Computational Analysis of Spatial Point Patterns for Cell Organelles

Master Thesis
Department of Computer Science
at ETH Zurich

Markus Sutter
April 2010 - September 2010

Supervision:
Prof. Dr. Ivo F. Sbalzarini
Dr. Jo A. Helmuth

Institute of Theoretical Computer Science
MOSAIC Group
## Contents

1 Abstracts
   1.1 Abstract .................................................. 1
   1.2 Zusammenfassung .......................................... 1

2 Introduction
   2.1 Background ................................................. 3
   2.2 Problem Statement ......................................... 3
   2.3 Objectives .................................................. 4
   2.4 Related Work ............................................... 4
   2.5 Overview ................................................... 6

3 Introduction of the Interaction Model
   3.1 Input Definition: Discrete Objects ....................... 7
   3.2 Nearest-Neighbor Interaction: Distances ................. 8
      3.2.1 Classical Co-Localization Measure .................. 8
      3.2.2 The Cellular Context ................................ 9
   3.3 Interaction Model .......................................... 9
   3.4 Estimation of Interaction Potentials ..................... 11
      3.4.1 Maximum-Likelihood Estimation ...................... 11
      3.4.2 Maximum-a-Posteriori Estimation ................. 12
      3.4.3 Non-Parametric Potentials ......................... 12
   3.5 Test for Interaction ....................................... 13
      3.5.1 Test for Interactions with Parametric Potentials . 13
      3.5.2 Non-Parametric Test of Interaction ............... 14
# Implementation of the Interaction Analysis Plugin (IAP)

## 4.1 Input: Data Preparation
- **4.1.1 Images from Fluorescence Microscopy**
- **4.1.2 Matlab Files**

## 4.2 Distance Calculations: Nearest-Neighbor Distances and the State Density

## 4.3 Estimation of Interaction Potentials
- **4.3.1 Sampling-based Optimization: CMA-ES**

## 4.4 Hypothesis Testing
- **4.4.1 Test for Interaction with Parametric Potentials**
- **4.4.2 Non-Parametric Test for Interaction**

# Extensions

## 5.1 Spatially Heterogenous Interaction Processes
- **5.1.1 Motivation**
- **5.1.2 Integration in IAP**

## 5.2 Uncertainty in Measurements of Distances
- **5.2.1 Motivation**
- **5.2.2 Estimation of Measurement Uncertainty**
- **5.2.3 Integration in Interaction Model**
- **5.2.4 Integration in IAP**

## 5.3 K-Nearest Neighbors

## 5.4 Three Classes of Objects

# Conclusions & Outlook

## 6.1 Interaction Analysis Plugin for ImageJ

## 6.2 Extensions

## 6.3 Outlook

# Software Documentation

## A.1 User Tutorial
- **A.1.1 Requirements**
- **A.1.2 Installation**
- **A.1.3 Work Flow**

## A.2 Implementation
- **A.2.1 Prerequisites to a Programmer**
- **A.2.2 Library Dependencies**
- **A.2.3 Packages**
- **A.2.4 Important Classes**

# Acknowledgements
Chapter 1

Abstracts

1.1 Abstract

Intra-cellular structures interact in numerous direct and indirect ways to fulfill cellular functions. Identifying relations between intra-cellular structures is of major interest. Since such interaction are often reflected in correlations between the locations of the interacting objects, multi-color microscopy images of structures can be analyzed in order to search for unknown interactions. A simple and widely used type of such analysis is co-localization analysis. Object-based co-localization analysis has recently been extensively studied and generalized to a statistic approach to interaction analysis [14]. This thesis has two main objectives: First, access of the cell biology community to these technological advances needs to be provided. Second, the theoretical foundation of the interaction analysis should be extended in several directions in order to relax current limitations. This thesis reviews the current theoretical basis of the interaction analysis, addresses computational challenges, discusses the software structure and implementation details. It also shows a number of extensions of the theoretical basis, and demonstrates the software on typical experimental data.

1.2 Zusammenfassung

Details der Implementierung der entwickelten Software wird dargelegt. Schliesslich wird die neue Software an einem praxisnahen Datensatz demonstriert.
Chapter 2

Introduction

In computational cell biology, one is often concerned with determining whether there is a correlation between two sets of structures. A potential application considers quantifying spatial correlations between viruses and organelles of the endocytic system in order to shed light on the endocytic pathway taken by specific virus strains, i.e., whether viruses are co-located with a type of organelle. Fluorescent proteins allow specifically labeling cellular structures and different objects can hence be distinguished in image-based analyses. The overall goal of this thesis is to develop and implement algorithms that allow to determine whether there is a statistically significant spatial correlation between two or more types of objects.

In Section 2.1, we first give a briefly sketch of the considered field of computational biology. Then, in Sections 2.2 and 2.3 we formally state the problem and describe the primary objectives we intend to achieve. Finally, we sketch the structure of the thesis in Section 2.5.

2.1 Background

It is generally accepted that cellular function results from interactions of sub-cellular structures in numerous direct and indirect ways in space and time. The location and the function of sub-cellular structures are therefore closely related. Obtaining position information of objects in cells is crucial to understand their role in biological processes and thus has attracted great attention [21, 4]. While molecular interactions crucially depend on close spatial proximity, other interactions typically cause spatial correlations between interacting structures. Such correlations are the target of microscopy-based co-localization analysis, which can provide hints of potential interactions. Recent advances in fluorescent microscopy have enabled probing interactions in cells, either directly or indirectly. However, large biological data sets usually hinder manual analysis of co-localization, which is tedious and unreliable. Computational co-localization analysis can open the door to systematic co-localization analysis on this type of data and a range of computational methods have been proposed in the literature. In Section 2.4 we give a short review of the important methods, for a comprehensive review refer to Bolte [5].

2.2 Problem Statement

Helmuth et al. [14] generalized classical co-localization analysis to a statistical framework that links observed localization patterns of two groups of objects to the parameters of an interaction
model. The framework is based on Gibbs point processes modeling effective pairwise interactions. In the model, we assume that in the absence of interactions between the two classes of objects, the objects would behave 'randomly', i.e., as independent stochastic variables. This expected random localization pattern constitutes the null hypothesis and a statistical model for this behavior can be formulated. If, however, there is an interaction between the two classes of objects, we assume that the observed localization pattern deviates from the one specified under the null hypothesis. Inferred interactions thus reflect all effects that are not explained by the null hypothesis. This model enables the usage of a wealth of known statistical methods for analyzing experimental data, which have been successfully used in other scientific fields like ecology. In cell biology, however, the approach is not commonly applied.

In the above mentioned study, Helmuth et al. [14] have shown that for certain Gibbs point processes, the mathematics involved are fairly simple and the estimation of interaction parameters is feasible. These claims hold under the following assumptions:

- Only nearest neighbors interact
- Spatial homogeneity of the stochastic process
- Independence of objects within groups

### 2.3 Objectives

The first objective is to implement the currently available algorithms in ImageJ, in order to provide access for a large group of potential users in biological sciences. To simplify their analysis process, the whole work flow should be integrated into one tool. More precisely, the ImageJ plugin should take as input multi-color fluorescence microscopy images, extract the sets of positions of two groups of sub-cellular structures, and perform the interaction analysis on the position data. The ImageJ plugin should produce as output a statement about the significance of a potentially observed interaction and details about inferred interaction potentials. That requires that all intermediate steps are implemented compatible to each other. Nevertheless, the different steps should be modular so that exchanging them by other algorithms or implementations does not affect other parts of the pipeline.

The second objective is to identify to what extent the above assumptions can be relaxed and to develop the corresponding parameter estimation algorithms. This includes the analysis of spatially heterogenous processes, interactions with more than one neighbor, and making the model more general in order to allow analyzing and detecting additional interactions processes. Furthermore, an approach to handle with uncertain object positions, as frequently produced by object localization techniques, should be developed.

### 2.4 Related Work

Interactions can be directly or indirectly probed in fluorescence microscopy. The direct approach is based on experiments that generate a signal upon the close proximity required for molecular interactions (FRET, BRET) [10] [15] [20].

The indirect approach relies on independently imaging two populations of objects in the same context and searching for clues of interaction in their spatial distributions. Images are only an
incomplete representation of reality and attention must be paid to the way in which the spatial information is collected from the sample. One needs to keep in mind that the limits of resolution in optical microscopy imply uncertainty of the localization of small observed objects.

If sub-cellular structures interact, it is typically manifested through statistical dependencies in their spatial distributions. Presence or absence of significant co-localization, however, does not necessarily imply presence or absence of interactions. Co-localization depends on the specific interaction mechanism and spatial proximity could also be caused by the structure of the environment.

Nevertheless, co-localization as an indicator for many types of sub-cellular interactions is a powerful approach. The computational methods using this paradigm can be grouped into two classes: intensity-correlation-based and object-based methods.

**Intensity-correlation** (IC) methods use the intensity values of the different color channels from the images to calculate a correlation score (coefficient). The fundamental premise is that spatial proximity of the imaged objects is manifested in correlations between the intensities of the different color channels in individual pixels of the images. If the correlation score is large, it implies a high degree of co-localization. Commonly used IC coefficients include the Pearson’s coefficient [8], Manders’ coefficient [15], Li’s coefficient [17] and cross-correlation [24], all compared in Bolte [5]. A wealth of co-localization analysis software is available, mainly due to the relative ease of implementing the software, as they use basic image analysis tools. Worth mentioning is Just Another Colocalization Plugin (JACoP) [5] for ImageJ. It implements the above mentioned IC coefficient methods and has a well-arranged user interface.

**Object-based** approaches to infer interactions in the physical space. This makes the statistics used to describe the interactions more intuitive, compared to IC methods which work in pixel intensity space. As the object-based methods depend on the nature of the co-localization event and also the fluorescent signal, they depend on reliable methods to extract object information from images. The co-localization measures are build upon these discrete objects. Even though these co-localization measures, e.g., counting overlapping objects [5], are more intuitive then their IC counterparts, it remains unclear to what extent a positive co-localization implies the presence of interactions.
2.5 Overview

Figure 2.5.1: Thesis Overview

The theoretical Chapter 3 contains a review of the work of Helmuth et al. [14], in which they link co-localization to interactions and builds the foundation of the implementation of this interaction model in ImageJ, which is described in Chapter 4.

Chapter 5 contains extensions to the existing model that relax some of the assumptions stated above. A schematic overview of the main Chapters 3, 4, and 5 is given in Figure 2.5.1.
Chapter 3

Introduction of the Interaction Model

This chapter describes the basic interaction model. The basic interaction model uses correlations in the spatial distributions of two or more objects for statistical inference of interactions. This approach is based on independently measuring the locations of two populations of objects and then searching for clues of interactions in their spatial location patterns. In order to do this, one needs to define the input data. As the application of this algorithm is in all biology, the input is typically consists of two microscopy images, one from each population of interest, see in Section 3.1. As the model assumes that only nearest neighbors (NN) interact (between the two populations of interest), as a next step, the nearest neighbors have to be identified. As the strength of the interaction depends on the distance, the nearest neighbor distance has to be calculated. This is described in Section 3.2. Section 3.3 describes then how the interaction between the two objects can be measured, e.g., how one can distinguish between a random pattern and one potentially caused by an interaction. In Section 3.4 we then explain how the potential, i.e., the strength or shape of the interaction, can be estimated using various statistical methods, and finally, Section 3.5 lays out statistical tests for interaction. Novel extensions to the basic interaction model will be discussed in Chapter 5.

Figure 3.0.1: Overview Chapter 3

3.1 Input Definition: Discrete Objects

Object-based approaches to interaction analysis quantify spatial relationships between sets of discrete objects. This requires information about the positions of the objects in order to establish a spatial relationship measure. In cell biology, the input is typically a set of microscopy images. Therefore, an intermediate step is required to extract geometric information from the images and map it to discrete objects. This extraction process is called image segmentation in computer vision and a widely studied field. How errors in this step will affect the interaction analysis will
be discussed in Section 5.2. The spatial distributions of the discrete objects are used to establish a co-localization measure. A statistical model based on the general binary Gibbs process links co-localization measures to interaction.

3.2 Nearest-Neighbor Interaction: Distances

The general model is based on the assumption that only nearest neighbors interact with each other. Below, we give a brief description of how this nearest neighbors assumption is motivated.

3.2.1 Classical Co-Localization Measure

In our model, every object is represented by a feature vector $x_i$ that holds the object’s position information. For non-point-like objects, this includes also other geometric properties, for example the object’s shape and size. Co-localization measures can be constructed for two sets of objects: $X = \{x_i\}_{i=1}^N$ and $Y = \{y_j\}_{j=1}^M$. Without loss of generality we assume the set $Y$ as a given reference and are interested in the interaction of objects from $X$ with the objects in $Y$. All objects are located in a bounded region $\Omega \subset \mathbb{R}^n$ with dimensionality $n$ and boundary $\partial \Omega$, e.g., a cell and its cell membrane. In Section 4.1, we will present how this information can be extracted from images.

A simple measure to express the interaction between objects in two sets is the distance to the nearest neighbor in the other group. We formally define the nearest-neighbor distance for $x_i$ to its nearest neighbor in $Y$ as:

$$d_i = \min_j \{d(x_i, y_j)\},$$

with $d$ a distance function in feature space. One can argue that if this distance is smaller than a given threshold, the two involved objects interact in some way. A classical nearest neighbor distance co-localization measure $C^t$ is defined by counting the distances below a problem specific threshold $t$ as follows:

$$C^t = \frac{1}{N} \sum_{i=1}^N 1(d_i < t) \begin{cases} 1(\text{TRUE}) = 1 \\ 1(\text{FALSE}) = 0 \end{cases} \overset{N \to \infty}{\longrightarrow} \int_{-\infty}^t p(d)dd. \tag{3.2.2}$$

A nearest-neighbor distance distribution with density $p(d)$ establishes if $N \to \infty$ and can be estimated from the set of distances $D = \{d_i\}_{i=1}^N$. $p(d) \cdot \Delta d$ is the probability for observing a distance $d$ within an interval $\Delta d$ about $d$ in the given context as caused by the interaction process.

The absolute location and the orientation of the objects play no role, as we use only distances between objects in our analysis. We therefore assume that the analyzed interaction process is translation and rotation invariant. In the step that reduces images to discrete objects, we need to ensure that the extraction of the relative distances is accurate. Based on this distance, only two categories of positions of the objects in $X$ are distinguished: either they are sufficiently close to one object in $Y$ to be considered interacting, or they are not. Furthermore, we assume that objects in $X$ interact with at most one object in $Y$ and they do not experience the presence of any other object in $Y$, unless they leave the distance threshold $t$ and cross it by another object in $Y$. The choice of $t$ reflects an (implicit) assumption about the length scale of the interaction to be detected and should be supported by prior knowledge or be chosen in a systematic way. We will relax the assumption about a fixed threshold $t$ later by introducing continuous interaction potentials.
3.2.2 The Cellular Context

Inferring interactions from an observed co-localization measure $C^t$ is not trivial as $C^t > 0$ does not necessarily imply any interaction between the objects. It may be caused by other factors such as the cellular context \( \{ \Omega, Y \} \). Each object $x_i \in X$, interacting with the objects in $Y$, will end up somewhere in $\Omega$. The associated nearest neighbor distance $d_i$ depends on the interaction and the frequency with which this distance occurs in $\Omega$ with $Y$. The relative frequencies of possible distances, called the state density, takes the cellular context into account. We choose $Y$ as reference and can then define the state density, which is fully determined by $\Omega$ and $Y$:

$$q(d) = \lim_{\Delta d \to 0} \frac{\text{Prob}(d_i \in [d, d + \Delta d] \mid \text{"no interaction"}, Y)}{\Delta d}. \quad (3.2.3)$$

In case of an interaction, some of the possible distances are favored over others, deforming the density $q(d)$ to $p(d)$. The interaction information is contained in the deviation from the expected base-level in the absence of the interaction. We get the base-level $C_0^t$ by letting $p(d) = q(d)$ in Equation (3.2.2). That’s what we expect to observe under the null hypothesis $H_0$: “no interaction”. With a formal hypothesis test, one can then assess the significance of the deviations from the case of no interaction.

3.3 Interaction Model

If there is no interaction between the objects in $X$ and $Y$, all objects in $X$ would be distributed in $\Omega$ according to a stochastic process that is independent of the objects in $Y$. So any statistical dependence between the objects in the two sets is a result of some interaction and this leads to the following generic definition:

**Definition 1. Interaction**

Interaction is the collection of all effects that cause significant correlations between the positions of the objects in $X$ and $Y$.

We seek an interaction analysis tool that returns true if a significant interaction is present and false otherwise. Our approach is to define and calculate an interaction score and then test for significance using statistical tests. Ideally, such an interaction score is independent of the cellular context and reflects variations of the true interaction strength in a monotonous fashion.

The first step in the interaction analysis is to define an interaction model, in which an interaction strength with these properties is used. The presented analysis is derived from the general binary Gibbs process with a fixed number of objects, which is a standard model in spatial point pattern analysis. The central component of the Gibbs process is an effective pair-wise interaction potential $\Phi(\cdot)$. As stated in Definition 1, the term interaction is used as an abstraction for all effects causing an observed correlated pattern, rather than a specific physical interaction. Nevertheless, the mathematical form of the Gibbs process relates to physical models of interaction potentials, which associate an energy level with each pair $\{i, j\}$ of objects. The probability density, called Gibbs measure, of the Gibbs process for two sets of objects, $X$ and $Y$, has the shape of a Boltzmann distribution:

$$p(X, Y) \propto \exp \left( - \sum_{i=1}^{N} \sum_{j=1}^{M} \Phi(x_i, y_j) \right). \quad (3.3.1)$$
The Gibbs measure (3.3.1) implies mutual independence of the objects within the sets \( X \) and \( Y \). For nearest-neighbor interactions, the interaction potential between \( x_i \) and \( y_j \) can be defined as:

\[
\Phi(x_i, y_j) = \begin{cases} 
\phi(d_i) & \text{if } y_j \text{ is } \text{NN} \text{ of } x_i \\
0 & \text{else} \end{cases}, \tag{3.3.2}
\]

where the function \( \phi(d) \) specifies the strength and distance dependence of the interaction. The shape of the potential can be modeled parametrically or non-parametrically. A specific choice constitutes a hypothesis or assumption about properties of the interaction, such as its range, and should be based on prior knowledge. The strength, shape, and range are independently represented in the following parametric model:

\[
\phi(d) = \epsilon \cdot f \left( \frac{d - t}{\sigma} \right). \tag{3.3.3}
\]

The parameter \( \epsilon \) is the interaction strength, \( f \) encodes the functional shape, \( \sigma \) defines the length scale (range), and \( t \) represents a shift along the distance axis of the interaction potential.

Let’s assume a cellular context \( \{\Omega, Y\} \) for the objects \( X \) is given. The probability \( p(X|\Omega, Y) \) for a potential defined in Equation (3.3.2) only depends on the nearest-neighbor distances \( D = \{d_i\}_{i=1}^N \), so \( p(X|\Omega, Y) \) could be written as \( p(D|\Omega, Y) \). An inner sum over all \( j \) as in Equation (3.3.1) is then not required, as all summands are equal to zero, except one. The mutual independence within \( X \) allows factorizing \( p(X|\Omega, Y) \) into terms that only depend on a single \( d_i \), each:

\[
p(X|\Omega, Y) = \prod_{i=1}^N p(x_i|\Omega, Y) \propto \prod_{i=1}^N \exp(-\phi(d_i)), \tag{3.3.4}
\]

where an explicit dependence of the potential on \( x_i \) is hidden by the nearest-neighbor distance \( d_i \). The probability \( p(x_i|\Omega, Y) \) of observing a certain \( x_i \) is proportional to \( \exp(-\phi(d_i)) \). That is because the Gibbs density only depends on the distance \( d_i \) associated with the object location \( x_i \), as a consequence of the definition of the potential (3.3.2) as nearest-neighbor potential.

The probability of observing a certain \( d \), however, also depends on how frequently an arbitrary object is found at any location \( x \) in the domain \( \Omega \) that is a distance \( d \) away from the nearest object in \( Y \). This frequency is given by the state density \( q(d) \) as stated in Equation (3.2.3). Straightforward calculations yield:

\[
p(d|\Omega, Y) = p(d|q) = Z^{-1} q(d) \exp(-\phi(d)), \tag{3.3.5}
\]

where \( Z \) is the normalization constant, called partition function, that renders \( p(d|q) \) a true probability density function. \( Z \) is defined by an integral over the range of all possible distances \([d_{\text{min}}, d_{\text{max}}]\) in the domain \( \Omega \):

\[
Z = \int_{d_{\text{min}}}^{d_{\text{max}}} q(d) \exp(-\phi(d))dd. \tag{3.3.6}
\]

For general Gibbs processes, parameter estimation is very involved, since the computation of the normalization constant \( Z \) requires evaluating a high-dimensional integral. As we only consider nearest-neighbor interactions, computing \( Z \) becomes feasible: \( Z \) is obtained by an one-dimensional (numerical) integration. This easily allows evaluating \( p(d) \). Thus, standard estimation techniques can be used in the present framework.
Definition 2. Interaction Model
The joint probability density of observations \( D = \{d_i\}^{N}_{i=1} \) can be specified as:

\[
p(D|q) = Z^{-N} \prod_{i=1}^{N} q(d_i) \exp(-\phi(d_i)) = Z^{-N} \prod_{i=1}^{N} q(d_i) \exp \left( -\epsilon \cdot f \left( \frac{d_i - t}{\sigma} \right) \right), \tag{3.3.7}
\]

where Equations (3.3.4) and (3.3.5) were used in the first step and Equation (3.3.3) in the second.

This is the central class of models used in this thesis for interaction analysis, which build as an extension of classical co-localization measures.

3.4 Estimation of Interaction Potentials

Now that we have defined our interaction model, the following questions arise: How can we fit the model to the given data and how can we determine if the interaction is significant?

3.4.1 Maximum-Likelihood Estimation

Let’s assume that the object data \( X = \{x_i\}^{N}_{i=1} \) and \( Y = \{y_j\}^{M}_{j=1} \) and the dependent quantities \( q(d) \) and \( D \) are given. We then try to find model parameters that maximize the probability that the present sample is observed. The method of maximum-likelihood [3] aims exactly at that: One tries to maximize the likelihood of the unknown parameters given the data. To simplify the calculation, we use the log-likelihood instead of the likelihood. This reformulation can be done without loss of generality as the logarithm is a monotonic function and the optima of a function and the logarithm of the function thus coincide. The log-likelihood of the parameters \( \Theta = \{\epsilon, \sigma, t\} \) of a given potential shape \( \phi \), given the observations \( D \) and the context \( q(d) \), is the logarithm of the joint probability density (3.3.7):

\[
l(\Theta|D, q, \phi) = \log \left( \prod_{i=1}^{N} p_\phi(d_i|q) \right) = -N \log(Z(\Theta)) + \sum_{i=1}^{N} \log(q(d_i)) - \phi(d_i; \Theta). \tag{3.4.1}
\]

The maximum-likelihood estimator (MLE) is defined as:

\[
\hat{\Theta}_{\text{MLE}} = \arg \max_{\Theta} l(\Theta|D, q, \phi). \tag{3.4.2}
\]

For the present interaction model, no general analytical solution for \( \hat{\Theta}_{\text{MLE}} \) can be found and numerical optimization techniques need to be used. These can be stochastic [11] or deterministic (e.g., gradient-based) [19] strategies.

Because the state density \( q(d) \) exists in many functional forms and is usually not known analytically, the normalization constant \( Z \) needs to be computed by numerical integration. As long as the interaction is not too strong, \( q(d) \exp(-\phi(d)) \) is a well-behaved function and basic numerical integration schemes are sufficiently robust and accurate.

The sum \( \sum_{i=1}^{N} \log(q(d_i)) \) of the log-state density valued at the locations of the data in Equation (3.4.1) does not have to be evaluated for parameter estimation (subscript “p.e.”), as it is not
a function of the unknown parameters $\Theta$.

\[ l_{p.e.}(\Theta) = -N \log(Z(\Theta)) - \sum_{i=1}^{N} \phi(d_i; \Theta), \]  
\[ \hat{\Theta}_{MLE} = \arg \max_{\Theta} l(\Theta) = \arg \max_{\Theta} l_{p.e.}(\Theta). \]  

So $l_{p.e.}(\Theta)$ can be used for parameter estimation instead of $l(\Theta)$. Nevertheless, $q(d)$ enters the likelihood through the normalization constant $Z$.

### 3.4.2 Maximum-a-Posteriori Estimation

Maximum-a-posteriori (MAP) estimators $\hat{\Theta}_{MAP}$ allow stabilizing parameter estimation in case of few data points or poor parameter determinability. The idea is to specify a prior $pr(\Theta)$ on the unknown parameters $\Theta$ and maximizing the posterior distribution:

\[ \hat{\Theta}_{MAP} = \arg \max_{\Theta} \frac{p(D|\Theta)pr(\Theta)}{\int_{\Theta} p(D|\Theta')pr(\Theta')d\Theta'} = \arg \max_{\Theta} p(D|\Theta)pr(\Theta) \]  
\[ \]  

instead of the likelihood defined in Section 3.4.1. This estimator can equivalently be written using the log-likelihood:

\[ \hat{\Theta}_{MAP} = \arg \max_{\Theta} \{ l(\Theta) + \log(pr(\Theta)) \}. \]

For certain models of interaction potentials, the parameter values are naturally bounded. But some potentials can have bad parameter determinability in limit cases of the range of the parameters. The Plummer potential, for example, yields a step function for $\sigma \to 0$. In such a situation, a simpler step function should be used instead. A prior on $\Theta$ can be used to avoid problems in such limit cases. In the example of the Plummer potential the prior can ensure a positive value for $\sigma$.

### 3.4.3 Non-Parametric Potentials

Maximum-a-posteriori estimation can also be used to control the smoothness of non-parametric estimates of interaction potential. A piece-wise linear (non-parametric) potential $\phi_{n.p.}(d)$ can be defined as a weighted sum of kernel functions $\kappa(\cdot)$ centered on $P$ support points $d_p$:

\[ \phi_{n.p.}(d) = \sum_{p=1}^{P} w_p \kappa(d - d_p) \]  
\[ \]  
\[ \text{with } \kappa(d') = \begin{cases} 
\frac{|d'|}{h} & \text{if } |d'| < h \\
0 & \text{else}
\end{cases} \]  
\[ \]  

where $h > 0$ denotes the constant spacing between support points $\{d_p\}_{p=1}^{P}$. Setting $w_P = 0$ enforces that the potential is zero at infinity. Using $\phi := \phi_{n.p.}$, the remaining weights can be estimated by numerically maximizing the penalized log-likelihood $l_{pen.}(\Theta)$:

\[ l_{pen.}(\Theta) = l(\Theta) + \sum_{p=1}^{P-1} \frac{\left( w_p - w_{p+1} \right)^2}{w_p - w_{p+1}} \log(pr(\Theta)), \]

\[ \]
with respect to $\Theta = (w_1, \ldots, w_{P-1})$. The quadratic penalty on the differences $\Delta w_p = w_p - w_{p+1}$ corresponds to the assumption of a Gaussian prior with zero mean and standard deviation $s$ on the differences (see Equation (3.4.5)):

$$
pr(\Theta) = pr(\Delta w_p) = \frac{1}{\sqrt{2\pi s^2}} \exp \left( -\frac{(\Delta w_p)^2}{2s^2} \right).
$$

The smoothness of $\phi_{u,p}$ is controlled by the parameter $s$, which is the standard deviation of the Gaussian prior. The larger $s$ the smoother the estimated potential. Among all potentials of similar global shape, the prior favors the one that has the least deviations of the weights from the global trend. This is the only weak assumption on the shape of the potential. This allows detecting structures in the data that would be missed with more restrictive potential shapes.

### 3.5 Test for Interaction

In Section 3.4.1 we assumed that a potential shape $\phi$ is given and we explored how the parameters could be estimated. In this Section, we describe how these estimated parameters should be interpreted and what can be done if no knowledge about the potential shape is available.

If a potential shape is given and its parameters are estimated, the statistical test described in Section 3.5.1 can be used to check for interaction, using the estimated parameters. If no prior knowledge about a potential is available, it may be beneficial to first perform a non-parametric hypothesis test for the presence of an interaction, described in Section 3.5.2, before attempting to fit an interaction potential. If the potential shape is unknown, Maximum-a-Posteriori estimators can be used to create non-parametric estimates of the interaction potential, as it was discussed in Section 3.4.3.

#### 3.5.1 Test for Interactions with Parametric Potentials

In the parametization of the interaction model in Equation (3.3.7), the presence of an interaction is equivalent to $\epsilon \neq 0$. Whether an observed estimate $\hat{\epsilon}$ is indicative of the actual presence of an interaction, however, has to be evaluated with statistical tests. Because the estimator $\hat{\epsilon}$ is a function of random variables in $D$, it is a random variable itself. Even if the hypothesis $H_0$: “no interaction” is true, a non-zero $\hat{\epsilon}$ may thus occur with finite probability. In other words, $\hat{\epsilon} \neq 0$ does not imply $\epsilon \neq 0$.

Inference about interactions requires finding a estimated critical interaction strength above which one can reject $H_0$ on a prescribed significance level $\alpha$. This critical interaction strength depends on the distribution (null distribution) of $\hat{\epsilon}$ under the null hypothesis $H_0$, which in turn depends on the sample size $N$, the state density $q$, and the prescribed significance level $\alpha$. For step potentials, a test statistic can be constructed based on the fact that the observed number of co-localized objects follows a binomial distribution. Helmuth et al. [14] show the derivation of a test statistic and a power analysis for the step potential. For general potentials, the null distribution is not explicitly known and such a reasoning is no longer valid. Nevertheless, tests for the presence of interaction can be constructed using different statistics.

The form of our interaction model (3.3.7) is a member of the exponential family. Therefore we have with $T$ a sufficient statistic for $\epsilon$:

$$
T = - \sum_{i=1}^{N} f \left( \frac{d_i - t}{\sigma} \right).
$$

(3.5.1)
**Definition 3. Sufficient Statistic**

A sufficient statistic is a function of the data that contains all the information which is in the data \[25\].

If we choose a sufficient statistic out of all possible statistics \(T = r(D)\) (for any function \(r(\cdot)\)), we have a statistic that carries all information available in \(D\) about the unknown strength \(\epsilon\) of the given interaction potential. For concluding something about the strength \(\epsilon\), knowing one sufficient statistic \(T\) is thus as good as knowing any other sufficient statistic or even the entire sample \(D\), a very powerful concept.

**Theorem 4. Factorization Theorem**

A statistic \(T = r(D)\) is sufficient if and only if the joint density \(p(D|\epsilon)\) of the observation \(D\) can be factorized into two non-negative functions \(u\) and \(v\) as:

\[
p(D|\epsilon) = u(D) \cdot v(T(D), \epsilon). \quad (3.5.2)
\]

The function \(u\) may depend on the full sample \(D\), but not on \(\epsilon\), while \(v\) can depend on \(\epsilon\), but the dependence on the data must only be through the value of the statistic \(T\) \[25\].

The joint probability \((3.3.7)\) can be re-written as:

\[
p(D, q) = \left( \prod_{i=1}^{N} q(d_i) \right) \cdot \left( Z(\epsilon)^{-N} \exp \left( -\epsilon \sum_{i=1}^{N} f \left( \frac{d_i - t}{\sigma} \right) \right) \right) \quad (3.5.3)
\]

which proves that \(T\) defined in Equation \((3.5.1)\) is a sufficient statistic for \(\epsilon\) in any potential parameterized as in Equation \((3.3.3)\).

A test for the presence of interactions can thus be constructed based on the distribution of \(T\) under \(H_0: \text{“no interaction”}\). We will discuss how to construct a test in Section \(4.4.1\).

The present interaction analysis framework allows testing using different potentials. In this model selection process, potentials with different shapes and numbers of parameters are fitted independently. The best potential can then be selected according to, for example, the Akaike or Bayesian information criterion, or the result of a likelihood ratio test \[6\].

**3.5.2 Non-Parametric Test of Interaction**

If no prior knowledge about an interaction potential is available, non-parametric tests can be designed without requiring a specific potential. As no potential is available, no interaction strength \(\epsilon\) value is available that can be used a test statistic. To test for significant deviation of the distance distribution \(p(D)\) from the state density \(q(d)\), the following distance counts can be used \[2\]:

\[
T = (T_1, \ldots, T_L)^t \quad \text{with} \quad T_l = \sum_{i=1}^{N} 1(t_l < d_i \leq t_{l+1}). \quad (3.5.4)
\]
The number of observed distances are counted in $L$ equi-sized bins defined by $L+1$ strictly increasing thresholds $t_l$ that span the entire non-zero range of $q(d)$ for a given context $\Omega$. Using these distance counts amounts to implicitly assuming that the potential is a piece-wise constant function. The lower the value of the potential in a given bin $l$, the higher the expected number of counts $T_l$. Hypothesis $H_0$: “no interaction” is equivalent to the potential being zero over the whole range of $d$ or, in other words, zero in all bins. The expected number of counts is then proportional to the integral of $q(d)$ over the bin considered. A deviation from the expected values of counts suggests that the true, but unknown, potential is non-zero in the regions spanned by the corresponding bins. As distance counts within different bins will be anti-correlated (if there are more counts in one bin, there have to be less in others), care must be taken not to over-estimate the significance of the collective deviation of multiple bins.

A rank-based non-parametric hypothesis test is constructed using the distance counts $T$. First, a Monte Carlo sample $\{T_k\}_{k=1}^K$ from the null distribution of $T$ is obtained by sampling $N$ distances $d_i$ from $q(d)$ using the inversion method and repeating this sampling procedure $K$ times. $N = |D|$ refers to the number of observations in $D$ that are subject to the test. The sample $\{T_k\}_{k=1}^K$ allows approximating the expectation $E_0(T)$ and co-variance $\text{Cov}_0(T)$ of the null distribution of the distance counts. The final test statistic $U$ is defined as:

$$U = (E_0(T) - T)^T \text{Cov}_0(T)^{-1} (E_0(T) - T)$$ (3.5.5)

As in Section 3.5.1 a rank-based test is used to judge the significance of the test statistic $U$. This requires that a ranking set $\{U_k\}_{k=1}^K$ is sampled from the null distribution. Additional Monte Carlo samples $\{T_k\}_{k=1}^K$ are generated as described above for expectation and co-variance. The ranking set $\{U_k\}_{k=1}^K$ is computed using these samples.

In the next step, $T_D$ and $U_D$ are computed from the set $D$ of experimentally observed distances that are subject to test. The observed $U_D$ is then ranked among the $\{(1-\alpha)K\}$-th test statistic in $\{U_k\}_{k=1}^K$. If it ranks higher than the $\{(1-\alpha)K\}$-th test statistic in $\{U_k\}_{k=1}^K$, $H_0$ is rejected on the significance level $\alpha$.

The number of bins $L$ influences the performance of the test. For $L = 2$, a test is recovered that relates to a parametric test with a step potential. Increasing $L$ allows resolving finer details in the structure of the observed distance distribution, and therefore provides the possibility to detect several types of deviation from $H_0$. If the true potential strongly differs from a step-like shape, a test with $L > 2$ will have more power. Too large values of $L$, however, reduce the statistical power of the test, as the expected number of the distances in a given bin will then be very low and only large deviations from the expectation will be significant.
Chapter 4

Implementation of the Interaction Analysis Plugin (IAP)

One important goal of this master thesis is to make the theoretical framework for quantifying interactions between intra-cellular objects accessible to biologists as a user-friendly tool. Nowadays, biologists can generate large amounts of digital image data and therefore, they need tools supporting them in their analysis. To get the most value out of analysis tools, they have to integrate smoothly into the workflow of the users and require only a small amount of introductory training. Furthermore, software needs to run fast. We have chosen ImageJ [1] as environment for our interaction analysis tool. We decided to implement it as a plugin for ImageJ, as it is a public domain image processing program, uses modern programming paradigms, has a broad user and developer community, runs on any platform, and is widely used in the cell biology community.

4.1 Input: Data Preparation

4.1.1 Images from Fluorescence Microscopy

The analysis in Section 3.1 is based on two sets of objects: $X = \{x_i\}_{i=1}^N$ and $Y = \{y_j\}_{j=1}^M$. Each object is represented by a feature vector $x_i$ or $y_j$, which holds the object’s position information. All objects are located in a bounded region $\Omega \subset \mathbb{R}^n$ with dimensionality $n$ and boundary $\partial \Omega$. To start the interaction analysis with our model, we first have to extract this information from the images.

For this purpose, the feature point detection algorithm of Sbalzarini et al. [22] is used. This algorithm relies on a minimum set of assumptions and a small set of user-defined parameters. It is
fast and efficient, while at the same time achieving an accuracy and precision that is comparable to more computationally intensive algorithms. It is therefore well-suited for our task. The feature point detection process consists of four steps:

**Feature Point Detection**

1. Image restoration
2. Estimation of point locations
3. Refinement of the point locations

The feature point detection algorithm is already implemented for ImageJ in the Particle Tracker plugin [22]. In this plugin point detection was closely coupled with point tracking and trajectory analysis part, therefore refactoring of the code to a more modular architecture was necessary to use the detection part alone.

**Domain Definition**

In addition to the object’s position, the context region $\Omega$ and its boundary $\partial \Omega$ have to be determined. The image dimensions give a bounding box for the domain, but often, only a section of the image belongs to $\Omega$. Hence, the user can define a (binary) mask or generate one based on the reference image. The automatic mask generation uses a macro which first adapts the contrast, then it applies a filter with Gaussian blur ($\sigma = 15$) before it converts the image to a binary mask. The IAP only uses the data within the mask.

**Chromatic Aberration**

As the IAP is especially targeted to analyze fluorescence microscopy images, it contains a position correction step to handle chromatic aberration [16]. Chromatic aberrations have to be considered in studies based on measurements of short distances between targets that are visualized using two different fluorescent stains: The best optical microscopes available today exhibit chromatic aberrations on the order of 100's nanometer. The correction of chromatic aberration is performed on the measured object positions with user-defined calibration function. Two independent linear models are used for this aberration correction for the vertical and horizontal dimension of the images. Especially for the estimation of the calibration function, a Chromatic Aberration (CA) plugins was developed. It takes the images of a dual-color beads as input and fits the parameter of the linear model to this input data. See Figure 4.1.1 for a visualization of chromatic aberration and Figure 4.1.2 shows the corresponding fit. The user can then use these estimates as input to the chromatic aberration correction step of IAP. During the correction, the first color/image is selected as reference and left intact whereas the second is transformed to match the reference.

**4.1.2 Matlab Files**

Users which already have the necessary input data available, i.e. they already have discrete object positions, can omit the first step of the procedure and directly use the analysis part of the IAP. If the input data is available as a Matlab file, this file can be imported into IAP and the raw object positions can be used for the interaction analysis without any modification.
Figure 4.1.1: Chromatic Aberration Shifts Visualized in CA

Figure 4.1.2: Linear Model for Chromatic Aberration Shifts in CA
4.2 Distance Calculations: Nearest-Neighbor Distances and the State Density

Given the objects $X = \{x_i\}_{i=1}^N$ and $Y = \{y_j\}_{j=1}^M$, the next step in the interaction analysis is to calculate the nearest-neighbor distances $D$ and the state density $q(d)$. For computing $D$, as well as for estimating $q(d)$, we need an algorithm that takes a distance function $d(x_i, y_j)$ and the objects in $X$ and $Y$ as inputs and returns for each object $x_i$ in $X$ the measured distance $d_i$ to it’s nearest neighbor in $Y$. Donald Knuth called it the “post-office problem” in 1973, referring to the application of assigning a residence to the nearest post office. The Voroni Diagramm in Figure 4.2.1 is a visualization of the problem. Nowadays, it is better known by its straightforward name, “nearest-neighbor search”.

![Figure 4.2.1: Voroni Diagramm: Nearest-Neighbor Domains](image)

The computational complexity of the naïve approach of measuring the distance from object $x_i$ to every object in $Y$ and then choose the smallest one is linear in the number of elements in $Y$ and has to be done for every object in $X$. The runtime is therefore $O(NM)$ with $N = |X|$ and $M = |Y|$ being the number of elements in the sets $X$ and $Y$, respectively.

To achieve better performance, a specialized data structure is used in IAP. A kd-tree is a space-partitioning data structure which iteratively bisects the search space into two regions containing half of the points of the parent region. Queries are then performed via traversal of the kd-tree from top to bottom and checking at each node, which subtree is closer to the query. For every object in $X$, the kd-tree is traversed once, so the total query time is in $O(N \log(M))$. The construction of a balanced kd-tree is in $O(M \log(M))$ because of the involved sorting step. As we also have to query nearest-neighbor distances to the objects $Y$ for calculating $q(d)$, it makes sense to invest some computational time in the beginning (for the construction of the tree) to get faster querying later.

Given knowledge about both $Y$ and $\Omega$, the state density $q(d)$ can be determined. We are using a straightforward sampling method, in which we sample exhaustively positions $z_k$ in $\Omega$ on a uniform Cartesian grid. The number of samples $Q = |Z|$ can be higher than the number of pixels of the images from which the objects $Y$ were extracted, which means that the spacing $h$ between grid points can be smaller than the size of a pixel. For each $z_k$, the distance $d_k$ to the nearest neighbor in $Y$ is then computed using the same kd-tree as before. If we sort this finite sample of distances $D_z = \{d_k\}_k$ into equally sized bins and normalize by the number of samples and bin size, we get a simple, but discontinuous approximation for $q(d)$. To get a smoother approximation, which
also converges faster to the true density than this histogram estimator, we use a kernel density estimator as explained in Wasserman [25]. So $D_z = \{d_k\}_k$ denotes the observed data, a sample from $q(d)$.

**Definition 5.** Given a kernel $K$ and a positive number $h$, called the smoothing bandwidth, the **kernel density estimator** is defined to be:

$$\hat{q}_Q(d) = \frac{1}{Qh} \sum_{k=1}^{Q} K \left( \frac{d - d_k}{h} \right).$$

(4.2.1)

At each point $d$, the kernel density estimator $\hat{q}_Q(d)$ is the average of the kernels centered on the data points $d_k$. We use the Gaussian (Normal) kernel $K(d) = \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{d^2}{2} \right)$. But a kernel could be any smooth function $K(d)$ that satisfies $K(d) \geq 0$, $\int K(d)dd = 1$, $\int dK(d)dd = 0$, and $\int d^2K(d)dd > 0$.

### 4.3 Estimation of Interaction Potentials

After having prepared the input data (Section 4.1) and calculated the nearest neighbor distances (Section 4.2), we fit our model to the data and estimate the model parameters. As discussed in Section 3.4.1 we find the parameters of the model with the maximum-likelihood method. The idea of this method is to find a value $\Theta$, such that $l(\Theta)$ is as large as possible for a given $l(\Theta)$ – a classical optimization problem. Depending on whether $\Theta$ is a scalar or a vector of discrete or continuous values, the possible constraints on $\Theta$, the objective function $l(\Theta)$, and how much information (e.g. gradient, convexity) about it is available, many different optimization strategies exist.

Recall that we can use $l_{p.e.}(\Theta)$ instead of the real likelihood $l(\Theta)$ for parameter estimation (see Section 3.4.1):

$$l_{p.e.}(\Theta) = -N \log(Z(\Theta)) - \sum_i \phi(\Theta; d_i)$$

In the present context, $\Theta$ is a vector of continuous values and the objective function $l_{p.e.}(\Theta)$ is continuous. Therefore sampling-based stochastic as well as gradient-based deterministic algorithms can be used. The latter requires to differentiate $l_{p.e.}(\Theta)$ with respect to $\Theta$, which requires, among other things, that the interaction potential $\Phi$ can be differentiated with respect to $\Theta$:

$$\frac{\partial l_{p.e.}(\Theta)}{\partial \Theta} = -N \frac{1}{Z(\Theta)} \frac{\partial Z(\Theta)}{\partial \Theta} + \sum_i \frac{\partial \phi(\Theta; d_i)}{\partial \Theta}.$$  

(4.3.1)

As we can see in Equation (4.3.1), the partial derivative of $Z(\Theta)$ with respect to $\Theta$ is required:

$$\frac{\partial Z(\Theta)}{\partial \Theta} = \int \frac{\partial}{\partial \Theta} q(d) \exp(-\phi(\Theta))dd$$

$$= \int q(d) \left( -\frac{\partial \phi(\Theta)}{\partial \Theta} \right) \exp(-\phi(\Theta))dd.$$  

(4.3.2)

The evaluation of the gradient of $l_{p.e.}$ is computational expensive. It requires two numerical integrations: one to compute $Z(\Theta)$ and another one for $\frac{\partial Z(\Theta)}{\partial \Theta}$. As such evaluations occur frequently in gradient-based optimization algorithms, these algorithm may not be the best choice.
4.3.1 Sampling-based Optimization: CMA-ES

A special group of sampling-based optimization algorithms are called evolutionary optimization. Candidate solutions to the optimization problem play the role of individuals in a population and the objective (fitness) function determines their chance to survive. The population evolves through mechanisms inspired by biology such as mutation, reproduction, recombination and selection. After several generations, the fittest individual is taken as the solution for the optimization problem.

A popular evolutionary optimization algorithm is Covariance Matrix Adaptation Evolution Strategy (CMA-ES) \cite{11}, it is more involved than the biological evolution operations and has a very high level of maturity. New individuals are sampled from a multivariate normal distribution. The covariance matrix of the distribution is updated during optimization. This adaption is based on two main principles: Firstly, a maximum likelihood principle is used to increase the probability of successful individuals. Secondly, the information gained from the recording of the time evolution of the distribution mean is included in the adaption and also used for step-size control.

CMA-ES optimization suits our needs, as it requires only a black box fitness function and is quasi parameter-free. Even if second-order-derivative-based methods may outperform it in some situation, for rugged search landscapes with local optima CMA-ES is the favorable choice. It could be applied to high dimensional search spaces with dimensions up to 100, for non-parametric potential estimation we normally have a search space with about 20 dimension.

4.4 Hypothesis Testing

As described in Section 3.5, depending on the prior knowledge about the interaction in question and its corresponding potential, different statistical test can be performed to test for the presence of an interaction.

Under the assumption that the shape of the interaction potential as well as the corresponding parameters \( \sigma \) and \( \tau \) (e.g., estimated from an experiment) are given, a test with high statistical detection power can be performed. The theoretical discussion about this test is explained in Section 3.5.1 and the design details are described in Section 4.4.1.

In many applications, no prior knowledge about the interaction potential is available. In this case, non-parametric tests can be used that do not require assuming a specific potential (see Section 3.5.2). Design details of the non-parametric test used in IAP follow in Section 4.4.2.

4.4.1 Test for Interaction with Parametric Potentials

The significance of a test statistic \( T \) \cite{3.5.1} is assessed in a rank-based test using independent and identically distributed (i.i.d) Monte Carlo samples. To perform this rank test, the potential shape \( f(\cdot) \), parameters \( \sigma \) and \( \tau \), and Monte Carlo samples from \( q(d) \) are required to calculate the test statistic \( T \) of the data and samples \( \{T_k\}_{k=1}^K \) from the null distribution of \( T \).

The Monte Carlo samples of \( q(d) \) are created using the inverse transform sampling procedure \cite{9} with the cumulative distribution function \( Q \) of \( q(d) \). The idea is that given a continuous uniform variable \( R \in [0, 1] \) and the cumulative distribution function \( Q \), the random variable \( X = Q^{-1}(R) \)
has the probability density \( q(d) \). As \( Q^{-1} \) is not given as an analytical expression, an empirical cumulative distribution created from \( q(d) \) is inverted instead. Piecewise linear interpolation is used to get a smooth approximation of the inverse cumulative distribution \( Q^{-1} \).

The test statistic samples \( \{T_k\}_{k=1}^K \) are obtained by sampling \( N = |D| \) distances \( d_i \) from \( q(d) \) using \( Q^{-1} \), computing \( T_k \), and repeating this procedure \( K \) times. Given these samples \( \{T_k\}_{k=1}^K \), the significance of the deviation of the test statistic \( T \) from the null distribution is assessed using a rank-based test. The observed value of \( T \) is ranked among the sorted \( \{T_k\}_{k=1}^K \). If and only if it ranks higher than \( \lceil (1 - \alpha)K \rceil \)-th statistic, \( H_0 \) is rejected on the significance level \( \alpha \). This test can be rendered arbitrarily accurate by increasing the number of Monte Carlo samples \( K \).

### 4.4.2 Non-Parametric Test for Interaction

The test statistic \( U \) used in the non-parametric rank test is defined as in Equation (3.5.5):

\[
U = (E_0(T) - T)^t \text{Cov}_0(T)^{-1}(E_0(T) - T). \tag{4.4.1}
\]

\( U \) is constructed using the distance counts:

\[
T = (T_1, \ldots, T_L)^t \quad \text{with} \quad T_i = \sum_{i=1}^{N} 1(t_i < d_i \leq t_{i+1}) \tag{4.4.2}
\]

in \( L \) equi-sized bins defined by \( L + 1 \) strictly increasing thresholds \( t_i \) that span the entire non-zero range of \( q(d) \).

A Monte Carlo sample \( \{T_k\}_{k=1}^K \) from the null distribution of \( T \) is obtained by sampling \( K \) times \( N = |D| \) distances \( d_i \) from \( q(d) \) and computing \( T_k \) by counting. The same inversion sampling method as for the parametric test in Section 4.4.1 is used. The samples \( \{T_k\}_{k=1}^K \) allow approximating the expectation \( E_0(T) \) and co-variance \( \text{Cov}_0(T) \) of the null distribution. A set \( \{U_k\}_{k=1}^K \) obtained from an additional Monte Carlo sample \( \{T_k\}_{k=1}^K \) is generated from the null distribution as described for the first Monte Carlo sample. Calculating \( U \), the expectation \( E_0(T) \), the co-variance \( \text{Cov}_0(T) \) and the inverse co-variance matrix \( \text{Cov}_0(T)^{-1} \) are basic linear algebra operations. For that the numeric linear algebra library “Scalala” is used.

In a next step, \( T \) is computed for the set \( D \) of experimentally observed distances and \( U \) is calculated from it. The observed \( U \) is then ranked among the sorted \( \{U_k\}_{k=1}^K \). \( H_0 \): “no interaction” is rejected on the significance level \( \alpha \), if and only if it ranks higher than \( \lceil (1 - \alpha)K \rceil \)-th.
Chapter 5

Extensions

We relax some of the assumptions stated in Section 2.2 and show for each extension, which changes are necessary in the theoretical interaction model and/or in the IAP implementation.

Figure 5.0.1: Overview Chapter 5

5.1 Spatially Heterogenous Interaction Processes

5.1.1 Motivation

The model in Chapter 3 is based on the assumption that the interaction is homogenous throughout the whole domain. In real applications, however, subdomains are often present, and we know that the interaction processes can not be homogenous across subdomains. Therefore, a way to model spatially heterogenous processes is desirable.

One way to approach this is to partition the space into independent subdomains and analyze them separately. Another idea is to formulate hyper-models that allow spatially varying parameter values. In our model, this would correspond to potential parameters, which are not constants anymore but rather a function of space (i.e., the location in the domain $\Omega$). This violates the assumption of a translational invariant interaction process, which is fundamental for the presented interaction model. The abstraction from object location to nearest neighbors distances is not possible without this assumption and the idea of an interaction potential is unfounded.
5.1.2 Integration in IAP

The first approach of partitioning the domain into subdomains requires no changes in the theoretical model of the interaction process in Chapter 3 at all. The only change concerns the model input handling. Instead of inputting the data of the whole domain at once, the data of the independent subdomains is given as an input to the model and the interaction analysis is executed separately for all subdomains. The results of the analyses of individual subdomains can then be compared with each other. Similarities lead to the conclusion that the process is spatially homogenous, whereas differences indicate the presence of a heterogenous interaction process.

The concept of a subdomain is called a Region of Interest (ROI) in the ImageJ terminology. ImageJ already supports different shapes of ROI’s in images, like rectangles, ovals, or polygons, the latter being a very flexible option. With the tool “ROI Manager”, ImageJ supports to save, open, and combine multiple ROI’s. This functionality is used in IAP to model heterogenous processes by partitioning the space. A ROI can be selected in the reference image and then only the distances in this subdomain are used. The user selects each subdomain of interest as a ROI and runs the interaction analysis for each of them. The analysis is concluded by the user with comparing the results of the different ROI’s.

5.2 Uncertainty in Measurements of Distances

5.2.1 Motivation

In our derivation of the interaction model, we assume for parameter estimation that the distances $D = \{d_i\}$ are known exactly. In most real applications, however, these distances are measurements that are corrupted by systematic and random errors. As long as the errors are small compared to the scale on which $p(d)$ changes significantly, they have little effect on the parameter estimates.

In cell biology, localization techniques are pushed to their limits in attempts to measure ever-shorther distances and scientists use these distances to understand small-scale dynamics. In this scenario, it is an unavoidable consequence that errors are significant and it becomes important to understand the measurement errors as the key to proper interpretation of the measured distances.

Object-base interaction analysis techniques share two crucial steps. The first is to determine the positions of the objects in the image. The second step involves a calculation of the Euclidian distance between positions of two objects. Although this calculation is simple in both 2D and 3D, its error analysis is demanding and has to be included correctly in the model.

The positions of two small fluorescent objects $x_1$ and $x_2$ are typically determined by fitting the photon count distribution (point spread function (PSF)) with Gaussians and using the means of these Gaussians as positions. Because of the finite number of detected photons, the results for $x_1$ and $x_2$ are not exact. They are estimates that differ from the true, but unknown positions by a normally distributed error. These errors have been discussed and analyzed for nanometer localization measurements by Thompson et al. [23]. The analysis of distance data based on these positions faces additional challenges, which are discussed in Stirling Churchman et al. [7]. Their main contribution, for which we give a summary in Section 5.2.2, is the derivation of the distribution of the Euclidean distance $|x_1 - x_2|$, given that the difference $x_1 - x_2$ is Gaussian distributed.
5.2.2 Estimation of Measurement Uncertainty

It is natural to assume that the distribution of errors is Gaussian when it appears Gaussian by eye, which it does in the case of distance measurements, especially for good signal/noise ratios. However, as the Euclidean distance $|x_1 - x_2|$ is a nonnegative number it follows that it cannot be Gaussian distributed. One would commit systematic errors on distance-related measurements and would lose the precision of modern microscopy techniques, if one would do so. But how are the errors of object distances distributed?

Two different objects with true positions $x_1$ and $x_2$ give rise to experimentally recorded positions $\hat{x}_1$ and $\hat{x}_2$ with Gaussian probability distributions

$$p(\hat{x}_i) = \frac{1}{2\pi\sigma_i^2} \exp\left(-\frac{(\hat{x}_i - x_i)^2}{2\sigma_i^2}\right), \text{ with } i = 1, 2. \quad (5.2.1)$$

Hence, $\tilde{d} = \hat{x}_1 - \hat{x}_2$ is Gaussian distributed with mean $d = x_1 - x_2$ and variance $\sigma^2 = \sigma_1^2 + \sigma_2^2$. We can write the Gaussian probability distribution for $\tilde{d}$ as a function of $\tilde{d}$ and the angle $\delta$ between $\tilde{d}$ and $d$, in which the true distance $d = |d|$ enters as a parameter:

$$p(\tilde{d}) = p(\tilde{d}, \delta) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(\tilde{d} - d)^2}{2\sigma^2}\right) \frac{1}{2\pi\sigma^2} \exp\left(-\frac{d^2 + \tilde{d}^2 - 2\tilde{d}d\cos(\delta)}{2\sigma^2}\right). \quad (5.2.2)$$

If we integrate $p(\tilde{d}, \delta)$ over the domain of $\delta$ with radius $\tilde{d}$, we obtain $p(\tilde{d})$, the distribution we where looking for. In 2D, we have to integrate over a circle and find:

$$2D: \ p(\tilde{d}) = \tilde{d} \int_0^{2\pi} p(\tilde{d}, \delta) d\delta = \frac{\tilde{d}}{2\pi\sigma^2} \exp\left(-\frac{\tilde{d}^2 + d^2}{2\sigma^2}\right) \int_0^{2\pi} \exp\left(\frac{\tilde{d} \cdot d \cos(\delta)}{\sigma^2}\right) d\delta. \quad (5.2.3)$$

The last integral is the modified Bessel function of integer order zero, $I_0$, and we can thus write:

$$2D: \ p(\tilde{d}) = \frac{\tilde{d}}{\sigma^2} \exp\left(-\frac{\tilde{d}^2 + d^2}{2\sigma^2}\right) I_0\left(\frac{\tilde{d} \cdot d}{\sigma^2}\right). \quad (5.2.4)$$

In 3D, we have to integrate over a surface of a sphere, which yields:

$$3D: \ p(\tilde{d}) = \sqrt{\frac{2}{\pi}} \frac{1}{\sigma d} \exp\left(-\frac{\tilde{d}^2 + d^2}{2\sigma^2}\right) \sinh\left(\frac{\tilde{d} \cdot d}{\sigma^2}\right). \quad (5.2.5)$$
Even if \( \sigma^2 \ll d^2 \), which means we have a good signal, the center of the Gaussian used to approximate \( p(d) \) is not a good estimate for \( d \). For \( \sigma^2 \ll d^2 \), we have a good approximation for \( p(d) \) in 2D:

\[
2D: p(\tilde{d}) \approx \frac{1}{\sqrt{2\pi \sigma}} \sqrt{\frac{d}{\tilde{d}}} \exp \left( -\frac{(\tilde{d} - d)^2}{2\sigma^2} \right), \tag{5.2.6}
\]

which looks similar to a Gaussian except for the factor \( \sqrt{\frac{d}{\tilde{d}}} \). This factor shifts the maximum of \( p(\tilde{d}) \) to approximately \( \tilde{d} = d \left(1 + \frac{\sigma^2}{2d^2}\right) \). This means we would systematically use too large values for \( d \) by a relative amount of \( \frac{\sigma^2}{2d^2} \) in our analysis. The complete derivation can be found in Stirling Churchman et al. [7] and more details about the overestimation of \( d \) in its Appendix B.

### 5.2.3 Integration in Interaction Model

In this Section, we will incorporate the understanding about the measurement error we gained from Section 5.2.2 into our model. The model density \( p_{\text{true}}(d) \) for the nearest neighbor distances is given in Equation (3.3.5). It is based on the assumption that we know the distances \( D = \{d_i\}_{i=1}^{N} \) exactly:

\[
p_{\text{true}}(d|\Omega, Y) = p_{\text{true}}(d|q) = Z^{-1}q(d) \exp(-\phi(d)). \tag{5.2.7}
\]

Let \( d \) denote the true distance and \( \tilde{d} \) the measured distance. As we have seen in Section 5.2.2, the probability distribution of a measured distance \( d \) around a true distance \( \tilde{d} \) is not Gaussian but has the following form:

\[
2D: p_{\text{uncert.}}(\tilde{d}) = \frac{\tilde{d}}{\sigma^2} \exp \left( -\frac{\tilde{d}^2 + d^2}{2\sigma^2} \right) I_0 \left( \frac{\tilde{d} \cdot d}{\sigma^2} \right) \approx \frac{1}{\sqrt{2\pi \sigma}} \sqrt{\frac{d}{\tilde{d}}} \exp \left( -\frac{(\tilde{d} - d)^2}{2\sigma^2} \right). \tag{5.2.8}
\]

As the true distance of any observed distance and the additive uncertainty of the observed distance are independent random variables, the blurring of the distances is mathematically described by a convolution, denoted by the operator \((*)\), of the true density with the density of the measurement uncertainty:

\[
p_n(\tilde{d}) = (p_{\text{true}} * p_{\text{uncert.}})(\tilde{d}) = \int_{-\infty}^{\infty} p_{\text{true}}(\tau) \cdot p_{\text{uncert.}}(\tilde{d} - \tau) \, d\tau. \tag{5.2.9}
\]

Inserting \( p_{\text{true}} \) into Equation (5.2.9) yields:

\[
p_n(\tilde{d}) = Z^{-1} \int_{-\infty}^{\infty} q(\tau) \exp(-\phi(\tau)) \cdot p_{\text{uncert.}}(\tilde{d} - \tau) \, d\tau, \tag{5.2.10}
\]

and inserting 2D: \( p_{\text{uncert.}}(\tilde{d}) \) results in:

\[
p_n(\tilde{d}) = Z^{-1} \int_{-\infty}^{\infty} q(\tau) \exp(-\phi(\tau)) \cdot \frac{\tilde{d} - \tau}{\sigma^2} \exp \left( -\frac{(\tilde{d} - \tau)^2 + d^2}{2\sigma^2} \right) I_0 \left( \frac{(\tilde{d} - \tau)d}{\sigma^2} \right) \, d\tau. \tag{5.2.11}
\]

This integral has to be solved numerically. Therefore we simplify the integration by inserting the approximation for \( p_{\text{uncert.}} \) if \( \sigma^2 \ll d^2 \), which results in:

\[
p_n(\tilde{d}) \approx Z^{-1} \int_{-\infty}^{\infty} q(\tau) \exp(-\phi(\tau)) \cdot \frac{1}{\sqrt{2\pi \sigma}} \sqrt{\frac{(\tilde{d} - \tau)d}{d}} \exp \left( -\frac{(\tilde{d} - \tau)^2 - d^2}{2\sigma^2} \right) \, d\tau. \tag{5.2.12}
\]
We assume only local support $\Omega$ of $p_{\text{true}}$ as the distances are from the finite domain $\Omega$:

$$p_m(d) \approx \frac{1}{\sqrt{2\pi\sigma Z}} \int_{\Omega} q(\tau) \sqrt{\frac{(d - \tau)}{d}} \exp \left( -\phi(\tau) - \frac{(d - \tau - d)^2}{2\sigma^2} \right) d\tau \tag{5.2.13}$$

With this probability of a single observed distance $\tilde{d}$, we can calculate the joint probability density of the observations $\tilde{D}$ respecting the uncertainty of the distance measurements:

$$p_m(\tilde{D}|q) \approx \left( \frac{1}{\sqrt{2\pi\sigma Z}} \right)^{-N} \prod_{i=1}^{N} \int_{\Omega} \exp \left( \log \left( \frac{(\tilde{d}_i - \tau)}{d_i} \right) - \phi(\tau) - \frac{(\tilde{d}_i - \tau - d_i)^2}{2\sigma^2} \right) d\tau. \tag{5.2.14}$$

Using this convoluted density, a log-likelihood can be constructed as shown in Section 3.4.1:

$$l_m(\Theta|\tilde{D}, q, \phi) \approx \log \left( \left( \frac{1}{\sqrt{2\pi\sigma Z}} \right)^{-N} \right) + \sum_{i=1}^{N} \log \left( \sum_{\tau_j \in \Omega} \exp \left( \log \left( \frac{(\tilde{d}_i - \tau_j)}{d_i} \right) - \phi(\tau_j) - \frac{(\tilde{d}_i - \tau_j - d_i)^2}{2\sigma^2} \right) \Delta \tau \right). \tag{5.2.16}$$

This means that for each distance $d_i$, an integral needs to be computed. As $q(\cdot)$ is not known in a compact analytical form, these integrals need to be computed numerically. They are approximated with a finite sum over points $\tau_j$ on a grid spacing $\Delta \tau$. We can write:

$$l_m(\Theta|\tilde{D}, q, \phi) \approx \log \left( \left( \frac{1}{\sqrt{2\pi\sigma Z}} \right)^{-N} \right) + \sum_{i=1}^{N} \log \left( \sum_{\tau_j \in \Omega} \exp \left( \log \left( \frac{(\tilde{d}_i - \tau_j)}{d_i} \right) - \phi(\tau_j) - \frac{(\tilde{d}_i - \tau_j - d_i)^2}{2\sigma^2} \right) \Delta \tau \right). \tag{5.2.16}$$

For maximum-likelihood estimations of model parameters $\Theta$, Equation (5.2.16) needs to be evaluated many times by the optimizer. Despite the double sum, the function can be efficiently computed, because the first sum is only over few (order 100) data points and the second sum is over the domain $\Omega$ within the finite support of $q(\cdot)$.

### 5.2.4 Integration in IAP

To take the uncertainty of the distance measurement into account the IAP needs to be modified in two places. The variances $\sigma_i$ of the positions must be available and we need to compute the convolved log-likelihood $l_m(\Theta|\tilde{D}, q, \phi)$ (5.2.16) instead of the original log-likelihood $l(\Theta|D, q, \phi)$ in Equation (3.4.1).

If the position data $X$ and $Y$ originate from special fluorescence microscopy techniques such as PALM or STORM [13], the variances $\sigma_i$ are directly available as an output of the single-molecule localization. IAP already uses the JMatIO-Library to read in discrete object positions. Therefore,
if the variances $\sigma_i$ are available in Matlab, the extension of IAP to read in variances is straightforward.

For maximum-likelihood estimation of model parameters $\Theta$, Equation (5.2.16) is the objective function of the optimizer. As CMA-ES optimizer on black box fitness functions, it is sufficient to implement the fitness function interface of the optimizer for $l_m(\Theta, D, q, \phi)$.

The term $\log \sum_j \exp(x_j)$ in $l_m(\Theta, D, q, \phi)$ requires special care, as it may lead to numerical underflow/overflow. A large $x_j$ can lead to overflow due to exponentiation; for negative values of $x_j$ with a large absolute value, the exponential term vanishes and taking the logarithm of this very small number results in underflow. A common trick based on a simple transformation can avoid these problems. One can write: $\exp(a) + \exp(b) = \exp(a - c) + \exp(b - c)) \exp(c)$ for any $c$. Using that for $\log \sum_j \exp(x_j)$ and with $\log(ab) = \log(a) + \log(b)$, one gets:

$$\log \left( \sum_j \exp(x_j) \right) = \log \left( \sum_j \exp(x_j - c) \exp(c) \right)$$

$$= \log \left( \left( \sum_j \exp(x_j - c) \right) \exp(c) \right)$$

$$= \log \left( \sum_j \exp(x_j - c) \right) + c$$

If one chooses $c$ to be the element of $x$ with the largest absolute value, one has created a more robust function implementation. This does, however, require to store all $x_i$ before evaluating $\log(\cdot)$.

### 5.3 K-Nearest Neighbors

The interaction model described in Chapter 3 assumes nearest neighbor interactions (3.3.2). We now relax the assumption of nearest-neighbor interactions to more, that is K-nearest neighbor interactions for $K \geq 2$. We still assume that objects from the same set do not interact with each other and we take the set $Y$ as a given reference. We are now interested in the interaction of objects from $X$ with, multiple objects in $Y$. Each object in $X$ can “feel” up to $K$ objects simultaneously, provided they are within interaction range.

For k-nearest neighbors the potential becomes:

$$\Phi(x_i, y_k) = \begin{cases} \phi(d_{i,k}) & \text{if } y_k \text{ is } k\text{-th NN of } x_i \text{ and } k \leq K \\ 0 & \text{else.} \end{cases} \quad (5.3.1)$$

Using the same potential for all $K$ neighbors corresponds to the assumption that the interaction to all $K$ neighbors is identical. We define $d_i = \{d_1, d_2, \ldots, d_K\}$ as the k-nearest neighbor distances of an object $x_i$. The inner sum, denoted by $\phi(d_i)$, over all $j$ in Equation (3.3.1) vanishes not completely, but considers only $K$ distances:

$$p(X, Y) \propto \exp \left( -\sum_{i=1}^{N} \sum_{k=1}^{K} \Phi(x_i, y_j) \right) \cdot \phi(d_i) = \sum_{k=1}^{K} \phi(d_{i,k}) . \quad (5.3.2)$$
For K-nearest neighbors, the state density $q(d)$ is a function of a vector $d$ and not only of a scalar distance. With the same arguments as in Section 3.3, we can specify the probability of observing a certain vector of distances $d$:

$$p(d|\Omega, Y) = p(d|q) = Z^{-1} q(d) \exp(-\phi(d)) .$$

(5.3.3)

The partition function $Z$ becomes a $K$-dimensional integral:

$$Z = \int q(d) \exp(-\phi(d)) \, dd$$

(5.3.4)

$$= \int_{d_{1,\text{max}}}^{d_{1,\text{min}}} \ldots \int_{d_{K,\text{max}}}^{d_{K,\text{min}}} q(d) \exp(-\phi(d)) \, dd_1 \ldots dd_K .$$

(5.3.5)

The interaction model for K-nearest neighbor interaction stated as the joint probability of observations $D = \{d_i\}_{i=1}^N$:

$$p(D|q) = Z^{-N} \prod_{i=1}^N q(d_i) \exp(-\phi(d_i)).$$

(5.3.6)

Parameter estimation for k-nearest neighbor works as in Section 3.4.1. The only difference is that for each evaluation of the log-likelihood, $K$ integrals have to be numerically evaluated to compute $Z$, instead of only one.

Instead of using the same potential $\phi(\cdot)$ for all neighbors, different potentials $\phi_k(d_k)$ with $k = 1, \ldots, K$ can be used for each of the $K$ neighbors:

$$\Phi(x_i, y_k) = \begin{cases} 
\phi_k(d_{i,k}) & \text{if } y_k \text{ is } k\text{-th NN of } x_i \text{ and } k \leq K \\
0 & \text{else}.
\end{cases}$$

(5.3.7)

Such a model assumes that depending on the degree of the relation to the corresponding neighbor different interactions act. The parameter space of such a model has $K$-times more dimensions than that for the case with only one potential, as each of the $K$ potentials has its own parameters.

The sufficient statistic $T$ in Equation 3.5.1 can be used again. The Factorization Theorem can be applied in both cases as in Section 3.5.1 and we can use the same rank-based hypothesis test.

### 5.4 Three Classes of Objects

The interaction model described in Chapter 3 models the interaction between two classes of objects. This Section discuss the case where one wants to detect presence of interactions in between three classes of objects. Depending on the type of the interaction, interactions of 3 classes of objects may be modeled and detected using a similar approach as for 2 classes and some ideas from K-nearest neighbors (Section 5.3). We will outline how the different types of interactions with 3 objects, A, B, and C can be modeled and detected. We use the notation AB if the classes A and B interact directly and !(AB) if they don’t.
In the first special case, only two of the three objects interact \((AB, !(AC), !(BC))\) with each other and the third class of objects \(C\) is not part of an interaction with the other two classes. This simple case can be treated with the existing model. One can run the interaction analysis three times, each time with a different pair of object classes. The run with the interacting pair \(AB\) will output a positive result, whereas the other two runs with \(!AC\) and \(!BC\) will return a negative result.

In a second special case, one object class is interacting with both other object classes, e.g. \(AB, AC\) and \(!BC\). Objects in class \(B\) and \(C\) do not directly interact. If the interactions are independent, one can use the same approach as in the first special case, but the runs return a positive result for \(AB\) and \(AC\) and only one negative for the run \(!BC\). If the two interactions are not independently, the interactions \(AB\) and \(AC\) must be analyzed together. Following the idea of 2-nearest neighbors, one can use a similar definition of the interaction potential as in Equation 5.3.7. Class \(A\) corresponds to the set \(X\) and classes \(B\) and \(C\) to the set of objects \(Y\). Instead of considering the nearest and second nearest neighbors, we model the interaction to the nearest neighbor in class \(B\) and the nearest neighbor in class \(C\). Objects from class \(A\) are in \(Y\) and objects of class \(B\) and \(C\) are in \(Y\). We use two potentials \(\phi_k(d_k)\) with \(k = B, C\) to define a combined interaction potential from class \(A\) to class \(B\) and \(C\):

\[
\Phi(x_i, y_k) = \begin{cases} 
\phi_B(d_i, B) & \text{if } y_k \text{ is NN of } x_i \text{ and in class } B \\
\phi_C(d_i, C) & \text{if } y_k \text{ is NN of } x_i \text{ and in class } C \\
0 & \text{else.}
\end{cases}
\]  

(5.4.1)

If we use \(d_i = \{d_B, d_C\}\) and \(K = 2\), the same derivation as for \(K\)-nearest neighbors (Section 5.3) can be used.

The general case \(AB, AC,\) and \(BC\), where all three objects interact with the other two could be treated with a similar approach. However, parameter estimation becomes very involved and it is not straightforward how to interpret the resulting model. Therefore a further discussion of this case is omitted.
Chapter 6

Conclusions & Outlook

This thesis is concerned with the development of computational methods and software for the analysis of interactions between intra-cellular structures. The theoretical basis of this thesis was developed in previous work by Helmuth et al. [14]. In the theory presented in [14] spatial point patterns of intra-cellular objects were assumed to result from nearest-neighbor interactions, which are modeled with an effective interaction potential. The input point patterns are typically extracted from dual-color fluorescence microscopy images of the structures of interest in a pre-processing step. The actual interaction analysis aims at estimating parameters of the interaction potential and judging their significance.

6.1 Interaction Analysis Plugin for ImageJ

The first objective of this thesis was to integrate the whole workflow – from importing unprocessed dual-color images to estimating interaction potentials for sets of discrete objects – into a single user-friendly tool. In order to guarantee optimal access of the cell biology community the algorithms were developed as plugins for ImageJ, which is a widely used open-source image analysis software. The first step in the analysis reduces images to sets of discrete objects. The point detection part of an existing feature point tracking plugin was reused for this purpose, which, however, required extensive refactoring and modularizing on the previous implementation. If object locations have been determined by other means they can be important into the plugin in the Matlab data file format. To prepare the output of the feature point detection for the interaction analysis chromatic aberrations had to be accounted for. Therefore a separate chromatic aberration correction plugin for ImageJ was developed. It estimates two linear models for the x- and y-components of the lateral shift between the images of dual-color beads (or any other small dual color object) in two different color channels. It reuses the refactored point detection implementation. The second step of the interaction analysis requires to compute distances $D$ between the interacting objects and the state density $q(d)$. A kd-tree data structure was introduced to compute these quantities in $O(N \log(M))$ time. For typical example data, the kd-tree reduced the runtime of estimating $q(d)$ to under one second, as compared to the 24 seconds of the naive ($O(NM)$) algorithm. Estimating $q(d)$ furthermore required to estimate the domain (cell) boundary from the reference image. This was achieved using standard ImageJ functions executed by a macro. Given the distances $D$ and the state density $q(d)$, the plugin can estimate parametric or non-parametric interaction potentials and perform parametric and non-parametric tests for the presence of interactions. The numerical optimization problem in Maximum-Likelihood and Maximum-a-Posteriori parameter estimation was solved using CMA-ES, which required to integrate an existing Java implementation in the present plugin. For optimal usability of the software, an extensive GUI was designed. It provides
an interface for setting user-provided parameters and options, visual feedback of intermediate steps of the analysis, plots of estimated interaction potentials, etc. Large portions of the present plugins were written in Scala and used Scalala for linear algebra operations. All developed software is modular and is based on modern programming paradigms.

6.2 Extensions

The second objective of this thesis was to identify to what extent the assumption of the original model can be relaxed and to develop corresponding extensions of the model and software. A first extension considers spatially heterogenous interaction processes, i.e. the process is no longer considered to be the same in all parts of the analyzed cell. A straightforward solution was presented, which partitions the cell domain into disjunct subdomains and performs the analysis on the data contained in the subdomains separately. The interaction analysis plugin can provide this functionality without any changes to the software by using functionalities of the ROI-manager of ImageJ. In a second extension of the previous model, the assumption of first-nearest-neighbor interactions is relaxed to interactions with $K > 1$ nearest neighbors. This thesis laid the theoretical foundations for treating this more complex case, discussed the changes required to the software implementation, and outlined emerging computational challenges. Furthermore a similar approach is suggested to treat the case of three classes of objects interacting in various ways. Finally, the parameter estimation is conceptually redesigned by taking into account the uncertainty in the estimated locations of the interacting objects. The uncertainty of distance measurements is not normally distributed, even if the uncertainty in the estimated locations is normally distributed. The consequences of this fact for estimating interaction potentials is treated theoretically. The computational challenges pertaining to this extension and guidelines for its implementation are presented.

6.3 Outlook

Future theoretical work is concerned with further broadening of the theoretical basis of the interaction analysis. In particular the case of multiple classes of objects interacting simultaneously should be investigated in more detail.

Some of the extensions presented in this thesis have not yet been implemented in IAP, since the focus was on creating an overall clean implementation rather than including extensions that might not be requested by potential users. The modular object-oriented design of the software implementation, however, allows easily extending the plugin’s functionality. In particular the K-nearest-neighbor interaction analysis can be readily implemented without major changes to the interface, functions, and data structures. In principle the plugin is also ready to be extended to handle location estimates with known distributions of the measurement uncertainties, as, e.g., typically produced by super-resolution microscopy techniques. Whereas handling the new input format is trivial, numerical integration routines need to be integrated to perform the convolution integral in Equation (5.2.9). If available, shape features of the analyzed objects could be taken into account to further refine distance measurements. The current implementation, however, is specifically tailored to point-like objects. Including shape information thus requires redesigning a number of functions and data types.

A further interesting question is whether the image-based interaction analysis approach – especially with the extension considering measurement uncertainties – can be tailored to detect and quantify molecular interactions on the nanometer length scale. Provided the position uncertainty and an
estimate for the interaction range is given it should be possible to compute a distance threshold for the step potential that yields test statistics with optimal power. Whether this optimal power can reach an acceptable level within the constraints imposed by the microscopy experiments (in particular chromatic aberrations) is an open question.

The user interface, user guidance through the typical workflow, and presentation of output could potentially be improved further. For this it is desirable to assess the plugin in biological labs with the goal of including users’ feedback in future versions of the plugin.
Appendix A

Software Documentation

A.1 User Tutorial

A.1.1 Requirements

Java

The Interaction Analysis Plugin (IAP) for ImageJ needs Java Runtime Environment (JRE) version 1.5 (sometimes denoted as Java 5) or higher. JRE can be downloaded from http://www.oracle.com/technetwork/java/javase/downloads in the current version.

ImageJ

This Interaction Analysis Plugin requires ImageJ (1.44e or later) to be installed on your machine. ImageJ is also included in the image processing package Fiji. One can check the currently installed version of ImageJ by starting ImageJ or Fiji and selecting About ImageJ… from the Help menu. In the same Help menu is also a menu item Update ImageJ … with which you can simply update your ImageJ installation to the current version.

If ImageJ is not installed, go to http://rsbweb.nih.gov/ij/ and download the latest version of ImageJ. On this webpage you can also find detailed installation instructions for different operating systems (Linux, Mac OS X, Windows). For new users of ImageJ, it is highly recommended to have a look at ImageJ’s user manual.

A.1.2 Installation

Once ImageJ is installed, it is simple to add the IAP plugin: Copy the IAP.jar file into the plugins folder of your ImageJ installation and restart ImageJ. The plugin will appear under the menu Plugins → Mosaic → IAP after a restart of ImageJ.

A.1.3 Work Flow

The interaction analysis with the IAP is split into two successive steps. The first step consists, additionally to input preparation, of computing the observed distances $D$ and approximating the
state density \( q(d) \). The second step contains the interaction analysis based on the measured distances.

![Plugin Workflow](image)

**Figure A.1.1: Plugin Workflow**

**Input & Distance Calculations**

The user of IAP can choose between images and Matlab file as input:

![Input type selection](image)

In both cases the user has to define the domain. For Matlab input this is a rectangle specified by the user. If the input is images, the user can generate automatically a binary mask from the reference image or he can open an existing binary mask. In addition he can specify a ROI in the reference image to run the analysis only within the ROI. If a mask and a ROI is set, the analysis is only performed within their intersection:

![Object-based Interaction Analysis Plugin](image)
Relative shifts in the position of the objects due to chromatic aberration can be correct by a linear calibration function. The IAP has the functionality to estimate the calibration function with the independent plugin CA:
If object position detection is needed (in case of images as input), the detection is performed interactively as the first step of the distance calculation:

The actual distance calculations run then fully automatically.

Interaction Analysis

In a second step, IAP offers the choice of three different interaction analyses of the measured distances to the user. The user can choose between:

- Non-Parametric Testing [A.1.5],
- Parametric Potential Estimation and Parametric Testing [A.1.6],
- Non-Parametric Potential Estimation [A.1.7].

All three analyses do not require any user interaction after setting the necessary parameters. After IAP has completed the selected analysis, it presents the results in a user-friendly format.
Figure A.1.4: Estimated state density

Figure A.1.5: Non-Parametric Testing

Figure A.1.6: Parametric Potential Estimation and Parametric Testing
The examples for illustrating the features of IAP are based on a 3D dual color confocal image of a HER 911 cell. Rab5-positive endosomes and Adenovirus serotype 2 particles have been fluorescently labeled and an interaction analysis of these two groups of intra-cellular objects has produced the output shown in Figures A.1.2, A.1.3, A.1.4, and A.1.8.

A.2 Implementation

This section gives a short introduction into the code but will not discuss all details of the implementation. For more details about the implementation of IAP, please refer to the documented source code.

A.2.1 Prerequisites to a Programmer

The plugin is implemented in the programming language Scala, so obviously Scala knowledge is an advantage. The transition from Java to Scala is relatively simple, so Java developers should have no problem to read and understand the code, especially if they have basic knowledge of functional programming. Basic concepts of Scala can be found on the official Scala web page [http://www.scala-lang.org](http://www.scala-lang.org) or in the online available book "Programming Scala" from O’Reilly. Information about writing an ImageJ plugin can be found in the developer documentation of ImageJ [http://rsbweb.nih.gov/ij/developer](http://rsbweb.nih.gov/ij/developer).
### A.2.2 Library Dependencies

IAP uses the following open source libraries:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weka</td>
<td>Data Mining Software in Java</td>
<td>kd-Tree for Nearest Neighbor Search</td>
</tr>
<tr>
<td>JMatIO</td>
<td>Matlab’s MAT-file I/O in JAVA</td>
<td>Reading .mat Matlab Files.</td>
</tr>
<tr>
<td>Scalala</td>
<td>Scala Linear Algebra library</td>
<td>Basic Linear Algebra Calculations</td>
</tr>
<tr>
<td>ScalaNLP</td>
<td>Scala library for Natural Language Processing, Machine Learning, and Statistics</td>
<td>Monte Carlo Sampling</td>
</tr>
<tr>
<td>CMA-ES</td>
<td>CMA-ES Optimization in Java</td>
<td>Potential Parameter Estimation</td>
</tr>
</tbody>
</table>

### A.2.3 Packages

IAP has a modular architecture. Each module is in a separate package:

- mosaic.plugins: ImageJ Plugins
- mosaic.interaction: Model Classes
- mosaic.interaction.input: Image Handling and Input Generation for Model
- mosaic.core.detection: Feature Point Detection and Linking
- mosaic.core.sampling: Monte Carlo Sampling and Hypothesis Tests
- mosaic.core.distance: Nearest Neighbor Distance Calculations
- mosaic.core.optimization: Potential Parameter Estimation

### A.2.4 Important Classes

The main classes of IAP are listed below:

- mosaic.interaction.InteractionModel: Main Model Functionality
- mosaic.interaction.InteractionGUI: Graphical User Interface and Work Flow
- mosaic.plugin.ChromaticAberration: Independent CA Plugin
Bibliography


Acknowledgements

I want to thank Jo Helmut for the idea of the thesis, the fertile collaboration and his important support. It was motivating and interesting for me to work together with him as my advisor. He was always patient, open for questions, and very flexible. Furthermore, I would like to thank Prof. Ivo Sbalzarini for the opportunity to write this thesis in his group and for mentoring me during my master studies.

When I got stuck during the implementation phase, it was Thomas Plüss who gave me another perspective of the problem and helped me to debug the program. I'm indebted for many invaluable discussions with Gabriel Krummenacher during lunch time and for sharing the experience of writing a master thesis. Stefan Schori and Thomas Sommer were both benevolent and provident, I enjoyed to be their flat mate.

This master thesis marks the end of my studies at ETHZ and the end of my privileged education. I feel grateful to have been in the lucky situation to study abroad and at very good schools which prepared me for my future life. Since I was born, it has always been my parents who supported me, in a liberal manner. They financed me and I want to thank my parents for that. Special thanks go to my family for everything these exceptional people have done for me. They taught me the courage to go self-reliant my way, the respect for my neighbors, and the discipline to work for my goals. Only this together with the necessary good luck made it possible to combine my studies with my passion for sport.