the interval that defines an optimal fractional position for the $S_{20,w}$ value of interest is calculated based on empirical considerations:

$$frac_{opt} = \begin{cases} n_{frac} * \frac{1}{2} & \text{if } n_{contaminants} \geq 1 \\ n_{frac} * \frac{2}{3} & \text{else} \end{cases} . \tag{3.15}$$

The tolerance interval is then defined as

$$\left[frac_{opt} - \frac{n_{frac}}{10}, frac_{opt} + \frac{n_{frac}}{10} \right] . \tag{3.16}$$

Under the assumption that a short runtime and a low osmolyte concentration is usually desirable, the starting value for t_{run} and c_{top} are initialized to the defined minimum, while v_{rotor} is set to v_{max}.

3.1.7.4 Positional Interpolation and Pre-Check

Linear interpolation between the two adjacent fractions holding the two $S_{20,w}$ values smaller and bigger than $S_{sample,20,w}$, as described in 3.1.3, yields its predicted position on the gradient. Only simulations where the $S_{sample,20,w}$ is predicted at a point within the tolerance interval 3.16 are considered acceptable conditions and are then further evaluated through calculation of an optimization score.

3.1.7.5 Optimization Score Calculation

The goal of the optimization score α is to provide a metric to compare the quality of different simulations that passed the pre-check. Its value is normalized to be within 0 and 1, with 1 being the best possible simulation. α is calculated based on the best possible results for a given set of restraints in an isolated fashion for every simulation. Compared to the alternative approach of relating every simulation to one another, the approach used here has the advantage of signifying the general feasibility of a density gradient centrifugation experiment.

If only one S value is provided by the user, the distance from the optimal position $|frac_{peak} - frac_{opt}|$ in relation to the distance and the calculated standard deviation is used to calculate the optimization score:

$$\alpha_{iso} = \frac{\left(1 - \frac{|frac_{peak} - frac_{opt}|}{(n_{frac}/10)}\right) + \left(1 - \frac{2 \cdot \sigma}{n_{frac}}\right)}{2} \, . \tag{3.17}$$

If two or more S values are supplied by the user, the positions of the calculated distributions can be taken into account. Due to the unreliability of the standard deviation prediction, the positions of the closest peaks on the left and the right of the peak position, μ_{left} and μ_{right}, of the S value of interest are calculated as coefficients of the maximum possible distance in the given direction:

$$\begin{aligned} p_{left} &= 1 - \left(\frac{\mu_{left}}{\mu_{samp}}\right) , \\ p_{right} &= 1 - \left(\frac{\mu_{right}}{n_{frac} - \mu_{samp}}\right) . \end{aligned} \tag{3.18}$$

The width of the predicted standard deviation is then weighted down due to its unreliability and combined with the peak distance coefficients to yield the final score:

$$\alpha_{mix} = \frac{9 \cdot p_{left} + 9 \cdot p_{right} + 2 \cdot \left(1 - \frac{2 \cdot \sigma}{n_{frac}}\right)}{20} \, . \tag{3.19}$$

α is rounded to two decimal places in order to not overinterpret the accuracy.

3.1.7.6 Looping and Abortion Criteria

If $S_{sample,20,w}$ passes the pre-check, it is added to the list of found solutions together with its optimization score and optimization metrics. If the non-exhaustive mode is chosen, the algorithm is aborted once the specified amount of solutions is found. If either the abortion criteria is not reached or exhaustive mode was chosen, the four parameters are changed one at a time in nested loops.

c_{top} and c_{bottom} of the tube are incrementally tried out first. As the next level, v_{rotor} is decremented in steps of 500 rpm, which for the three considered rotors corresponds to not more than 120 possible values. Last, t_{run} is increased in steps of 10 minutes. Assuming that runs will reasonably last between 6 and 48 hours, the maximum amount of possible values is therefore 252. The increment/decrement steps for time and rotor speed were empirically chosen in agreement with users.

3.1.7.7 Computational Complexity

The possible simulation parameters are combinatorial, so the product of the maximum values yields the upper limit for run simulations the optimization algorithm has to perform and check. Without any constraints, the maximum number of osmolyte combinations would be:

$$n_{Comb.,Osmolyte} = \sum_{i=2}^{30} i \cdot (80 - i) = 11426\,. \tag{3.20}$$

This naive calculation is based on the supported minimum (5%) and maximum (30%) values for the top concentration and minimum (20%) and maximum (60%) for the bottom concentration from table A.1 and A.2, only considering integer values and assuming a minimum concentration distance of 15 while ignoring outlier values from the tables. Assuming a step size of 1 for v_{rotor} and t_{run}, the unreduced search space would be

$$max_{unreduced}(n_{sim}) = n_{Comb.,Osmolyte} \cdot max(v_{rotor}) \cdot max(t_{run}) \approx 2 \cdot 10^{12}\,. \tag{3.21}$$

For a given rotor, the average amount of available gradients for the two currently supported osmolytes is 14. Therefore for the reduced space, the maximum amount of simulations is reduced to

$$max(n_{sim}) = n_{Comb.,Osmolyte,red.} \cdot \frac{max(v_{rotor})}{500} \cdot \frac{max(t_{run})}{10} \approx\ 420000\,. \tag{3.22}$$

The complexity reduction is therefore already $\approx\ 5 \cdot 10^6$ on average before taking into account thresholds supplied by the user, which makes the brute-force approach computationally feasible.

The computational complexity is again linearly dependent on the number of fractions and dependent on the number of possible conditions to be checked. This dependency is shown in figure 3.13. Due to the optimization algorithm judging every simulation result in an isolated manner, the algorithm has a linear running time of $O(n \cdot m)$, where n is the number of fractions and m is the number of conditions to simulate. As to be expected, the runtime is increased majorly compared to a singular simulation.

3.1.7.8 Simulation Optimization Results

Exploring the whole parameter space of the optimization was not feasible, therefore, the previously used 20S Proteasome iand the SW40 Rotor was again chosen as the test sample. First, it was tested how many acceptable conditions where the sample is predicted to peak within the tolerance interval defined in 3.16 could be found depending on the number of possible conditions. Here, sucrose was chosen as osmolyte and thresholds were set to the highest values in order to allow all possible combinations. Starting from the mean allowed runtime of $360 + \frac{2880-360}{2} = 1620$ minutes, the minimum allowed time was decreased while the maximum allowed time was increased, both in steps of 126 minutes for every data point. The number of found good conditions was then determined for the 20S sample as well as for 15S and 25S samples, as plotted in figure 3.14. A quasi linear dependence of found good

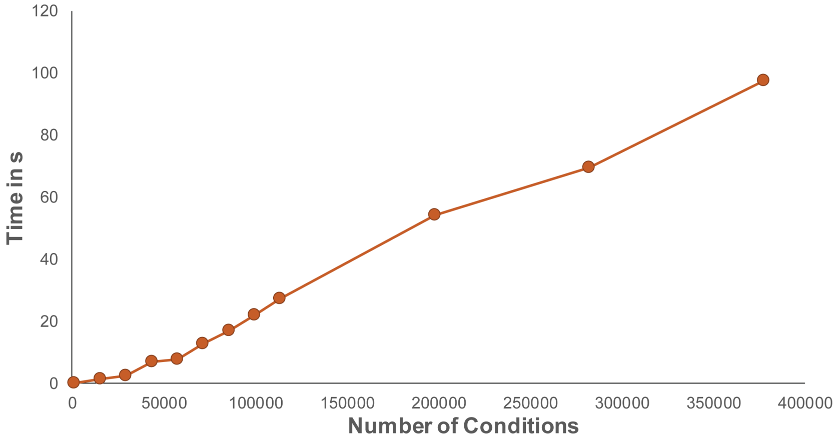


FIGURE 3.13: Runtime of the optimization algorithm depending on input size. Shown are measurements of runtime depending on the number of conditions allowed by the user-defined thresholds, yielding a linear relationship.

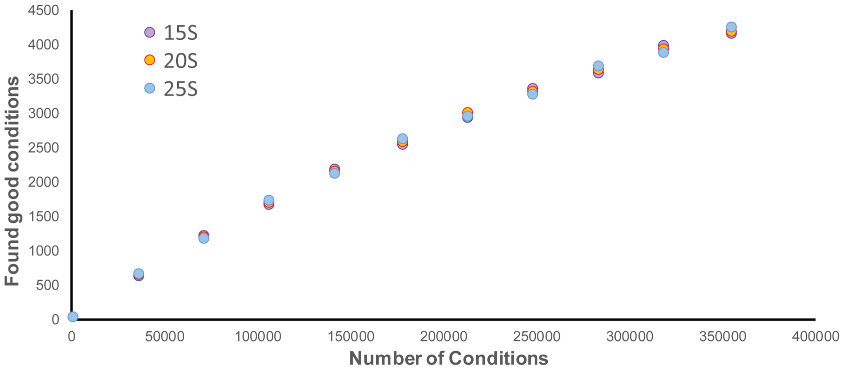


FIGURE 3.14: Found good conditions for single S values depending on number of conditions. Simulations were performed in a SW40 rotor with a sample volume of 0.1ml, ρ_p=1.4 and non-thresholded sucrose concentrations while the possible runtimes were increased equally in both directions.

conditions on the number of possible runtimes could be observed for all three S values.

Next, optimization simulations were run with a 20S sample and 15S+25S contaminants and again with 5S+35S contaminants. Sucrose was again chosen as the osmolyte with no concentration thresholds. The runtime constraints were modified for every data point as for the previous simulation. The biggest found peak distance to both sides and the calculated optimization score for both optimization scenarios are shown in figure 3.15.

It can be seen that in this particular scenario, the increased number of possible conditions doesn't result in overall bigger separation of the species, which here has an upper limit based on the S value difference between the species reflected in the optimization score. A bigger parameter space resulting in more conditions with the same optimization score can still have advantages based on user preferences concerning osmolyte concentration and runtime.

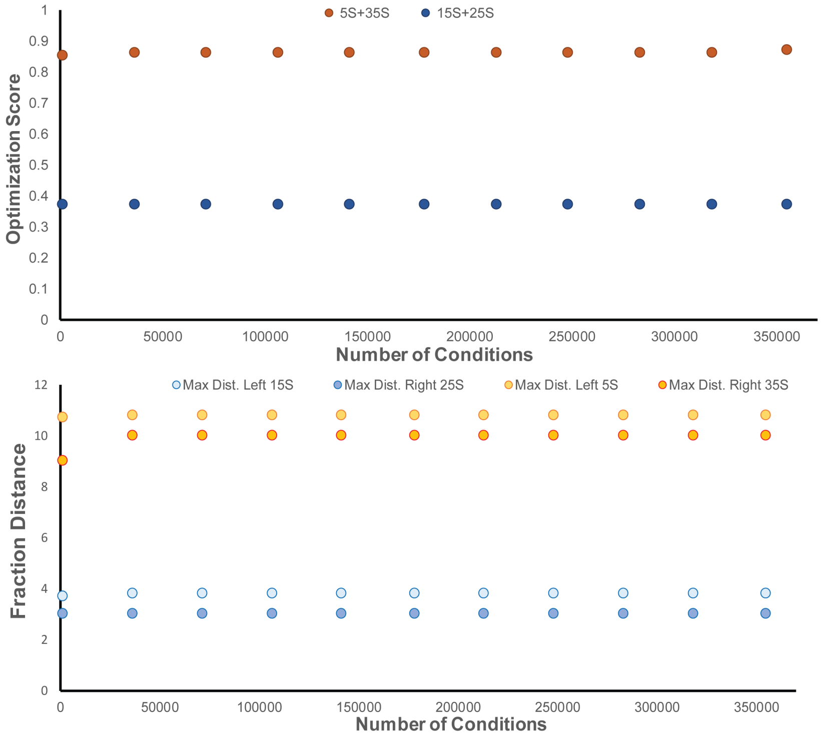


FIGURE 3.15: Peak distance and optimization scores for two optimization simulations with different number of conditions. One simulation was run with a 20S sample to be separated from a 15S and a 25S contaminant, the other with a 20S sample to be separated from a 5S and 35S sample. Shown are the best optimization score and the maximum fractional distance for different number of conditions that were tested. Simulations were performed in a SW40 rotor with a sample volume of 0.1ml, ρ_p=1.4 and non-thresholded sucrose concentrations while the possible runtimes were increased equally in both directions.

3.1.8 CowGrace Frontend

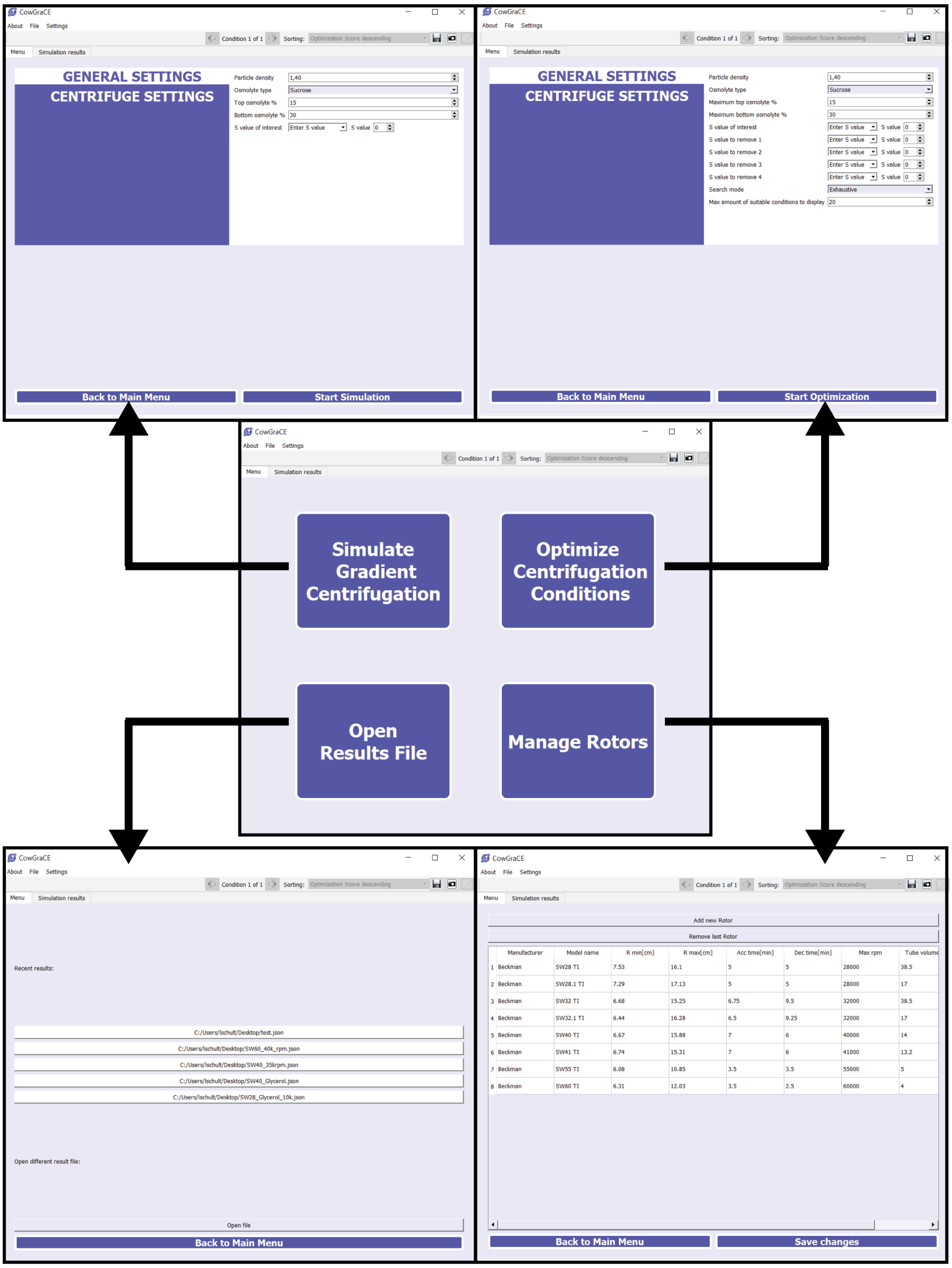
The CowGrace frontend was developed with a focus on convenience and usability. CowGraCE consists of two tabs: One holds the main menu with four different functionalities and the second holds the result display that is initiated the first time a simulation is performed or loaded after the software is opened. The different parts of the GUI are shortly explained in the following.

3.1.8.1 Rotor Management

To ensure extendability of CowGraCE, a feature to add and remove swing-out rotors was implemented. The user interface of this is shown in fig. 3.16. The rotor data is saved as a human readable JSON file in the "share" subfolder of the application that can be transferred to another computer or modified by hand, if so desired. The software is equipped with a standard list of rotors based on the available models in the Department of Structural Dynamics.

3.1.8.2 Result Loading

Previously saved simulation or optimization results (see 3.1.8.5) can be loaded from the corresponding option in the main menu as shown in fig. 3.16. The software keeps track of the last five saved result files and displays them, along with an option to open files not shown. Once a file is selected, the software opens the result display and the loaded parameters are entered as simulation settings.



Particle density: 1,40
Osmolyte type: Sucrose
Top osmolyte %: 15
Bottom osmolyte %: 30
S value of interest: Enter S value; S value: 0

Particle density: 1,40
Osmolyte type: Sucrose
Maximum top osmolyte %: 15
Maximum bottom osmolyte %: 30
S value of interest: Enter S value; S value: 0
S value to remove 1: Enter S value; S value: 0
S value to remove 2: Enter S value; S value: 0
S value to remove 3: Enter S value; S value: 0
S value to remove 4: Enter S value; S value: 0
Search mode: Exhaustive
Max amount of suitable conditions to display: 20

Recent results:

C:/Users/lschult/Desktop/test.json
C:/Users/lschult/Desktop/SW60_40k_rpm.json
C:/Users/lschult/Desktop/SW40_35krpm.json
C:/Users/lschult/Desktop/SW40_Glycerol.json
C:/Users/lschult/Desktop/SW28_Glycerol_10k.json

Open different result file:

	Manufacturer	Model name	R min[cm]	R max[cm]	Acc time[min]	Dec time[min]	Max rpm	Tube volume
1	Beckman	SW28 TI	7.53	16.1	5	5	28000	38.5
2	Beckman	SW28.1 TI	7.29	17.13	5	5	28000	17
3	Beckman	SW32 TI	6.68	15.25	6.75	9.5	32000	38.5
4	Beckman	SW32.1 TI	6.44	16.28	6.5	9.25	32000	17
5	Beckman	SW40 TI	6.67	15.88	7	6	40000	14
6	Beckman	SW41 TI	6.74	15.31	7	6	41000	13.2
7	Beckman	SW55 TI	6.08	10.85	3.5	3.5	55000	5
8	Beckman	SW60 TI	6.31	12.03	3.5	2.5	60000	4

FIGURE 3.16: CowGraCE main window. Shown in the middle is the main menu from which the result loading widget (bottom left), the rotor management widget (bottom right) and the simulation and optimization modes (top left and top right) can be started.

3.1.8.3 Simulation Mode

The basic simulation mode requires the user to provide all needed values. These are separated in two categories, general and centrifuge settings. General settings are temperature, particle density, osmolyte type, osmolyte top and bottom concentration and a $S_{20,w}$ value of interest whose predicted distribution will be displayed in the result window, which can either be entered directly or predicted from an entered molecular weight by formula 2.59. Centrifuge settings consist of temperature and rotor related settings that include gradient volume, sample volume, centrifuge speed, run time and number of fractions which can either be entered as a number or as volume per fraction. Based on the rotor selected from the drop-down menu, the input fields are restricted to valid values based on the rotor information.

Furthermore, an option to calculate rotor settings based on a rotor's k-factor was implemented. The k-factor is a measure of rotor efficiency which is defined as

$$k = \frac{2.533 \cdot 10^5 \cdot ln(\frac{r_{min}}{r_{max}})}{(\frac{rpm}{1000})^2} . \tag{3.23}$$

Through the k-factor, runtimes necessary for the same sedimentation behavior in two rotors can be calculated through the relation

$$\frac{t_1}{k_1} = \frac{t_2}{k_2} . \tag{3.24}$$

This feature is useful if scaling up a purification and switching to a bigger rotor, for example. Here, three modes are offered:

1. **Calculate Runtime from known rpm**: The runtime of the current rotor is calculated from the selected old rotors' runtime and rpm and the current rotor's rpm.
2. **Calculate rpm from known runtime**: The rpm of the current rotor is calculated from the selected old rotors' runtime and rpm and the current rotor's runtime. Due to the maximum rotor speed, it is possible that no valid value can be found, in this case, -1 is entered into the rpm field.
3. **Find closest rpm and runtime from runtime preference**: In this newly developed mode, there are two unknown variables that are to be found. To solve this problem, the user can enter a preferred runtime and specify whether it's to be treated as the minimum, the maximum or neither. Starting at the preferred runtime, it is evaluated whether a rotor speed within the rotor's speed limits can be found. The simple algorithm for this is shown in fig. 3.17.

Once all values are set, clicking the corresponding button starts the simulation, creates the result window if needed and switches to it.

3.1.8.4 Optimization Mode

In optimization mode, a slightly different user input is needed. General settings now include max top concentration and max bottom osmolyte concentration, and up to four contaminating $S_{20,w}$ values in the sample that can be entered. Also, the user can choose how many suitable conditions should be displayed later and whether the search should be exhaustive or be aborted after the amount of conditions is found.

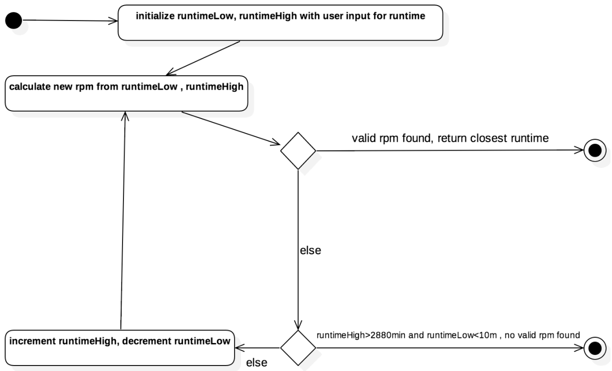

initialize runtimeLow, runtimeHigh with user input for runtime
calculate new rpm from runtimeLow , runtimeHigh
valid rpm found, return closest runtime
else
runtimeHigh>2880min and runtimeLow<10m , no valid rpm found
increment runtimeHigh, decrement runtimeLow
else

FIGURE 3.17: Simple algorithm for finding the closest runtime. Shown is the bidirectional search, for the initial runtime set to maximum or minimum a one directional search is employed. The runtime increment/decrement was chosen to be 10 minutes. The maximum runtime of 48 hours was an empirically chosen value.

Centrifuge settings now include min and max runtime and have no user input for rpm.

3.1.8.5 Result Display

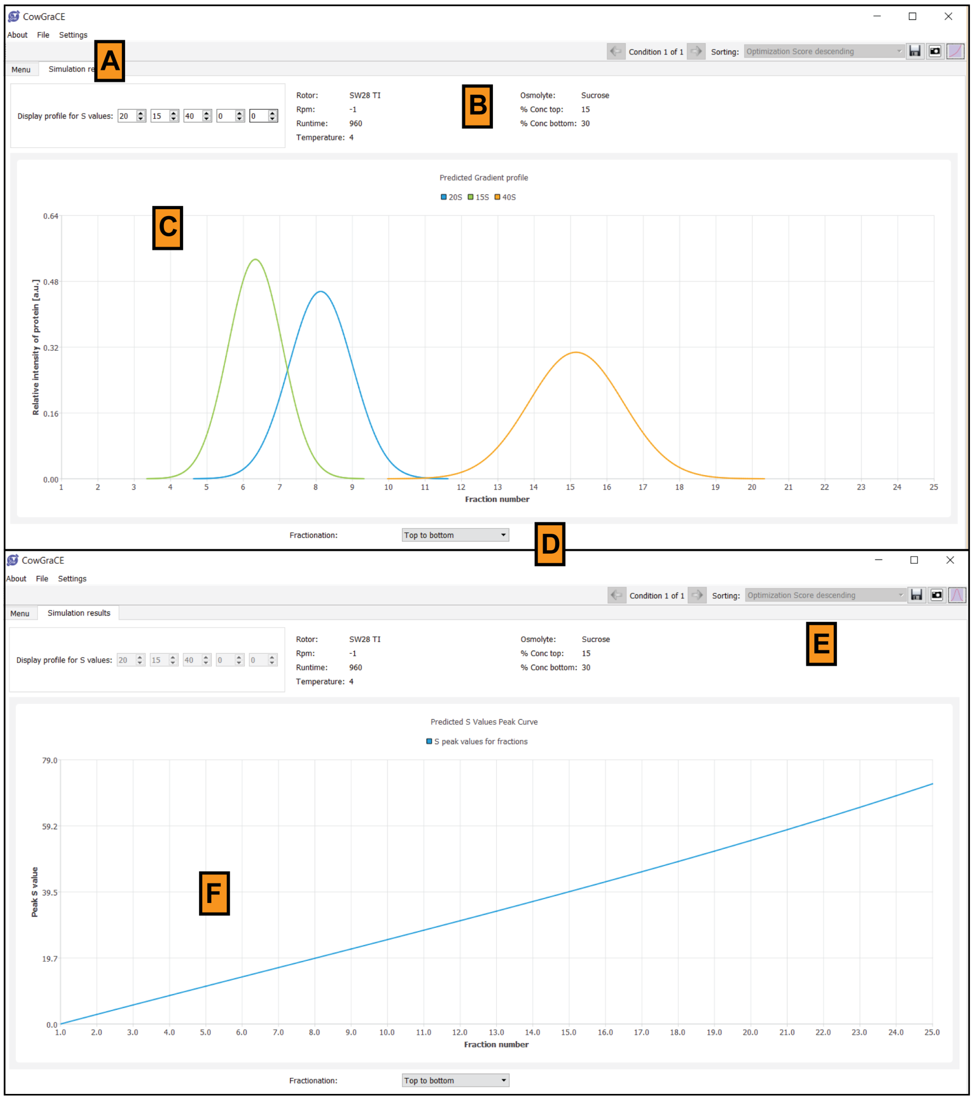
In the result display, predicted protein distributions along the fractions are shown together with multiple customization options and information. Up to 5 $S_{20,w}$ values can be chosen to be displayed along with their particle distribution as fraction number against relative intensity, also the optimal peak position based on equation 3.15 is shown. The fractionation direction can be set which inverts the graph. The plot can be changed to display fraction number against peak $S_{20,w}$ values at the fractions. Also, the display can be saved as an image and the data can be saved as a JSON file. The interface of the result display for simulation with set parameters is shown in fig. 3.18.

In optimization mode, the display usually holds more than one simulation result. The run conditions that were judged as suitable can be sorted by the user based on their optimization score, distance from optimal position(in descending manner), maximum top and bottom osmolyte percentage and runtime (in ascending and descending manner). Also, the optimization metrics are displayed.

3.1.9 CowGraCE Backend

The backend of the cross-platform object oriented software is split into two parts: The dynamically linked simulatorLibrary that handles all algorithmic operations and the CowGraCE application itself that handles GUI aspects and data management. An overview over the class relations is given in figure 3.19. The cowGraceApplication namespace is shortened to CGA in the following, and the simulatorLibrary namespace is shortened to SL.

The CGA::GraceMainWindow spawns all the other widgets and facilitates communication between them. The CGA::RotorManagementWidget utilizes a std::vector of a rotor struct to save, load and pass the rotor data to the other widgets. The



Condition 1 of 1
Sorting: Optimization Score descending

Display profile for S values: 20 15 40 0 0

Rotor: SW28 TI
Rpm: -1
Runtime: 960
Temperature: 4

Osmolyte: Sucrose
% Conc top: 15
% Conc bottom: 30

Fractionation: Top to bottom

Condition 1 of 1
Sorting: Optimization Score descending

Display profile for S values: 20 15 40 0 0

Rotor: SW28 TI
Rpm: -1
Runtime: 960
Temperature: 4

Osmolyte: Sucrose
% Conc top: 15
% Conc bottom: 30

Fractionation: Top to bottom

FIGURE 3.18: Result display for a simulation with set parameters. A: $S_{20,w}$ values selection, B: Important run parameters, C: Display area of h chosen distributions, D: Fractionation direction choice, E: Save buttons and switch between F and C, F: Display of peak $S_{20,w}$ values at the fractions.

CGA::SimulationsOptionsWidget holds different options depending on the simulation mode and stores the options that are returned to the CGA::GraceMainWindow once simulation start is requested in an SL::ArgumentContainer, which is a modified parameter object from the CowLib. The CGA::GraceMainWindow then creates a CGA::SimulationDataView object that represents the result display. Also, a CGA::SimulationHandlerThread is spawned with the simulation options that, as the name indicates, runs in a thread separate from the GUI thread. This has the advantage that the GUI is still responsive while calculations take place, and also, a crash in the calculation thread doesn't result in shutdown of the whole software. In the CGA::SimulationHandlerThread, either a SL::Simulator or a SL::ConditionOptimizer object is spawned with the simulation options that return a vector of a specialized struct that holds simulation results and optimization metrics which are left empty in case of a simple simulation. This struct is then sent to the CGA::SimulationDataView.

The SL::ConditionOptimizer and SL::Simulator implement the previously introduced algorithms in a speed-optimized manner, relying solely on std::vector and struct objects. Based on the calculation speed, no parallelization was deemed necessary to be implemented.

The CGA::SimulationDataView initializes a primitive data model with the results, thus implementing the Model-View-Controller (MVC) programming concept. From the held peak $S_{20,w}$ values and the estimated standard deviation values, particle distributions of the user-decided $S_{20,w}$ values of interest are calculated and displayed dynamically.

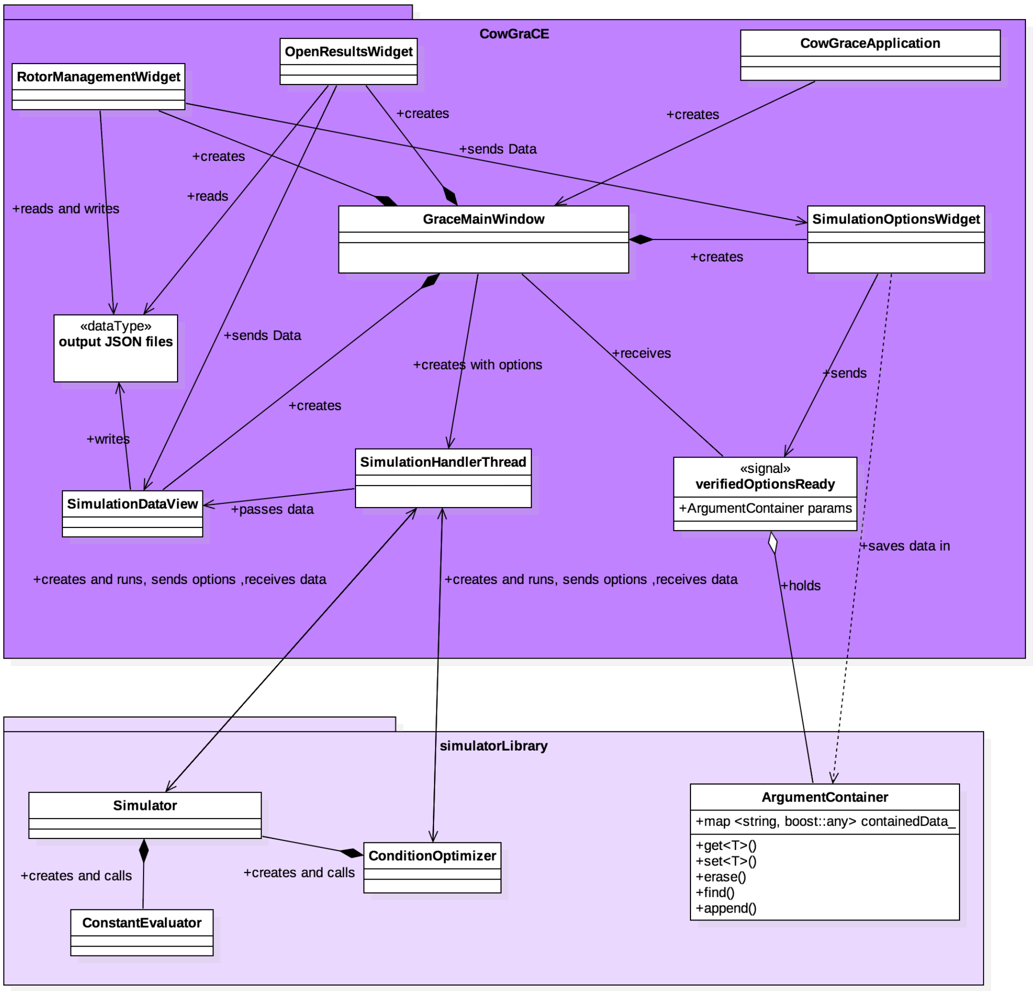


FIGURE 3.19: Diagram of CowGraCE architecture in UML 2 notation. Shown are the CowGraCE application, the dynamic simulatorLibrary and their most important classes and relations.

3.2 Micrograph Quality Checker

The Micrograph Quality Checker is a preprocessing tool that helps users process their data in parallel to recording. As of now, the MQC generates training data for a machine learning based approach for image classification through analysis of the user's binary decisions on the quality of a micrograph and several developed metrics for image quality that are calculated during processing. An early prototype of the software was previously implemented by Dr. Boris Busche. The goal of this project apart from the live preprocessing aspects was to test the validity of the SVM classification methodology with different image metrics and to test on selected datasets whether a general model for good and bad images can be developed or whether a live training of a model for every dataset is necessary. For both of these cases, the minimum amount of data necessary was also to be investigated.

The MQC acts as the new entry point for the COW ecosystem. It continuously detects newly acquired data and presents real space image, the power spectrum and, if available, CTF parameters to the user who can then make a decision whether this micrograph should be retained for further processing or not. Due to the high computational demand of working with large micrographs, the software was designed to be run on a large computer cluster that has access to data recorded by the microscope. Because of the reliance of the algorithmic approach on the user's decisions and the data pipeline, the backend of the software is presented first, followed by the workflow of processing that is reliant on the software architecture. Afterwards, the machine learning based classification method and its results on chosen datasets is described, as well as its generalization capabilities.

The functional parts of the software are shown in figure 3.20. Currently, the SVM based prediction of image judgments is performed externally, while the data that comes from user decisions and from the background calculations is saved as files.

3.2.1 MQC Backend

The MQC software architecture consists of multiple objects that communicate information between one another. A schematic class diagram is shown in figure 3.21. The main principle is as follows: The mqc::Sniffer searches for data unknown to it, starts calculation jobs, sends information to the mqc::GraphCleanWidget that then displays images and metainformation about the micrograph and awaits user decisions. Once a user gives the finish signal, the judged micrographs are then separated based on their judgment.

3.2.1.1 Main Window

The mqc::mainWindow class acquires input parameters from the user, starts all independent operations and facilitates communication between the threads. The necessary parameters are:

1. **Source Folder**: The folder where the recorded data is deposited.
2. **Output Folder**: The folder where temporary files and the output location of the classified micrographs are stored.
3. **File type**: The type of file that the mqc::Sniffer searches for. As the files need to be loaded into a cow::Data object, only the supported file formats be chosen.

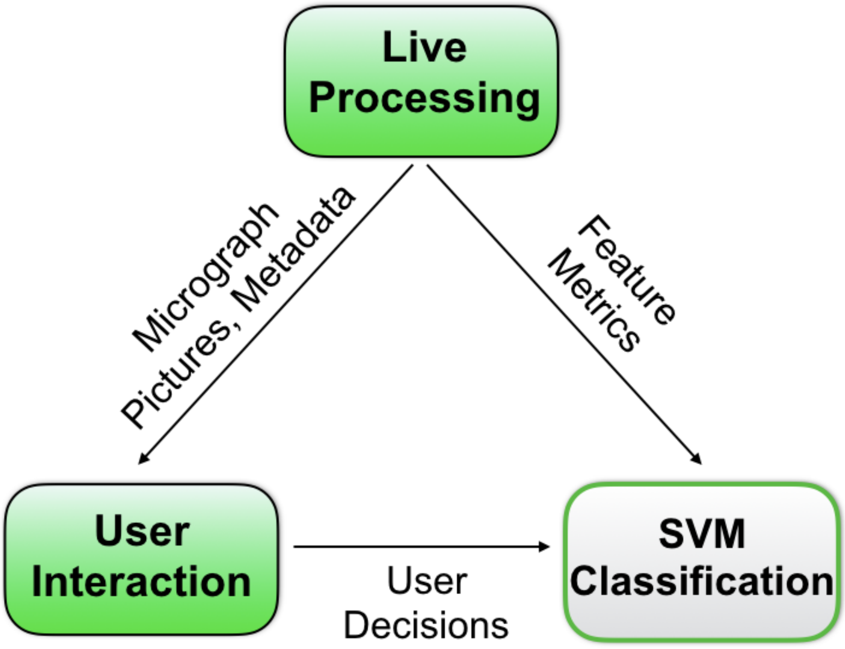


FIGURE 3.20: Overview of the Micrograph Quality Checker. Shown are the different functional components and the interaction between them.

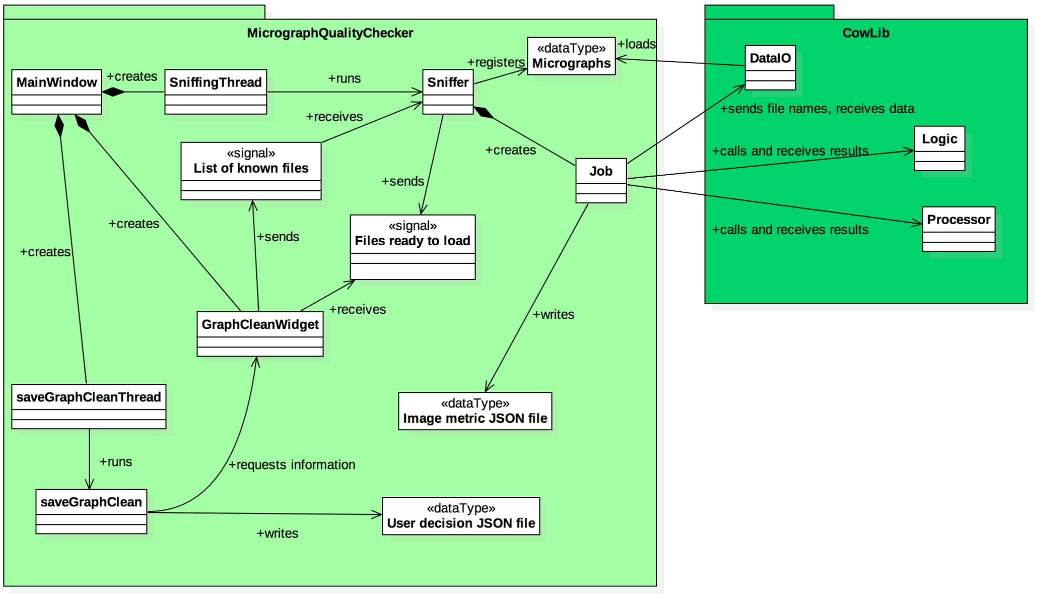


FIGURE 3.21: UML2 class diagram of the MQC backend.

4. **Result loading**: If chosen, a result file, if available in the output folder, will be loaded.
5. **GCtf results loading**: If chosen, GCtf results for every micrograph are loaded.

After receiving the start signal from the user, a parameter sanity check is performed for the parameters and upon passing, the mqc::Sniffer, that handles the live processing, receives the parameters and is initialized. The mqc::GraphCleanwidget is also initialized. If a valid result file could be detected and result loading was chosen by the user, this widget receives a signal to load the file, while the sniffer receives a signal to not start sniffing before it is passed the list of files that were successfully loaded by the mqc::GraphCleanWidget. The inner workings of result files are further elaborated on in section 3.2.1.4.

3.2.1.2 Live Processing

The live processing takes place in a separate background thread in order to not block the GUI thread. Upon initiation, the mqc::Sniffer creates the following folder structure in the output folder:

1. **Good/** : The folder where the micrographs classified as good are moved to
2. **Bad/** : The folder where the micrographs classified as bad are moved to
3. **SmallPictures/** : The folder where the small pictures of the input data is stored. This is elaborated on in the following.

Afterwards, the data sniffing operation is initiated. If no previous results were loaded, an empty list of known images is created. If results were loaded, the received file list is used as the starting point. The sniffing thread then acquires the list of files contained in the source folder, filters for the chosen file type and compares the files to the list of known files. If the file is not known, its name is saved.

It should be noted that for Titan Krios images that are recorded as frames which need to be aligned, an implementation of the MotionCor2 algorithm[210] was deemed desirable. This was attempted in collaboration with Dr. Boris Busche, but at the time of writing had not been completely finished despite showing promising results with small tweaks resulting in slightly better contrasts compared to MotionCor2. Similarly, implementation and improvement of the GCtf algorithm[208] within the COW framework could not be completed due to time constraints. Therefore these processing steps, if desired by the user, for now can only be done through a bash script written by Karl Bertram in parallel to the sniffing operation. To accommodate this, the sniffer takes into account whether a log and a result file produced by GCtf is available for a given micrograph before declaring it as known and including it in a calculation job. Due to the result file being continuously written to, an empirical value of 10 seconds was chosen for the difference between time of being last modified and declaring a file ready to be processed further.

After a list of files that are to be processed further is acquired, it is divided up into calculation jobs that each hold up to 10 filenames. These jobs are passed to a thread pool that launches the jobs in first in, first out (FIFO) order when a CPU becomes available. After the jobs are launched, the mqc::sniffer starts its sniffing operation again after a sniffing interval timeout that can be chosen by the user has passed.

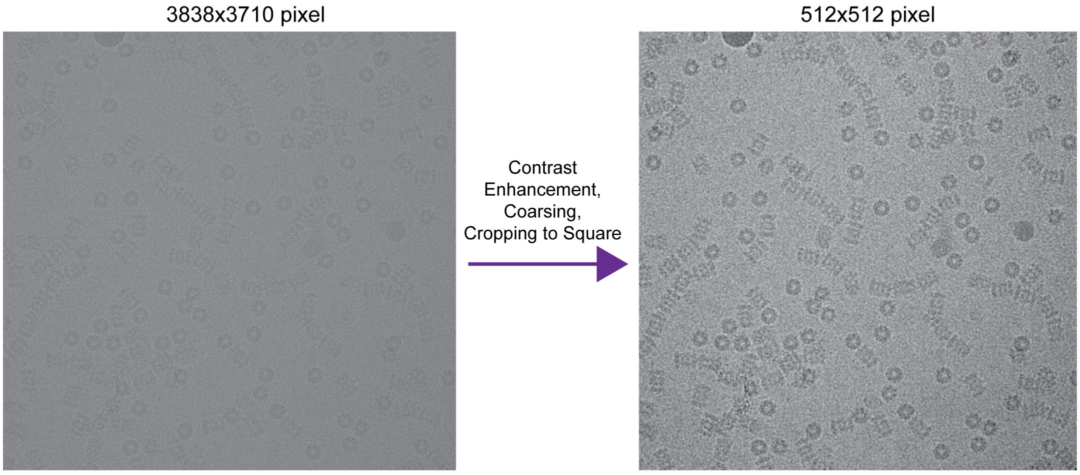


FIGURE 3.22: Comparison of original image to contrast enhanced image. Shown on the left is a cryo micrograph with 20S Proteasomes and on the right the resulting image that is presented to the user with clearly increased contrast. Both images are shown in the same size for clarity.

3.2.1.3 Calculation Jobs

A mqc::Job, each running in a separate thread, has the goal of creating a contrast-enhanced, size-reduced image and a power spectrum for every micrograph as well as optional CTF parameter determination.

First, the micrographs are loaded from the list of filenames. Then, depending on whether GCtf data is to be used or not, a power spectrum is either loaded from there or calculated from the micrograph. In the latter case, a segment-averaged power spectrum to achieve better contrast is created. From this, a boxcar filter smoothened power spectrum is subtracted as a form of background correction. Lastly, the power spectrum is convoluted with a natural logarithm function to increase visibility. The micrographs in real space are normalized, histogram equalized, outlier adjusted and coarsed to 512 pixels x 512 pixels. In case of K2 images, the images are cropped to square size. The results of this procedure are shown in figure 3.22.

The real space images and power spectra are then saved in the *SmallPictures/* folder in the PNG format and the filenames are sent back to the mqc::sniffer that sends them to the mqc::GraphCleanWidget, which presents them to the user. The advantage of saving coarsed images to the hard drive lies in the ability to reload them at any given time and to avoid straining the memory demands with the original size micrographs. Only sending file names instead of the images themselves reduces the computational load. In case of available GCtf data, the log files are parsed to yield determined $\triangle d_u$ and $\triangle d_v$ values (see equation 1.16) that signify the present astigmatism in the micrograph's power spectrum and the resLimit parameter which describes to which resolution the CTF could be fitted and therefore can be used to approximate the realistically maximum achievable resolution for particles from the micrographs. These parameters are then also sent to the mqc::GraphCleanWidget via the mqc::Sniffer.

Then, several statistical image metrics are calculated from the real space micrograph and the power spectrum that are later used for the SVM classification. These metrics are explained and evaluated in detail in section 3.2.3 and saved in a JSON file in the output folder.

3.2.1.4 User Interaction

The user interaction is handled in the mqc::GraphCleanWidget, which is spawned as soon as sniffing is initiated. The main function of this widget is to display contrast enhanced images and power spectra to the user who can then judge them as good or bad. Once a decision is made, the micrograph is marked according to the judgment and the next image is displayed, bearing some resemblance to popular dating apps. Navigating between pictures in both directions is also possible.

When starting from scratch, this widget receives data in form of filenames of processed micrographs and their power spectra that are located in the *SmallPictures/* folder as well as optional CTF parameters and stores them internally. The data is ordered chronologically by arrival time and updated dynamically. In order to minimize system requirements while avoiding delays for the user, 10 pictures in each direction of the current position in the image list are cached. Whenever the current image position is changed, the cache's contents are updated.

In the background, the mqc::saveGraphClean class is constantly updating a result file that is saved as a JSON document in the output folder. This threaded saving was deemed necessary due to the software usually running on a computer cluster that the user connects to, which can lead to spontaneous aborts. Another layer of data loss prevention is the usual approach of creating a backup file before writing to the original one. After having updated the result file, the mqc::saveGraphClean class checks the mqc::GraphCleanWidget every 30 seconds if new data to be written is available. The result file not only saves user decisions on micrographs, but also measures the time that the user spends on a micrograph between image display and occurred judgment.

In case of a result file being found in the output folder during initiation and the corresponding loading option being selected by the user, the results can be loaded by the widget, including previous user decisions as long as the PNG images in the corresponding folder are available. If so desired, the images can be sorted according to their quality judgment as performed by the user.

When the finish signal is initiated by the user, the micrographs corresponding to the judged images are then moved to the subfolder for either good or bad images.

3.2.2 MQC Frontend

The starting screen of the MQC is shown in figure 3.23. Here, a user can input the parameters detailed in the previous section and commence the live processing through a start button. Furthermore, in a settings window, common options such as maximum amount of processors to use can be defined. The micrograph judgment process can be done via keyboard with the option to configure the keys associated with the four possible operations (Next, previous, good, bad).

As soon as live processing is initiated, the screen switches to the so called GraphClean window that is also shown in figure 3.23. As the user's decision are used for the supervised learning approach, unnecessary distractions were avoided in the GUI. The user is shown a picture of a micrograph and of the power spectrum or the power spectrum together with the fitted CTF spectrum. In the ladder case, the determined CTF parameters are also shown above the pictures, together with the current filename. A progress bar shows the judgment progress and informs the user about the amount of processed micrographs, while the log area displays information about the current state of the live processing unit.

3.2.3 Image Metrics

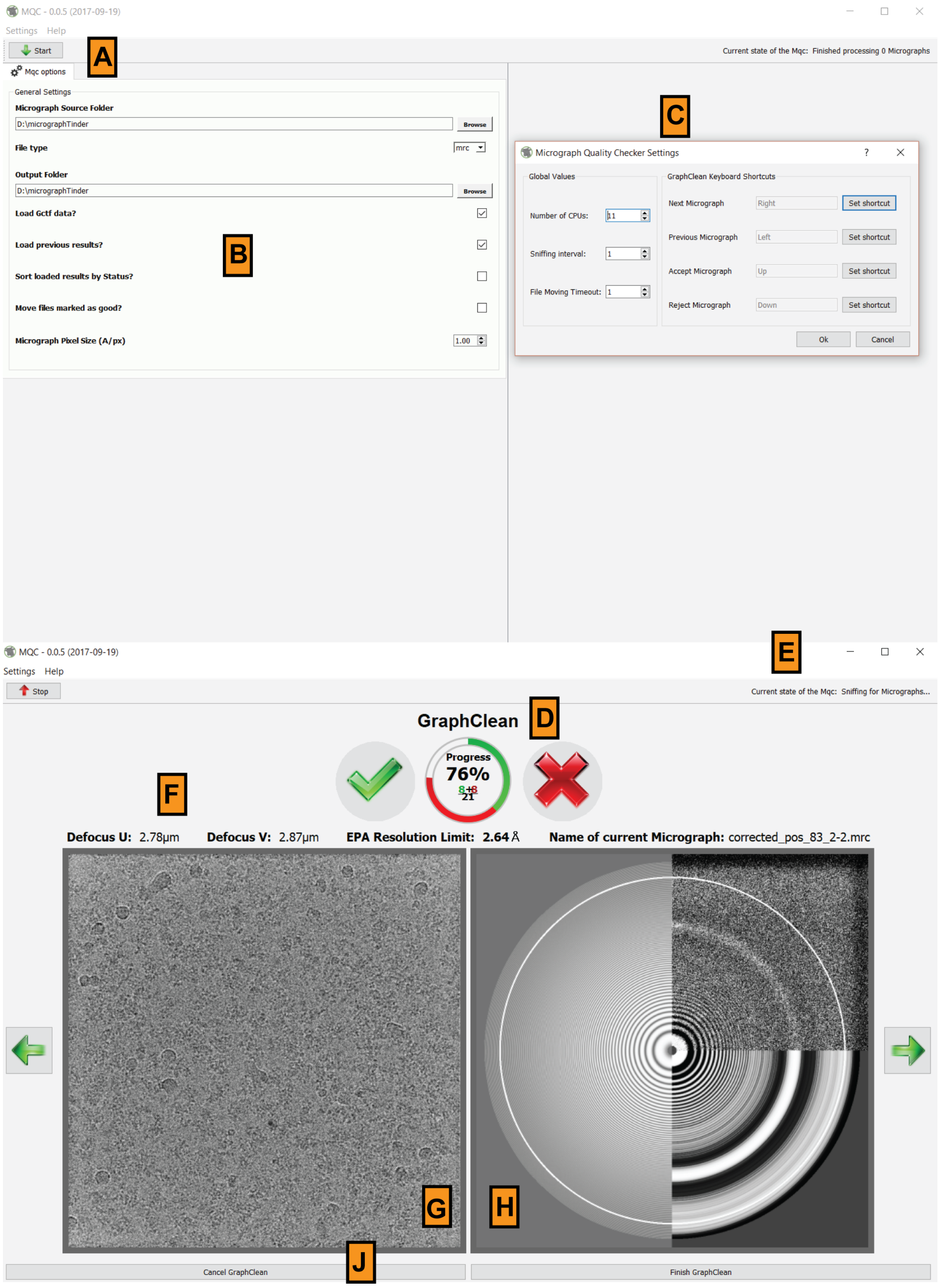
As described in section 3.2.1.3, several statistical metrics are calculated for every image during a calculation job. An overview over the metrics and the acquisition procedure is given in figure 3.24. The goal of these calculations is to accurately describe the quality of a recorded micrograph in the metric space to understand a user's decision behavior quantitatively. As the power spectrum of a micrograph contains information about the frequency-dependent signal of all particles on a micrograph which is highly relevant for high-resolution structure determination, a large share of metrics are derived from Fourier space information.

From a loaded micrograph, a segment-averaged, logarithmized power spectrum is calculated to increase the signal. The first two used metrics are its pixel value mean μ_{PS} and variance σ^2, calculated according to formulas 2.1 and 2.4. Afterwards, the power spectrum is transformed into cylindrical coordinates, which results in an image with all values at an equal distance from the center in one line, representing the signal at a spatial frequency. From this, other metrics are derived: The variance for every of the n_l lines is calculated and the maximum line variance $max(\sigma_l^2)$ determined, as well as the sum of variances $sum(\sigma_l^2)$. Treating every variance as an input value, the variance of the sum of line variances and the mean of the sum of line variances are calculated by means of equation 2.5, denoted σ_{sum}^2 and μ_{sum}. Last, the deviation from the mean, denoted d_{mean}, is determined as

$$d_{mean} = \frac{abs\left(max(\sigma_l^2) - \frac{\Sigma_{\sigma_l^2}}{n_l}\right)}{\sqrt{\sigma_{sum}^2}} . \tag{3.25}$$

In real space, the image is first subjected to a boxcar filter and then to a DoG filter. This procedure enhances low resolution information and, as can be seen in figure 3.25 accentuates image defects and contaminations. The variance of the resulting image is then calculated to be used as an image metric and denoted $\sigma_{Box,DoG}^2$.

If available, the CTF parameters $\triangle d_u$ and resLimit were also considered image metrics and used as such. Out of the two defocus values, the defocus difference was calculated as this metric holds information about the astigmatism of the micrograph. As the last metric, the judgment time of the user in milliseconds was used. It should be noted that this metric can obviously only be used to classify micrographs that have been visually inspected by a qualified user, but through possible correlation to other metrics could serve as an indicator of whether the employed metrics are relevant for a given made decision.



Current state of the Mqc: Finished processing 0 Micrographs

General Settings

Micrograph Source Folder: D:\micrographTinder

File type: mrc

Output Folder: D:\micrographTinder

Load Gctf data? ☑

Load previous results? ☑

Sort loaded results by Status? ☐

Move files marked as good? ☐

Micrograph Pixel Size (A/px): 1.00

Global Values

Number of CPUs: 11

Sniffing interval: 1

File Moving Timeout: 1

GraphClean Keyboard Shortcuts

Next Micrograph: Right

Previous Micrograph: Left

Accept Micrograph: Up

Reject Micrograph: Down

Current state of the Mqc: Sniffing for Micrographs...

Defocus U: 2.78µm **Defocus V:** 2.87µm **EPA Resolution Limit:** 2.64 Å **Name of current Micrograph:** corrected_pos_83_2-2.mrc

FIGURE 3.23: MQC GUI. Shown are the MQC main window that servers as the starting screen in the top half and the MQC GraphClean GUI in the bottom half. A: Start live processing button, B: Parameter area, C: Opened settings window, D: Progress bar and accept/reject buttons, E: Log display area, F: CTF parameter and file name display, G: Micrograph display, H: Power spectrum display, in this case with CTF fit included, J: Control buttons to cancel or finish

3.2.4 SVM Data Analysis

Based on data acquired through the manual judgment of complete datasets in combination with acquisition of statistical image metrics in the MQC, the machine learning analysis of chosen datasets is presented in the following sections. First, the selected datasets are introduced, followed by data preprocessing, model training and tuning

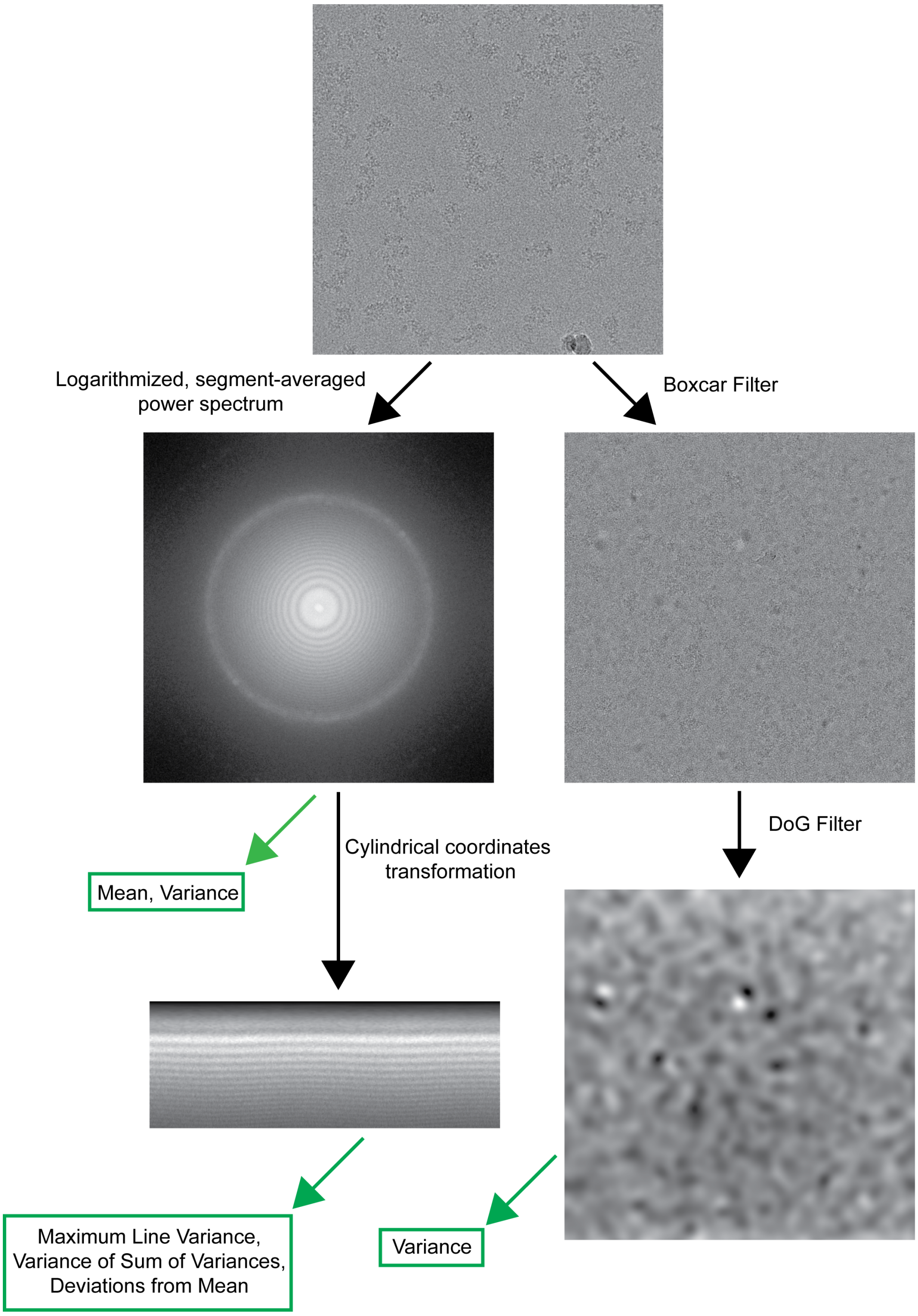

Logarithmized, segment-averaged power spectrum
Boxcar Filter
Mean, Variance
Cylindrical coordinates transformation
DoG Filter
Maximum Line Variance, Variance of Sum of Variances, Deviations from Mean
Variance

FIGURE 3.24: Statistical image metric acquisition procedure. Shown are the steps to acquire the statistical metrics for real space and Fourier Space metrics. Calculated metrics are shown in green boxes. The original micrograph contains a Spliceosomal complex sample and contains crystalline ice.

methodology and results. Lastly, the investigation into the generalization capabilities of the SVM approach are presented.

The data analysis was done in RStudio with custom written R scripts utilizing the library Dplyr[200] for data cleaning and manipulation, the library Caret[110] for correlation analysis and the library e1071[121] for SVM modeling. The latter was

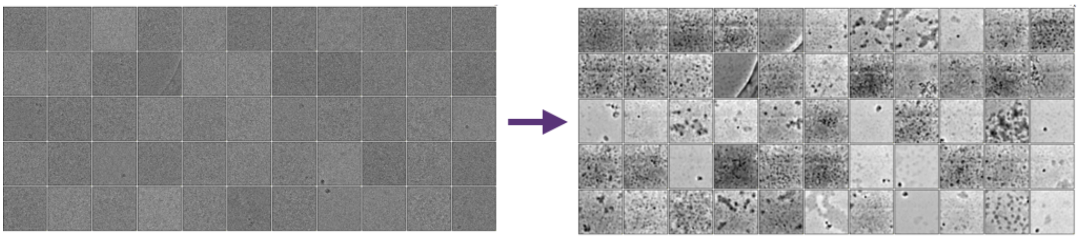


FIGURE 3.25: Effects of the Boxcar and DoG filter procedure. Shown are 55 cryo micrographs containing Spliceosomal complexes before and after the filter procedure.

chosen from the many SVM libraries due to presenting a R interface for the high-performance and award-winning C++ SVM library libSVM [31] which could easily be integrated into the COW framework at a later date.

3.2.4.1 Datasets Chosen for Modeling Analysis

Three datasets, which are summarized in table 3.4, were selected for further analysis. Two contained well-characterized protein complexes and were recorded on a Titan Krios microscope under usage of cryo preparation and different cameras, while the last dataset contained a CM200 negative stain screen of a protein complex denoted Protein X. Every dataset had a different amount of overall images and different quality judgment rates and was rated by the person working on the specific protein complex, ensuring judgment reliability. The Titan Krios datasets were rated by the same person. Exemplary micrographs that were rated good and bad for datasets A and B are shown in figures B.1 and B.2 in the Appendix.

TABLE 3.4: Overview of datasets selected for machine learning analysis

	Dataset A	**Dataset B**	**Dataset C**
Protein Complex	Spliceosomal B Complex	20S Proteasome	Protein X
Microscope + Camera	Titan Krios+ Falcon III	Titan Krios + Gatan K2	CM200
Magnification	120k	160k	88k
Pixelsize	1.16	0.826	2.5
Frames	20	60	-
Preparation	Cryo	Cryo	Negative Stain
Foil	Yes	Yes	Yes
CTF information	Yes	Yes	No
Number of judged images	8870	2262	1000
Share of images judged as good	62.6%	74.3%	35.6%
Rater ID	1	1	2

To estimate the reliability of the training data sets, rater 1 was asked to re-rate 262 random images from dataset B, corresponding to around 12% of the dataset, which were then compared to the previous ratings, as shown in table 3.5.

TABLE 3.5: Confusion Matrix of Intra-Rater Consistency

	Bad in first Rating	**Good in first Rating**
Bad in second Rating	52	17
Good in second Rating	6	187

The calculated κ from table 3.5 was 0.82, which signifies excellent agreement by both thresholds proposed in the literature. Generalizing based on this sample, the datasets were assumed to be reliably rated. The rating accuracy of 91.3% was regarded as a reasonable estimate of the human error or inconsistency.

3.2.4.2 Data Preprocessing

First, the calculated statistical metrics were combined with the user decisions and decision time, which were capped at 100 seconds.

Then, a correlation matrix (CM) was calculated from the statistical metrics, the CTF information (if available) and the decision times. Metrics with a maximum Pearson Correlation Coefficient(PCC)>0.75 were iteratively removed from the dataset as described in section 2.2.3.1. Then, the logarithm of the eliminated metrics was calculated to see whether rescaling would lessen their correlation to other metrics. The correlation matrix was then calculated again with the logarithmized metrics, again iteratively removing metrics with PCC>0.75 from the dataset. The complete first and second correlation matrices for all three datasets are shown in appendix B. Elimination results are shown in table 3.6. Furthermore, it was seen that when including the position of the micrograph in the matrix, no strong correlation to any other variable could be seen for any of the three datasets, which enabled random partition of the labeled data in the further analysis steps.

TABLE 3.6: Removed metrics from the datasets based on correlation values. Listed are the eliminated redundant metrics after each of the correlation matrix determination steps and the amount of metrics left for SVM model training.

	Dataset A	**Dataset B**	**Dataset C**
Removed in CM1	$max(\sigma_l^2)$, $sum(\sigma_l^2)$, μ_{sum}	$max(\sigma_l^2)$, $sum(\sigma_l^2)$, μ_{sum}	$max(\sigma_l^2)$, $sum(\sigma_l^2)$, μ_{sum} , σ^2
Removed in CM2	$log_{10}(max(\sigma_l^2))$, $log_{10}(\mu_{sum})$	$log_{10}(sum(\sigma_l^2))$, $log_{10}(\mu_{sum})$	$log_{10}(sum(\sigma_l^2))$, $log_{10}(\mu_{sum})$, $log_{10}(\sigma^2)$
Number of Metrics after CM2	10	10	6

For all three datasets, only the power spectrum statistical metrics correlated highly with one another, which could be expected due to their non-independent determination. Lastly, the metrics were each normalized to a range between 0 and 1 to ensure no weighting bias due to the SVM approach taking into account the absolute value distance.

3.2.4.3 Hyperparameter Tuning and Performance Evaluation

For each dataset, the approach of randomly separating the dataset into 70% training data, on which an optimized model is trained, and 30% testing data, on which the performance of the model is evaluated, was chosen, as illustrated in figure 3.26. This was repeated 10 times for each dataset to quantify the effects of the random separation into the two sets. While random, each selection was checked to have both images classified as good and images classified as good in both training and testing data. Then, the hyperparameters were tuned on the training data with an adaptive grid search. In this analysis, the radial Gaussian kernel function was used in SVM modeling, for which the free parameter γ needed to be tuned in addition to the general SVM cost parameter C.

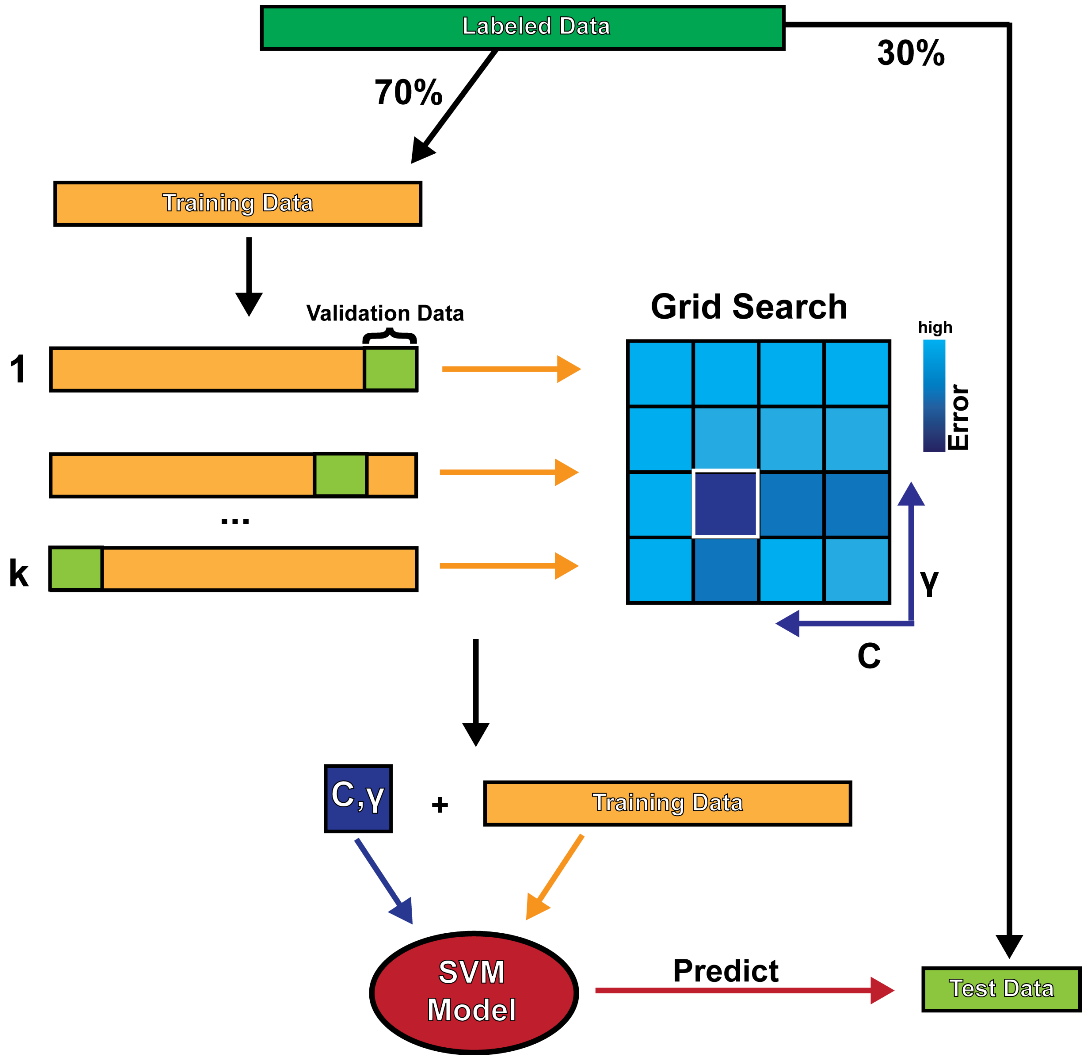


FIGURE 3.26: Illustration of the SVM classification workflow. After splitting the data into training and test set, 5-fold cross-validated grid search is performed and the resulting model's performance is evaluated on the test set. This procedure was repeated 10 times for every dataset.

Hyperparameter tuning was done through an adaptive grid search with 5-fold cross-validation. The grid search is necessitated through the interdependence of the parameters. For the tuning search range, $C \in [10^{-1}, 10^{4}]$ and $\gamma \in [10^{-4}, 10^{1}]$ were used. Exemplary visualizations of the grid search results are shown in figure B.3 in the Appendix. For all three datasets, different hyperparameter combinations proved to be optimal for minimizing the *RMSE*, as shown in table B.7. The lowest *RMSE* was observed for dataset B.

Then, the optimal hyperparameters were used for training a SVM model on the whole training data, which was subsequently used to predict the testing data. The confusion matrices of the average prediction results and their standard deviation on all three datasets are shown in table 3.7. These values and all confusion matrix values in the following were rounded to integers.

It could be observed that the prediction accuracy for dataset B was the highest and

TABLE 3.7: Prediction performance of the hypertuned models on the testing data of each dataset. Listed are the rounded average values and their standard deviation, both rounded to integers.

Dataset A	**Label -**	**Label +**
Predicted -	505±16	115±11
Predicted +	513±19	1528±28

Dataset B	**Label -**	**Label +**
Predicted -	113±5	9±4
Predicted +	63±6	494±10

Dataset C	**Label -**	**Label +**
Predicted -	161±6	43±7
Predicted +	36±7	61±6

the prediction accuracy for dataset C was the lowest on average. The relevant performance metrics, as described in section 2.2.3.2, were calculated from the confusion matrices, rounded to two decimal places. They are listed in table 3.9 and are visualized for easier comparison in figure 3.27.

TABLE 3.8: Average performance metrics of the hypertuned models of the three datasets. Listed are the average values of the rates and their standard deviation, both rounded to two decimal places. These values signify how accurate the predictions are overall(Acc), how well negative and positive images are recognized(FNR, FPR) and how accurate a positive or negative prediction is(FPR, FNR).

	Dataset A	**Dataset B**	**Dataset C**
Acc	0.76±0.01	0.89±0.01	0.74±0.03
FPR	0.5±0.01	0.36±0.03	0.18±0.03
FNR	0.07±0.01	0.02±0.01	0.41±0.05
NPV	0.81±0.02	0.93±0.03	0.79±0.03
PPV	0.75±0.01	0.89±0.01	0.63± 0.06

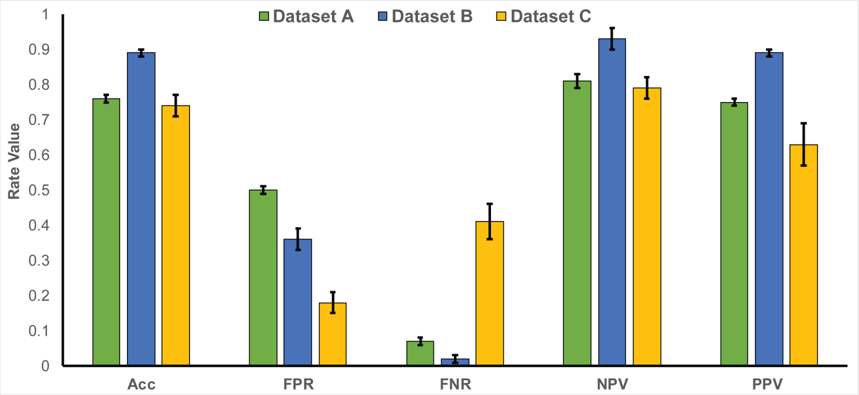


FIGURE 3.27: Performance metric comparison for the SVM models trained on the three datasets. Shown are the mean values of the accuracy rate Acc, the false negative rate FNR, the false positive rate FPR, and the positive and negative predictive values, PPV and NPV, each on a scale of 0 to 1. Also shown are the determined standard deviations from the 10 independent runs as error bars.

Due to the main objective of the MQC being the elimination of bad micrographs that negatively influence the subsequent data processing workflow, the most relevant performance metrics are, apart from the overall accuracy Acc, the false negative rate FNR that signifies the share of bad images not recognized as bad, where the desired value would be zero, and specially the positive predictive value PPV, where the desired value would be 1, that describes the share of true positive images in the images that were predicted as good. The highest Acc, PPV and NPV values on average were observed for dataset B, while dataset C showcased the lowest FPR, which means a larger share of bad images predicted as good, despite having by far the highest FNR, thus predicting too many usable images as bad. Dataset A and B both show a higher FPR than FNR. This can be explained by the larger share of positively judged images in the datasets as opposed to dataset C, which makes a misclassification of the minority class more likely.

Overall, all models predicted their assigned testing sets with a higher accuracy than pure guessing would. Also, the PPV surpassed the share of good images for every dataset, which would therefore result in an output data set with a better good to bad image ratio than the input data. It also can be noted that the standard deviation of every described metric was very low, which implies that with the given amount of images from the datasets, a singular 70-30 split might be sufficient to train and tune a model on the training set to accurately predict the test data set.

3.2.4.4 Learning Curves

Carefully interpreting the acquired SVM model performance as a success, learning curves were recorded for the 10 hypertuned models of all three datasets to be able to investigate how many images were needed to reach the peak prediction performance and to analyze the amount of overfitting that occurs. For this, the same training and test datasets from the previous section were utilized.

In steps of 5% of the training set size, a growing amount of random images were drawn from the training set and subsequently used to train a SVM model, using the optimal hyperparameters determined previously. With this model, the test data and the training data itself were predicted and the relevant metrics $Accuracy$, PPV and FPR for both were determined. For every amount of images, the drawing and prediction procedure was repeated 5 times to minimize the influence of random drawing on the results, with the average value for all three metrics being recorded. The three metrics were then again averaged over the 10 independent runs to minimize the influence of the splitting into training and test data. The averaged learning curves for all three datasets are shown in figure 3.28.

For all three datasets, a variance convergence between testing data and training data could be observed for all metrics. Dataset C showed the biggest variance at maximum training data size, which points towards effects of overfitting the training data. This can be explained with the smaller number of metrics that the model is trained on. For this dataset, the number of micrographs needed for variance convergence was estimated to be around 600.

Dataset A showed a small variance for all three performance metrics after convergence at around 2500 images, pointing to little overfitting and rather a performance limit of the model itself. Dataset B converged to its very high accuracy and PPV after around 750 images with very little visible overfitting effects.

3.2.4.5 Computational Complexity of the MQC Workflow

The MQC workflow as presented involves two different computational steps: Data acquisition and SVM model training and tuning. The computational cost of these is

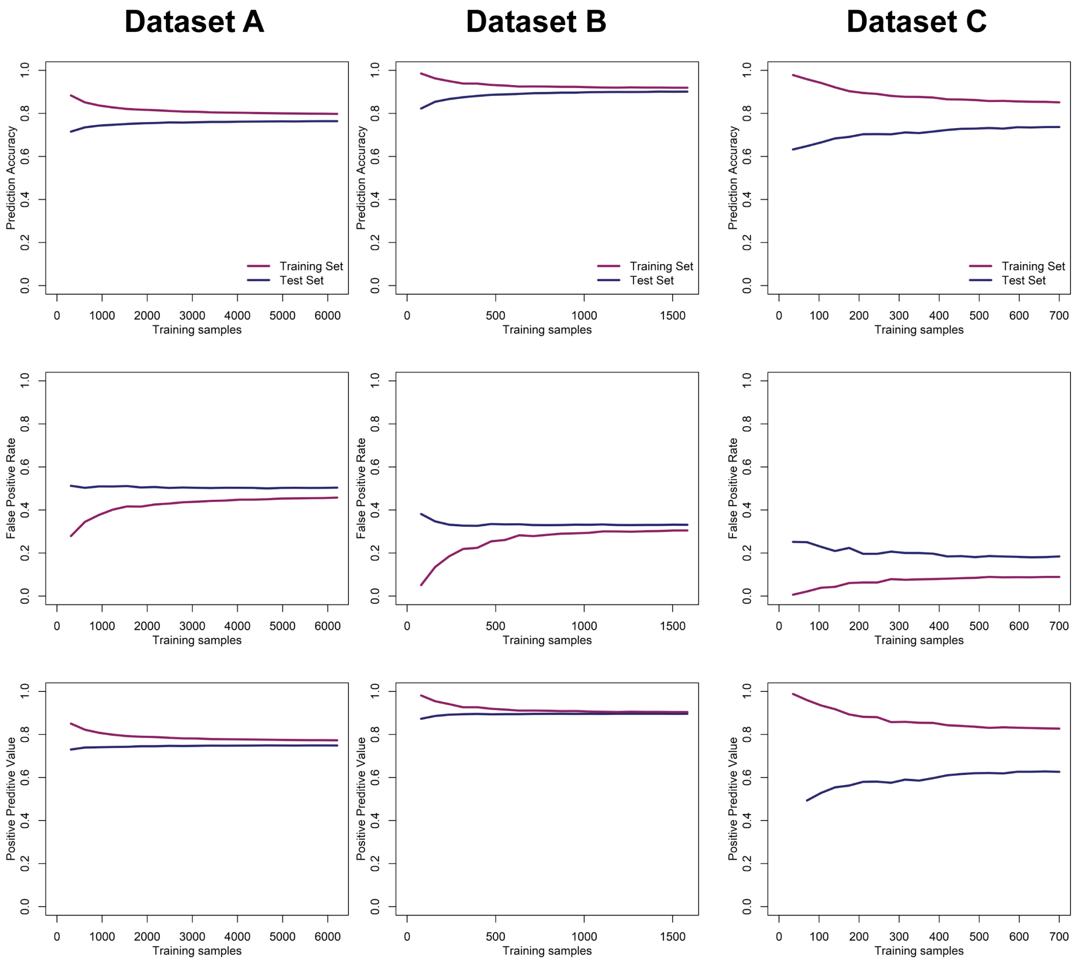


FIGURE 3.28: Learning curves of the SVM models. Shown are learning curves for *Accuracy*, *PPV* and *FNR*, which are plotted on a scale of 0 to 1 that were acquired for a growing training data set size with testing being done on a fixed test set. Acquired metric values were averaged over 5 random drawings of the amount of images from the training set and again averaged over the 10 independent runs.

shown in figure 3.29. Image processing, including data presentation to the user, was not found to be linear to the amount of CPU cores used, probably due to the non-threaded operations that are performed in the Software as well as mutexed result writing operations. The data analysis computations that were performed manually from the acquired data were an order of magnitude less computationally expensive than the preceding data acquisition, highlighting the performance benefits of using a SVM approach. In a future C++ implementation, even more performance enhancements are possible through efficient parallelization and GPU computing utilization.

The last part of the MQC workflow, acquisition speed of labeled training data through user decisions, could naturally not be quantified in a testing environment as easily as the pure computational aspects, but in the presented datasets, the average decision time per micrograph was around 2 seconds. Overall, the efficiency of the machine learning computations enable background live training possibilities for the software.

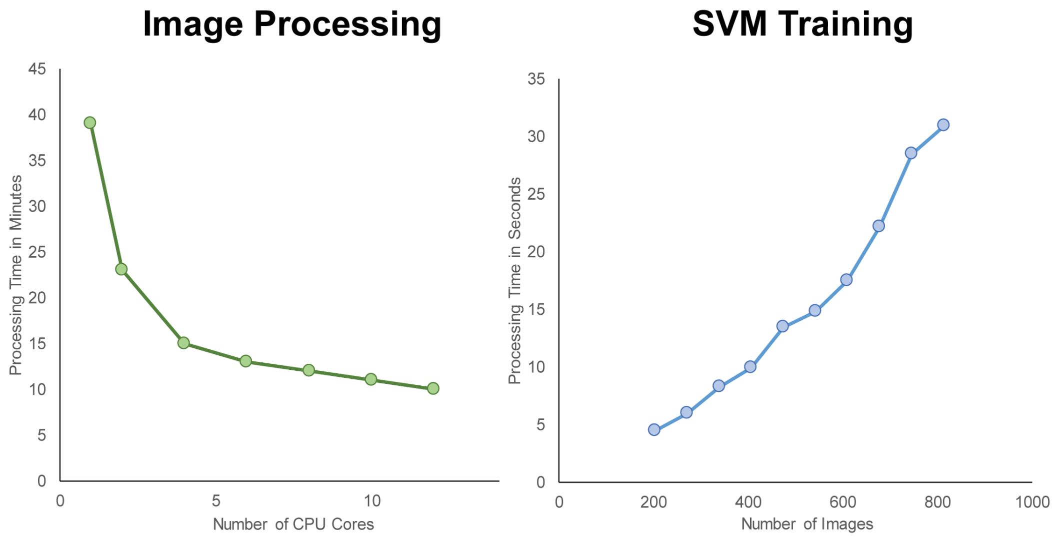


FIGURE 3.29: Computational speed of the two parts of the MQC data processing. Shown on the left is the time needed for the software to process a dataset of 662 3878 pixel x 3710 pixel MRC images on the Dell Workstation depending on the number of cores. Processing included data presentation to the user and calculation of the statistical metrics. Shown on the right is the amount of time needed for the processing algorithm shown in figure 3.26 depending on the number of images on one CPU core, as no multicore implementation was available in the R interface. The model was trained and tuned on 10 metrics in the same hyperparameter search space as the models in the previous section.

3.2.4.6 Generalization Capabilities

One of the main questions for the MQC project was the generalized applicability of models trained on a dataset, e.g. whether the SVMs trained in the previous section can recognize good and bad images in general. To investigate this, three models, each trained on one dataset, were used to predict the full datasets they were not trained on. As dataset C did not have the CTF parameters that were included in the models trained on datasets A and B, dataset C could not be predicted by the other models. All other combinations of model and dataset were computed. The results are shown in table 3.9. The prediction results could safely be called abysmal. In all four combinations, almost all images were predicted as bad, with the questionable achievement of none of the few images predicted as good being true positives. Therefore, no manifold repetitions were performed and no performance metrics were calculated.

TABLE 3.9: Prediction performance of SVM models trained on one dataset on the other datasets. In every table, it is denoted which dataset was predicted by a model trained on which other dataset.

A by B	**Label -**	**Label +**
Predicted -	3373	5494
Predicted +	3	0

A by C	**Label -**	**Label +**
Predicted -	3375	5494
Predicted +	1	0

B by A	**Label -**	**Label +**
Predicted -	579	1681
Predicted +	2	0

B by C	**Label -**	**Label +**
Predicted -	565	1681
Predicted +	16	0

Next, it was investigated whether model training on dataset combinations could improve the capability of the MQC to improve the quality of its predictions, meaning as little bad images as possible being classified as good while recognizing true positives. Dataset C was left out due to the aforementioned lack of CTF parameters. First, datasets A and B were joined in full and the full dataset preprocessing, hyperparameter tuning and performance evaluation procedure was executed with creation of new testing data, ensuring images from both datasets being present in it. Again, this procedure was repeated 10 times to minimize the randomness in the dataset splitting. Performance evaluation was done on images from the two datasets separately with the confusion matrices being shown in table 3.10. Due to the different size of the two datasets, dataset B was also joined with a reduced dataset A of the same size as dataset B. Again, the full dataset preprocessing, hyperparameter tuning and performance evaluation procedure was executed in 10 independent runs. The prediction results of this approach are shown in table 3.11.

TABLE 3.10: Prediction performance of the models trained on joined dataset on the test data split by dataset Origin. Listed are the average values and their standard deviation, both rounded to integers.

Dataset A	**Label -**	**Label +**
Predicted -	437±23	109±20
Predicted +	574±30	1548±25

Dataset B	**Label -**	**Label +**
Predicted -	104±7	12±3
Predicted +	67±7	490±14

TABLE 3.11: Prediction performance of the models trained on the equal number joined dataset on the test data split by dataset origin. Listed are the average values and their standard deviation, both rounded to integers.

Dataset A	**Label -**	**Label +**
Predicted -	107±10	24±7
Predicted +	158±8	385±16

Dataset B	**Label -**	**Label +**
Predicted -	101±12	7±3
Predicted +	74±9	502±15

The previously used performance metrics were calculated for the two joined models and are shown in table 3.12. To compare the joined models' prediction capabilities to the separately trained ones, the performance metrics on the datasets from all three model sources are compared in figure 3.30.

TABLE 3.12: Prediction performance metrics of the hypertuned models of the joined dataset on the datasets split by origin. Listed are the average values of the rates and their standard deviation rounded to two decimal places. These values signify how accurate the predictions are overall(Acc), how well negative and positive images are recognized(FNR and FPR) and how accurate a positive or negative prediction is(FPR, FNR).

	Joined Model on A	**Joined Equal Model on A**	**Joined Model on B**	**Joined Equal Model on B**
Acc	0.74±0.01	0.73±0.01	0.88±0.01	0.88±0.02
FPR	0.57±0.02	0.6±0.03	0.4±0.03	0.42±0.04
FNR	0.07±0.01	0.06±0.02	0.03±0.01	0.01±0.01
NPV	0.8±0.03	0.82±0.03	0.89±0.02	0.94±0.03
PPV	0.73±0.01	0.71±0.01	0.88±0.01	0.87±0.02

For dataset A, the SVM model trained only on the dataset itself showed slightly better or equal performance in all measured metrics, but most deviations were close to the range of the standard deviation. While the model trained on the dataset that was joined in equal numbers had a smaller overall size than dataset A itself, the size

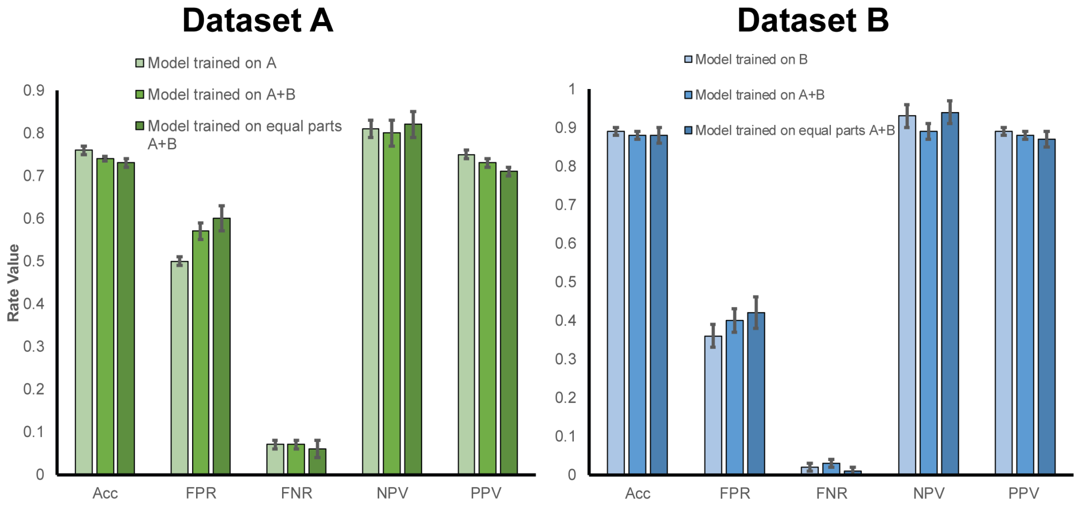


FIGURE 3.30: Performance metric comparison for the differently trained SVM models on the two datasets. Shown are the mean values of the accuracy rate Acc, the false negative rate FNR, the false positive rate FPR, and the positive and negative predictive values, PPV and NPV, each on a scale of 0 to 1 for the three model sources. Also shown are the determined standard deviations from the 10 independent runs as error bars.

was still above the convergence number seen in figure 3.28. The model trained on the whole joined dataset obviously had a larger sample size. In both cases, no clear improvement in the quality of the positive classifications could be observed from a dataset combination. The false positive rate FPR increased for the models containing images from Dataset B, which means that images that were labeled as bad were less likely to be recognized as such, which decreases the quality of a classified output dataset.

Dataset B showed similar tendencies, with the the most important measures PPV, the accuracy of positive predictions, and FPR, the misclassification rate for images labeled as bad, being within the error range from one another for the different model sources. These results indicate that the generalization capabilities of the presented SVM approach are limited, at least for the analyzed datasets, as despite the larger sample size, no improvement in the relevant values could be observed, but rather a trend towards worse performance, which would ultimately result in datasets containing more images that were supposed to be thrown out. Also, pure pre-trained models applied on different datasets performed way worse than even pure guessing, rendering the generalization approach questionable.

These findings seem to confirm the initial hypothesis that due to the noisy micrographs that have specific properties due to the contained samples, the varying imaging conditions, different detectors and many more experimental factors, individual live training for every dataset is preferable, specially taking into account the different quality thresholds explicitly applied by raters of the datasets.

Chapter 4

Discussion

The cryo-EM field has experienced remarkable growth in the last years [27], with more and more 3D structures being determined at resolutions that allow interpretations in atomic detail. Nevertheless, a large share of published structures never reach those resolutions, specially structures of protein complexes that are not well understood and characterized or not as highly symmetrical as for example virus particles. The results of this thesis offer improvements for two of the factors that limit obtainable resolution: Sample preparation and elimination of low quality TEM images.

Two new softwares were developed to tackle these problems that can limit resolution and biological interpretability of structures determined by cryo-EM were introduced: CowGraCE, a software containing new algorithmic approaches to optimize rate-zonal centrifugation experiments and the Micrograph Quality Checker, a preprocessing tool that can utilize machine learning to classify micrographs in parallel to data recording to improve dataset quality. Their results and implications are discussed in the following and put it into a broader context.

4.1 CowGraCE

In the CowGraCE project, a light-weight standalone software that aids users in optimizing their experimental conditions for density gradient centrifugation in swing-out rotors was successfully implemented and tested. In the following, software architecture, speed and simulation results are discussed.

4.1.1 Implementation and Simulation Speed

The software that can be used on MacOS, Windows and Linux was designed to be fast, simple and convenient through a linear workflow and extendability in terms of rotor choices. The speed of a single simulation was shown to be in the μs range(see section 3.1.4), enabling the brute-force search in optimization mode which therefore has a runtime in the low minutes range, even in an unrestricted parameter space.

4.1.2 Simulation Results and Optimization Capatibilities

Based on an adapted version of the indirect approach to simulation of rate-zonal centrifugation experiments[178], the simulated results of the CowGraCE algorithm compared to experimental data very favorably and showed a very good approximation of peak fraction values for samples with either known sedimentation coefficients or only the molecular weight(see 3.1.6). In 9 out of 10 tests, the prediction surpassed the results from COMPASS [177], which is the only other available software with this simulation functionality.

To estimate the particle distribution in an osmolyte gradient of known composition, a new approach for predicting standard deviations of the assumed normal distributions was developed and tested (see section 3.1.2). As opposed to previously published approaches[178, 156], no prior information about diffusive behavior or other properties of the protein complex of interest is needed in the new CowGraCE method. Accurate distribution prediction in absence of specific physicochemical information about the protein sample proved to be a challenging task, as comparisons to experimental data showed larger deviations than for the peak S values, but an improvement compared to COMPASS results could still be observed (see section 3.1.6.3). Future improvements to the algorithm will focus on the distribution prediction for complex samples, where known distribution broadening factors such as overall protein concentration can be taken into account to estimate particle distributions more closely.

A reduced-space brute-force optimization algorithm was successfully developed that can aid in finding centrifugation conditions that result in removal of sample contaminations through user-set thresholds and an optimization score that judges the simulations based on the reliable peak value fraction positions and in small parts through predicted peak widths(see section 3.1.7.5). Simulations found a large number of conditions that predicted a protein sample within the optimal fraction range of the tube. Through separation optimization simulations, it was found that the optimal optimization score can be found even in a small parameter space and for multiple parameter combinations, which is beneficial to a user who for example wants to minimize osmolyte concentration or runtime (see 3.1.7.8).

Chromatography-free purification through density gradient centrifugation is steadily gaining traction, as shown by recent biological successes [163, 17, 16] and integration into methodological developments such as GraDeR[77] and GraFix[101]. Overall, more accurate rate-zonal centrifugation simulations by CowGraCE have the potential to reduce wasting valuable experimentation time and sample material and help users achieve purer and more stable samples for various methods in structural biology.

4.2 Micrograph Quality Checker and COW

The MQC that was developed for this thesis covers an aspect of the single particle cryo-EM data processing that had been neglected until now by tapping into the potential of live processing and into the benefits of eliminating suboptimal TEM images in parallel to their acquisition. In the following, the implementation, speed, classification results and the integration into the COW are discussed.

4.2.1 Implementation, Processing Workflow and Speed

The implementation of the Micrograph Quality Checker enables live preprocessing of TEM images together with a proposed machine learning based image classification and metadata acquisition that can theoretically be used in the later stages of image processing. The highly parallelized software, as shown in figure 3.29, processes images reasonably fast and the proposed SVM model training and tuning can be computed on a time scale that is an order of magnitude below the image processing itself.

As soon as the implementation of frame alignment and CTF parameter determination are completed, the whole preprocessing workflow will be covered, which opens up possibilities to improve processing speed due to better combination of the algorithmic steps. For example, frame alignment, CTF parameter determination and parts of the image metric acquisition all take place in Fourier space. Right now, micrographs are transformed into Fourier Space and back multiple times, as well as loaded into memory and written to disk. These steps can be combined to reduce the computational cost.

4.2.2 SVM Model Performance on Chosen Datasets

Despite the low number of metrics, binary classification on the three presented datasets was fairly successful, with prediction accuracy ranging from 74% to almost 90%, despite the amount of images being on the low side of typical dataset sizes(see section 3.2.4.3). The early variance convergence showed the current limitations of the SVM models, but shows the applicability of live training due to the fairly low amount of images needed to reach peak model performance (see section 3.2.4.4). For the second dataset, the prediction accuracy of nearly 90% was very close to the measured user consistency of 91% (see section 3.2.4.1). For the other datasets, no user consistency numbers were available, but it is likely that the current metric space does not cover all possible reasons for accurately modeling the classification behavior of an experienced user. Despite this, all positive predictive values exceeded the input ratios of good to bad images, therefore improving the dataset composition. The observed standard deviations of classification performance were very low, which can be interpreted as little overfitting effects being present and may allow for less independent verification repetitions.

To improve the models in the future, more image metrics are needed, presumably real space metrics, as of now, only the variance of images after filtering with a Boxcar and a DoG filter was used. Also, the lack of being apply to use the decision time as a metric for classification of not-judged micrographs needs to be taken into account. A closer analysis of falsely predicted positive images should be informative in terms of which micrograph contaminations and defects are currently not described well enough by the available metrics. Metrics that need to be integrated into the model are behavior of the frames during frame alignment, for example large deviation of image patches from one another is expected to correlate with a suboptimal image quality. Further patch-based real-space metrics might also help identify contaminations and defects in specific areas of a micrograph.

4.2.3 Generalization of Models

The generalization capabilities of the trained models proved to be virtually nonexistent. Training SVM models on datasets resulted in prediction performance similar to the models trained on a singular dataset, but in all relevant metrics, a slight performance decline was observed despite the larger training data sample size, which means that the subset of the data classified as good by the joined model would be worse than having trained a model on a separate dataset(see 3.2.4.6). Therefore, it can be concluded that the SVM approach of this thesis with a relatively low amount of features does not reflect a general recognition of good and bad images. Due to the statistical metrics employed, the noisy nature of TEM micrographs, the different protein complexes and differences in the imaging and sample preparation process, it remains to be seen whether such a generalization is possible at all. Further analysis should focus on datasets of the same protein complex, imaged under slightly different conditions, i.e. different cameras, defocus conditions and investigate what factors influence model transferability.

4.2.4 Proposed Live Training Implementation

Due to the non-generalizable model performance at this stage of the project, a SVM model needs to be trained in parallel to data acquisition and user judgment. Also, allowing a user to define his own expectation baselines of what constitutes a good image from a dataset is advantageous due to enabling individualized outputs.

As the processing speed measurements showed, live training is computationally feasible. Theoretically, one could obviously build a database of models, have every instance of the MQC connect to it and check whether any of these pre-trained models predict data during the acquisition process well enough, but this approach would be problematic in terms of confidential data and is therefore neglected in favor of live training. The proposed algorithm for this is summarized in figure 4.1.

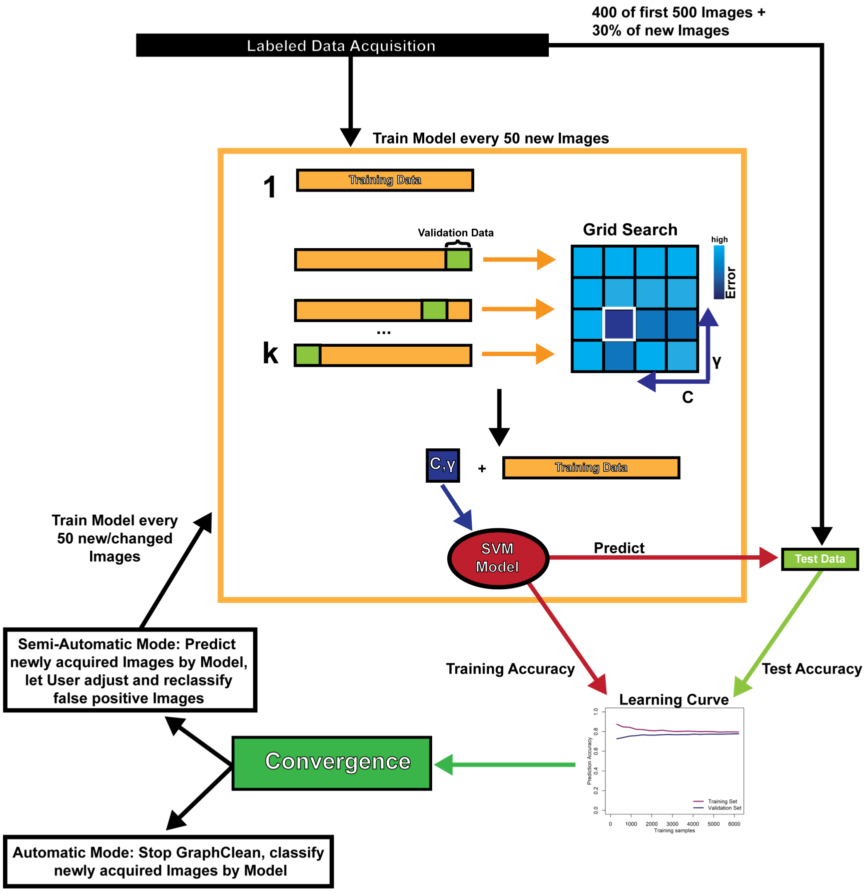


FIGURE 4.1: Proposed Live SVM Model Training Algorithm

Variance convergence that could be seen in the learning curves was observed at different amount of images in the analyze datasets- at around 600, 750 and 2500 images. Due to the testing sets having different sizes, the absolute numbers should not be over interpreted, but no convergence could be observed at less than a third of the dataset used as training size. Due to no information about dataset size being available in case of live processing and different convergence speeds depending on the dataset, heuristically a value of 500 images is proposed for the start of modeling. 400 of the 500 images are then to be set aside as test data, with the remaining images being used as training data. All new arriving images are then continuously added to the training data. Every 50 new Images, the complete model tuning and evaluation algorithm previously described in figure 3.26 will then be computed and tested on the test data. A continuously updated learning curve can then show convergence of the modeling.

Then, two possible operation modes are proposed: An automatic mode where the user input and live training is shut off and all further acquired images are classified by the converged SVM model, and a semi-automatic mode. Here, the idea is to predict newly acquired data and to let the user confirm or reject the prediction. Also, it might be beneficial to let the user re-rate false positive images in case of the need to overturn the previous judgment. This changing of training data might lead to a higher prediction performance of the model.

Due to the self-referential data processing in high-resolution cryo-EM, careful selection of data as early as possible can potentially open doors to a better understanding of biological systems by improving the achievable quality of structures and supporting users in diverse applications. For example, the automatic mode of the live training can be used if the user is satisfied with the model prediction and does not want to invest further time into manual selection. The semi-automatic mode with continuous user input could on the other hand be used to teach the MQC to only specifically accept the very best images, even in absence of image contaminations or defects. For example, once the machine learning model is good enough, all micrographs with a non-desired particle distribution could be rejected. Thinking even further into the future, a distinction between conformational substates of particles on micrographs could theoretically be made possible through accurate model training.

Comparing the amount of particles that are used for published structures over the years, a steady increase can be seen. While only a couple of years ago, the amount of commonly used particles was in the tens of thousands range[66], nowadays datasets routinely consist of more than 500.000 particles[75]. Due to needing more and more particles for high-resolution structures, it is likely that this trend continues into the future. This also necessitates effective automated micrograph processing, and the MQC represents a step forward towards this.

4.2.5 COW Framework and Integration of MQC

The MQC is the first part of the COW framework that is able to do live processing. It represents the interface to data acquisition that starts the subsequent data processing workflow. The current components of the COW and their interactions are shown in figure 4.2. The COW is currently in the closed beta stage and will be released to the

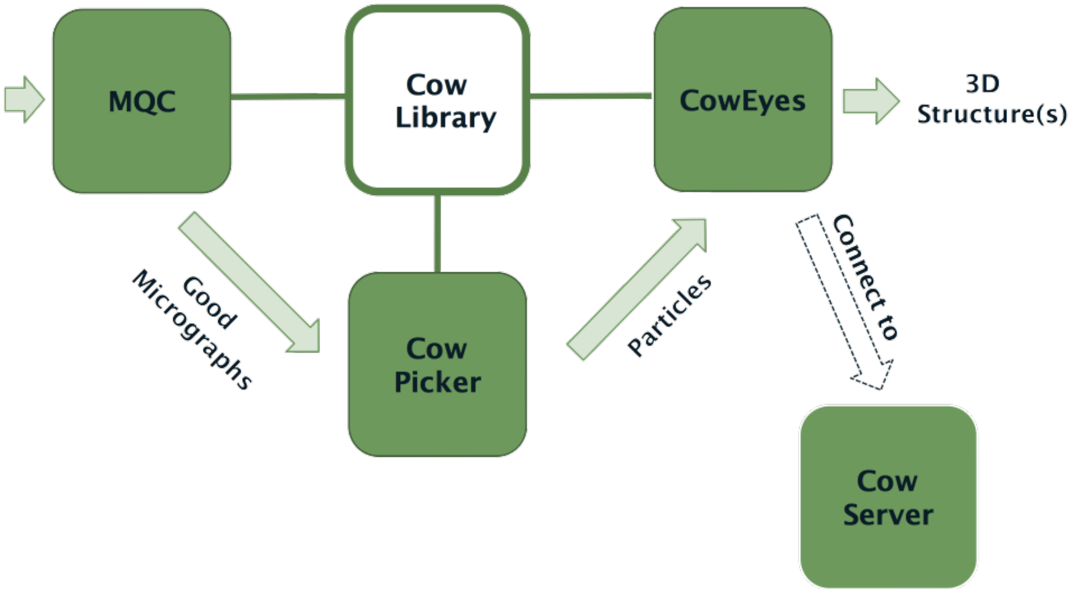


FIGURE 4.2: Structure of the COW. Shown are the front ends that all connect to the CowLib and cover the whole workflow from data recording to final 3D structure(s).

public in the near future.

While currently the MQC and the CowPicker work on normal files and the preferable cow project structure only exists within CowEyes, it is planned to integrate both of these frontends into CowEyes as interactive logics that operate on cow projects. With this, live processing can be brought to all stages of image processing, which enables for example calculating and continuously updating a 3D structure while the microscope is still producing data, as long as the processing steps are computationally fast enough. Through this, feedback loops to earlier processing steps and possibly even to the data acquisition stage will be feasible in the future.

Combined with the usage of modern computing techniques to deal with the ever-increasing amount of data, this opens the door to potentially being able to automate sample screening, to prevent the microscopes from recording suboptimal data over an extended amount of time and to quickly identify the image processing bottlenecks, for which the COW framework gives users all tools to overcome.

4.3 Outlook

The amount of atomic level biological information made available through cryo-EM holds the potential to connect basic research to applied studies. High-resolution structures and their energy landscapes that are changed based of the chemical environment[74] can potentially be used to better understand the mechanism of drugs that target large protein complexes, thereby gaining e.g. pharmacological relevance.

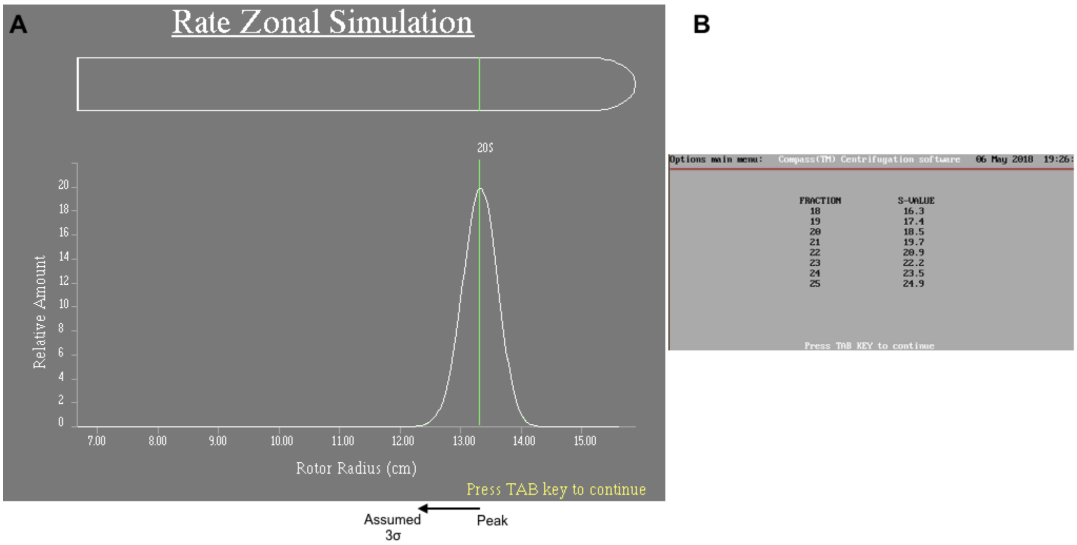
The results of this thesis, through more effective sample purification assisted by CowGraCE and flexible micrograph preprocessing and classification by the MQC, present methodological advancements in the cryo-EM field, which, combined with other technological improvements, can help make further steps for a better understanding of life's inner workings at the lowest level of biology.

Appendix A

Additional Data for CowGraCE

TABLE A.1: Supported sucrose gradients in the BioComp Gradient-Maker. The columns signify the rotor, the rows the concentration at the top of the tube and the values the supported bottom concentrations.

Rotor	2% -	5% -	10% -	15% -	20% -	25% -	26% -	30% -	32% -	60% -
SW28	27%	20%,25%,30%,40%,45%	25%,30%,35%,40%,45%,50%	27%,30%,35%,40%,45%	50%,60%,70%,	40%,60%	65%	-	60%	80%
SW40	-	20%,25%,29%,30%,40%,45%	25%,30%,40%,45%	30%,40%,45%,60%	50%	50	-	60%	-	-
SW60	-	20%,25%,30%,40%,45%,50%	25%,30%,40%,45%	30%,40%,45%	-	-	-	-	-	-



```
Options main menu:   Compass(TM) Centrifugation software   06 May 2018  19:26:

          FRACTION           S-VALUE
             18                16.3
             19                17.4
             20                18.5
             21                19.7
             22                20.9
             23                22.2
             24                23.5
             25                24.9

                  Press TAB KEY to continue
```

FIGURE A.1: Determination of mean and standard deviation from COMPASS. Shown is the available output from the software. A shows the simulated distribution from which the standard deviation was extracted, B shows the tabular results from which the mean position was interpolated.

TABLE A.2: Supported Glycerol gradients in the BioComp Gradient-Maker. The columns signify the rotor, the rows the concentration at the top of the tube and the values the supported bottom concentrations.

Rotor	5% -	10% -	15% -	30% -	45% -
SW28 +SW60	20%,30%	30%,40%,50%,,60%,80%	35%	70%	80%
SW40	20%,30%,40%	30%,40%,50%,60%	40%	-	-

TABLE A.3: Predicted and experimentally determined S Values

Protein	**Molecular Weight**	**Experimental S**	**Predicted S (rounded)**
Proteasome	700 kDa	20	20
α-Amylase	97.6 kDa	5.2-8.6	5
Alcohol Dehydrogenase	150 kDa	6.7-8.6	7
β-Amylase	152 kDa	8.9-9.2	7
Glucoronidase	390 kDa	14-15.5	13
Apoferritin	441 kDa	16.3-17.6	15
Thyroglobulin	669 kDa	19.4-20.9	19
Ribosome	2.5 MDa	70	47

Appendix B

Additional Data for the MQC

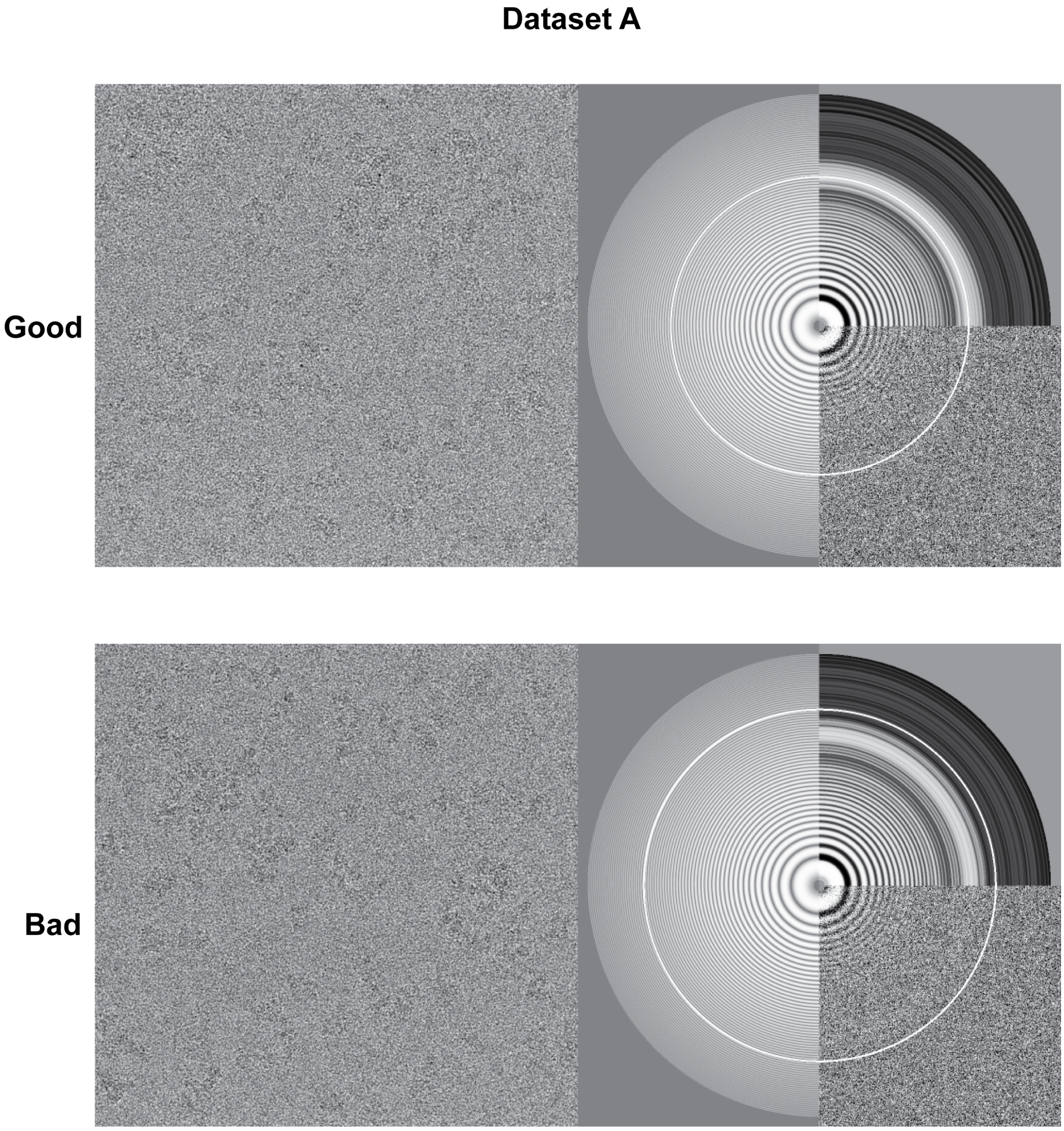


FIGURE B.1: Exemplary Good and Bad Micrographs and their Power Spectrum with Ctf Fit from Dataset A

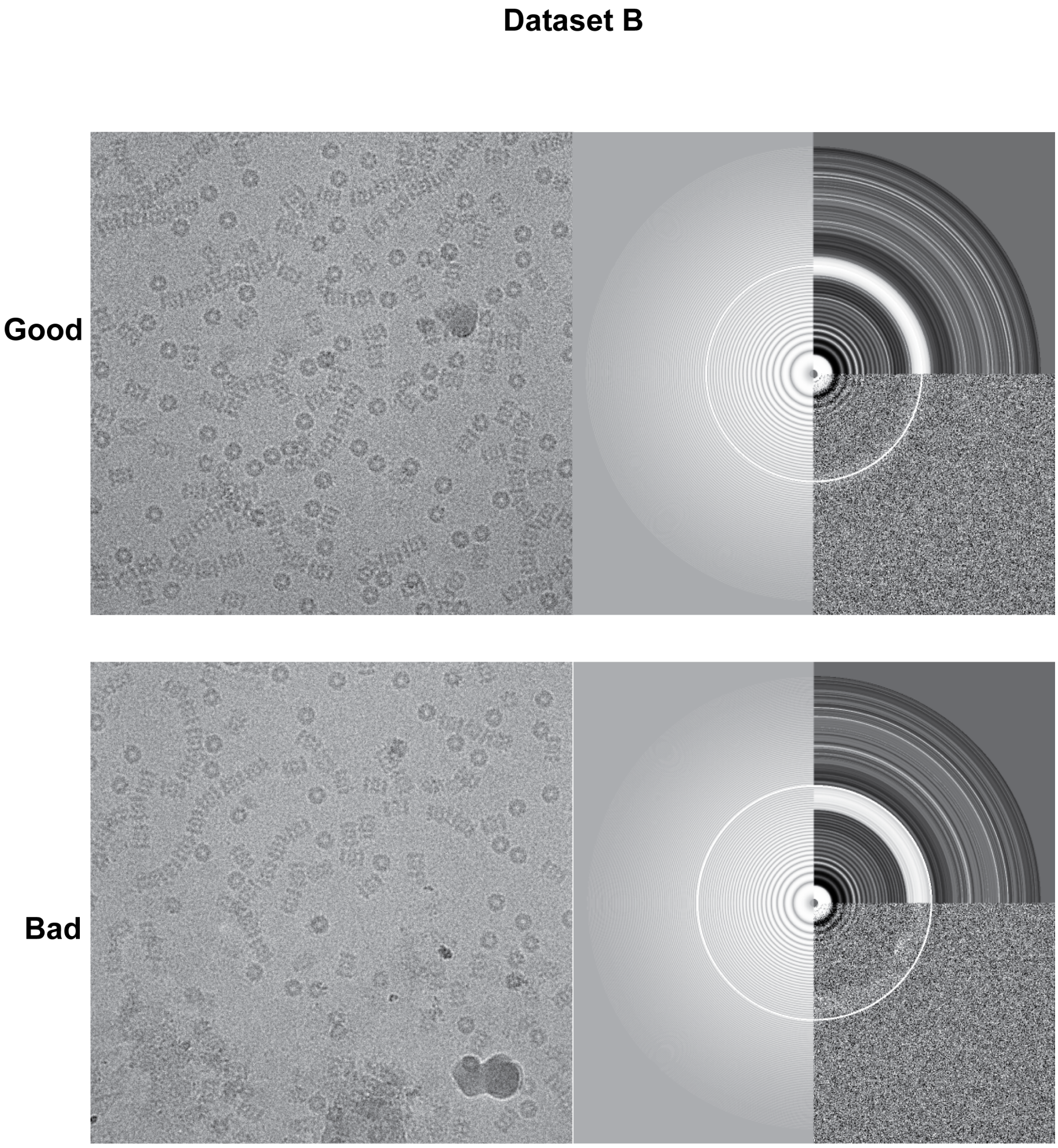


FIGURE B.2: Exemplary Good and Bad Micrographs and their Power Spectrum with Ctf Fit from Dataset B

TABLE B.1: Correlation matrix 1 from dataset A. The metrics that were eliminated are shown in bold.

	boxDogVariance	defocusU	deviationsFromMean	imageMean	imageVariance	**maxLineVariance**	**meanOfSumOfVariances**	**sumOfVariances**	varianceOfSumOfVariances	resLimit	decisionTimee	defocusDiff
boxDogVariance	1.00	-0.15	-0.00	-0.15	0.10	0.02	0.05	0.05	-0.02	0.08	0.00	0.00
defocusU	-0.15	1.00	-0.28	0.64	0.02	0.08	0.03	0.03	0.03	-0.01	-0.01	0.04
deviationsFromMean	-0.00	-0.28	1.00	-0.27	-0.13	-0.29	-0.21	-0.21	-0.06	-0.04	0.01	-0.22
imageMean	-0.15	0.64	-0.27	1.00	-0.31	0.16	0.15	0.15	0.06	0.02	-0.01	0.10
imageVariance	0.10	0.02	-0.13	-0.31	1.00	0.24	0.29	0.29	0.15	0.17	-0.00	0.03
maxLineVariance	0.02	0.08	-0.29	0.16	0.24	1.00	0.97	0.97	0.90	0.24	-0.00	0.26
meanOfSumOfVariances	0.05	0.03	-0.21	0.15	0.29	0.97	1.00	1.00	0.87	0.26	-0.01	0.21
sumOfVariances	0.05	0.03	-0.21	0.15	0.29	0.97	1.00	1.00	0.87	0.26	-0.01	0.21
varianceOfSumOfVariances	-0.02	0.03	-0.06	0.06	0.15	0.90	0.87	0.87	1.00	0.19	-0.00	0.15
resLimit	0.08	-0.01	-0.04	0.02	0.17	0.24	0.26	0.26	0.19	1.00	0.00	0.24
decisionTime	0.00	-0.01	0.01	-0.01	-0.00	-0.00	-0.01	-0.01	-0.00	0.00	1.00	0.00
defocusDiff	0.00	0.04	-0.22	0.10	0.03	0.26	0.21	0.21	0.15	0.24	0.00	1.00

TABLE B.2: Correlation matrix 2 from dataset A. The metrics that were eliminated are shown in bold.

	boxDogVariance	defocusU	deviationsFromMean	imageMean	imageVariance	varianceOfSumOfVariances	resLimit	decisionTime	defocusDiff	logMaxLineVariance	**logMeanOfSumOfVariances**	**logSumOfVariances**
boxDogVariance	1.00	-0.15	-0.00	-0.15	0.10	-0.02	0.08	0.00	0.00	0.07	0.13	0.13
defocusU	-0.15	1.00	-0.28	0.64	0.02	0.03	-0.01	-0.01	0.04	0.17	0.08	0.08
deviationsFromMean	-0.00	-0.28	1.00	-0.27	-0.13	-0.06	-0.04	0.01	-0.22	-0.56	-0.35	-0.35
imageMean	-0.15	0.64	-0.27	1.00	-0.31	0.06	0.02	-0.01	0.10	0.24	0.22	0.22
imageVariance	0.10	0.02	-0.13	-0.31	1.00	0.15	0.17	-0.00	0.03	0.33	0.42	0.42
varianceOfSumOfVariances	-0.02	0.03	-0.06	0.06	0.15	1.00	0.19	-0.00	0.15	0.46	0.46	0.46
resLimit	0.08	-0.01	-0.04	0.02	0.17	0.19	1.00	0.00	0.24	0.19	0.23	0.23
decisionTime	0.00	-0.01	0.01	-0.01	-0.00	-0.00	0.00	1.00	0.00	-0.01	-0.01	-0.01
defocusDiff	0.00	0.04	-0.22	0.10	0.03	0.15	0.24	0.00	1.00	0.40	0.27	0.27
logMaxLineVariance	0.07	0.17	-0.56	0.24	0.33	0.46	0.19	-0.01	0.40	1.00	0.90	0.90
logMeanOfSumOfVariances	0.13	0.08	-0.35	0.22	0.42	0.46	0.23	-0.01	0.27	0.90	1.00	1.00
logSumOfVariances	0.13	0.08	-0.35	0.22	0.42	0.46	0.23	-0.01	0.27	0.90	1.00	1.00

TABLE B.3: Correlation matrix 1 from dataset B. The metrics that were eliminated are shown in bold.

	boxDogVariance	defocusU	deviationsFromMean	imageMean	imageVariance	**maxLineVariance**	**meanOfSumOfVariances**	**sumOfVariances**	varianceOfSumOfVariances	resLimit	decisionTime	defocusDiff
boxDogVariance	1.00	0.05	0.13	0.02	0.03	0.36	0.50	0.50	0.57	0.07	0.03	0.15
defocusU	0.05	1.00	-0.31	0.22	-0.13	0.40	0.20	0.20	0.24	0.22	-0.03	0.37
deviationsFromMean	0.13	-0.31	1.00	-0.08	0.15	-0.31	0.13	0.13	0.01	0.04	0.00	0.03
imageMean	0.02	0.22	-0.08	1.00	-0.38	-0.31	-0.31	-0.31	-0.02	0.33	-0.06	0.16
imageVariance	0.03	-0.13	0.15	-0.38	1.00	0.24	0.44	0.44	0.19	0.33	0.01	0.33
maxLineVariance	0.36	0.40	-0.31	-0.31	0.24	1.00	0.80	0.80	0.80	0.08	0.02	0.25
meanOfSumOfVariances	0.50	0.20	0.13	-0.31	0.44	0.80	1.00	1.00	0.85	0.17	0.02	0.31
sumOfVariances	0.50	0.20	0.13	-0.31	0.44	0.80	1.00	1.00	0.85	0.17	0.02	0.31
varianceOfSumOfVariances	0.57	0.24	0.01	-0.02	0.19	0.80	0.85	0.85	1.00	0.18	-0.00	0.27
resLimit	0.07	0.22	0.04	0.33	0.33	0.08	0.17	0.17	0.18	1.00	-0.07	0.57
decisionTime	0.03	-0.03	0.00	-0.06	0.01	0.02	0.02	0.02	-0.00	-0.07	1.00	-0.03
defocusDiff	0.15	0.37	0.03	0.16	0.33	0.25	0.31	0.31	0.27	0.57	-0.03	1.00

TABLE B.4: Correlation matrix 2 from dataset B. The metrics that were eliminated are shown in bold.

	boxDogVariance	defocusU	deviationsFromMean	imageMean	imageVariance	varianceOfSumOfVariances	resLimit	decisionTime	defocusDiff	logMaxLineVariance	**logMeanOfSumOfVariances**	**logSumOfVariances**
boxDogVariance	1.00	0.05	0.13	0.02	0.03	0.57	0.07	0.03	0.15	0.20	0.26	0.26
defocusU	0.05	1.00	-0.31	0.22	-0.13	0.24	0.22	-0.03	0.37	0.25	0.08	0.08
deviationsFromMean	0.13	-0.31	1.00	-0.08	0.15	0.01	0.04	0.00	0.03	-0.34	0.09	0.09
imageMean	0.02	0.22	-0.08	1.00	-0.38	-0.02	0.33	-0.06	0.16	-0.64	-0.76	-0.76
imageVariance	0.03	-0.13	0.15	-0.38	1.00	0.19	0.33	0.01	0.33	0.29	0.48	0.48
varianceOfSumOfVariances	0.57	0.24	0.01	-0.02	0.19	1.00	0.18	-0.00	0.27	0.47	0.48	0.48
resLimit	0.07	0.22	0.04	0.33	0.33	0.18	1.00	-0.07	0.57	-0.12	-0.07	-0.07
decisionTime	0.03	-0.03	0.00	-0.06	0.01	-0.00	-0.07	1.00	-0.03	0.04	0.05	0.05
defocusDiff	0.15	0.37	0.03	0.16	0.33	0.27	0.57	-0.03	1.00	0.08	0.12	0.12
logMaxLineVariance	0.20	0.25	-0.34	-0.64	0.29	0.47	-0.12	0.04	0.08	1.00	0.89	0.89
logMeanOfSumOfVariances	0.26	0.08	0.09	-0.76	0.48	0.48	-0.07	0.05	0.12	0.89	1.00	1.00
logSumOfVariances	0.26	0.08	0.09	-0.76	0.48	0.48	-0.07	0.05	0.12	0.89	1.00	1.00

TABLE B.5: Correlation matrix 1 from dataset C. The metrics that were eliminated are shown in bold.

	boxDogVariance	deviationsFromMean	imageMean	**imageVariance**	**maxLineVariance**	**meanOfSumOfVariances**	**sumOfVariances**	varianceOfSumOfVariances	decisionTime
boxDogVariance	1.00	-0.16	-0.23	0.44	0.30	0.36	0.36	0.19	-0.01
deviationsFromMean	-0.16	1.00	-0.10	-0.24	-0.58	-0.53	-0.53	-0.40	0.03
imageMean	-0.23	-0.10	1.00	-0.78	-0.11	-0.18	-0.18	-0.06	0.00
imageVariance	0.44	-0.24	-0.78	1.00	0.56	0.64	0.64	0.43	-0.01
maxLineVariance	0.30	-0.58	-0.11	0.56	1.00	0.98	0.98	0.93	-0.01
meanOfSumOfVariances	0.36	-0.53	-0.18	0.64	0.98	1.00	1.00	0.89	-0.02
sumOfVariances	0.36	-0.53	-0.18	0.64	0.98	1.00	1.00	0.89	-0.02
varianceOfSumOfVariances	0.19	-0.40	-0.06	0.43	0.93	0.89	0.89	1.00	-0.01
decisionTime	-0.01	0.03	0.00	-0.01	-0.01	-0.02	-0.02	-0.01	1.00

TABLE B.6: Correlation matrix 2 from dataset C. The metrics that were eliminated are shown in bold.

	boxDogVariance	deviationsFromMean	imageMean	varianceOfSumOfVariances	decisionTime	logMaxLineVariance	**logMeanOfSumOfVariances**	**logSumOfVariances**	**logImageVariance**
boxDogVariance	1.00	-0.16	-0.23	0.19	-0.01	0.39	0.50	0.50	-0.35
deviationsFromMean	-0.16	1.00	-0.10	-0.40	0.03	-0.71	-0.56	-0.56	0.14
imageMean	-0.23	-0.10	1.00	-0.06	0.00	-0.17	-0.30	-0.30	0.91
varianceOfSumOfVariances	0.19	-0.40	-0.06	1.00	-0.01	0.67	0.64	0.64	-0.36
decisionTime	-0.01	0.03	0.00	-0.01	1.00	-0.03	-0.04	-0.04	0.01
logMaxLineVariance	0.39	-0.71	-0.17	0.67	-0.03	1.00	0.95	0.95	-0.51
logMeanOfSumOfVariances	0.50	-0.56	-0.30	0.64	-0.04	0.95	1.00	1.00	-0.63
logSumOfVariances	0.50	-0.56	-0.30	0.64	-0.04	0.95	1.00	1.00	-0.63
logImageVariance	-0.35	0.14	0.91	-0.36	0.01	-0.51	-0.63	-0.63	1.00

TABLE B.7: Determined Optimal Hyperparameters for the Datasets in the different independent tuning runs

	Dataset A	Dataset B	Dataset C
Run 1	$\gamma = 0.01, C = 10000$	$\gamma = 0.1, C = 10$	$\gamma = 1, C = 100$
Run 2	$\gamma = 0.1, C = 10$	$\gamma = 0.01, C = 1000$	$\gamma = 0.1, C = 10000$
Run 3	$\gamma = 0.1, C = 10$	$\gamma = 0.1, C = 10$	$\gamma = 1, C = 10$
Run 4	$\gamma = 0.1, C = 10$	$\gamma = 0.01, C = 1000$	$\gamma = 1, C = 10$
Run 5	$\gamma = 0.1, C = 10$	$\gamma = 0.1, C = 10$	$\gamma = 0.1, C = 1000$
Run 6	$\gamma = 0.1, C = 10$	$\gamma = 0.1, C = 100$	$\gamma = 1, C = 10$
Run 7	$\gamma = 0.1, C = 10$	$\gamma = 0.1, C = 100$	$\gamma = 0.1, C = 1000$
Run 8	$\gamma = 0.01, C = 10000$	$\gamma = 0.01, C = 10000$	$\gamma = 1, C = 100$
Run 9	$\gamma = 0.1, C = 10$	$\gamma = 11, C = 100$	$\gamma = 1, C = 100$
Run 10	$\gamma = 0.1, C = 10$	$\gamma = 0.01, C = 10000$	$\gamma = 10, C = 1$

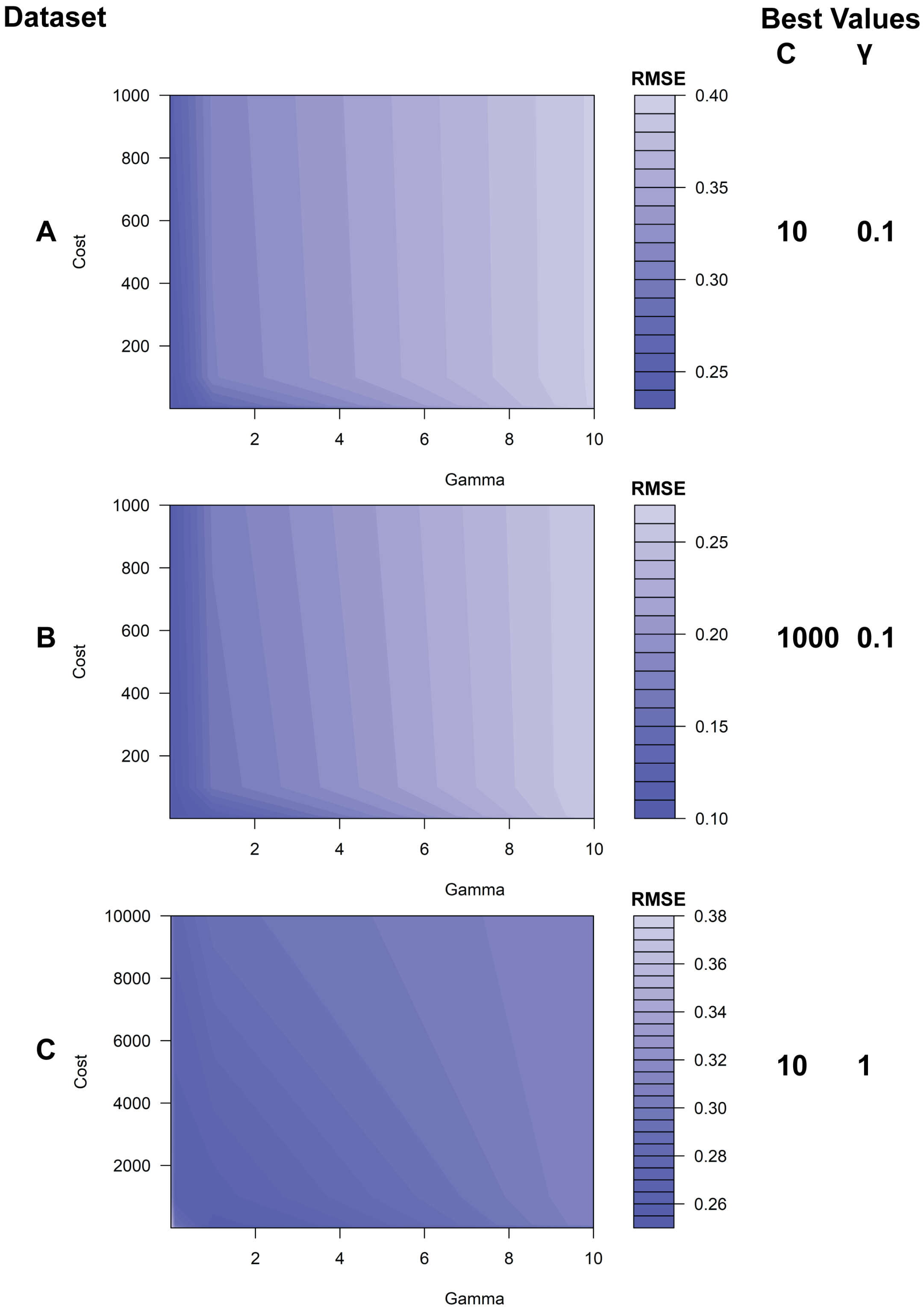


FIGURE B.3: Exemply Results of hyperparameter tuning. Shown are the results of a 5-fold cross-validation adaptive grid search tuning for the two hyperparameters on the three datasets. Areas are colored according to the determined mean RMSE for the parameter combination.

Bibliography

[1] E Abbe. “Beiträge zur Theorie des Mikroskops und der mikroskopischen Wahrnehmung”. In: *Archiv für mikroskopische Anatomie* 9.1 (1873), pp. 413–418. ISSN: 0176-7364. DOI: `10.1007/BF02956173`. URL: `https://doi.org/10.1007/BF02956173`.

[2] Marc Adrian et al. “Cryo-negative staining”. In: *Micron* 29.2 (1998), pp. 145–160. ISSN: 0968-4328. DOI: `https://doi.org/10.1016/S0968-4328(97)00068-1`. URL: `http://www.sciencedirect.com/science/article/pii/S0968432897000681`.

[3] Christopher W Akey and Stuart J Edelstein. “Equivalence of the projected structure of thin catalase crystals preserved for electron microscopy by negative stain, glucose or embedding in the presence of tannic acid”. In: *Journal of Molecular Biology* 163.4 (1983), pp. 575–612. ISSN: 0022-2836. DOI: `https://doi.org/10.1016/0022-2836(83)90113-4`. URL: `http://www.sciencedirect.com/science/article/pii/0022283683901134`.

[4] B J Alder and T E Wainwright. “Studies in Molecular Dynamics. I. General Method”. In: *The Journal of Chemical Physics* 31.2 (1959), pp. 459–466. ISSN: 0021-9606. DOI: `10.1063/1.1730376`. URL: `https://doi.org/10.1063/1.1730376`.

[5] Ethem Alpaydin. *Introduction to Machine Learning*. MIT Press, 2010.

[6] A H Andersen and A C Kak. “Simultaneous Algebraic Reconstruction Technique (SART): A superior implementation of the ART algorithm”. In: *Ultrasonic Imaging* 6.1 (1984), pp. 81–94. ISSN: 0161-7346. DOI: `https://doi.org/10.1016/0161-7346(84)90008-7`. URL: `http://www.sciencedirect.com/science/article/pii/0161734684900087`.

[7] D P Anderson. “BOINC: a system for public-resource computing and storage”. In: *Fifth IEEE/ACM International Workshop on Grid Computing*. 2004, pp. 4–10. ISBN: 1550-5510 VO -. DOI: `10.1109/GRID.2004.14`.

[8] Christian B. Anfinsen. “Principles that Govern the Folding of Protein Chains”. In: *Science* 181.4096 (1973).

[9] Jim Arlow. *UML 2 and the unified process: practical object-oriented analysis and design*. Pearson Education, 2005.

[10] Xiao-chen Bai, Greg McMullan, and Sjors H W Scheres. “How cryo-EM is revolutionizing structural biology”. In: *Trends in Biochemical Sciences* 40.1 (2015), pp. 49–57. ISSN: 0968-0004. DOI: `10.1016/j.tibs.2014.10.005`. URL: `http://dx.doi.org/10.1016/j.tibs.2014.10.005`.

[11] Lindsay A Baker et al. “The resolution dependence of optimal exposures in liquid nitrogen temperature electron cryomicroscopy of catalase crystals”. In: *Journal of Structural Biology* 169.3 (2010), pp. 431–437. ISSN: 1047-8477. DOI: `https://doi.org/10.1016/j.jsb.2009.11.014`. URL: `http://www.sciencedirect.com/science/article/pii/S1047847709003189`.

[12] T S Baker and R Henderson. "Electron cryomicroscopy". In: *International Tables for Crystallography Vol. F* F (2001), pp. 451–463. DOI: `10.1107/97809553602060000703`. URL: `http://it.iucr.org/Fb/ch19o6v0001/`.

[13] E.J. Barber. "Calculation of Density and Viscosity of Sucrose Solutions as Function of Concentration and Temperature". In: *Natl. Cancer Inst. Monogr.* 21.219 (1966), p. 239.

[14] Susan M Baxter et al. "Scientific Software Development Is Not an Oxymoron". In: *PLOS Computational Biology* 2.9 (2006). DOI: `10.1371/journal.pcbi.0020087`.

[15] Daniele Bellavia et al. "A homemade device for linear sucrose gradients". In: *Analytical Biochemistry* 379.2 (2008), pp. 211–212. ISSN: 0003-2697. DOI: `https://doi.org/10.1016/j.ab.2008.05.010`. URL: `http://www.sciencedirect.com/science/article/pii/S000326970800314X`.

[16] Karl Bertram et al. "Cryo-EM structure of a human spliceosome activated for step 2 of splicing". In: *Nature* 542.7641 (2017), pp. 318–323. ISSN: 14764687. DOI: `10.1038/nature21079`. URL: `http://dx.doi.org/10.1038/nature21079`.

[17] Karl Bertram et al. "Cryo-EM Structure of a Pre-catalytic Human Spliceosome Primed for Activation". In: *Cell* 170.4 (2017), 701–713.e11. ISSN: 10974172. DOI: `10.1016/j.cell.2017.07.011`.

[18] B Bishop. "Digital computation of sedimentation coefficients in zonal centrifuges". In: *Natl. Cancer Inst. Monogr.* (1966).

[19] Nikhil Biyani et al. "Focus: The interface between data collection and data processing in cryo-EM". In: *Journal of Structural Biology* 198.2 (2017), pp. 124–133. ISSN: 10958657. DOI: `10.1016/j.jsb.2017.03.007`. URL: `http://dx.doi.org/10.1016/j.jsb.2017.03.007`.

[20] Charles Blilie. "Patterns in scientific software: An introduction". In: *Computing in Science and Engineering* 4.3 (2002), p. 48. ISSN: 15219615. DOI: `10.1109/5992.998640`.

[21] Maxime Bocher. *Plane Analytic Geometry: With Introductory Chapters on the Differential Calculus*. H.Holt, 1915.

[22] Lisa Borland and Marin van Heel. "Classification of image data in conjugate representation spaces". In: *Journal of the Optical Society of America A* 7.4 (1990), p. 601. ISSN: 1084-7529. DOI: `10.1364/JOSAA.7.000601`. URL: `https://www.osapublishing.org/abstract.cfm?URI=josaa-7-4-601`.

[23] Myron K Brakke. "Density Gradient Centrifugation: A New Separation Technique". In: *Journal of the American Chemical Society* 73.4 (1951), pp. 1847–1848. ISSN: 0002-7863. DOI: `10.1021/ja01148a508`. URL: `https://doi.org/10.1021/ja01148a508`.

[24] A F Brilot et al. "Beam-Induced Motion of Vitrified Specimen on Holey Carbon Film". In: *Journal of structural biology* 177.3 (2012), pp. 630–637. ISSN: 08966273. DOI: `10.1097/MCA.0000000000000178.Endothelial`. arXiv: `NIHMS150003`.

[25] Zimei Bu and David J E Callaway. "Chapter 5 - Proteins MOVE! Protein dynamics and long-range allostery in cell signaling". In: *Protein Structure and Diseases*. Ed. by Rossen B T Advances in Protein Chemistry Donev and Structural Biology. Vol. 83. Academic Press, 2011, pp. 163–221. ISBN: 1876-1623. DOI: `https://doi.org/10.1016/B978-0-12-381262-9.00005-7`. URL: `http://www.sciencedirect.com/science/article/pii/B9780123812629000057`.

[26] Boris Busche. "New Algorithms for Automated Processing of Electronmicroscopic Images". PhD thesis. 2013, p. 158. ISBN: 9783844022629.

[27] Ewen Callaway. "The Revolution Will Not Be Crystallized". In: *Nature* 525 (2015), pp. 172–174. ISSN: 0163-6545. DOI: `10.1215/01636545-2009-008`.

[28] Melody G. Campbell et al. "Movies of ice-embedded particles enhance resolution in electron cryo-microscopy". In: *Structure* 20.11 (2012), pp. 1823–1828. ISSN: 09692126. DOI: `10.1016/j.str.2012.08.026`. URL: `http://dx.doi.org/10.1016/j.str.2012.08.026`.

[29] John M. Chambers. *Programming With Data- A Guide to the S Language*. 1998.

[30] Tony F. Chan, Gene H. Golub, and Randall J. LeVeque. *Algorithms for Computing the Sample Variance: Analysis and Recommendations*. 1983. DOI: `10.2307/2683386`. URL: `http://www.jstor.org/stable/2683386?origin=crossref`.

[31] Chih-chung Chang and Chih-jen Lin. "LIBSVM : A Library for Support Vector Machines". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 2 (2013), pp. 1–39. ISSN: 21576904. DOI: `10.1145/1961189.1961199`. arXiv: `0-387-31073-8`.

[32] Scott Chapon and Ben Straub. *Pro Git 2nd Edition*. 2014.

[33] Ashwin Chari et al. "ProteoPlex : stability optimization of macromolecular complexes by sparse-matrix screening of chemical space". In: *Nature Methods* 12.9 (2015), pp. 859–867. DOI: `10.1038/nmeth.3493`.

[34] Anchi Cheng et al. "MRC2014: Extensions to the MRC format header for electron cryo-microscopy and tomography". In: *Journal of Structural Biology* 192.2 (2015), pp. 146–150. ISSN: 10958657. DOI: `10.1016/j.jsb.2015.04.002`.

[35] Nian-Sheng Cheng. "Formula for Viscosity of Glycerol-Water Mixture". In: *Industrial and Engineering Chemistry Research* 47.9 (2008), pp. 3285–3288. ISSN: 01628828. DOI: `10.1063/1.2978249`. URL: `http://www.sciencemag.org/content/299/5613/1719.abstract`.

[36] W Chiu, K H Downing, and J Dubochet. "Cryoprotection in electron microscopy". In: *Journal of Microscopy (Oxford)* 141.pt3 (1986), pp. 385–391.

[37] Corinna Cortes and Vladimir Vapnik. "Support Vector Networks". In: *Machine Learning* 20.3 (1995), 273˜–˜297. ISSN: 08856125. DOI: `10.1007/BF00994018`. arXiv: `arXiv:1011.1669v3`. URL: `http://link.springer.com/10.1007/BF00994018`.

[38] Rachel Courtland. *Gordon Moore: The Man Whose Name Means Progress*. 2015. URL: `https://spectrum.ieee.org/computing/hardware/gordon-moore-the-man-whose-name-means-progress`.

[39] Francis Crick. "Central dogma of molecular biology". In: *Nature* 227.5258 (1970), pp. 561–563. ISSN: 00280836. DOI: `10.1038/227561a0`. arXiv: `arXiv:1011.1669v3`.

[40] Nello Cristianini and Bernhard Schölkopf. "Support Vector Machines and Kernel Methods: The New Generation of Learning Machines". In: *Artificial Intelligence Magazine* 23.3 (2002), pp. 31–42. ISSN: 0738-4602. DOI: `http://dx.doi.org/10.1609/aimag.v23i3.1655`.

[41] R.A. Crowther, D.J. DeRosier, and A. Klug. "The reconstruction of a three-dimensional structure from projections and its application to electron microscopy". In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 317.1530 (1970), 319 LP –340. URL: `http://rspa.royalsocietypublishing.org/content/317/1530/319.abstract`.

[42] Radostin Danev et al. "Volta potential phase plate for in-focus phase contrast transmission electron microscopy". In: *Proceedings of the National Academy of Sciences* 111.44 (2014), 15635 LP –15640. URL: `http://www.pnas.org/content/111/44/15635.abstract`.

[43] Louis De Broglie. "Waves and Quanta". In: *Nature* 112 (1923), p. 540. URL: `http://dx.doi.org/10.1038/112540a0http://10.0.4.14/112540a0`.

[44] C. Ding and X. He. "K-means clustering via principal component analysis". In: *Proceedings of the 21st International Conference on Machine Learning*. 2004, pp. 225–232.

[45] Christopher M Dobson. "Protein folding and misfolding". In: *Nature* 426 (2003), p. 884. URL: `http://dx.doi.org/10.1038/nature02261http://10.0.4.14/nature02261`.

[46] Jan Drenth. *Principles of Protein X-Ray Crystallography*. Springer, 2007.

[47] P. Dube et al. "The portal protein of bacterio- phage SPP1: a DNA pump with 13-fold symmetry". In: *The EMBO journal* 12.4 (1993), pp. 1303–1309.

[48] Nadav Elad et al. "Detection and separation of heterogeneity in molecular complexes by statistical analysis of their two-dimensional projections". In: *Journal of Structural Biology* 162.1 (2008), pp. 108–120. ISSN: 1047-8477. DOI: `https://doi.org/10.1016/j.jsb.2007.11.007`. URL: `http://www.sciencedirect.com/science/article/pii/S1047847707002912`.

[49] Hans Elmlund, Dominika Elmlund, and Samy Bengio. "PRIME: Probabilistic Initial 3D Model Generation for Single-Particle Cryo-Electron Microscopy". In: *Structure* 21.8 (2013), pp. 1299–1306. ISSN: 0969-2126. DOI: `10.1016/j.str.2013.07.002`. URL: `http://dx.doi.org/10.1016/j.str.2013.07.002`.

[50] J. J. Fernández et al. "Sharpening high resolution information in single particle electron cryomicroscopy". In: *Journal of Structural Biology* 164.1 (2008), pp. 170–175. ISSN: 10478477. DOI: `10.1016/j.jsb.2008.05.010`.

[51] Emil Fischer. "Einfluss der Configuration auf die Wirkung der Enzyme". In: *Berichte der deutschen chemischen Gesellschaft* 27.3 (1894), pp. 2985–2993. ISSN: 0365-9496. DOI: `10.1002/cber.18940270364`. URL: `https://doi.org/10.1002/cber.18940270364`.

[52] Joseph L Fleiss, Bruce Levin, and Myunghee Cho Paik. *Statistical Methods for Rates and Proportions*. John Wiley and Sons.

[53] Agner Fog. *Optimizing software in C++*. 2014, pp. 1–160. URL: `http://www.agner.org/optimize/optimizing{\_}cpp.pdf`.

[54] *Format descriptions for still images*. URL: `https://www.loc.gov/preservation/digital/formats/fdd/still{\_}fdd.shtml`.

[55] Message P Forum. *MPI: A Message-Passing Interface Standard*. Tech. rep. Knoxville, TN, USA, 1994.

[56] J.B. Joseph Fourier. *Théorie analytique de la chaleur*. Paris, 1822.

[57] Martin Fowler. *UML Distilled Third Edition*. 2003, pp. 1–118.

[58] J Frank. "Image analysis of single macromolecules". In: *Electron Microsc. Rev.* 2.1 (1989), pp. 53–74. ISSN: 08920354. DOI: `10.1016/0892-0354(89)90010-5`.

[59] Joachim Frank. *Three-Dimensinal Electron Microscopy of Macromolecular Assenblies*. Oxford University Press, 2013. ISBN: 9788578110796. DOI: `10.1017/CBO9781107415324.004`. arXiv: `arXiv:1011.1669v3`.

[60] James S Fraser et al. "Hidden alternative structures of proline isomerase essential for catalysis". In: *Nature* 462 (2009), p. 669. URL: `http://dx.doi.org/10.1038/nature08615http://10.0.4.14/nature08615https://www.nature.com/articles/nature08615{\#}supplementary-information`.

[61] H Frauenfelder, S G Sligar, and P G Wolynes. "The energy landscapes and motions of proteins". In: *Science* 254.5038 (1991), 1598 LP –1603. URL: `http://science.sciencemag.org/content/254/5038/1598.abstract`.

[62] B Freitag et al. "Breaking the spherical and chromatic aberration barrier in transmission electron microscopy". In: *Ultramicroscopy* 102.3 (2005), pp. 209–214. ISSN: 0304-3991. DOI: `https://doi.org/10.1016/j.ultramic.2004.09.013`. URL: `http://www.sciencedirect.com/science/article/pii/S0304399104001834`.

[63] Daissy H Garces, William T Rhodes, and Nestor M Peña. "Projection-slice theorem: a compact notation". In: *Journal of the Optical Society of America A* 28.5 (2011), pp. 766–769. DOI: `10.1364/JOSAA.28.000766`. URL: `http://josaa.osa.org/abstract.cfm?URI=josaa-28-5-766`.

[64] C.F. Gauss. "Theoria interpolationis methodo nova tractata". In: *Werke*. Göttingen: Königliche Gesellschaft der Wissenschaften, 1866, pp. 265–330.

[65] Peter Gilbert. "Iterative methods for the three-dimensional reconstruction of an object from projections". In: *Journal of Theoretical Biology* 36.1 (1972), pp. 105–117. ISSN: 0022-5193. DOI: `https://doi.org/10.1016/0022-5193(72)90180-4`. URL: `http://www.sciencedirect.com/science/article/pii/0022519372901804`.

[66] Monika M. Golas et al. "3D Cryo-EM Structure of an Active Step I Spliceosome and Localization of Its Catalytic Core". In: *Molecular Cell* 40.6 (2010), pp. 927–938. ISSN: 10972765. DOI: `10.1016/j.molcel.2010.11.023`.

[67] Richard Gordon, Robert Bender, and Gabor T Herman. "Algebraic Reconstruction Techniques (ART) for three-dimensional electron microscopy and X-ray photography". In: *Journal of Theoretical Biology* 29.3 (1970), pp. 471–481. ISSN: 0022-5193. DOI: `https://doi.org/10.1016/0022-5193(70)90109-8`. URL: `http://www.sciencedirect.com/science/article/pii/0022519370901098`.

[68] Timothy Grant and Nikolaus Grigorieff. "Measuring the optimal exposure for single particle cryo-EM using a 2.6 Å reconstruction of rotavirus VP6". In: *eLife* 4 (2015). Ed. by Wesley I Sundquist, e06980. ISSN: 2050-084X. DOI: `10.7554/eLife.06980`. URL: `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4471936/`.

[69] N. Grigorieff. *Frealign: An Exploratory Tool for Single-Particle Cryo-EM*. 1st ed. Vol. 579. Elsevier Inc., 2016, pp. 191–226. ISBN: 9780128053829. DOI: `10.1016/bs.mie.2016.04.013`. URL: `http://dx.doi.org/10.1016/bs.mie.2016.04.013`.

[70] Sydney R. Hall. "The STAR File: A New Format for Electronic Data Transfer and Archiving". In: *Journal of Chemical Information and Computer Sciences* 31.2 (1991), pp. 326–333. ISSN: 00952338. DOI: `10.1021/ci00002a020`.

[71] G. Harauz and M. van Heel. "Exact filters for general geometry three dimensional reconstruction". In: *Optik* 73 (1986), pp. 146–156.

[72] J.Robin Harris and Dirk Scheffler. "Routine preparation of air-dried negatively stained and unstained specimens on holey carbon support films: a review of applications". In: *Micron* 33.5 (2002), pp. 461–480. ISSN: 0968-4328. DOI: `https://doi.org/10.1016/S0968-4328(01)00039-7`. URL: `http://www.sciencedirect.com/science/article/pii/S0968432801000397`.

[73] David Haselbach. "Conformational Dynamics of Large Proteins and Protein Complexes". PhD thesis. 2014.

[74] David Haselbach et al. "Long-range allosteric regulation of the human 26S proteasome by 20S proteasome-targeting cancer drugs". In: *Nature Communications* 8.May (2017), pp. 1–8. ISSN: 20411723. DOI: `10.1038/ncomms15578`. URL: `http://dx.doi.org/10.1038/ncomms15578`.

[75] David Haselbach et al. "Structure and Conformational Dynamics of the Human Spliceosomal BactComplex". In: *Cell* 172.3 (2018), 454–464.e11. ISSN: 10974172. DOI: `10.1016/j.cell.2018.01.010`.

[76] T; Tibshirani Hastie. "The Elements of Statistical Learning". In: *Math. Intell.* 27.2 (2009), pp. 83–85. ISSN: 03436993. DOI: `10.1007/b94608`. arXiv: `arXiv:1011.1669v3`. URL: `http://www.springerlink.com/index/10.1007/b94608`.

[77] Florian Hauer et al. "GraDeR: Membrane Protein Complex Preparation for Single-Particle Cryo-EM". In: *Structure* 23.9 (2015), pp. 1769–1775. ISSN: 18784186. DOI: `10.1016/j.str.2015.06.029`.

[78] Elke Haustein and Petra Schwille. "Single-molecule spectroscopic methods". In: *Current Opinion in Structural Biology* 14.5 (2004), pp. 531–540. ISSN: 0959-440X. DOI: `https://doi.org/10.1016/j.sbi.2004.09.004`. URL: `http://www.sciencedirect.com/science/article/pii/S0959440X04001447`.

[79] W A Havelka, R Henderson, and D Oesterhelt. "Three-dimensional structure of halorhodopsin at 7 Å resolution". In: *Journal of Molecular Biology* 247.4 (1995), pp. 726–738. ISSN: 0022-2836. DOI: `https://doi.org/10.1016/S0022-2836(05)80151-2`. URL: `http://www.sciencedirect.com/science/article/pii/S0022283605801512`.

[80] M. van Heel et al. "Arthropod hemo- cyanin studies by image analysis". In: *Structure and Function of Invertebrate Respiratory Proteins*. Ed. by E. Wood. Harwood Academic, 1982, pp. 69–73.

[81] Marin van Heel and George Harauz. *Resolution criteria for three dimensional reconstruction*. 1986.

[82] Marin van Heel and Wilko Keegstra. "IMAGIC: A fast, flexible and friendly image analysis software system". In: *Ultramicroscopy* 7.2 (1981), pp. 113–129. ISSN: 03043991. DOI: `10.1016/0304-3991(81)90001-2`.

[83] Marin van Heel and Michael Schatz. "Fourier shell correlation threshold criteria". In: *Journal of Structural Biology* 151.3 (2005), pp. 250–262. ISSN: 1047-8477. DOI: `https://doi.org/10.1016/j.jsb.2005.05.009`. URL: `http://www.sciencedirect.com/science/article/pii/S1047847705001292`.

[84] Burkhard Clemens Heisen. "New Algorithms for Macromolecular Structure Determination". PhD thesis. 2009.

[85] Stefan W Hell and Jan Wichmann. "Breaking the diffraction resolution limit by stimulated emission: stimulated-emission-depletion fluorescence microscopy". In: *Optics Letters* 19.11 (1994), pp. 780–782. DOI: `10.1364/OL.19.000780`. URL: `http://ol.osa.org/abstract.cfm?URI=ol-19-11-780`.

[86] Richard Henderson. "Image contrast in high-resolution electron microscopy of biological macromolecules: TMV in ice". In: *Ultramicroscopy* 46.1 (1992), pp. 1–18. ISSN: 0304-3991. DOI: `https://doi.org/10.1016/0304-3991(92)90003-3`. URL: `http://www.sciencedirect.com/science/article/pii/0304399192900033`.

[87] Richard Henderson et al. "Outcome of the First Electron Microscopy Validation Task Force Meeting". In: *Structure* 20-330.2 (2012), pp. 205–214. ISSN: 0969-2126. DOI: `10.1016/j.str.2011.12.014`. URL: `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3328769/`.

[88] Katherine A Henzler-Wildman et al. "Intrinsic motions along an enzymatic reaction trajectory". In: *Nature* 450 (2007), p. 838. URL: `http://dx.doi.org/10.1038/nature06410http://10.0.4.14/nature06410https://www.nature.com/articles/nature06410{\#}supplementary-information`.

[89] G T Herman and M Kalinowski. "Classification of heterogeneous electron microscopic projections into homogeneous subsets". In: *Ultramicroscopy* 108.4 (2008), pp. 327–338. ISSN: 0304-3991. DOI: `https://doi.org/10.1016/j.ultramic.2007.05.005`. URL: `http://www.sciencedirect.com/science/article/pii/S0304399107001374`.

[90] J Bernard Heymann and David M Belnap. "Bsoft: Image processing and molecular modeling for electron microscopy". In: *Journal of Structural Biology* 157.1 (2007), pp. 3–18. ISSN: 1047-8477. DOI: `https://doi.org/10.1016/j.jsb.2006.06.006`. URL: `http://www.sciencedirect.com/science/article/pii/S1047847706001997`.

[91] J. Bernard Heymann, Mónica Chagoyen, and David M. Belnap. "Common conventions for interchange and archiving of three-dimensional electron microscopy information in structural biology". In: *Journal of Structural Biology* 151.2 (2005), pp. 196–207. ISSN: 10478477. DOI: `10.1016/j.jsb.2005.06.001`.

[92] William Hirst and Robert A. Cox. "A method for predicting the location of particles sedimenting in sucrose gradients". In: *Analytical Biochemistry* 131.1 (1983), pp. 51–68. ISSN: 10960309. DOI: `10.1016/0003-2697(83)90135-5`.

[93] Chih-Wei Hsu, Chih-Chungg Chang, and Chih-Jen Lin. "A Practical Guide to Support Vector Classification". Taipei, 2010.

[94] Peter J. Huber. *Robust Statistics*. Vol. 82. 3. 1981, p. 308. ISBN: 9780470434697. DOI: `10.1002/9780470434697`. URL: `http://doi.wiley.com/10.1002/9780470434697`.

[95] *ISO/IEC 14882:2005 Information technology — Programming languages — C++*. Tech. rep. 2005. URL: `ftp://std.dkuug.dk/JTC1/SC22/WG21/prot/14882fdis/n1577.pdf`.

[96] *ISO/IEC 14882:2011 Information technology — Programming languages — C++*. Tech. rep. 2012, 1338 (est.) URL: `http://www.iso.org/iso/iso{\_}catalogue/catalogue{\_}tc/catalogue{\_}detail.htm?csnumber=50372`.

[97] Arjen J Jakobi, Matthias Wilmanns, and Carsten Sachse. "Model-based local density sharpening of cryo-EM maps". In: *eLife* 6 (2017). Ed. by Axel T Brunger, e27131. ISSN: 2050-084X. DOI: `10.7554/eLife.27131`. URL: `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC5679758/`.

[98] I.T. Jolliffe. *Principal Component Analysis*. Second. Springer, 2002.

[99] LN Joppa et al. "Troubling Trends in Scientific Software Use". In: *Science* 340.May (2013). ISSN: 15449173. DOI: `10.1371/journal.pbio.1001745`. arXiv: `1210.0530`. URL: `http://arxiv.org/abs/1210.0530{\%}0Ahttp://dx.doi.org/10.1371/journal.pbio.1001745`.

[100] L. Joyeux and P. a. Penczek. "Efficiency of 2D alignment methods". In: *Ultramicroscopy* 92.2 (2002), pp. 33–46.

[101] Berthold Kastner et al. "GraFix: Sample preparation for single-particle electron cryomicroscopy". In: *Nature Methods* 5.1 (2008), pp. 53–55. ISSN: 15487091. DOI: `10.1038/nmeth1139`.

[102] Lewis E Kay. "NMR studies of protein structure and dynamics". In: *Journal of Magnetic Resonance* 173.2 (2005), pp. 193–207. ISSN: 1090-7807. DOI: `https://doi.org/10.1016/j.jmr.2004.11.021`. URL: `http://www.sciencedirect.com/science/article/pii/S1090780704003854`.

[103] James Keeler. *Understanding NMR Spectroscopy*. December. 2002, p. 211. ISBN: 1119964938. DOI: `10.1007/SpringerReference_67582`. arXiv: `arXiv:1011.1669v3`. URL: `http://www.springerreference.com/index/doi/10.1007/SpringerReference{\_}67582`.

[104] E. Kellenberger and J. Kistler. "The physics of specimen preparation". In: *Advances in Structure Research by Diffraction Methods*. Ed. by W Hoppe and R Mason. Wiesbaden: Vieweg, 1979, pp. 49–79.

[105] Diane F. Kelly. "A software chasm: Software engineering and scientific computing". In: *IEEE Software* 24.6 (2007), pp. 0–2. ISSN: 07407459. DOI: `10.1109/MS.2007.155`.

[106] Maryam Khoshouei et al. "Cryo-EM structure of haemoglobin at 3.2 Å determined with the Volta phase plate". In: *Nature Communications* 8.May (2017), pp. 1–6. ISSN: 20411723. DOI: `10.1038/ncomms16099`. URL: `http://dx.doi.org/10.1038/ncomms16099`.

[107] Jan-Martin Kirves. "New Algorithms for Single Particle Cryo Electron Microscopic Image Processing". PhD thesis. 2014.

[108] R Kohavi. "A study of cross validation and bootstrap for accuracy estimation and model selection". In: *14th International Joint Conference on Artificial Intelligence (IJCAI)* 2 (1995), pp. 1137–1144.

[109] Alp Kucukelbir, Fred J Sigworth, and Hemant D Tagare. "The Local Resolution of Cryo-EM Density Maps". In: *Nature methods* 11.1 (2014), pp. 63–65. ISSN: 1548-7091. DOI: `10.1038/nmeth.2727`. URL: `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3903095/`.

[110] Max Kuhn et al. *Package 'caret' Classification and Regression Training Description Misc functions for training and plotting classification and regression models*. 2017. URL: `https://cran.r-project.org/web/packages/caret/caret.pdf`.

[111] Jack Kyte. *Structure in protein chemistry*. New York & London: Garland, 1995. DOI: `10.1016/S0307-4412(96)80028-8`. URL: `https://doi.org/10.1016/S0307-4412(96)80028-8`.

[112] J Richard Landis and Gary G Koch. "The Measurement of Observer Agreement for Categorical Data". In: *Biometrics* 33.1 (1977), pp. 159–174. ISSN: 0006341X. DOI: `10.2307/2529310`.

[113] Leon S Lasdon. *Optimization Theory for Large Systems*. New York: The Macmillan Company, 1970.

[114] Michael Levitt and Arieh Warshel. "Computer simulation of protein folding". In: *Nature* 253 (1975), p. 694. URL: `http://dx.doi.org/10.1038/253694a0http://10.0.4.14/253694a0`.

[115] Tony Lindeberg. *Image Matching Using Generalized Scale-Space Interest Points*. Vol. 52. 1. Springer US, 2015, pp. 3–36. ISBN: 1085101405410. DOI: `10.1007/s10851-014-0541-0`. URL: `http://dx.doi.org/10.1007/s10851-014-0541-0`.

[116] Stanley B Lippman, Josée Lajoie, and Barbara E. Moo. *C++ Primer , Fifth Edition*. 2004, pp. 1–202. ISBN: 0672326965. DOI: `10.1017/CBO9781107415324.004`. arXiv: `arXiv:1011.1669v3`.

[117] Mario Luettich. "Analytische Methoden zur hochauflösenden Strukturbestimmung in der Kryo-Elektronen-Mikroskopie". PhD Thesis. Göttingen, 2007.

[118] Roberto Marabini et al. "CTF Challenge: Result Summary". In: *Journal of structural biology* 190.3 (2015), pp. 348–359. ISSN: 1047-8477. DOI: `10.1016/j.jsb.2015.04.003`. URL: `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4672951/`.

[119] Mary L McHugh. "Interrater reliability: the kappa statistic". In: *Biochemia Medica* 22.3 (2012), pp. 276–282. ISSN: 1330-0962. URL: `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3900052/`.

[120] G McMullan et al. "Comparison of optimal performance at 300keV of three direct electron detectors for use in low dose electron microscopy". In: *Ultramicroscopy* 147 (2014), pp. 156–163. ISSN: 0304-3991. DOI: `https://doi.org/10.1016/j.ultramic.2014.08.002`. URL: `http://www.sciencedirect.com/science/article/pii/S030439911400151X`.

[121] David Meyer. *Support Vector Machines: interface to libsvm in e1071*. 2017. DOI: `10.1002/wics.049`. arXiv: `arXiv:1011.1669v3`. URL: `http://www.csie.ntu.edu.tw/{~}cjlin/papers/ijcnn.ps.gz.{\%}0Ahttp://www.csie.ntu.edu.tw/{~}cjlin/papers/ijcnn.ps.gz.{\%}0Ahttp://www.tandfonline.com/doi/abs/10.1198/jasa.2003.s269`.

[122] Gordon E. Moore. "Cramming more components onto integrated circuits". In: *Proceedings of the IEEE* 86.1 (1965), pp. 82–85. ISSN: 00189219. DOI: `10.1109/JPROC.1998.658762`.

[123] Toshio Moriya et al. "High-resolution Single Particle Analysis from Electron Cryo-microscopy Images Using SPHIRE". In: *Journal of Visualized Experiments* 123 (2017), pp. 1–11. ISSN: 1940-087X. DOI: `10.3791/55448`. URL: `https://www.jove.com/video/55448/high-resolution-single-particle-analysis-from-electron-cryo`.

[124] NVIDIA. *CUDA*. URL: `https://developer.nvidia.com/cuda-zone` (visited on 04/19/2018).

[125] H Nyquist. "Certain Topics in Telegraph Transmission Theory". In: *Transactions of the American Institute of Electrical Engineers* 47.617-644 (1928).

[126] Linus Pauling, Robert B Corey, and H R Branson. "The structure of proteins: Two hydrogen-bonded helical configurations of the polypeptide chain". In: *Proceedings of the National Academy of Sciences* 37.4 (1951), 205 LP –211. URL: `http://www.pnas.org/content/37/4/205.abstract`.

[127] Karl Pearson. "On lines and planes of closest fit to systems of points in space". In: *Philosophical Magazine Series 6* 2.11 (1901), pp. 559–572. ISSN: 1941-5982. DOI: `10.1080/14786440109462720`. URL: `http://www.tandfonline.com/doi/abs/10.1080/14786440109462720`.

[128] P. A. Penczek, R. A. Grassucci, and J. Frank. "The ribosome at improved resolution: new techniques for merging and orientation refinement in 3D cryo-electron microscopy of biological particles". In: *Ultramicroscopy* 53.3 (1994), pp. 251–270.

[129] Pawel a Penczek. "Fundamentals of three-dimensional recostruction from projections". In: *Methods in enzymology* 482.10 (2010), pp. 1–28. ISSN: 1557-7988. DOI: `10.1016/S0076-6879(10)82001-4.Fundamentals`. URL: `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3165033{\&}tool=pmcentrez{\&}rendertype=abstract`.

[130] Pawel A Penczek. "Resolution Measures in Molecular Electron Microscopy". In: *Cryo-EM, Part B: 3-D Reconstruction*. Ed. by Grant J B T Methods in Enzymology Jensen. Vol. 482. Academic Press, 2010, pp. 73–100. ISBN: 0076-6879. DOI: `https://doi.org/10.1016/S0076-6879(10)82003-8`. URL: `http://www.sciencedirect.com/science/article/pii/S0076687910820038`.

[131] Pawel A Penczek. "Three-dimensional spectral signal-to-noise ratio for a class of reconstruction algorithms". In: *Journal of Structural Biology* 138.1 (2002), pp. 34–46. ISSN: 1047-8477. DOI: `https://doi.org/10.1016/S1047-8477(02)00033-3`. URL: `http://www.sciencedirect.com/science/article/pii/S1047847702000333`.

[132] Pawel A Penczek, Joachim Frank, and Christian M T Spahn. "A method of focused classification, based on the bootstrap 3D variance analysis, and its application to EF-G-dependent translocation". In: *Journal of Structural Biology* 154.2 (2006), pp. 184–194. ISSN: 1047-8477. DOI: `https://doi.org/10.1016/j.jsb.2005.12.013`. URL: `http://www.sciencedirect.com/science/article/pii/S1047847705002881`.

[133] Pawel A Penczek, Marek Kimmel, and Christian M T Spahn. "Identifying conformational states of macromolecules by eigen-analysis of resampled cryo-EM images". In: *Structure(London, England:1993)* 19.11 (2011), pp. 1582–1590. ISSN: 0969-2126. DOI: `10.1016/j.str.2011.10.003`. URL: `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3255080/`.

[134] Pawel A Penczek, Robert Renka, and Hermann Schomberg. "Gridding-based direct Fourier inversion of the three-dimensional ray transform." eng. In: *Journal of the Optical Society of America. A, Optics, image science, and vision* 21.4 (2004), pp. 499–509. ISSN: 1084-7529 (Print).

[135] John S. Philo. "A method for directly fitting the time derivative of sedimentation velocity data and an alternative algorithm for calculating sedimentation coefficient distribution functions". In: *Analytical Biochemistry* 279.2 (2000), pp. 151–163. ISSN: 00032697. DOI: `10.1006/abio.2000.4480`.

[136] John S. Philo. "Limiting the sedimentation coefficient for sedimentation velocity data analysis: Partial boundary modeling and g(sz.ast;) approaches revisited". In: *Analytical Biochemistry* 412.2 (2011), pp. 189–202. ISSN: 00032697. DOI: `10.1016/j.ab.2011.01.035`. URL: `http://dx.doi.org/10.1016/j.ab.2011.01.035`.

[137] John S Philo. "Measuring Sedimentation, Diffusion, and Molecular Weights of Small Molecules by Direct Fitting of Sedimentation Velocity Concentration Profiles". In: *Modern Analytical Ultracentrifugation* (1994), pp. 156–170. DOI: `10.1007/978-1-4684-6828-1_9`. URL: `www.jphilo.mailway.com/svedberg.pdf`.

[138] James B Procter and Andreas Prlic. "Ten Simple Rules for the Open Development of Scientific Software". In: *PLOS Computational Biology* 8.12 (2012), pp. 8–10. DOI: `10.1371/journal.pcbi.1002802`.

[139] Friedrich Pukelsheim. "The three sigma rule". In: *American Statistician* 48.2 (1994), pp. 88–91. ISSN: 15372731. DOI: `10.1080/00031305.1994.10476030`.

[140] Jeff Racine. "Gnuplot 4.0: A portable interactive plotting utility". In: *Journal of Applied Econometrics* 21.1 (2006), pp. 133–141. ISSN: 08837252. DOI: `10.1002/jae.885`.

[141] J Radon. "On the determination of functions from their integral values along certain manifolds". In: *IEEE Transactions on Medical Imaging* 5.4 (1986), pp. 170–176. ISSN: 0278-0062 VO - 5. DOI: `10.1109/TMI.1986.4307775`.

[142] Johann Radon. "Mengen konvexer Körper, die einen gemeinsamen Punkt enthalten". In: *Mathematische Annalen* 83.1 (1921), pp. 113–115. ISSN: 1432-1807. DOI: `10.1007/BF01464231`. URL: `https://doi.org/10.1007/BF01464231`.

[143] Alain Rakotomamonjy. "Variable Selection Using SVM-based Criteria". In: *Journal ofMachine Learning Research* 3 (2003), pp. 1357–1370. ISSN: 15324435. DOI: `10.1162/153244303322753706`. arXiv: `1111.6189v1`. URL: `http://delivery.acm.org/10.1145/950000/944977/3-1357-rakotomamonjy.pdf?ip=195.220.108.113{\&}id=944977{\&}acc=PUBLIC{\&}key=7EBF6E77E86B478F.399DC7EE7AC85DEE.4D4702B0C3E38B35.4D4702B0C3E38B35{\&}CFID=426401126{\&}CFTOKEN=82029792{\&}{\_}{\_}acm{\_}{\_}=1395658938{\_}a7ebbb259e6b9309b92a8`.

[144] Ludwig Reimer and Helmut Kohl. *Transmission Electron Microscopy Physics of Image Formation*. Vol. 51. 2008, p. 587. ISBN: 9780387400938. DOI: `10.1007/978-0-387-34758-5`. arXiv: `arXiv:1011.1669v3`. URL: `http://link.springer.com/10.1007/978-0-387-40093-8{\%}5Cnhttp://books.google.com/books?id=o-0nI8hxEpMC{\&}pgis=1`.

[145] David Rickwood. *Centrifugation: A practical approach*. OXford: IRL Press, 1984.

[146] Guillaume Rigaut et al. "A generic protein purification method for protein complex characterization and proteome exploration". In: *Nature Biotechnology* 17 (1999), p. 1030. URL: `http://dx.doi.org/10.1038/13732http://10.0.4.14/13732`.

[147] Alexis Rohou and Nikolaus Grigorieff. "CTFFIND4: Fast and accurate defocus estimation from electron micrographs". In: *Journal of Structural Biology* 192.2 (2015), pp. 216–221. ISSN: 1047-8477. DOI: `https://doi.org/10.1016/j.jsb.2015.08.008`. URL: `http://www.sciencedirect.com/science/article/pii/S1047847715300460`.

[148] Peter B Rosenthal and Richard Henderson. "Optimal Determination of Particle Orientation, Absolute Hand, and Contrast Loss in Single-particle Electron Cryomicroscopy". In: *Journal of Molecular Biology* 333.4 (2003), pp. 721–745. ISSN: 0022-2836. DOI: `https://doi.org/10.1016/j.jmb.2003.07.013`. URL: `http://www.sciencedirect.com/science/article/pii/S0022283603010222`.

[149] Royal Swedish Academy of Sciences. *The Nobel Prize in Chemistry 2017*. Stockholm, 2017. URL: `https://www.nobelprize.org/nobel{\_}prizes/chemistry/laureates/2017/press.pdf`.

[150] Rachel S Ruskin, Zhiheng Yu, and Nikolaus Grigorieff. "Quantitative characterization of electron detectors for transmission electron microscopy". In: *Journal of Structural Biology* 184.3 (2013), pp. 385–393. ISSN: 1047-8477. DOI: `https://doi.org/10.1016/j.jsb.2013.10.016`. URL: `http://www.sciencedirect.com/science/article/pii/S1047847713002815`.

[151] Artur L Samuel. "Some studies in machine learning using the game of checkers". In: *IBM Journal of research and development* 3.3 (1959), pp. 210–229. ISSN: 0018-8646. DOI: `10.1147/rd.33.0210`.

[152] B Sander, M M Golas, and H Stark. "Automatic CTF correction for single particles based upon multivariate statistical analysis of individual power spectra". In: *Journal of Structural Biology* 142.3 (2003), pp. 392–401. ISSN: 1047-8477. DOI: `https://doi.org/10.1016/S1047-8477(03)00072-8`. URL: `http://www.sciencedirect.com/science/article/pii/S1047847703000728`.

[153] B. Sander, M. M. Golas, and H. Stark. "Corrim-based alignment for improved speed in single-particle image processing". In: *Journal of Structural Biology* 143.3 (2003), pp. 219–228. ISSN: 10478477. DOI: `10.1016/j.jsb.2003.08.001`.

[154] Bjoern Sander et al. "An Approach for De Novo Structure Determination of Dynamic Molecular Assemblies by Electron Cryomicroscopy". In: *Structure* 18.6 (2010), pp. 667–676. ISSN: 09692126. DOI: `10.1016/j.str.2010.05.001`.

[155] F Sanger and H Tuppy. "The amino-acid sequence in the phenylalanyl chain of insulin. 1. The identification of lower peptides from partial hydrolysates". In: *Biochemical Journal* 49.4 (1951), 463 LP –481. URL: `http://www.biochemj.org/content/49/4/463.abstract`.

[156] W. K. Sartory, H. B. Halsall, and J. P. Breillatt. "Simulation of gradient and band propagation in the centrifuge". In: *Biophysical Chemistry* 5.1-2 (1976), pp. 107–135. ISSN: 03014622. DOI: `10.1016/0301-4622(76)80029-4`.

[157] Sjors H W Scheres. "RELION: Implementation of a Bayesian approach to cryo-EM structure determination". In: *Journal of Structural Biology* 180.3 (2012), pp. 519–530. ISSN: 1047-8477. DOI: `10.1016/j.jsb.2012.09.006`. URL: `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3690530/`.

[158] Sjors H W Scheres. "RELION: Implementation of a Bayesian approach to cryo-EM structure determination". In: *Journal of Structural Biology* 180.3 (2012), pp. 519–530. ISSN: 10478477. DOI: `10.1016/j.jsb.2012.09.006`. URL: `http://dx.doi.org/10.1016/j.jsb.2012.09.006`.

[159] Sjors H W Scheres and Shaoxia Chen. "Prevention of overfitting in cryo-EM structure determination". In: *Nature methods* 9.9 (2012), pp. 853–854. ISSN: 1548-7091. DOI: `10.1038/nmeth.2115`. URL: `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4912033/`.

[160] Sjors H W Scheres et al. "Disentangling conformational states of macromolecules in 3D-EM through likelihood optimization". In: *Nature Methods* 4 (2006), p. 27. URL: `http://dx.doi.org/10.1038/nmeth992http://10.0.4.14/nmeth992https://www.nature.com/articles/nmeth992{\#}supplementary-information`.

[161] Sjors H W Scheres et al. "Modeling Experimental Image Formation for Likelihood-Based Classification of Electron Microscopy Data". In: *Structure* 15.10 (2007), pp. 1167–1177. ISSN: 0969-2126. DOI: `https://doi.org/10.1016/j.str.2007.09.003`. URL: `http://www.sciencedirect.com/science/article/pii/S0969212607003279`.

[162] Martin Schmeisser et al. "Parallel, distributed and GPU computing technologies in single-particle electron microscopy". In: *Acta Crystallographica Section D: Biological Crystallography* 65.Pt 7 (2009), pp. 659–671. ISSN: 0907-4449. DOI: `10.1107/S0907444909011433`. URL: `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2703572/`.

[163] J Schrader et al. "The inhibition mechanism of human 20". In: *Science* 353.6299 (2016), pp. 595–598. DOI: `10.1126/science.aaf8993`.

[164] Peter Schuck. "Diffusion-deconvoluted sedimentation coefficient distributions for the analysis of interacting and non-interacting protein mixtures". In: *Modern Analytical Ultracentrifugation: Techniques and Methods* 301 (2005), pp. 26–51.

[165] J Segal. "Some Problems of Professional End User Developers". In: *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2007)*. 2007, pp. 111–118. ISBN: 1943-6092 VO -. DOI: `10.1109/VLHCC.2007.17`.

[166] Judith Segal. *Models of scientific software development*. 2004.

[167] F. J. Sigworth et al. "An introduction to maximum-likelihood methods in cryo-EM." In: *Methods in enzymology* 482.10 (2010), pp. 263–294.

[168] Amit Singer et al. "Detecting Consistent Common Lines in Cryo-EM by Voting". In: *Journal of structural biology* 169.3 (2010), pp. 312–322. ISSN: 1047-8477. DOI: `10.1016/j.jsb.2009.11.003`. URL: `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2826584/`.

[169] Roy D Sleator. "Proteins". In: *Bioengineered* 3.2 (2012), pp. 80–85. ISSN: 2165-5979. DOI: `10.4161/bbug.18303`. URL: `https://doi.org/10.4161/bbug.18303`.

[170] Steven W Smith. "Continuous Signal Processing". In: *The Scientist and Engineer's Guide to Digital Signal Processing*. 1997. Chap. 13, pp. 243–260. ISBN: 0-9660176-3-3. DOI: `10.1016/B978-0-7506-7444-7/50050-9`.

[171] C O S Sorzano et al. "A review of resolution measures and related aspects in 3D Electron Microscopy". In: *Progress in Biophysics and Molecular Biology* 124 (2017), pp. 1–30. ISSN: 0079-6107. DOI: `https://doi.org/10.1016/j.pbiomolbio.2016.09.005`. URL: `http://www.sciencedirect.com/science/article/pii/S0079610716300037`.

[172] Holger Stark. "Chapter Five - GraFix: Stabilization of Fragile Macromolecular Complexes for Single Particle Cryo-EM". In: *Cryo-EM Part A Sample Preparation and Data Collection*. Ed. by Grant J B T Methods in Enzymology Jensen. Vol. 481. Academic Press, 2010, pp. 109–126. ISBN: 0076-6879. DOI: `https://doi.org/10.1016/S0076-6879(10)81005-5`. URL: `http://www.sciencedirect.com/science/article/pii/S0076687910810055`.

[173] Holger Stark et al. “Large-Scale Movement of Elongation Factor G and Extensive Conformational Change of the Ribosome during Translocation”. In: *Cell* 100.3 (2000), pp. 301–309. ISSN: 0092-8674. DOI: `https://doi.org/10.1016/S0092-8674(00)80666-2`. URL: `http://www.sciencedirect.com/science/article/pii/S0092867400806662`.

[174] Holger Stark et al. “The 70S Escherichia coli ribosome at 23 å resolution: fitting the ribosomal RNA”. In: *Structure* 3.8 (1995), pp. 815–821. ISSN: 0969-2126. DOI: `https://doi.org/10.1016/S0969-2126(01)00216-7`. URL: `http://www.sciencedirect.com/science/article/pii/S0969212601002167`.

[175] J A Stark. “Adaptive image contrast enhancement using generalizations of histogram equalization”. In: *IEEE Transactions on Image Processing* 9.5 (2000), pp. 889–896. ISSN: 1057-7149 VO - 9. DOI: `10.1109/83.841534`.

[176] J. Steensgaard, N.P.R. Moller, and L. Funding. “Rate zonal centrifugation: Quantative aspects”. In: *Centrifugal Separations in Molecular and Cell Biology*. Ed. by G.B. Birnie and D. Rickwood. 1978.

[177] J Steensgaard, D Rickwood, and S Humphries. *COMPASS(TM) CENTRIFUGATION Software*. 1987.

[178] Jens Steensgaard and Niels Peter Hundahl Mller. “Computer Simulation of Density-Gradient Centrifugation”. In: (1979).

[179] Bjarne Stroustrup. *The C++ programming language*. 1988, pp. 1–288. ISBN: 0131158171.

[180] Tilo Strutz. *Data Fitting and Uncertainty- A practical introduction to weighted least squares and beyond*. Springer Vieweg, 2016. ISBN: 978-3-658-11455-8.

[181] Louis Sullivan. *The tall office building artistically considered*. 1896.

[182] Joel L. Sussman et al. “Protein Data Bank (PDB): Database of three-dimensional structural information of biological macromolecules”. In: *Acta Crystallographica Section D: Biological Crystallography* 54.6 I (1998), pp. 1078–1084. ISSN: 09074449. DOI: `10.1107/S0907444998009378`.

[183] Kenneth A. Taylor and Robert M. Glaeser. “Electron Microscopy of Frozen Hydrated Biological Specimens at the level of atomic resolution is a major”. In: *Journal of Ultrastructure Research* 456 (1976), pp. 448–456.

[184] Thermo Scientific. *Compass Centrifugation Software Instruction Manual*. 2014.

[185] Martin Thoma. “Analysis and Optimization of Convolutional Neural Network Architectures”. In: May (2017). arXiv: `1707.09725`. URL: `http://arxiv.org/abs/1707.09725`.

[186] P N T Unwin and R Henderson. “Molecular structure determination by electron microscopy of unstained crystalline specimens”. In: *Journal of Molecular Biology* 94.3 (1975), pp. 425–440. ISSN: 0022-2836. DOI: `https://doi.org/10.1016/0022-2836(75)90212-0`. URL: `http://www.sciencedirect.com/science/article/pii/0022283675902120`.

[187] Schaaf A. Van der and J. H. van Hateren. “Modelling the power spectra of natural images: statistics and information”. In: *Vision Res.* 36.17 (1996), pp. 2759–2770. URL: `pm:8917763`.

[188] M. Van Heel. “Three-dimensional reconstruction from projections with unknown an-gular relationships”. In: *Proceedings of the 8th European Congress on Electron Microscopy (Budapest)*. 1984, pp. 1347–1348.

[189] Marin Van Heel. "Angular reconstitution: A posteriori assignment of projection directions for 3D reconstruction". In: *Ultramicroscopy* 21.2 (1987), pp. 111–123. ISSN: 0304-3991. DOI: `https://doi.org/10.1016/0304-3991(87)90078-7`. URL: `http://www.sciencedirect.com/science/article/pii/0304399187900787`.

[190] Marin Van Heel et al. "A new generation of the IMAGIC image processing system". In: *Journal of Structural Biology* 116.1 (1996), pp. 17–24. ISSN: 10478477. DOI: `10.1006/jsbi.1996.0004`.

[191] K. E. Van Holde and Wolfgang O. Weischet. "Boundary analysis of sedimentatio velocity experiments with monodisperse and paucidisperse solutes". In: *Biopolymers* 17.6 (1978), pp. 1387–1403. ISSN: 10970282. DOI: `10.1002/bip.1978.360170602`.

[192] Charles Van Loan. *Computational Frameworks for the Fast Fourier Transform*. 1992.

[193] Moshe Y. Vardi. "Science has only two legs". In: *Communications of the ACM* 53.9 (2010), pp. 5–5. ISSN: 00010782. DOI: `10.1145/1810891.1810892`. URL: `http://portal.acm.org/citation.cfm?doid=1810891.1810892`.

[194] W.N. Venables and D.M. Smith. *An Introduction to R*. 2018, pp. 1–102. ISBN: 3900051127. DOI: `10.1016/B978-0-12-381308-4.00001-7`. arXiv: `arXiv:1011.1669v3`.

[195] Iris Vessey. "Problems Versus Solutions: The Role of the Application Domain in Software". In: *Papers Presented at the Seventh Workshop on Empirical Studies of Programmers*. ESP '97. New York, NY, USA: ACM, 1997, pp. 233–240. ISBN: 0-89791-992-0. DOI: `10.1145/266399.266419`. URL: `http://doi.acm.org/10.1145/266399.266419`.

[196] Lakshmi Sumitra Vijayachandran et al. "Robots, pipelines, polyproteins: Enabling multiprotein expression in prokaryotic and eukaryotic cells". In: *Journal of Structural Biology* 175.2 (2011), pp. 198–208. ISSN: 1047-8477. DOI: `https://doi.org/10.1016/j.jsb.2011.03.007`. URL: `http://www.sciencedirect.com/science/article/pii/S1047847711000724`.

[197] Feng Wang et al. "DeepPicker: A deep learning approach for fully automated particle picking in cryo-EM". In: *Journal of Structural Biology* 195.3 (2016), pp. 325–336. ISSN: 1047-8477. DOI: `https://doi.org/10.1016/j.jsb.2016.07.006`. URL: `http://www.sciencedirect.com/science/article/pii/S1047847716301472`.

[198] J. H. J. Ward. "Hierarchical Grouping to Optimize an Objective Function". In: *Journal of the American Statistical Association*. Vol. 58. 1963, pp. 236–244.

[199] Zeyi Wen et al. "Improving Efficiency of SVM k-fold Cross-validation by Alpha Seeding". In: ii (2016), pp. 2768–2774. arXiv: `1611.07659`. URL: `http://arxiv.org/abs/1611.07659`.

[200] H Wickham. *Package Dplyr*. 2018. DOI: `10.1093/biostatistics/1.4.465>`.

[201] Artur Wilin. "Gene selection for cancer classification". In: (2002), pp. 389–422. DOI: `10.1108/03321640910919020`.

[202] Greg Wilson et al. "Best Practices for Scientific Computing". In: *PLoS Biology* 12.1 (2014). ISSN: 15449173. DOI: `10.1371/journal.pbio.1001745`. arXiv: `arXiv:1210.0530v4`.

[203] Yifan Xiao and Guangwen Yang. "A fast method for particle picking in cryo-electron micrographs based on fast R-CNN". In: *AIP Conference Proceedings* 1836.1 (2017), p. 20080. ISSN: 0094-243X. DOI: `10.1063/1.4982020`. URL: `https://aip.scitation.org/doi/abs/10.1063/1.4982020`.

[204] Zong Xie, Qing Hu, and Da Yu. "Improved Feature Selection Algorithm Based on SVM and Correlation". In: *Advances in Neural Networks, Third International Symposium on Neural Networks* (2006), pp. 1373–1380.

[205] Katznelson Yitzhak. *An introduction to harmonic Analysis*. Dover, 1976.

[206] F Zemlin et al. "Coma-free alignment of high resolution electron microscopes with the aid of optical diffractograms". In: *Ultramicroscopy* 3 (1978), pp. 49–60. ISSN: 0304-3991. DOI: `https://doi.org/10.1016/S0304-3991(78)80006-0`. URL: `http://www.sciencedirect.com/science/article/pii/S0304399178800060`.

[207] Kai Zhang. *Gautomatch (unpublished)*. 2016. URL: `https://www.mrc-lmb.cam.ac.uk/kzhang/Gautomatch/`.

[208] Kai Zhang. "Gctf: Real-time CTF determination and correction". In: *Journal of Structural Biology* 193.1 (2016), pp. 1–12. ISSN: 10958657. DOI: `10.1016/j.jsb.2015.11.003`. URL: `http://dx.doi.org/10.1016/j.jsb.2015.11.003`.

[209] J. Zhao, M. A. Brubaker, and J. L. Rubinstein. "TMaCS: a hybrid template match- ing and classification system for partially-automated particle selection". In: *Journal of structural biology* 181.3 (2013), pp. 234–242.

[210] Shawn Q Zheng et al. "MotionCor2: anisotropic correction of beam-induced motion for improved cryo-electron microscopy". In: *Nature Methods* 14.4 (2017), pp. 332–333. ISSN: 15487105. DOI: `10.1038/nmeth.4193`. URL: `http://dx.doi.org/10.1038/nmeth.4193`.

Curriculum Vitae

Lukas Schulte

born 15th June 1988 in Eckernförde, Germany

Am Weißen Steine 3, 37085 Göttingen
Phone: +49 (0) 551 201-1410
E-Mail: lukasschulte@posteo.de

University education

06/2014 - 09/2018	Max Planck Institute for biophysical Chemistry, Göttingen: Doctoral studies within the doctoral degree programme "Biomolecules: Structure - Function - Dynamics" at the Georg-August University School of Science (GAUSS). PhD thesis: *New Computational Tools for Sample Purification and early-stage Data Processing in high-resolution cryo EM*
10/2010 - 10/2013	University of Lübeck, Germany: Studies in Molecular Life Science (M.Sc.). Master Thesis: *Studies of the Conformational Dynamics of the Adenylate Kinase under Magnesium Withdrawal* at the Institute of Physics, University of Lübeck
10/2007 - 10/2010	University of Lübeck, Germany: Studies in Molecular Life Science (B.Sc.). Bachelor Thesis: *Focussing on the Equilibrium: Single Molecule Spectroscopic Studies of the Conformational Dynamics of the Adenylate Kinase* at the Institute of Physics, University of Lübeck

---

School education

08/2005 - 07/2007	Abitur/A-levels at Secondary school Jungmannschule Eckernförde
07/2004 - 06/2005	Exchange Student at Valparaiso High School, Valparaiso, Indiana, USA

Research Experience

08/2011 - 12/2011	Student Researcher : Department of Cell Biology, Harvard Medical School, Boston, MA, USA, with Prof. Dr. Tom Walz: *Electron-crystallographic Studies on Aquaporin-0 Lipid Interactions and Development of automated Noise Detection Algorithms in spEM images*
12/2011 - 04/2012	Student Researcher : Department of Biochemistry, Brandeis University, Waltham, MA, USA, with Prof. Dr. Dorothee Kern: *Computational Statistical Phylogeny of a new Kind of Bacterial Kinases*
03/2006 - 04/2005	Student Researcher : Institute of Physics, University of Lübeck, Lübeck, Germany, with Prof. Dr. Christian Hübner: *New Algorithms for statistical Determination of very slow Conformational Rates with Single Molecule Spectroscopy*

Extracurricular Graduate School Activities

08/2015 - 08/2017	Student Representative for the doctoral degree programme "Biomolecules: Structure - Function - Dynamics"
08/2015 - 08/2016	Member of the Editorial Board for Layout and Media for the Student Newspaper GGNB Times
06/2015 and 06/2016	Student Organizer of the GGNB Summer Games