BACHELOR

Investigation of a potentially novel method to investigate protein-protein interactions in protein coronas on nanoparticles

Gerritsen, Ruben H.J.

Award date:
2017

TU/e Technische Universiteit Eindhoven

Bachelor Thesis

Applied Mathematics

Investigation of a potentially novel method to investigate protein-protein interactions in protein coronas on nanoparticles

Author:
Ruben Gerritsen
r.h.j.gerritsen@student.tue.nl

Supervisors:
Prof.dr. R.W. van der Hofstad
R. Fitzner

In cooperation with:
Prof.dr.ir. M.W.J. Prins
Junhong Yan

13th July 2017

Contents

Chapter 1

Introduction

In this work we evaluate a potentially novel method to investigate protein-protein interactions in the protein corona on nanoparticles. The protein corona on nanoparticles heavily influences how a nanoparticle behaves inside the body [1]. Therefore a thorough understanding of the protein corona is necessary. Different techniques for analyzing protein coronas exist, but at the moment no proper way exists to find protein-protein interactions in the corona. We will investigate a novel method that uses DNA-assisted protein assays that could give inside into the protein-protein interactions.

In this first chapter we briefly discuss nanoparticles and the protein corona of a nanoparticle. Thereafter we present the potentially novel method for investigating protein-protein interactions in the protein coronas. We also discuss the challenges encountered with this method. To conclude the chapter we will give an outline of the report.

1.1 Nanoparticles and Protein Corona

Nanoparticles typically are particles with a diameter in the nanometer range, usually between 1nm and 100nm. They are widely used in modern medicine in a wide range of applications such as contrast agents and drug delivery systems [2]. Nanoparticles in living organisms adsorb proteins on their surface and a protein adsorption layer (the "protein corona") forms on the particle. This protein corona heavily influences the interaction between the particle and its surroundings. Hence a thorough understanding of the protein corona is necessary to fully understand the effect and potential of nanoparticles in the human body and other living organisms [1].

To enhance the understanding of the protein corona a lot of experimental studies have been conducted using varies techniques. The techniques used can give inside into the thickness of the protein corona [3] and the protein corona composition, which proteins make up the protein corona [4]. However, currently no research is done in protein-protein interactions within the corona. Also no suitable technique exists to measure this interaction. In this work we will investigate a potentially new way of studying the protein-protein interactions in the protein coronas on nanoparticles.

1.2 Project Background

The novel method in this work uses the idea of DNA-assisted protein assays. In which proteins are labeled with DNA (or other nucleic acids). Due to the amplification ability of DNA a high sensitivity for the DNA labels can be achieved via PCR. Therefore proteins that can not be measured directly can be measured via their DNA labels. This idea is exploited in the new method.

The Novel Method

To investigate protein-protein interactions in the protein corona, proteins are labeled with DNA tags, see Figure 1.1a. For every protein type there are two different DNA tags. The two tags are due to the fact that DNA is not symmetric and can end either with a 5' carbon or a 3' carbon. Only 5' carbon can react with 3' carbon and vice versa. Interactions of the type 5'-5' and 3'-3' are not possible. To avoid the continues use of 3' and 5'carbon we will refer to the two types as starters and enders. Every protein type has its own DNA label. A single protein can carry multiple starter or ender type tags, but not both.

After the proteins are labeled they are placed in solution with the nanoparticles and adsorb to the surface, Figure 1.1b. Once the protein corona has formed the proteins are fixed to the surface using mild crosslinking, Figure 1.1c.

When the proteins are fixed to the surface the sample is divided into two parts. In one of the two parts DNA tags are ligated, Figure 1.1d. Which means that if two proteins are close enough for their tags to interact they form a connection. Due to the properties of DNA ligation one starter tag can only react with exactly one ender tag. The other part of the sample is used later on.

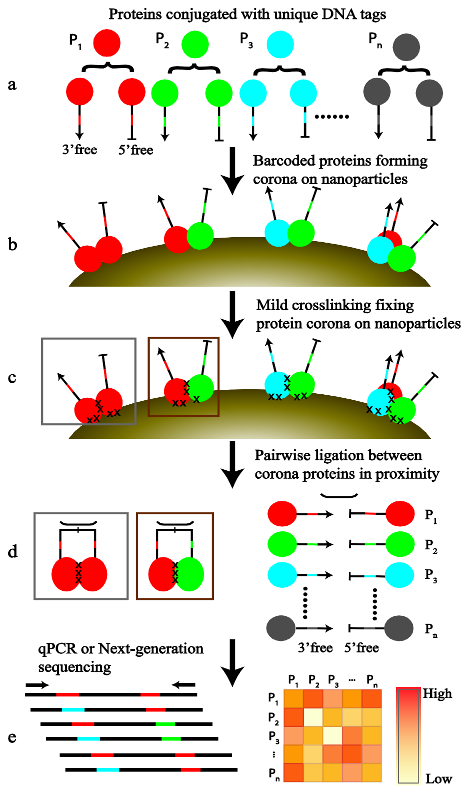


Figure 1.1: Workflow summary

After the tags are ligated the ligated tags can be quantified using qPCR (quantative Polymerase Chain Reaction, see also [5]), Figure 1.1e. More precise the ratio between ligated pairs can be measured. For example suppose we have 2 different starters s_1 and s_2 and two different enders e_1 and e_2. Then the ratio between ligated pairs, s_1e_1, s_1e_2 etc. can be measured.

In the non-ligated sample the individual tags are measured using the same technique. This gives the ratio between for example s_1 and s_2 or e_1 and e_2. Due to a limitation of the technique the ratio between starters and enders can not be measured.

Since only tags on proteins that are in close proximity of one another can ligate it is expected that from the relative amounts of ligated pairs the interaction between the proteins in the corona can be predicted. For example, if a certain protein attracts another protein one would expect more ligated pairs between these two proteins.

A Data Example

In Figure 1.2 we see an example of the data obtained using our technique. We see the data for three different proteins, B, C and D. With different ender and starter tags indicated by the 1 and 2 respectively. In the central figure we can see the relative frequencies of the ligated pairs, C1B2 stands for a starter on protein C connected to an ender on protein B. And on the right we can see the separately measured frequencies of starter tags and ender tags. Note that the separate starters and enders both sum to one. So that there is no data on the ratio between the number of starters and enders.

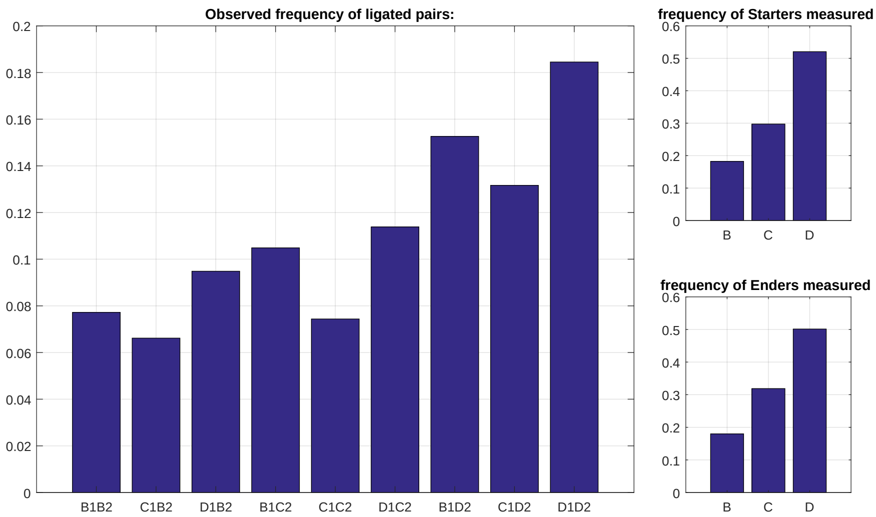


Figure 1.2: An example of the data obtained by the method

Challenges in the Method

Two major challenges have to be overcome to use this novel method succesfully. The first challenge being that the only data on the proteins on the surface and the connections between tags is given in ratio's. This means that nothing can be said about concentrations and surface coverage of the proteins.

The second challenge is that a protein can have multiple tags. The tags will be of the same kind, but the number of tags per protein is unknown. Since different proteins have different sizes a different number of tags is expected on different types of proteins.

1.3 Project Objectives and Report Outline

The main objective of the current work is to investigate the feasibility of the current approach to help in characterizing protein-protein interactions in the protein corona. The main question is: is it possible to deduce from the ratios of ligated pairs and non-ligated tags, the attraction between proteins on the surface of a nanoparticle?

To answer this question we start by setting up a very simple basic model in Chapter 2. This model is compared to observed data. We will see that there are large deviations between the simple model and the data. At the end of Chapter 2 we discuss the complete approach to answer the main question.

Chapter 3 is devoted to "mislabeling" which is a type of measurement error occuring during the experiment. We investigate to what extent this measurement error can influence the outcome of the experiment.

In Chapters 4 and 5 a program to simulate the experiment will be introduced and the effect of different parameters on the system will be analyzed. Furthermore, the simulation will be used to fit certain parameters on the available data. With the fitted parameters we hope that the affinities (attraction) between the proteins in the corona can be determined.

We conclude this work in Chapter 6. Conclusions will be summarized and an outlook on future work will be given.

Chapter 2

Simple Mathematical Model

In this chapter we present a basic model for the protein ligation on the surface of a nanoparticle. We first describe how the experimental parts are translated into math and introduce the necessary notation. Then, we derive some exact mathematical results based on two different approaches. In the first approach it is assumed that every starter has an infinite amount of tags and can therefore interact with all neighbours that are close enough. In the second approach we will assume that every starter can only react with one ender or, in other words, has only one tag.

2.1 An Independent Model

2.1.1 Formalizing the Experiment

There are different proteins, every type of protein is uniquely tagged with ender and starter tags, as described in the previous chapter. We will refer to the different proteins with the natural numbers (1,2,...) and to starter and ender type tags with s and e respectively. For example s_k would mean a protein of type k with a starter type label. Pairs can only be formed between starters and enders if they are "close enough". To model "close enough" we introduce the interaction distance R. Particles s and e can interact if $\texttt{dist}(s, e) \leq R$. If there is a connection (or interaction) we use $s_k e_l$ to refer to the connection between a starter on protein k and an ender on protein l.

To derive a base model of the system we will assume that all proteins are points on a surface. We will use a homogeneous spatial Poisson process to model the location of the proteins on the surface. Poisson processes N and M are used for starters and ender respectively. The rate for the processes are given by α (proteins / unit area) and β respectively. This rate is the sum of the rates of the different proteins, so $\alpha = \alpha_1 + \alpha_2 + \ldots$, where α_k is the rate for s_k.

Table 2.1 gives a summary of all relevant variables, parameters and the used notation. The general simplifying assumptions are summarized as follows:

- Proteins can be considered as points
- And the protein locations on the surface of a nanoparticle are independent and uniform (homogeneous spatial Poisson process).

In the following two section, we will derive some basic results, from this model setup.

2.1.2 Every starter can react with every ender in range

In this section we will consider the case where every starter or ender protein has exactly enough tags to react with every other possible protein in range ones.

	Description	Estimated value
N, M	Poisson processes for starters and enders respectively	
s	A starter type	
e	An ender type	
k	Protein type of starter	
l	Protein type of ender	
$s_k e_l$	An interaction (connection) between starter k and ender l	
α, β	Rate's of the Poisson processes for starters and enders respectively	
α_k	Rate of starter k	
x_k, y_l	Fractions of k and l, so $\alpha \cdot x_k = \alpha_k$ and $\beta \cdot y_l = \beta_l$	
R	Interaction range of a protein, the maximum distance over which a DNA tag can interact with another DNA tag	< 30nm
X	Number of total interactions	
$X_{k,l}$	Number of interactions between starter k and ender l	
$\hat{X}, \hat{X}_{k,l}$	An estimate for X, $X_{k,l}$ respectively	
$D_{particle}$	Diameter of the nanoparticle	≈ 1000nm
$D_{protein}$	Diameter of an average protein	$\approx$ 3-15nm

Table 2.1: Summary of variables and parameters

Expected number of total interactions

Under this assumption the expected number of total interactions per unit area is given by:

$$\mathbb{E}[\hat{X}] = \sum_{s \in N} \text{Number of enders in range of } s.$$

The number of enders in range is simply the number of enders in a circle around s with radius R. Since the number of enders in area $A = \pi R^2$ is $\sim$ Pois(βA), we have:

$$\mathbb{E}[\hat{X}] = \sum_{s \in N} \beta \pi R^2 = \alpha \beta \pi R^2. \tag{2.1}$$

Expected number of reaction products $\mathbb{E}[X_{k,l}]$

In a similar fashion we find the expected number of reaction products per unit area:

$$\mathbb{E}[X_{k,l}] = \sum_{s_k \in N} \text{Number of } e_l \text{ in range of } s_k = \alpha_k \beta_l \pi R^2. \tag{2.2}$$

An other approximation

By assuming that every starter can react with every ender in range we overcount the real number of reactions, especially if a protein can only make one connection to another protein. To get a more accurate idea of the expected number of interactions another approach can be adopted. We will look at the following approximation for the expected number of interactions:

$$\mathbb{E}[\hat{X}_s] = \sum_{s \in N} \text{is there an ender in range of } s.$$

In this case every starter reacts only once but enders can react multiple times. The probability that there is an ender in range of s is given by:

$$\mathbb{P}(B_R(s) \text{ contains an ender}) = \mathbb{P}(\texttt{dist}(s, M) \leq R) = 1 - e^{-\beta \pi R^2}. \tag{2.3}$$

Where $\texttt{dist}(s, M)$ is the distance between starter s and the closest point of M (the closest ender). This is the *Contact Distribution Function* or *Empty Space Function* [6]. This chance is also the fraction of proteins that react so we get for the expected number of interactions:

$$\mathbb{E}[\hat{X}_s] = \mathbb{E}[N] \cdot \left(1 - e^{-\beta \pi R^2}\right) = \alpha \left(1 - e^{-\beta \pi R^2}\right) \tag{2.4}$$

Or if we assume enders react only once and starters multiple times:

$$\mathbb{E}[\hat{X}_e] = \mathbb{E}[M] \cdot \left(1 - e^{-\alpha \pi R^2}\right) = \beta \left(1 - e^{-\alpha \pi R^2}\right) \tag{2.5}$$

For small $\beta \pi R^2$ and $\alpha \pi R^2$ these equations are very close to 2.1, since for small x, $1 - \exp(-x) \approx x$. By replacing α by α_k and β by β_l in equations 2.4 and 2.5, we get approximations of the number of reaction products.

Variance

From the estimates (2.4) and (2.5) we compute the variance of the number of interactions. Equation (2.4) actually arises from the product of a Poisson random variable and a Bernoulli random variable with parameter $p = 1 - e^{-\beta \pi R^2}$. This product has a Poisson distribution so that for the variance is given by:

$$\mathrm{Var}(\hat{X}_s) = p\alpha = \mathbb{E}[\hat{X}_s] \tag{2.6}$$

And for the variance of the number of reaction products we get:

$$\mathrm{Var}(X_{k,l}) = \alpha_k \cdot \left(1 - e^{-\beta_l \pi R^2}\right) \approx \mathbb{E}[X_{k,l}] \tag{2.7}$$

Note that this equation is only an approximation to the real variance as the formula is not symmetric in starters and enders. However for a small interaction range or small β_l this formula reduces to $\alpha_k \beta_l \pi R^2$ which is symmetric in enders and starters.

An upperbound on the true number of interactions

Assuming that every starter can react with every ender in range gives rise to an overcount of the real number of interactions. Therefore the formulas derived in this section give an upperbound on the true number of interactions. The simplest upperbound is given by equation 2.1:

$$\mathbb{E}[X] \leq \mathbb{E}[\hat{X}] = \alpha \beta \pi R^2 \tag{2.8}$$

A sharper upperbound is given by the minimum of 2.4 and 2.5:

$$\mathbb{E}[X] \leq \min\left(\mathbb{E}[\hat{X}_s], \mathbb{E}[\hat{X}_e]\right) \tag{2.9}$$

2.1.3 Every starter and ender can only react with one other protein

In this section we will look at the expected number of interactions under the assumption that every starter can react with only one ender and vice versa or, in other words, has only one tag. Since there is no easy way to count pairs between two different Poisson processes we assume that every starter reacts to the closest available ender.

Chance for a starter to react to the closest ender

A starter s can only react to the closest ender if there is no other starter closer to the ender e than s. Figure 2.1 describes this idea, suppose the red dot is the starter s and the blue one an ender e. For these two dots to react we need that $r_1 \leq R$, where r_1 is the distance from the starter to the closest ender. Furthermore, we need that the ender has no starter closer by than s, this means the blue circle should not contain any other red dots. The chance that the blue circle does not contain another starter is the chance that the circle is empty. This is given by:

$$\mathbb{P}(\text{Red} \cap B_{r_1}(e) = \emptyset) = e^{-\alpha\pi r_1^2}. \tag{2.10}$$

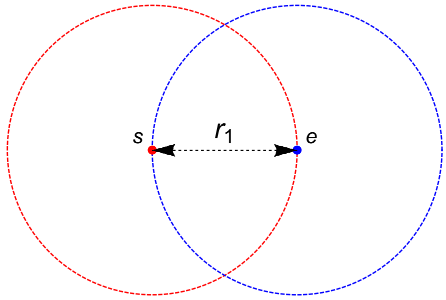


Figure 2.1: Illustration of particles interacting ($r_1 \leq R$)

The chance that the closest neighbor of s is a distance r_1 away from s is given by the *Contact Distribution Function*:

$$\mathbb{P}(\texttt{dist}(s, M) < r) = 1 - e^{-\beta\pi r^2} \tag{2.11}$$

From this it follows that the pdf for the closest neighbor is given by:

$$f_d(r) = 2\beta\pi r e^{-\beta\pi r^2}. \tag{2.12}$$

Call the event that a starter interacts with its first neighbor I_1, then we have:

$$\mathbb{P}(I_1) = \int_0^R f_d(r_1) \cdot \mathbb{P}(\text{Red} \cap B_{r_1}(e) = \emptyset)\, \mathrm{d}r_1 \tag{2.13}$$

$$= \frac{\beta}{\beta + \alpha} \cdot \left(1 - e^{-(\alpha+\beta)\pi R^2}\right) \tag{2.14}$$

And for an estimate of the expected number of interactions per unit area we get:

$$\mathbb{E}[\hat{X}_1] = \mathbb{E}[N] \cdot \mathbb{P}(I_1) \tag{2.15}$$

$$= \frac{\alpha\beta}{\beta + \alpha} \cdot \left(1 - e^{-(\alpha+\beta)\pi R^2}\right) \tag{2.16}$$

Note the interchangeability of α and β in this formula as one would expect from this problem.

Furthermore, the variance of $\hat{X}_1$ is equal to $\mathbb{E}[\hat{X}_1]$, since we can view this random variable as N tries of a bernoulli experiment (see section 2.1.2).

Reaction Products

For the reaction products we find:

$$\mathbb{E}[X_{k,l}] = \frac{\alpha_k \beta_l}{\beta + \alpha} \cdot \left(1 - e^{-(\alpha+\beta)\pi R^2}\right) \tag{2.17}$$

A lower bound for the true number of interactions

From Section 2.1.3 a lower bound for the true expected value of the number of interactions follows. Since formula (2.16) undercounts the number of interactions we have:

$$\mathbb{E}[X] \geq \mathbb{E}[\hat{X}_1] \tag{2.18}$$

The distance between the lower and upper bound

Taking the simplest upper bound for $\mathbb{E}[X]$ which is $\alpha\beta\pi R^2$ we have that the distance between this upper bound and the lower bound from the previous section is:

$$\mathbb{E}[\hat{X}] - \mathbb{E}[\hat{X}_1] = \alpha\beta\pi R^2 - \frac{\alpha\beta}{\beta+\alpha} \cdot \left(1 - e^{-(\alpha+\beta)\pi R^2}\right) \tag{2.19}$$

$$= \frac{\alpha\beta}{\beta+\alpha} \cdot \sum_{n=2}^{\infty} (-1)^n \frac{((\alpha+\beta)\pi R^2)^n}{n!} \tag{2.20}$$

In Figure 2.2 we see this error relative to $\mathbb{E}[\hat{X}]$ (distance between upper and lower bound divided by the expected value of $\hat{X}$)plotted for fixed R, and variable $\alpha = \beta$.

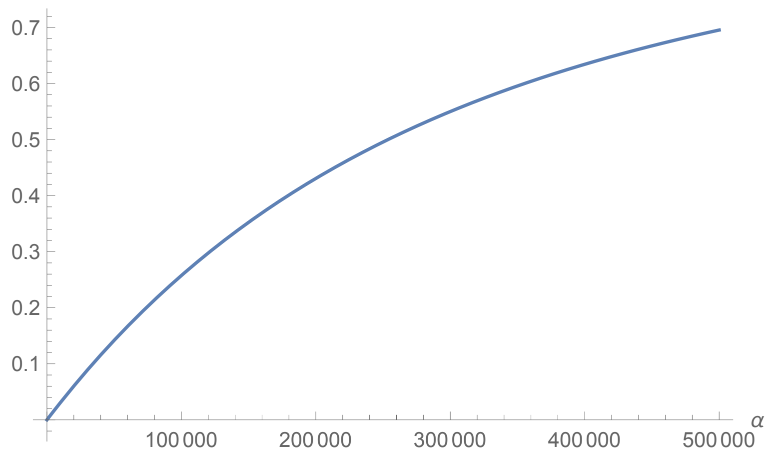


Figure 2.2: Maximum error for fixed R as function of $\alpha = \beta$ relative to $\mathbb{E}[\hat{X}]$

2.1.4 Converting the numbers to ratio's

In the current experimental setup no rates or interaction ranges can be measured. The only available data is ratio's between reaction products and ratio's of starting products. From these ratio's the relative frequencies can be determined. We use the following notation for these relative frequencies, x_k for the fraction of starters of type k and y_l for the fraction of enders of type l. Here we will convert the main result on the expected number of reaction products to relative frequencies.

To do so, we normalize equations (2.2) and (2.17). To make the normalization process easier, we substitute $\alpha_k = x_k \cdot \alpha$ and $\beta_l = y_l \cdot \beta$. We then find, for equations (2.2) and (2.17) respectively, the sum over all possible pairs (k, l),

$$\sum_{k,l} x_k y_l \alpha\beta\pi R^2 = \alpha\beta\pi R^2 \tag{2.21}$$

$$\sum_{k,l} \frac{x_k y_l \alpha\beta}{\beta+\alpha} \cdot \left(1 - e^{-(\alpha+\beta)\pi R^2}\right) = \frac{\alpha\beta}{\beta+\alpha} \cdot \left(1 - e^{-(\alpha+\beta)\pi R^2}\right). \tag{2.22}$$

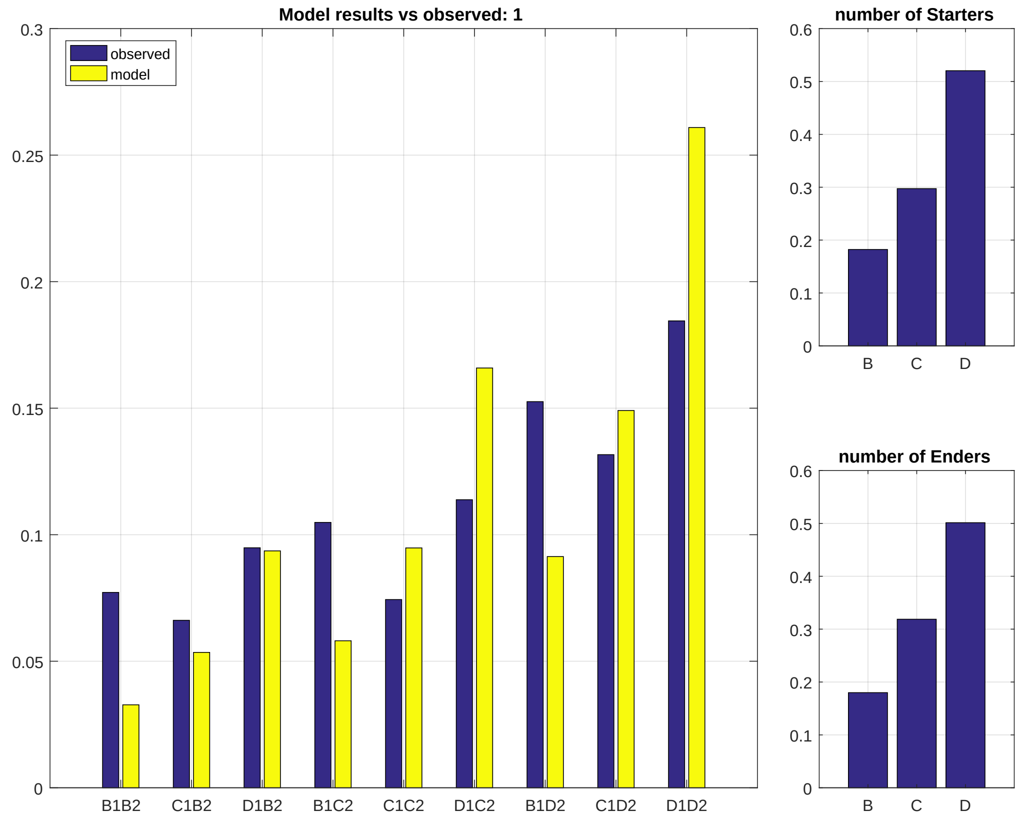


Figure 2.3: The data set from Figure 1.2, but now with the results of the independent model next to the observed results

Using these sums to normalize the two equations, we find for both equations, that the expected fraction of interactions of type $s_k e_l$ is given by:

$$\mathbb{E}[X_{k,l}] = x_k y_l \tag{2.23}$$

Where we now used $X_{k,l}$ to indicate the fraction of interactions of type $s_k e_l$.

2.2 Model vs. Data

The result from the previous section gives us the reference we need for the rest of the work. We can now compare the data to the completely independent and uniform case. In Figure 2.3 we see the same figure as Figure 1.2 but now with the predictions of the simple model alongside of it. We see that the simple model corresponds with what one expects from the separate starters and enders. At the same time we can see that the observed values deviate largely from this prediction as is observed for all available data. It is hoped that exactly this deviation can give a clue on the protein-protein interactions in the protein corona. However there are multiple factors that need to be taken into account.

First of all the deviations could be due to measurement errors. It is known that, while analyzing the samples, some tags are misread as others tags. This phenomena will be called "mislabeling". A second explanation may be found in the unknown number of tags a protein has. In Figure 2.3 for example the "number of Starters" measured is the frequency of starter tags measured, not proteins. So a priori it is not clear whether these frequencies say anything about the actual number of proteins close to each other. Finally there is the hoped for effect, the effect of affinities. With affinities we mean the attraction between two protein types. If two protein types attract one another the relative frequency of this ligated pair should increase.

2.3 Project Approach

The first step in this project is investigating the influence of possible measurement errors. Once they are excluded as the source of the deviations, tags and affinities can be considered. Since the mathematics of describing proteins with a certain number of tags and affinities gets rather involved the choice has been made to proceed numerically. A simulation will be setup to describe the proteins on the surface of a nanoparticle, including the number of tags per protein and the possible affinities.

The main goal of this project is to find the affinities between proteins on the nanoparticle surface. To achieve this, we first need to know more about the number of tags per protein. If a certain protein type has twice as many tags as another type, it will be counted more often in the experimental results. This could lead us to conclude that a certain affinity exists between two types, while in fact it is just the number of tags that made the difference in the data. Therefore knowledge of the number of tags is necessary to account for them in the simulations.

To find the average number of tags per protein a set of special experiments will be conducted with only one protein present in solution but with different tags. In this way, affinities between different types of proteins are excluded as the source of deviations in the frequencies of reaction products. Hence, we can use these data sets to fit the number of tags on.

Given the number of tags per protein, we can look at the affinities. Again experiments will be conducted, but now with multiple proteins. To eliminate as much uncertainty as possible a series of experiments is used with only 2 different protein types. The model affinities are fitted to these experimental results, taking into account the already found number of tags. In this way we expect that the relative affinities between different protein types can be determined.

Chapter 3

Mislabeling

From the start of the project it was known that some ligated pairs were mislabeled as other pairs. In this chapter we will investigate the influence of this mislabeling.

3.1 Mislabeling

Mislabeling is said to occur if one pair of ligated tags is counted as another type of tags. Mislabeling is fortunately relatively easy to measure. One can just measure one type of pairs and see which readouts it gives. We measure for example only pair s_1e_2 and see how often it is readout as any other pair s_ke_l. In this way for every pair the amount of mislabeling can be determined and a matrix as in Figure 3.1 can be made, where an intenser color stands for an intenser readout. As one can see the diagonal is red or orange corresponding with a high readout, for example two proteins B_B are readout as B_B the most, which is good. Looking at proteins A_B, however, we see that they are readout as B_B quite often, which is undesirable.

To make mathematical sense of the matrix presented in Figure 3.1 the rows are normalized. In this way a right stochastic matrix is obtained. This matrix will be called the "mislabel matrix" A. This matrix can be used to convert model results to "real" results. Suppose we have the frequencies of the reaction products predicted by the model in a row vector $\mathbf{y}_{model}$. We obtain the same mislabeling as in the experiments by simply multiplying $\mathbf{y}_{model}$ by A.

$$\mathbf{y}_{model,cor} = \mathbf{y}_{model} \cdot A.$$

Where $\mathbf{y}_{model,cor}$ is the model result corrected for the mislabeling in the experiment. In Figure 3.2a we see ones again Figure 1.2 but now with the model results and the model results corrected for mislabeling. Interestingly enough no real effect is apparent. In Tabel 3.1 we see the mislabel matrix for this figure and see that the fractions of mislabeling are very small. So the effect is only apparent if the measured quantity is itself very small. This is seen in Figure 3.2b where the C1C2 interaction is very small and the mislabeling therefore has a larger effect. Note however that in Figure 3.2b a log scale is used for all the frequencies. The C1C2 interaction predicted by the model is almost ten thousand times to small compared to the observed value. In the data corrected for mislabeling we see a very small improvement. Similar observations were made for all other experiments, were the mislabeling had either no observable effect or only a very small effect, but only in cases were the interactions were already very small.

3.2 Other Mislabel Matrices

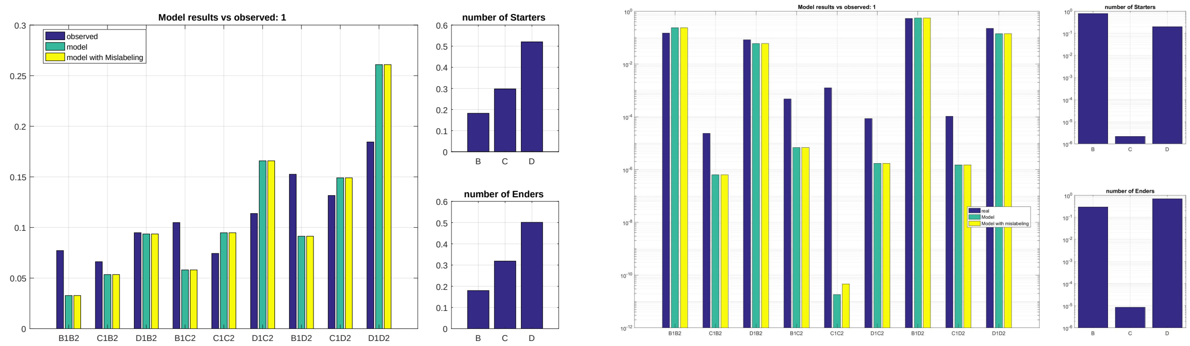
The current experimentally obtained mislabel matrix can obviously not explain the deviations from the independent behavior and has only a very small effect on the frequency of interactions.

		output															
	input	A_A	B_A	C_A	D_A	A_B	B_B	C_B	D_B	A_C	B_C	C_C	D_C	A_D	B_D	C_D	D_D
input	A_A	7,6E+09	2,1E+02	1,0E+00	1,0E+00	8,4E+01	1,0E+00	1,0E+00	1,0E+00	1,0E+01	1,0E+00	1,0E+00	1,0E+00	4,7E+02	1,0E+00	1,0E+00	1,0E+00
	B_A	8,2E+01	1,3E+10	1,0E+00	1,0E+00	1,0E+00	5,6E+02	1,0E+00	1,0E+00	1,0E+00	1,2E+01	1,0E+00	1,0E+00	1,0E+00	4,2E+02	1,0E+00	1,0E+00
	C_A	2,8E+02	8,5E+01	2,0E+10	2,3E+01	1,0E+00	1,0E+00	4,1E+02	1,0E+00	1,0E+00	1,0E+00	4,0E+01	1,0E+00	1,0E+00	1,0E+00	6,3E+02	1,0E+00
	D_A	8,8E+01	4,1E+01	1,0E+00	2,0E+10	1,0E+00	1,0E+00	1,0E+00	5,0E+02	1,0E+00	1,0E+00	1,0E+00	7,0E+01	1,0E+00	1,0E+00	1,0E+00	9,0E+02
	A_B	7,4E+02	1,0E+00	1,0E+00	1,0E+00	1,3E+10	3,9E+05	1,0E+00	1,0E+00	2,9E+03	1,0E+00	1,0E+00	1,0E+00	3,6E+02	1,0E+00	1,0E+00	1,0E+00
	B_B	3,9E+01	3,6E+02	1,0E+00	1,0E+00	1,0E+00	3,4E+10	1,0E+00	1,0E+00	1,0E+00	2,0E+03	1,0E+00	1,0E+00	1,0E+00	4,6E+02	1,0E+00	1,0E+00
	C_B	2,9E+01	1,0E+00	4,7E+02	1,0E+00	5,9E+00	1,8E+05	2,6E+10	3,7E+01	1,0E+00	1,0E+00	4,5E+03	1,0E+00	1,0E+00	1,0E+00	4,4E+02	6,0E+00
	D_B	1,0E+00	1,0E+00	1,0E+00	7,5E+02	6,4E+00	2,7E+05	1,0E+00	3,7E+10	1,0E+00	1,0E+00	1,0E+00	6,1E+03	1,0E+00	1,0E+00	1,0E+00	3,9E+02
	A_C	1,3E+01	1,0E+00	1,0E+00	1,0E+00	1,2E+02	1,0E+00	1,0E+00	1,0E+00	9,4E+09	4,0E+00	1,9E+00	7,4E+00	2,2E+02	1,0E+00	1,0E+00	1,0E+00
	B_C	1,0E+00	3,8E+00	1,0E+00	1,0E+00	1,0E+00	4,8E+02	1,0E+00	1,0E+00	7,0E+01	1,7E+10	2,0E+00	1,0E+00	1,0E+00	4,7E+02	1,0E+00	1,0E+00
	C_C	1,0E+00	1,0E+00	1,9E+01	1,0E+00	1,0E+00	1,0E+00	3,7E+02	1,0E+00	6,0E+01	2,4E+01	1,8E+10	4,3E+01	1,0E+00	1,0E+00	4,4E+02	1,0E+00
	D_C	1,0E+00	1,0E+00	1,0E+00	1,6E+01	1,0E+00	1,0E+00	1,0E+00	5,1E+02	8,8E+01	5,5E+01	1,0E+00	2,5E+10	1,0E+00	1,0E+00	1,0E+00	7,6E+02
	A_D	3,7E+01	1,0E+00	1,0E+00	1,0E+00	4,5E+01	1,0E+00	1,0E+00	1,0E+00	5,6E+01	1,0E+00	1,0E+00	1,0E+00	2,1E+10	2,7E+01	2,7E+00	1,2E+01
	B_D	1,0E+00	1,4E+01	1,0E+00	1,0E+00	1,0E+00	4,0E+02	1,0E+00	1,0E+00	1,0E+00	1,0E+00	1,0E+00	1,0E+00	4,2E+01	3,7E+10	1,0E+00	1,0E+00
	C_D	1,5E+01	1,0E+00	2,4E+01	1,0E+00	1,0E+00	1,0E+00	2,6E+02	1,0E+00	1,0E+00	1,0E+00	1,0E+00	1,0E+00	2,3E+01	2,7E+01	2,8E+10	6,3E+01
	D_D	1,0E+00	1,0E+00	1,0E+00	5,6E+01	1,0E+00	1,0E+00	1,0E+00	3,7E+02	1,0E+00	1,0E+00	1,0E+00	1,0E+00	2,7E+01	7,2E+00	1,0E+00	3,4E+10

Figure 3.1: mislabeling counts

I\O	B1B2	C1B2	D1B2	B1C2	C1C2	D1C2	B1D2	C1D2	D1D2
B1B2	1	0	0	5.98e-08	0	0	1.34e-08	0	0
C1B2	7.11e-06	1	1.45e-09	0	1.76e-07	0	0	1.69e-08	2.31e-10
D1B2	7.2e-06	0	1	0	0	1.65e-07	0	0	1.04e-08
B1C2	2.78e-08	0	0	1	1.13e-10	0	2.73e-08	0	0
C1C2	0	2.02e-08	0	1.34e-09	1	2.4e-09	0	2.46e-08	0
D1C2	0	0	2.07e-08	2.21e-09	0	1	0	0	3.05e-08
B1D2	1.08e-08	0	0	0	0	0	1	0	0
C1D2	0	9.04e-09	0	0	0	0	9.53e-10	1	2.22e-09
D1D1	0	0	1.1e-08	0	0	0	2.14e-10	0	1

Table 3.1: The experimentally obtained mislabel matrix used in Figure 3.2



(a) The example from Figure 1.2 now with the model corrected for mislabeling added.

(b) A more extreme experiment with larger deviations and a visible effect of the mislabeling.

Figure 3.2: The effect of mislabeling

This begs the question how a mislabel matrix would look that could explain the deviations from the independent behavior. In this section we will answer this question with a very coarse approach. Namely by generating such a matrix.

To generate this mislabel matrix A the following iterative scheme was used. Starting with A being the identity matrix of size 9×9. The matrix is updated in the following fashion for a given number of cycles. Where $\mathbf{y}_{model}$ initially refers to the basic result of the independent model.

while *cycle* $<$ *number of cycles* **do**
  M is set to the identity matrix of size 9×9.
  for i *from 1 to 9* **do**
    for j *from 1 to 9* **do**
      if i *equals* j **then**
        $M(i,j) = \min(1, \frac{\mathbf{y}_{observed}(i)}{\mathbf{y}_{model}(i)})$
      else
        $M(i,j) = \frac{1}{8} \cdot (1 - \min(1, \frac{\mathbf{y}_{observed}(i)}{\mathbf{y}_{model}(i)}))$
      end
    end
  end
  $\mathbf{y}_{model} = \mathbf{y}_{model} \cdot M$
  $A = A \cdot M$
  update the cycle number
end

The matrix A obtained in this fashion is a mislabel matrix that could account for the deviations observed between the basic model and the experimental results, as can been seen in Figure 3.3. In Figure 3.3 we see the same case as before but now with the mislabel matrix applied obtained by the iterative scheme with a total of 30 cycles. As can been seen the model now fits the data quite well. This is expected since we fit more parameters then data points and in such a situation a good fit always exists and is not unique.

There are, however, two uses for this matrix. The first is getting an idea of the mislabeling needed to explain the data. The second use is to see if the “transition” matrix obtained also works on other experiments. All experiments were conducted at different concentrations such that a set of the same experiments at different concentrations exists. What we tested is if one mislabel matrix obtained from one experiment would improve the other experiments in the same set. This has been tested extensively for all experimental data sets. However it was found that in general only the experiment from which the matrix was obtained benefited from the mislabel matrix considerably. The other experiments in the same set only benefited little or

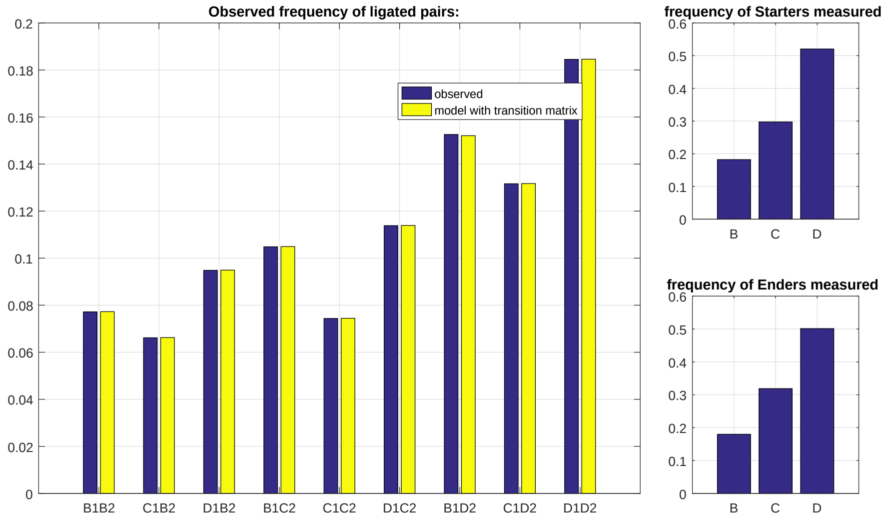


Figure 3.3: The example data but now with the generated mislabel matrix from Table 3.2

I\O	B1B2	C1B2	D1B2	B1C2	C1C2	D1C2	B1D2	C1D2	D1D2
B1B2	0.832	0.0201	0.0207	0.0209	0.0203	0.021	0.0224	0.0212	0.0215
C1B2	0.0556	0.545	0.0533	0.059	0.0505	0.0551	0.0659	0.0565	0.0589
D1B2	0.0505	0.0429	0.596	0.0536	0.0446	0.0491	0.0601	0.0505	0.0529
B1C2	0.0134	0.0132	0.0136	0.89	0.0134	0.0137	0.0145	0.0139	0.014
C1C2	0.0745	0.0582	0.0643	0.0795	0.42	0.0683	0.0895	0.0702	0.0757
D1C2	0.0704	0.053	0.0582	0.0753	0.0533	0.47	0.0849	0.0644	0.0705
B1D2	0	0	0	0	0	0	1	0	0
C1D2	0.0502	0.0402	0.0443	0.0535	0.0411	0.0467	0.0602	0.612	0.0514
D1D2	0.0586	0.0435	0.0477	0.0626	0.0436	0.0516	0.0707	0.053	0.569

Table 3.2: The numerically obtained mislabel matrix used in Figure 3.3

not at all. For examples, see Appendix A.1.

To get an idea of the mislabeling necessary to explain the data, Table 3.2 presents the mislabel matrix obtained by the iterative scheme used in Figure 3.3. We can see that the mislabeling needed is quite large. And from the experimentally obtained mislabel matrix we know that it can't be that large therefore it is almost impossible that the deviations are explained by mislabeling.

Other Transition Matrices

In the previous subsection we looked at a transition matrix that could explain the difference between the simple model and the experimental data. Besides the matrices obtained by the previously mentioned iterative scheme other type of matrices were tested as well. Transition matrices based on the identity matrix with a random error were considered and matrices based on "smart guesses" where the amount of mislabeling was chosen such that the diagonal elements of the matrix would be close to one. This was done to investigate the possibility of a transition matrix between the independent model and the data of different experiments. However no such transition matrix was found. Therefore, it is assumed that this transition approach is very unlikely to succeed in explaining the deviations observed.

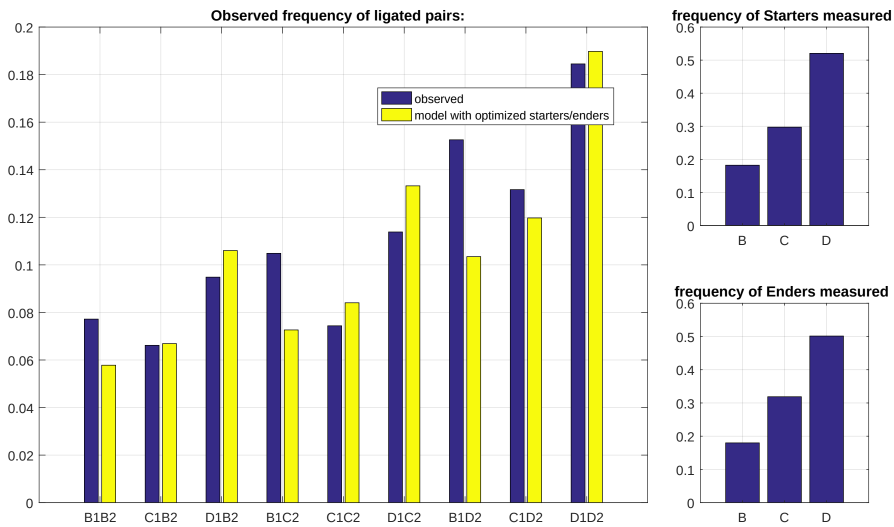


Figure 3.4: The example data but now with the fitted starters and enders

3.3 Another Type of Measurement Error

By considering the transition matrices of the previous section we looked at measurement errors in the measurements of the reaction products (ligated pairs). Another measurement error could be found in the measurement of the separate starters and enders. This error was investigated in a similar way as before. But now constants were fitted to the starters and enders to minimize the difference between the simple model and experimental results. For example the fraction of s_k is given by x_k and the expected fraction of $s_k e_l$ is given by $x_k y_l$. In the fitting approach we substitute $x_{k,new} = c_k x_k$ for x_k and try to find values for c_k that minimize the difference between observed and model values.

The fitting of all c_k's gives an improvement in the model, as can be seen in Figure 3.4, but not as big as the mislabel matrix did. Furthermore, the same problem as with the transition matrix occurred namely that c_k's found on the basis of one experiment did not always improve another experiment in the same set. For an example see Appendix A.2. Therefore it is believed that there is no systematic error in the starter and ender measurements.

3.4 Conclusion

In this chapter we investigated the role of mislabeling and similar measurement errors. It was found that mislabeling in the current form can not account for the deviations seen between the independent model and the data from experiments. We also looked if a systematic error in the measurement of the reaction products or the separate tags could have influenced the data. Using sets of the same experiment at different concentrations, we searched for the measurement errors necessary to enhance one of the experiments in the set. This same measurement error was then used to predict the outcomes of the other experiments in the same set. It was found that in some cases this approach improved the other experiments but that in most cases no improvement or even a deterioration was observed. Therefore we conclude that it is very unlikely that there is a systematic error in the measurements.

Chapter 4

The Simulation

In this chapter the simulation and fitting approaches for the proteins on nanoparticles are described. In Chapter 1 we saw that the experiment consists of two general stages. The first being the adsorption of the proteins to the surface and the second the ligation of the proteins. In the model these two stages will also be treated separately. We will start the chapter by a conceptual discussion of the model.

4.1 Conceptual Model

Suppose that proteins are little sphere's and that the nanoparticle is a way larger sphere that can be approximated by a flat square. Then if proteins would not interact, we could consider the protein adsorption as the uniform raining down of proteins onto the surface. However we know that the proteins do interact, in this model we will assume that the proteins interact due to a change in the surface properties of the nanoparticle. Suppose we have a protein A that attracts other proteins A. Then we will say that the protein A changes the surface of the nanoparticle around the protein in such a way that the probability of another A protein arriving in the area around the already present A protein increases. In this way also repulsions can be modeled by lowering the probability around a protein. With the current approach protein adsorption reduces to placing proteins onto the surface taking into account already present proteins and there influence on the probability of the location for the following protein.

Once all proteins are placed on the surface, pairs should be formed that can form. The requirements for pair formation are that the two proteins are in range of one another, that one of the proteins has a starter tag and the other an ender and that the tags are not yet connected to some other tags. To model this all possible pairs are searched, all proteins with starter tags in range of proteins with ender tags. Then the closest possible tag pair is connected and this is continued until all possible pairs are connected. Counting the number of connected pairs gives and normalizing the counts gives the data is in the experiment.

4.2 Model Setup

4.2.1 The Nanoparticle Surface

Since the nanoparticle is significantly larger than the proteins the surface is almost flat from the perspective of a protein. Therefore, the nanoparticle surface is modeled by a square surface. To model the surface properties "probability grids" are used. With a probability grid we mean a very fine grid over the surface. Every grid cell gets a value and the probability of an arriving protein landing in that grid cell will be proportional to the value of that cell.

Since there are different types of proteins every protein needs its own probability grid. For example suppose we have two proteins A and B. And A attracts A and B attracts B. Then a location with a lot of B will be attractive for another B protein, but neutral for an A protein. Therefore, the change is surface properties for every protein type is recorded in a separate probability grid.

4.2.2 The Proteins

The proteins are modeled as hard 2D disks on the surface. A protein is nothing more than a circle with a certain radius $r_{p,i}$ depending on the protein type i. Every protein has a list of properties: if it is carrying starter or ender type tags, how many tags it carries and what type of protein it is.

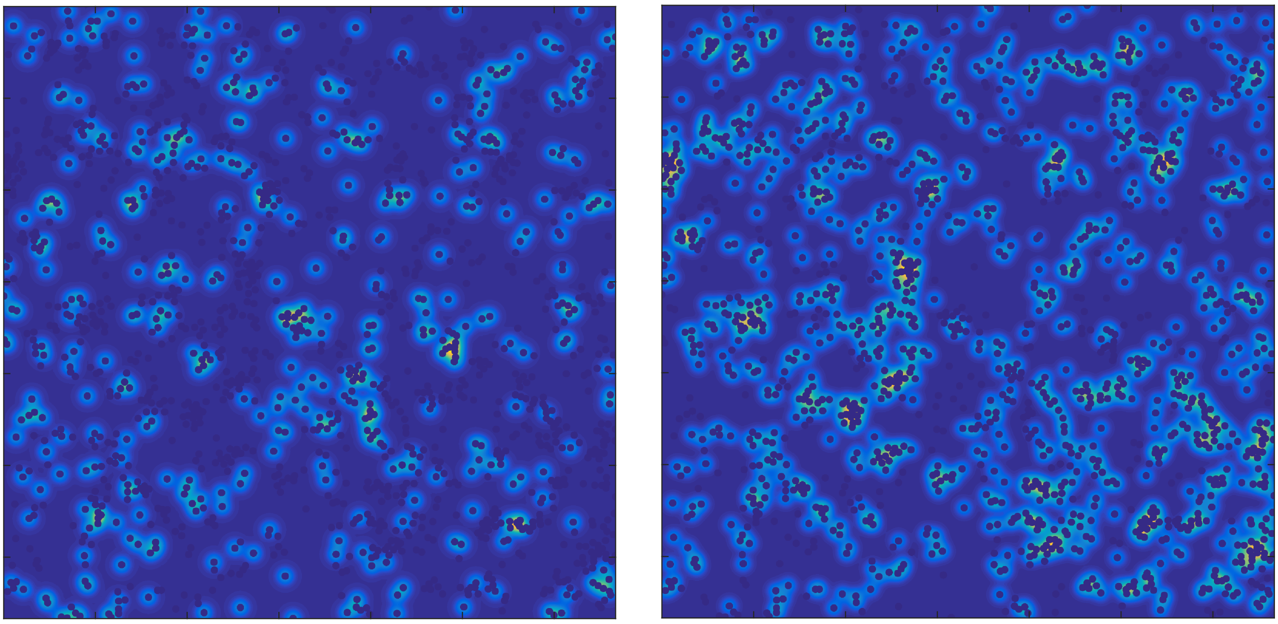
4.2.3 Protein Adsorption and Affinities

The protein adsorption is modeled as a dynamic process. At the start the probability grids are completely uniform so every cell has the same value. Then a first protein is created, random numbers are drawn to determine if the protein is carrying starter or ender type tags and which type of protein it is going to be, also the number of tags is determined. Once the type of protein is known a random number will be drawn between 0 and 1 and multiplied by the total value of the probability grid of that protein type. Starting from the first cell in the grid, the grid cell values are subtracted from the random number until the remaining number is zero or negative. The grid cell for which this happens is the grid cell were the protein is place. After placing the protein the probability grids are updated. Suppose that we have placed a B protein from the previous example (B attracts B), than the probability grid of the A protein will not be updated and the values around the B protein in the B probability grid will be increased. An example of this can be seen in Figure 4.1 where one can see the probability grids for A and B type proteins, if A attracts A and B attracts B. In the current model a 2D gaussian distribution is used with a standard deviation of two averaged protein radii. The exact amount of value added is determined by a multiplicative constant by which the distribution is multiplied. This constant depends on the protein on the surface and to which probability grid the value is added. We will call these added values affinities, $a_{i,j}$ between protein type i and protein type j. In the example of Figure 4.1 we would have that $a_{A,A}$ and $a_{B,B}$ are high and $a_{A,B}$ and $a_{B,A}$ are zero. Note finally that in the example the proteins themselves are dark blue disks where the probability of an arriving protein is zero. This is because we do not allow overlap between proteins. So it is not possible for a protein to arrive at a location that is already taken by another protein.

4.2.4 "Ligation" of Tags

Ones the proteins are placed on the surface the pairs should be "ligated" as in the experiment. This comes down to forming pairs between proteins if they are close enough, there are tags to interact and we are dealing with a starter and ender tag and not two starters or enders. To model close enough an interaction range R is used. All proteins with a separation distance greater than R can not interact.

Every protein has a certain number of tags attached to it. This is a Poisson distributed number with mean γ_i. A protein can not make more connections than there are tags. To form pairs a greedy algorithm is used. First all possible pairs are listed, where possible pairs are all pairs of proteins that have a separation distance smaller than R and where one of the two has starter type tags and the other ender type tags. This list of possible pairs is then sorted based



(a) The probability grid as perceived by an A type protein

(b) The probability grid as perceived by a B type protein

Figure 4.1: An example of the probability grids for the case where A proteins attract A and B attract B. A higher intensity in color means a higher probability for a protein to land there.

on the separation distance. Starting from the smallest distance the proteins are connected. For every connection made the number of remaining tags on the two proteins is lowered by one. A connection is only made if there is a tag left to connect. The pair forming is stopped as soon as there are no tags left or no possible pairs.

4.2.5 Input Parameter Summary

In Table 4.1 we present an overview of all input parameters and the basis values used.

	Description	Value
N_{prot}	Total number of proteins	5000
N_{type}	Number of different protein types	2
ρ_s	fraction of starters	0.5
R	interaction range	6nm
x_{max}	size of the surface	1000nm
x_k	chance of being a starter of type k	-
y_l	chance of being an ender of type l	-
γ_i	mean number of tags for protein type i	0.1 - 3
$r_{p,i}$	radius or protein i	3nm
$a_{i,j}$	Affinity between protein i and j given in added value to the grid	-

Table 4.1: Summary of model parameters and the basis values used

4.2.6 Model Output

Given the input as in the previous section the model gives the number of pairs of each type found as output. This is given in relative frequencies for easy comparison with the data.

4.3 Fitting Approach

4.3.1 The Fitting of the Number of Tags

To fit the number of tags the simulation is used. The simulation was run for a certain number of times with an initial guess of the number of tags. Based on the average result of the simulation compared to the data the number of tags was updated depending on a set of rules and the simulation was rerun. This process was continued until the simulation results where in a $x\%$ deviation range, there was no possibility for enhancing the result or until the number of maximum simulation evaluations was reached. We will present the updating approach for the case of two different proteins and therefore for four different protein-tag pairs, namely: s_1, s_2, e_1, e_2.

Updating the number of tags

The number of tags was updated according to the follow scheme. First the ratios of the experimental results to the simulation results are computed. So for every starter-ender pair a ratio r_{s_k,e_l} is obtained. Suppose that r_{s_2,e_1} and r_{s_2,e_2} are both greater than one than in the experiment s_2 was more prominent than it was in the simulation, therefore it is reasonable to assume that the number of tags on s_2 should be increased. Reasoning along similar lines gives a set of rules for updating the number of tags, assuming both ratios are either greater or smaller than one.

To not overdo the updating of the number of tags it was decided that only the most deviating protein-tag pair would be updated between two simulation rounds. To quantify "most deviating" the maximum of $r_{s_1,e_1} \cdot r_{s_1,e_2}$ and $\frac{1}{r_{s_1,e_1} \cdot r_{s_1,e_2}}$ was taken for s_1 and a similar approach was used for the other protein-tag pairs, the so obtained number will be called the "deviation number" for a protein-tag pair. Since there is only a straightforward manner of updating the number of tags in the case that both ratios for a protein-tag pair are either both greater or both smaller than one, the protein-tag pair with that largest deviation number and both ratio's greater or smaller than one was updated. The update was done linearly using the ratio with the minimal distance to one. Suppose that all this was the case for protein-tag pair s_k and ratio r_{s_k,e_l} we would get:

$$N_{tags,s_k,new} = r_{s_k,e_l} \cdot N_{tags,s_k,old} \tag{4.1}$$

If no protein-tag pair has both ratio's greater or smaller than one, the fitting procedure stops. Further, we have added a maximum number of tags, no more than four tags on average are allowed per protein. This is based on the experimental setup in which it is impossible to have more than four tags on average per protein.

4.3.2 Fitting of Affinities

The fitting of affinities is done in exactly the same manner as the number of tags. The only difference is the way of updating the affinities. We will describe the affinity fitting based on the same setting as before for two different proteins and four different protein-tag pairs. For the remainder of this report it turns out that an alternative notation for starters and enders is more useful. Since we work with only two proteins, say A and B, and since we only have one starter A and one ender B we can use a short hand notation namely AB for $s_A e_B$ etc. From here on this notation will be used and AB will refer to a connection between A and B and the frequency of interaction AB will be refered to with AB.

Updating the Affinities

Since all ratio's always sum to one the four pairs of reaction products are linked and only three seperate values need to be fitted. Which are:

$$d_1 = \frac{AA}{AB}, \qquad d_2 = \frac{AA}{BA}, \qquad d_3 = \frac{AA}{BB}, \tag{4.2}$$

We will use d_1 for the values obtained from the experimental data and s_1 to refer to the same quantity but then from the simulator. To determine how much the observed quantities deviate from the simulator we introduce more ratio's namely:

$$r_1 = \frac{d_1}{s_1}, \qquad r_2 = \frac{d_2}{s_2}, \qquad r_3 = \frac{d_3}{s_3}, \tag{4.3}$$

From these ratio's we determine how to update the affinities. If all ratios are larger than one then it is safe to assume that there is to little AA and the affinity between the A's has to be increased. If this is not the case we look at the ratio's separately. And find the i for which r_i is the smallest. Note that since the ratio's should sum to one "the smallest" is always smaller than one. If r_i is smaller than one it means there is to little of the denominator of s_i, which for r_1 would mean to little AB. Hence we increase the affinity for the term in the denominator of s_i with the smallest r_i. The new affinity is determined as follows:

$$a_{B,A,new} = \left(\frac{BA_{observerd}}{BA_{simulation}}\right)^n \cdot a_{B,A,old}. \tag{4.4}$$

Where we assumed that it is the affinity between B and A that needs to be increased and n is some number which determines the speed of the increase of the affinity. Note that by this approach we only increase affinities and never decrease affinities. There is a special reason for this. In the current simulator the effect between a very low affinity or zero affinity is only very small. Therefore if we would allow for decreasing affinities, the algorithm can get stuck on updating an already low affinity to an even lower affinity etc. without ever enhancing the fit. To avoid this, the choice is made to only increase affinities during the fitting process.

Chapter 5

Simulation Results

In this chapter we present the results obtained with the simulation. In the first section we will look at the simulation and the influence of different parameters on the outcome of the model. Thereafter we will look at the fitting of the number of tags and affinities.

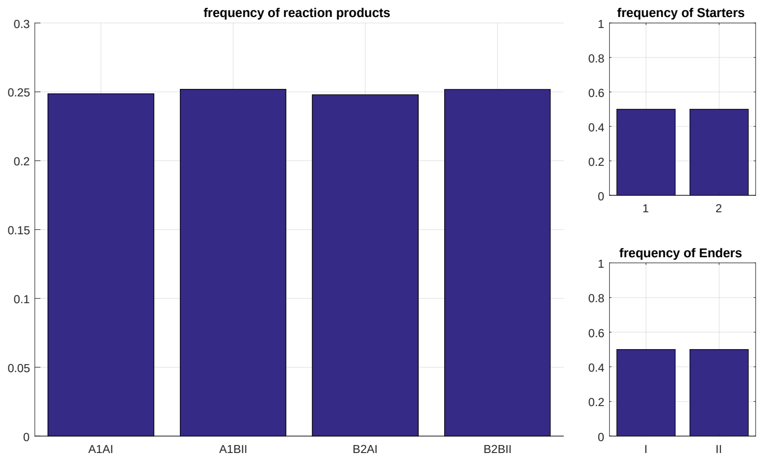
As mentioned before for the fitting of the number of tags and the affinities some special experiments were conducted. These experiments had at most two proteins A and B present and four tags. Two starter tags, which we will refer to with 1 and 2 and two ender tags, I and II were used. Also if there was more than one protein in the experiment, the experiment was setup such that one of the two starters was attached to an A protein and one to a B protein and the same for the starters. We use throughout this chapter the following notation for the protein-tag pairs and reaction product. A1 stands for an A protein with starter "1". The pair A1BII is the reaction product of a "1" starter on an A protein with ender II on a B protein. Furthermore, we will also always use the base parameters as in Table 4.1 unless mentioned otherwise.

5.1 General Simulation Results

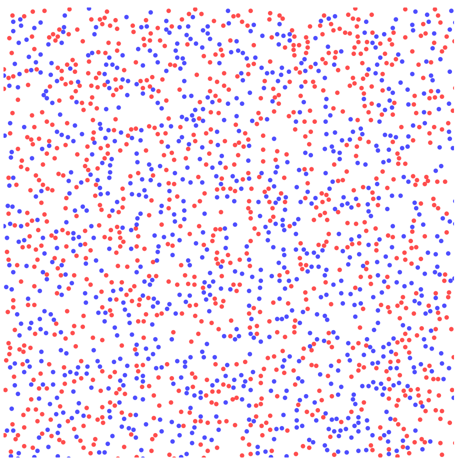
In this section we will investigate the influence of the number of tags on a protein and the protein-protein affinities on the model outcomes. To do this we will start with the most basic case. Where we have two proteins A and B and both in equal proportions so 50% of the starters are A and 50% of the enders. In Figure 5.1a we see this basic case with on average one tag per protein and no affinities averaged over 20 runs. The model behaves as expected by the simple math model and we see that all reaction products are equal. We also see in Figure 5.1b that on the surface the proteins are distributed uniformly.

In Figure 5.2 we see a more interesting case in which the starter A and ender B have 3 tags instead of one on average. This results in a spike at the AB interaction and we see that the BA interaction "suffers" the most from this effect. While the AA and BB interactions remain more or less equal. It is especially interesting to compare Figure 5.2a to 5.2b where the number of tags for the starter A and ender B are heightened to thirty, we see almost no effect compared to the case with only three tags on average. This is explained by the fact that there are on average just a few proteins in range to interact with. Increasing the average number of tags beyond this point will have no effect on the outcome of the model. This could solve the problem of not knowing how much tags per proteins are present. If every protein has more than enough, it can be assumed that a protein will interact with all its neighboring proteins.

From the number of tags we move to the affinities. In Figure 5.3 we see the same case as in chapter 4, A attracts A and B attracts B, but A-B and B-A interactions are neutral. We see the nanoparticle surface and can clearly see the cluster formation of the proteins especially if the affinities increase. For this case the number of tags per protein has been brought back

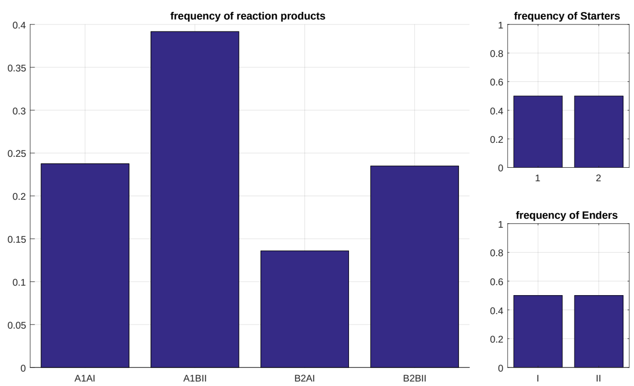


(a) Frequencies of the starters, enders and reaction products according to the simulation

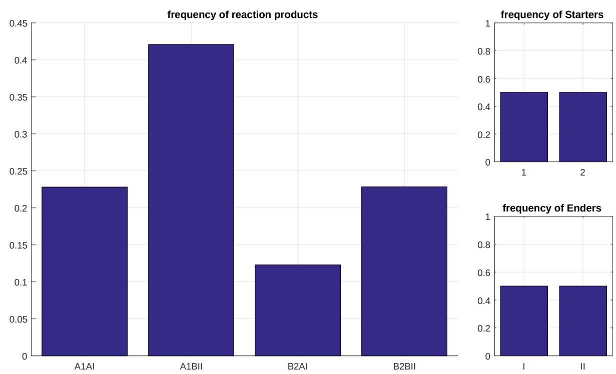


(b) A part of the nanoparticles surface with proteins A (red) and B (blue).

Figure 5.1: Result of a case where everything is equal, on average one arm per protein and no affinities. The A1AI label can be read as AA, but to find the corresponding frequency of starters for A one needs to look at the 1.



(a) Three tags on A1 and BII



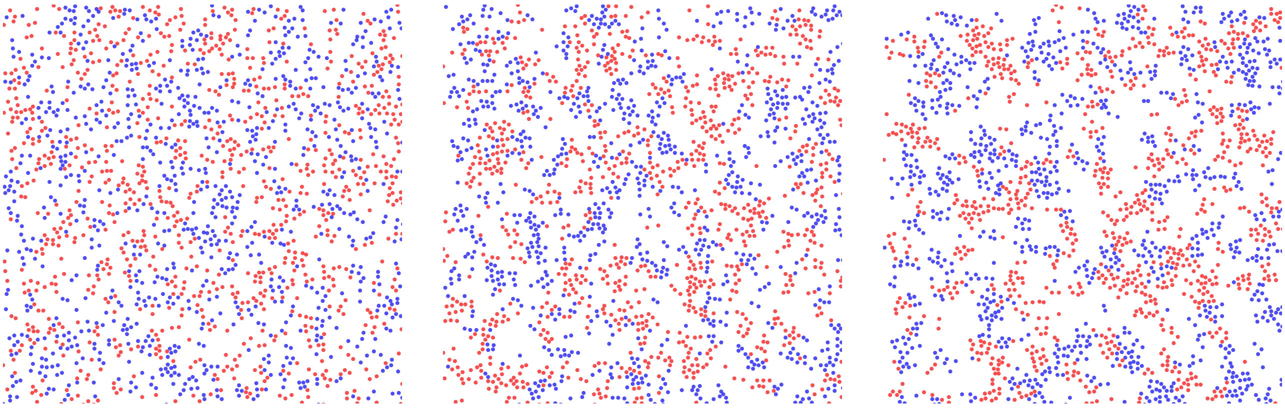
(b) Thirty tags on A1 and BII

Figure 5.2: The same as for Figure 5.1 but now starter A1 and ender BII have three and thirty tags respectively instead of one on average.

to one. From Figure 5.4 it is clear that with the affinities between two proteins the number of interactions can be influenced directly. An increase in affinity between a pair of proteins will increase the number of interactions in the simulations.

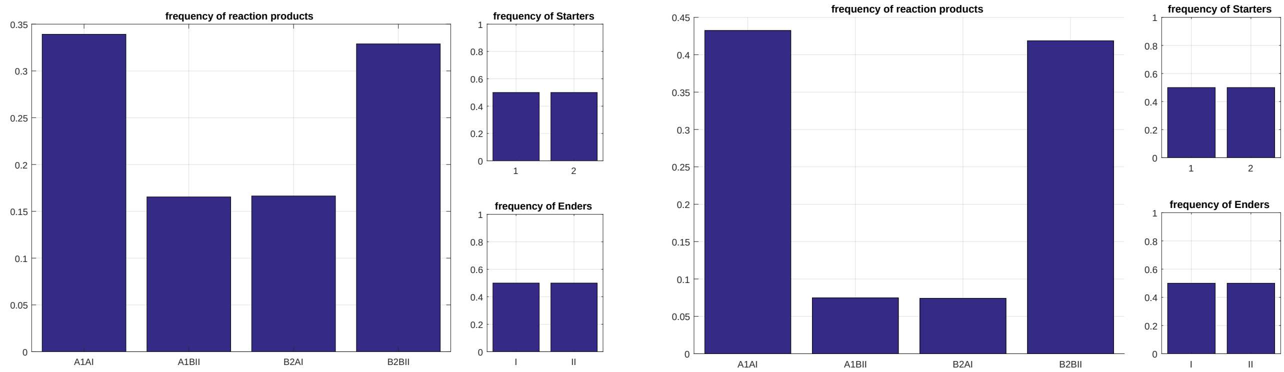
In Figure 5.5 we see the effect of increasing the affinities between all proteins. We see that this has a drastic effect on cluster formations on the surface, but also see that it has no effect on the relative frequencies of reaction products. This illustrates one of the challenges in the novel method already mentioned in Chapter 1. Namely the fact that from ratio's no information on concentrations or surface coverage can be obtained. From the result in Figure 5.5 it follows that also no information on the absolute attraction between proteins can be obtained. We will only be able to see relative affinities.

Lastly we mention that giving, for example, only AB a high affinity will result in getting high interaction frequencies between AB, but also between BA. This is explained by the fact that proteins first adsorb to the surface and then form pairs. Therefore, it does not matter if it is A that attracts B to the surface or the other way around. An A next to a B that interact has a fifty fifty change of being detected as an AB or BA.



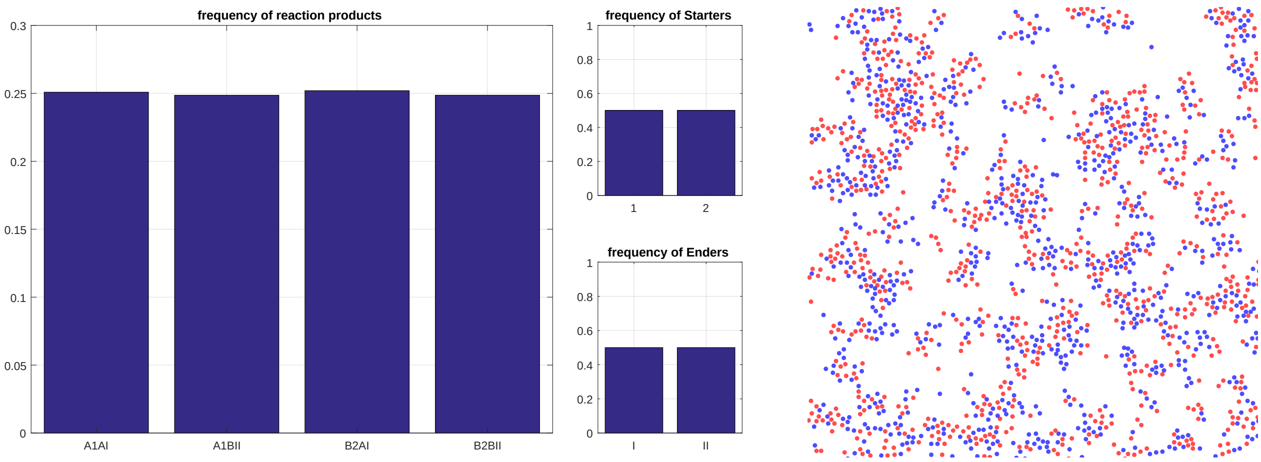
(a) $a_{A,A} = a_{B,B} = 10000$ (b) $a_{A,A} = a_{B,B} = 50000$ (c) $a_{A,A} = a_{B,B} = 100000$

Figure 5.3: The effect of the affinities on protein locations on the surface, where every protein is attracted to its own kind. In this case A attracts A and B attracts B.



(a) $a_{A,A} = a_{B,B} = 10000$ (b) $a_{A,A} = a_{B,B} = 100000$

Figure 5.4: The bar charts corresponding to Figures 5.3a and 5.3c respectively. We can see that the affinities influence the relative frequencies of reaction products directly.



(a) Frequencies of the starters, enders and reaction products.

(b) A part of the nanoparticles surface with proteins A (red) and B (blue).

Figure 5.5: In this case all affinities were equal but high. We see clustering on the surface but no effect in the bar charts. The used affinity between every protein is 100000. The average number of tags is equal to one per protein.

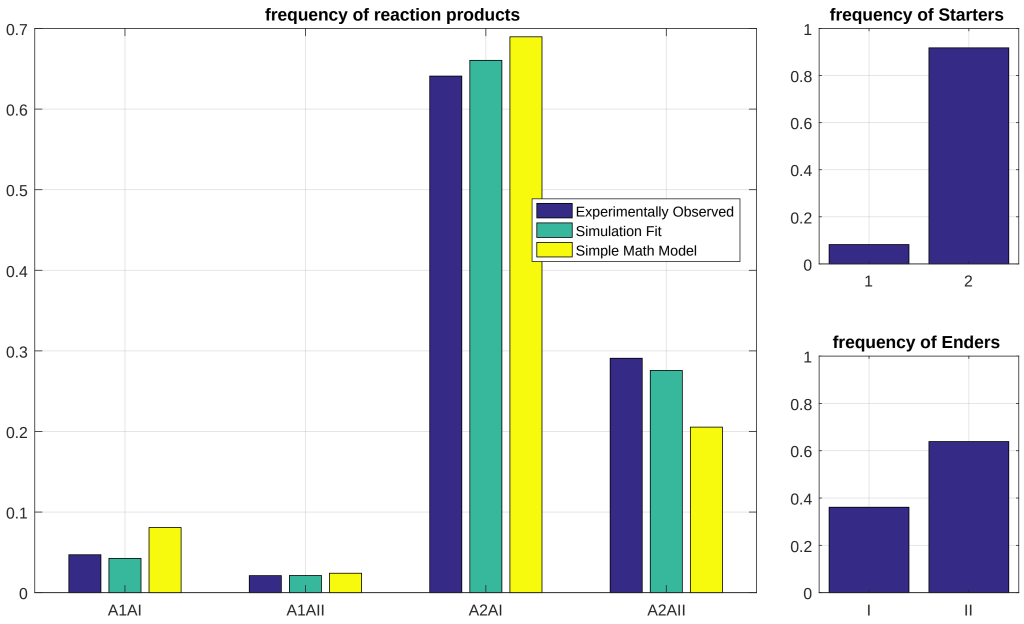


Figure 5.6: The observed values, fitted values and predicted values by the simple math model for protein A at a high concentration.

5.2 Number of Tags Fitting

In this section we look at the fitting of the number of tags. The tags were fitted based on experiments with only A or only B proteins. For every case, experiments with two different concentrations were used. We will refer to them as high and low concentrations. In Figure 5.6 we see the arm fitting results compared to the observed values and values predicted by the simple math model for protein A at a high concentration. We see that the fitted model is quite close to the observed values, but not equal. We also see that this fitted simulation is a big improvement over the basic math model.

In Tables 5.1 and 5.2 we see some of the results of the tags fitting summarized. Since the fitting procedure needs an initial guess different initial guesses were tried. An initial guess of 0.1 means that all protein-tag pairs will have on average 0.1 tags initially.

From the tables it is immediately apparent that the current fitting approach does not give unique best solutions. For almost every initial guess a different number of tags is found. Furthermore, the deviations between the fitted model and the observed values are still quite large, ranging from one to thirty two percent. We can therefore conclude that the fitting of the number of tags does not give us the accuracy we hoped for at the beginning of the project. To be able to fit the affinities a high accuracy in the number of tags is needed. An accuracy that, with the current approach, is not achieved. Therefore a different method to determine the number of arms will be necessary to make this approach successful.

It should be noted however that the initial guesses give the same trend as a result. Take for example the low concentration fitting of the A protein. We see that the A2 and AI remain unchanged in most cases and A1 is higher than the intitial guess and and AII is a bit higher than A1.

Concentration	Initial Guess	A1	A2	AI	AII	Max deviation (%)
High	0.1	0.063	0.1	0.1	0.14	32
	0.3	0.17	0.3	0.3	0.46	9
	0.7	0.39	0.7	0.58	1.01	1
	1	0.51	1.0	0.72	1.36	8
	2	0.9	2.0	1.39	3.96	4
Low	0.1	0.12	0.1	0.1	0.17	9.3
	0.3	0.37	0.3	0.3	0.46	13
	0.7	0.96	0.7	0.7	1.24	4
	1	1.49	1.0	1.0	2.072	5
	2	3.96	2.0	1.5	3.9	14

Table 5.1: The results of the number of tags fitting for the A protein. The numbers in the colums A1-AII are the mean number of tags per protein. The initial guess is the mean number of tags for every protein-tag pair.

Concentration	Initial Guess	B1	B2	BI	BII	Max deviation (%)
High	0.1	0.41	0.1	0.11	0.11	9
	0.3	2.22	0.3	0.31	0.3	4
	0.7	3.92	0.35	0.73	0.68	4
	1	3.96	0.37	1.03	1	6
	2	4	0.4	2.12	2	3
Low	0.1	0.46	0.1	0.1	0.1	21
	0.3	2.07	0.3	0.3	0.35	16
	0.7	3.98	0.37	0.7	0.9	17
	1	3.92	0.38	0.86	1.22	11
	2	3.99	0.41	1.72	2.82	15

Table 5.2: The results of the number of tags fitting for the B protein. The numbers in the colums B1-BII are the mean number of tags per protein.

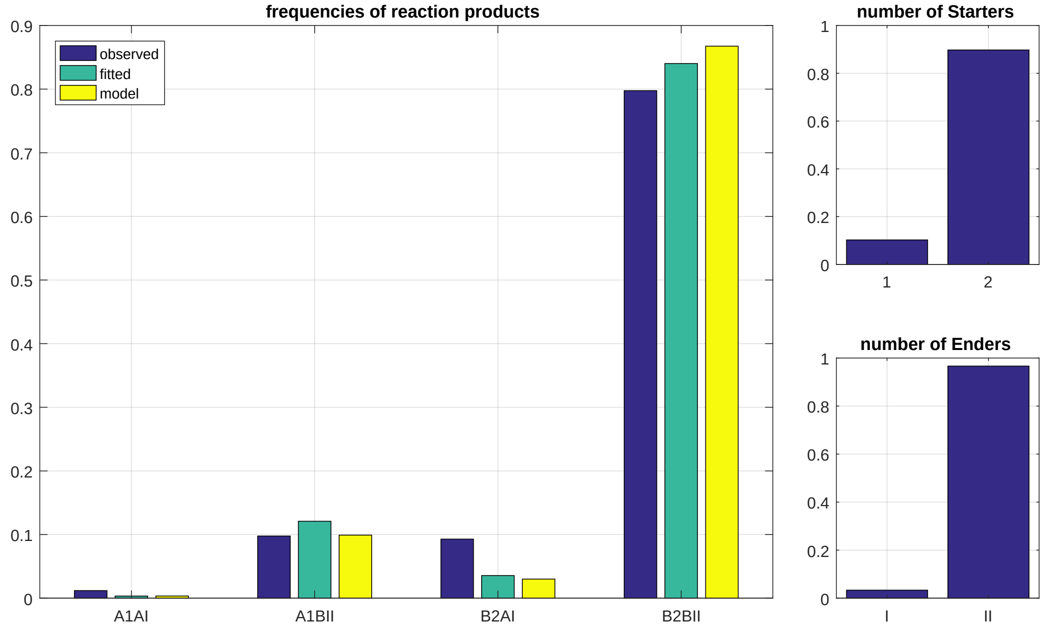


Figure 5.7: One of the experiments for the affinity fitting. We have the observed values from experiment, the fitted values by affinity fitting and the simple math model outcomes.

5.3 Fitting of Affinities

As we saw in the previous section the current number of tags fitting approach is not sufficient to continue with the affinity fitting from that data. Therefore we can not present any real results of fitting the affinities. Nonetheless, we perform an example fitting. For the example case we use different experiments with protein-tag pairs A1, B2, AI and BII to fit. However, for the average number of tags we don't use any data and just assume they are all equal to one. One of the experiments used in the fitting can be seen in Figure 5.7, note the extreme low values of the A1 and AI protein-tag pairs.

$a_{A,A}$	$5.10 \cdot 10^{13}$
$a_{A,B}$	100.00
$a_{B,A}$	$3.15 \cdot 10^{14}$
$a_{B,B}$	100.00

Table 5.3: Fitted Affinities: Based on an initial guess of 100.00 for each affinity

In the fitting process it turned out that these kind of extreme low values are problematic. In Figure 5.7 we also see the result of the fitting of the data, the fitted values for the affinity can be found in Table 5.3. We see that there are extreme high affinities, but that in the result of the fitting this makes little difference for the B2AI and B2BII interaction. This is explained by the fact that there is to little A1 and BI in the system. Even if all A1 and BI would react, there would still be more A2BII, simply because the A2 and BII protein-tag pairs are more present in the system. This means that it becomes near impossible to fit and find the affinities in the case that there is to little of some protein-tag pair in the system. Combining the fitting of affinities with the fitted number of tags, however, could still give a solution in some cases. For the current case if the number of tags for B2 and BII would be very low compared to A1 and AI a fit is again possible.

5.4 Conclusion

From the simulation results we saw that the number of tags influences the outcome of the model significantly. The simulation results also suggest that increasing the number of tags beyond the number of proteins to react with in range, has no effect on the model outcome.

This could potentially clear the problem of the number of tags introduced in Chapter 1, where we presented, as one of the challenges, the determination of the number of tags per protein. If experiments can be conducted with enough tags per protein such that the protein can react with all its neighboring proteins, the necessity of knowing the exact amount of tags vanishes.

We also saw that increasing the affinity between two protein-tag pairs highly influences the frequency of the corresponding interaction. However, when all affinities are increased we see no effect on interaction frequencies. This is an indication of the limitations of the current approach. We will see a difference in the relative affinities, but can not determine the extend of attraction between proteins.

From the tag fitting we learned that the current approach is not sufficient to fit the number of tags. We saw that the fitting depends to heavily on the initial guess. We also saw that even with the initial guess equal but a different concentration in the experiment the results differ. Which could be an indication of measuring errors, an inaccuracy of the fitting approach or a concentration dependence in the number of tags per protein. The latter is supported by the fact that in the experimental results large differences were observed in reaction products for the same experiment at different concentrations.

Lastly we saw that the current fitting approach for the affinities gave us a limitation of the current simulation. We found that if there are to little protein-tag pairs present of some kind or to much of another, it is impossible to fit the affinities. At the same time, combining the tag fitting and the affinities could give the desired result.

Chapter 6

Conclusion and Outlook

6.1 Conclusion

A novel method for the investigation of protein-protein interactions in the protein corona on nanoparticles has been proposed. The aim of this thesis was to investigate the possibility of the novel method to find protein-protein interactions and in particular attractions in the protein corona.

To do so a simple math model was derived to describe the experiment. From the model we saw that if all protein-tag pairs would behave equal and independently the reaction product frequencies would scale with the product of the protein-tag pair frequencies forming the reaction product. The simple math model was compared to the data and it was observed that a large deviation between the simple model and data was present. This is an indication of the protein-protein attractions in the corona.

In Chapter 3 we discussed the effect of systematic measurement errors and in particular the phenomenon called "mislabeling". It was found that mislabeling or a systematic error in the measurement of the protein-tag pairs was very unlikely to be the cause of the deviations observed between the data and the simple model.

After excluding the measurement errors as a source of the deviations, we investigated the role of the number of tags and affinities. For this we proceeded numerically. A simulator was build which could account for the number of tags on a protein and affinities (attractions) between proteins. The general project approach was as follows. Starting from a set of special experiments in which affinities were excluded the number of arms could be fitted. Given the number of arms from the fit, we could fit the affinities based on other experiments and thus obtaining the attraction between the proteins in the experiment.

During the fitting procedure difficulties were encountered. It turned out that the fitting of the number of tags had non-unique solutions and that the outcome of the fitting depended heavily on the initial guesses chosen. Furthermore, it was observed that fitting the affinities without considering the arms presents difficulties if certain protein-tag pairs are present with very small frequencies.

Overall the main question of this thesis can not be answered at this point. We have seen that mislabeling is not the cause of the deviations observed. At the same time, we were not able to show that the deviations were caused by the number of tags per protein or the affinities between proteins. This was mainly due to the lack of knowledge on the number of tags per protein.

6.2 Outlook

In this thesis a start was given for the investigation of the novel method to investigate protein-protein interactions in the protein corona. The investigation is not finished at this point. Further work is necessary and could focus on the following points:

- The approach used throughout this thesis, of fitting the number af tags and then affinities, failed. This was mainly due to the lack of information on the number of tags per protein. In this work the tags were fitted on experiments with one protein. Therefore no knowledge was obtained on the ratio of the number of arms between an A or B protein. If this ratio could be obtained the current simulations could already be improved. Even better would be to have an absolute number on the number of tags per protein. In that case all uncertainty due to the number of arms vanishes. It should be noted that a successful determination of the affinities without proper knowledge on the number of tags per protein is very unlikely.

- A different solution to the "number of tags"-problem was found through the simulation. Conducting experiments with an excess of tags would imply that every protein can react with every other protein in range. Therefore, the number of tags becomes irrelevant for the determination of the number of reaction products. This implies that affinities could be fitted directly on the obtained experimental results.

- In our approach fitting procedures were only run for single datasets and we were mostly over fitting. In future work a way of fitting multiple datasets to the same simulation is necessary, to get more certainty on the obtained results. This also means that multiple data sets under the same circumstances are needed.

- To check the validity of the novel method another experimental method is necessary to check the results obtained. The main idea of the novel method is to test multiple proteins on the surface. The verifying method only has to be able to measure two, since it is only a verification.

- Some effort should be made to investigate the ratio between starters and enders. Throughout this thesis 50% starters and enders were assumed, but this is only based on guesses. An exact number or estimate of the actual ratio would also improve the simulations.

Bibliography

[1] P. del Pino, B. Pelaz, Q. Zhang, P. Maffre, G. U. Nienhaus, and W. J. Parak, "Protein corona formation around nanoparticles – from the past to the future," *Mater. Horiz.*, vol. 1, no. 3, pp. 301–313, 2014.

[2] S. K. Murthy, "Nanoparticles in modern medicine: state of the art and future challenges." *International journal of nanomedicine*, vol. 2, no. 2, pp. 129–41, 2007.

[3] C. D. Walkey and W. C. W. Chan, "Understanding and controlling the interaction of nanomaterials with proteins in a physiological environment," *Chem. Soc. Rev.*, vol. 41, no. 7, pp. 2780–2799, 2012.

[4] D. Docter, U. Distler, W. Storck, J. Kuharev, D. Wünsch, A. Hahlbrock, S. K. Knauer, S. Tenzer, and R. H. Stauber, "Quantitative profiling of the protein coronas that form around nanoparticles," *Nat. Protocols*, vol. 9, no. 9, pp. 2030–2044, sep 2014.

[5] Genetic Science Learning Center., "PCR," 2013. [Online]. Available: http://learn.genetics.utah.edu/content/labs/pcr/

[6] A. Baddeley, "Spatial Point Processes and their Applications," in *Stochastic Geometry: Lectures given at the C.I.M.E. Summer School held in Martina Franca, Italy, September 13–18, 2004*, W. Wolfgang, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, vol. 1892, pp. 1–75.

Appendices

Appendix A

Examples of the measurement errors tested in Chapter 3

A.1 Mislabeling: An Example

In Section 3.2 we discussed "mislabelmatrices" obtained by an iterative scheme. We used mislabelmatrices obtained from one concentration in a set of experiments to see whether they would improve the other experiments in the same set. To quantify "improve" we will use the mean absolute deviation in percent. This is, the absolute value of the percent deviation averaged over all products. To see whether we are dealing with an improvement or a deterioration, we compare the mean absolute deviation between the independent model with mislabeling and the observed values and between the independent model and the observed values.

In Table A.1 we present the results of an experiment that shows all observed behaviors. We see the mean absolute percent deviation with mislabelmatrices obtained from all four concentrations. From the results it follows that the mislabelmatrix obtained by the iterative scheme is very successful in making a perfect fit between observed data and our model, leaving a mean error of less then 0.001% in all cases. It also follows that in some cases the mislabelmatrix improves all other experiments, for example the mislabel matrix obtained from concentration 2. In other cases the mislabelmatrix deteriorates the fits of the other experiments, which is the case for the mislabel matrix obtained from concentration 4. And the mislabelmatrices obtained from concentration 1 and 3 improve some and deteriorate another.

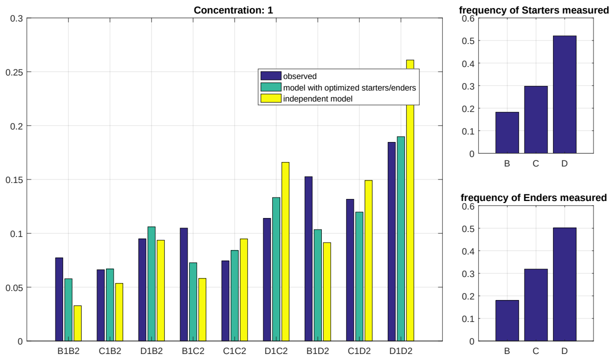
Concentration	Used con. 1	Used con. 2	Used con. 3	Used con. 4	independent
1	<0.001	54.68	26.73	118.36	75.82
2	133.48	<0.001	112.32	336.14	49.39
3	37.29	42.63	<0.001	105.19	74.92
4	40.32	53.53	26.79	<0.001	99.54

Table A.1: Mean absolute deviations in percent (%) between the model corrected for mislabeling with the mislabelmatrix from Section 3.2 and observed values. "Used con. x" means that the mislabelmatrix used is the matrix obtained from concentration x. In the last column the deviations observed for the simple independent math model are presented as a comparison. A hundred cycles were used to obtain the matrices.

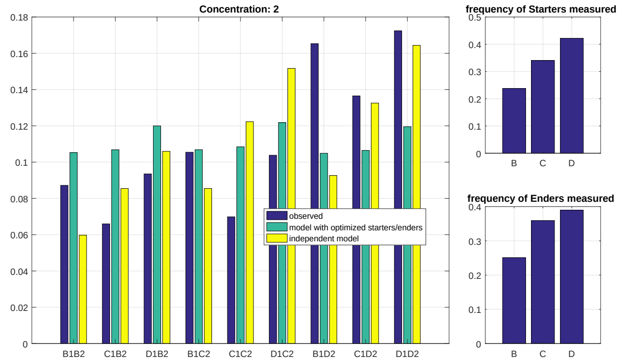
A.2 Another Type of Measurement Error: An Example

Here we show an example of the test mentioned in Section 3.3. In general every experiment was performed at four different concentrations, which we have named 1 through 4. The idea of the test is to fit one set of coefficients, c_k's, to an experiment at one of the four concentrations. For this example fit, the coefficients were fitted to the first concentration (concentration 1). We use these coefficients to predict the outcome of the experiments at concentrations 2, 3 and 4. The result of this approach can be seen in Figure A.1.

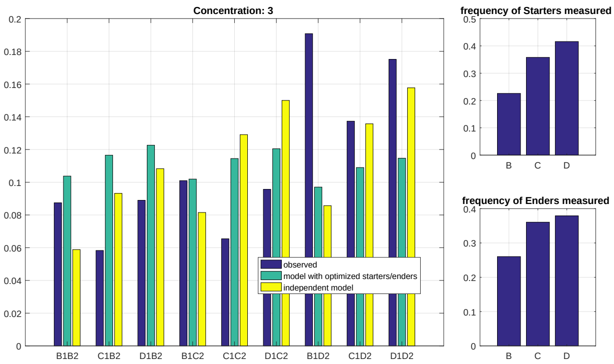
As a measure for the improvement caused by the coefficients, we compare the mean absolute deviation between the observed values and the independent model from Chapter 2 with the mean deviation between the observed values and the independent model corrected with the coefficients, c_k's. The result of this can be found in Table A.2 where we see that only the experiment from which the coefficients were obtained benefits from the coefficients found. For the experiments at different concentrations almost no improvement is observed compared to the independent model and sometimes even a deterioration. Fitting the c_k's to another concentration, for example the fourth, we obtain a different result, see Table A.3. Where every concentration is improved, although only slightly. Similar behavior was observed for all experiments, in some cases there was a concentration that improved all experiments in a set a bit, and in other cases no improvement or a deterioration was observed.



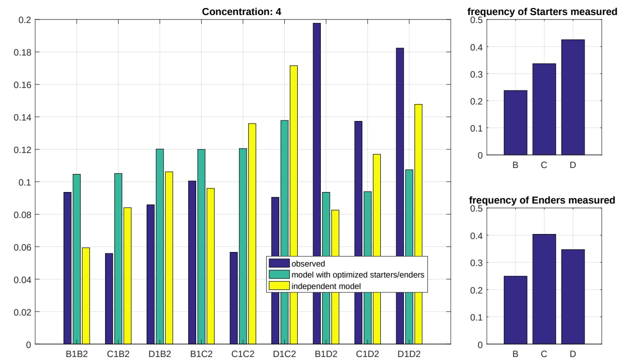
(a) Concentration 1



(b) Concentration 2



(c) Concentration 3



(d) Concentration 4

Figure A.1: The result of fitting the coefficients c_k to the first concentration and then using the coefficients to predict the outcome of the experiments at concentrations 2 through 4

optimized vs observed (%)	independent vs observed (%)
15.87	32.28
30.46	29.56
40.27	39.30
50.02	48.57

Table A.2: Mean deviations in percent between the fitted (optimized) model and observed values and between the independent model and observed values. The coefficients used for this result were obtained by fitting on the first concentration.

optimized vs observed (%)	independent vs observed (%)
19.81	32.28
23.91	29.56
32.92	39.30
38.99	48.57

Table A.3: Mean deviations in percent between the fitted (optimized) model and observed values and between the independent model and observed values. The coefficients used for this result were obtained by fitting on the fourth concentration.