

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Matija Čufar

**Modeliranje kinetike vezave  
monoklonskih protiteles na receptor  
 $Fc\gamma RIIIa$**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM  
PRVE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: prof. dr. Blaž Zupan

SOMENTOR: prof. dr. Neža Mramor Kosta

Ljubljana, 2016

*Besedilo je oblikovano z urejevalnikom besedil L<sup>A</sup>T<sub>E</sub>X.*

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Modeliranje kinetike vezave monoklonskih protiteles na receptor  $Fc\gamma RIIIa$   
Kinetic modelling of binding between monoclonal antibodies and the  $Fc\gamma RIIIa$   
receptor

V nalogi razvijte matematični model, ki popiše dinamiko izbranega biološkega sistema in, konkretno, obravnava vezavo protiteles na receptor. Predlagajte strukturo modela in poiščite optimizacijsko tehniko, ki na podlagi eksperimentalni meritev računsko učinkovito določi parametre modela. Modele ovrednotite na podatkih in poročajte o uspešnosti modeliranja in optimizacijskih pristopov.













# Kazalo

**Povzetek**

**Abstract**

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Pregled sorodnega dela</b>	<b>3</b>
2.1	Modeliranje v biologiji . . . . .	3
2.2	Primer računskega modeliranja . . . . .	5
2.3	Ocenjevanje parametrov modela iz podatkov . . . . .	6
<b>3</b>	<b>Opis ozadja problema in metode pridobivanja podatkov</b>	<b>7</b>
3.1	Protitelesa in njihovo delovanje . . . . .	7
3.2	Interferometrija z biološkimi plastmi. . . . .	9
3.3	Predstavitev podatkov . . . . .	9
<b>4</b>	<b>Model in metoda ocene parametrov iz podatkov</b>	<b>13</b>
4.1	Matematični model interakcije . . . . .	13
4.2	Ocena parametrov iz podatkov . . . . .	17
<b>5</b>	<b>Rezultati in diskusija</b>	<b>21</b>
5.1	Rezultati . . . . .	21
5.2	Preverjanje robustnosti metode . . . . .	26

**6 Zaključek**

**33**

**Literatura**

**35**

# Povzetek

**Naslov:** Modeliranje kinetike vezave monoklonskih protiteles na receptor  $Fc\gamma RIIIa$

**Avtor:** Matija Čufar

**Povzetek:** Matematično modeliranje postaja vedno bolj pomembno področje sodobne biologije. V diplomskem delu smo razvili model, ki opisuje dinamiko vezave med dvema vrstama protiteles in receptorjem  $Fc\gamma RIIIa$ . Naš cilj je bil najti metodo, ki bo iz eksperimentalnih podatkov določila parametre modela. Za najboljšo metodo se je izkazal Levenberg-Marquardtov algoritem, ki deluje zelo učinkovito in zanesljivo ter poišče parametre modela tako, da se obnašanje modela dobro ujema s podatki. Rezultati metode predstavljajo ocene pomembnih fizikalnih parametrov, ki bi se lahko izkazali za uporabne pri karakterizaciji biološkega zdravila.

**Ključne besede:** matematično modeliranje, imunski sistem, monoklonsko protitelo, receptor, interferometrija z biološkimi plastmi, optimizacija, Levenberg-Marquardtov algoritem, prilagajanje funkcij.



# Abstract

**Title:** Kinetic modelling of binding between monoclonal antibodies and the Fc $\gamma$ RIIIa receptor

**Author:** Matija Čufar

**Abstract:** Mathematical modelling is becoming an increasingly important part of life sciences. This thesis deals with a mechanistic biological model of binding between two kinds of antibodies and the Fc $\gamma$ RIIIa receptor. Our goal was to develop a method that would determine the model parameters to match experimental data. The best method turned out to be the Levenberg-Marquardt algorithm, which is surprisingly reliable and fitted the model to the data well. The results give us an estimate of important physical parameters of the antibody that can be useful in the characterization of a biopharmaceutical drug.

**Keywords:** mathematical modelling, immune system, monoclonal antibody, receptor, bio-layer interferometry, optimization, Levenberg-Marquardt algorithm, curve fitting.



# Poglavje 1

## Uvod

Priča smo hitremu razvoju matematičnega modeliranja v biologiji. To je med drugim omogočil razvoj tehnologij, sploh na področju merilnih tehnik in zajema ter skladiščenja podatkov. Da bi te eksperimentalne podatke razumeli in uporabili pridobljeno znanje, jih je potrebno modelirati [1].

Opis pojava z matematičnim modelom omogoči globlje razumevanje, saj v modelu poskušamo zajeti njegove najbolj pomembne dejavnike. Pogosto se lahko s pomočjo mehanističnih modelov tudi izognemo kompleksnim, dražjim in zamudnim eksperimentalnim metodam. Model namreč omogoča, da neko lastnost procesa določimo, ne da bi jo merili neposredno.

Primer biološkega procesa, kjer bi nam modeli omogočili vpogled v podatke, je vezava protitelesa na receptor, kot ga preučujejo v farmacevtskem podjetju Lek d.d. na Oddelku za biofarmacevtiko. S pomočjo matematičnega modela bomo iz meritev, izmerjenih pri interferometriji z biološkimi plastmi, angl. *Bio-layer interferometry* ali BLI [2], določili pomembne lastnosti določenega protitelesa.

V tem diplomskem delu bomo predstavili mehanistični model, ki opisuje interakcijo med dvema vrstama monoklonskih protiteles in receptorjem Fc $\gamma$ RIIIa. Z njim lahko iz podatkov, izmerjenih z metodo BLI, napovemo konstante kinetike vezave in razmerje med količinama obeh različnih protiteles v vzorcu. Predstavili bomo tudi metodo prilagajanja funkcij, s katero

bomo parametre določili. Ocenili bomo tudi natančnost in zanesljivost metode.

Parametre kinetike vezave protiteles na receptor, ki jih opisuje predstavljen model, predstavljajo lastnosti, ki so pombembne za karakterizacijo protiteles. Uporabljajo se kot eden izmed dejavnikov v procesu razvoja biološkega zdravila, bolj natančno v procesu izbire celičnih klonov, pri katerem se iz nabora različnih celic, izbere tisto, ki je najbolj primerna za nadaljni razvoj. Uspešen izračun parametrov in s tem določitev modela bi proces izbire klonov pohitрил, saj je metoda BLI, s katero pridobivamo podatke, ki jih modeliramo, preprostejša, cenejša in hitrejša od nekaterih drugih metod, s katerimi lahko določimo iste lastnosti proteina.



# Poglavje 2

## Pregled sorodnega dela

Z besedo “model” opisujemo več različnih, a povezanih konceptov. Lahko so fizični (npr. leseni modeli mostov) ali izmišljeni objekti (npr. Bohrov model atoma, ki je predstavljen podono kot sistem zvezde in planeta), strukture iz teorije množic, opisi, enačbe ali pa kombinacije teh stvari. V vseh teh primerih model predstavlja nek objekt ali pojav iz realnega sveta in nam pomaga pri njegovem razumevanju [3]. V tem razdelku se bomo osredotočili na matematično in računsko modeliranje v biologiji, opisovanje bioloških sistemov z uporabo enačb, algoritmov in podatkovnih struktur.

### 2.1 Modeliranje v biologiji

Modeliranje v biologiji združuje metode in spoznanja iz različnih naravoslovnih ved, kot so matematika, fizika, kemija in računalništvo in jih uporablja za opisovanje bioloških procesov. Modeliranje uporabljamo v najrazličnejših problemskih domenah, od modeliranja rasti populacij in interakcij med vrstami do opisovanja celičnih procesov in interakcij med posameznimi proteini [4].

Prepletanje med znanostmi o življenju in matematiko ima dolgo zgodovino. Znani matematiki in drugi znastveniki, kot so Pitagora, Aristotel, Fibonacci, Cardano, Bernoulli, Euler, Fourier, Laplace, Gauss, von Helmholtz,

Riemann, Einstein, Thompson, Turing, Wiener, von Neumann, Thom, in Keller so na področju biologije uporabljali matematične metode, nekateri izmed njih pa so iz živih sistemov dobili navdih za odkritje novih znanj v matematiki [5].

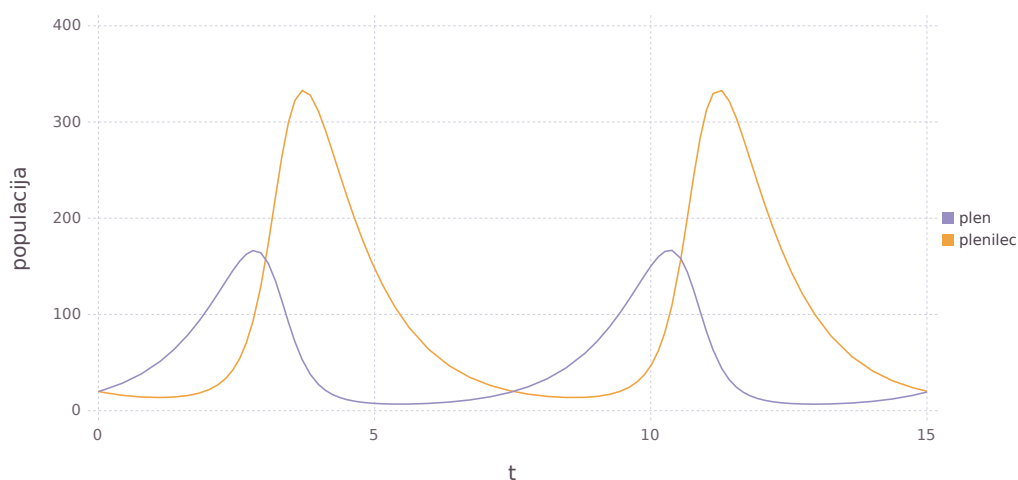
Eden izmed bolj pomembnih zgodnjih matematičnih modelov je razlaga dedovanja lastnosti med križanimi rastlinami, ki jo je leta 1866 razvil Gregor Mendel [6]. To velja za prvo delo na področju genetike. S svojim modelom je opisoval dedovanje sedmih karakteristik rastlin graha: višino rastline, obliko in barvo stroka, obliko in barvo ploda in pozicijo in barvo cveta. Ugotovil je, da ima potomec rastline z zelenim plodom in rastline z rumenim plodom vedno rumen plod, v naslednjih generacijah pa se včasih ponovno pojavi zelen. Pojav je razložil z dominantnimi in recesivnimi faktorji, ki jih danes imenujemo geni.

Čeprav je bilo modeliranje v biologiji prisotno že od samega začetka razvoja te znanstvene vede, je bilo do pred kratkim med biologi veliko bolj razširjeno kvalitativno opisovanje pojavov. Matematično in računsko modeliranje je postalo zares razširjeno šele proti koncu 20. stoletja z razvojem systemske, računske in matematične biologije in z razvojem bioinformatike. Ta razvoj sta omogočila napredka v računalništvu in merilnih tehnikah, ki sta pospešila, pocenila in pohitrila pridobivanje in analizo podatkov [1, 5].

Ta nova področja biologije so nam omogočila doslej nepredstavljivo globoko in natančno razumevanje nekaterih bioloških procesov. Primeri teh spoznanj so opisi in simulacije metabolnih poti v celicah, ki jih je prinesel razvoj systemske biologije, preučevanje strukture in delovanja genov s katerimi se ukvarjata bioinformatika in računska biologija, modeli delovanja možganov iz nevroznanosti in natančni modeli delovanja in izločanja zdravil iz telesa iz farmakodinamike in farmakokinetike.

## 2.2 Primer računskega modeliranja

V tem razdelku bomo kot primer uspešne uporabe matematičnega modeliranja v biologiji predstavili preprost model dinamike populacij plenilca in plena. Model sta neodvisno razvila Alfred J. Lotka, ki ga je leta 1925 uporabil za opis kemijskih reakcij, in italjanski matematik Vito Volterra leta 1926, ki je z njim opisal oscilacije nekaterih vrst rib v Jadranskem morju [7, 8].



Slika 2.1: Primer dinamike plenilca in plena. Na abscisni osi je čas, na ordinatni pa je populacija.

Model opisuje dinamiko dveh vrst, kjer je ena plenilec in druga plen. Temelji na ideji, da večja populacija plena pomeni več hrane za plenilca, kar pomeni, da se lahko tudi plenilec bolj razmnoži, obratno pa večja populacija plenilca pomeni manjše možnosti za razmnoževanje plena. To povzroči nihanje v obeh populacijah.

Če z  $x(t)$  označimo populacijo plena (npr. zajcev) in z  $y(t)$  populacijo plenilca (npr. lisic), je Lotke-Volterrina enačba modela [4]:

$$\begin{aligned}\frac{dx}{dt} &= x(a - by) \\ \frac{dy}{dt} &= y(cx - d),\end{aligned}\tag{2.1}$$

kjer  $\frac{dx}{dt}$  in  $\frac{dy}{dt}$  predstavljata spremembi populacij  $x$  in  $y$  v času,  $a$ ,  $b$ ,  $c$  in  $d$  pa pozitivne parametre modela.

Nekatere rešitve teh enačb so periodične in spominjajo na nihanje, druge pa predstavljajo izumrtje ene od vrst. Primer nihajoče rešitve teh enačb je na sliki 2.1. Omenili bi še, da je ta model zelo preprost in da so bili kasneje razviti modeli, ki bolje opišejo dinamiko populacij.

## 2.3 Ocenjevanje parametrov modela iz podatkov

Ko enkrat imamo matematični model, za katerega se nam zdi, da bi lahko dobro opisal izbrani pojav, moramo oceniti vrednosti njegovih parametrov. Parametri modela so navadno spremenljivke v enačbah, ki predstavljajo lastnosti preučevanega sistema. V primeru tega diplomskega dela bomo določali parametre, ki predstavljajo lastnosti protiteles.

Parametre modela iz eksperimentalnih podatkov določimo tako, da poiščemo kombinacijo vrednosti parametrov, pri katerih se napovedi, ki jih določa model karseda dobro prilegajo eksperimentalnim podatkom. Čeprav je načinov za doseganje tega cilja več, je najpogostejši pristop za reševanje tega problema metoda najmanjših kvadratov.

Metoda najmanjših kvadratov je metoda, pri kateri iščemo take vrednosti parametrov, da je vsota kvadratov razlik med napovedmi modela in podatki najmanjša. Metoda je med starejšimi in najbolj uporabljanimi metodami sodobne statistike [9].

Problem najmanjših kvadratov rešujemo kot sistem enačb, njegova rešitev pa predstavlja oceno parametrov modela. Ta sistem enačb je lahko linearen ali nelinearen. Linearni sistemi enačb so lahko rešljivi, v primeru nelinearnih pa moramo uporabiti iterativne algoritme, ki delujejo tako, da začnejo z začetno oceno parametrov in jo z računanjem novih približkov izboljšujejo. Teh algoritmov je več, vsak pa ima svoje prednosti in slabosti [10].

## Poglavje 3

# Opis ozadja problema in metode pridobivanja podatkov

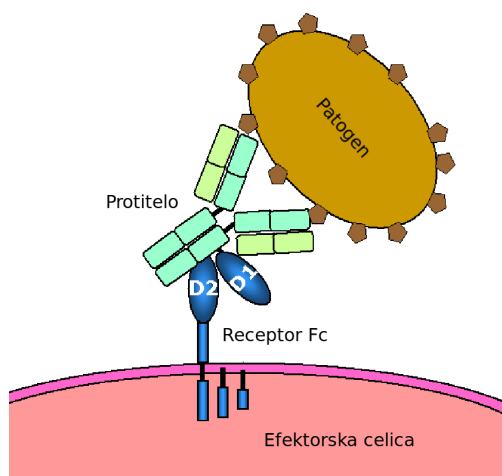
V tem poglavju bomo na kratko opisali ozadje problema, za katerega smo v nalogi razvili model. Predstavili bomo protitelesa in njihovo delovanje, nato pa nekaj povedali še o metodi s katero so bili izmerjeni podatki, na podlagi katerih smo ocenjevali parametre modela. Na koncu bomo tudi na kratko predstavili uporabljene podatke.

### 3.1 Protitelesa in njihovo delovanje

Protitelo ali imunoglobulin je protein, ki nastaja v plazemskih celicah (limfocitih) in telesu služi za prepoznavo in nevtralizacijo patogenov, kot so virusi in bakterije. To stori tako, da se na tujek ali okuženo celico veže in ga s tem onemogoči, ali pa proti njemu usmeri druge celice imunskega sistema.

Vezava protitelesa poteka na antigenu - molekuli, ki v telesu sproži imunski odziv. Antigen je lahko tudi del tujka, npr. del membrane bakterije, ki povzroča bolezen. Vsako protitelo se lahko veže samo na eno, točno določeno vrsto antigena, zato ima organizem za vsako telesu poznano bolezen svojo vrsto protiteles, vsaka od njih pa nastaja v specifični vrsti limfocita [11].

Protitelesa imajo značilno obliko, ki spominja na črko Y. Z zgornjima



Slika 3.1: Skica vezave protitelesa na patogen in efektorsko celico imunskega sistema.

dvema, lahkima verigama, se protitelo veže na antigen oziroma tarčo, s spodnjo, težko verigo pa na receptor Fc na celici imunskega sistema in jo s tem usmeri proti tarči. Proces vezave je opisan na sliki 3.1.

Monoklonska protitelesa so protitelesa, katere pridobivamo iz med seboj identičnih celic - klonov ene celice. S tem zagotovimo, da so vsa pridobljena protitelesa enaka in tako delujejo na isti antigen. Monoklonska protitelesa se uporabljajo pri terapiji različnih bolezni, kot so nekatere vrste raka in nekatere avtoimunske bolezni [11].

Čeprav imajo vsa protitelsa, ki nastanejo v klonih iste celice, enako kemijsko strukturo, se v procesu glikozilacije, ki poteka v celici, nanje vežejo glikani, velike skupine preprostih sladkorjev (monosaharidov). Količino in vrsto glikanov vezanih na protitelo imenujemo glikozilacijski profil protitelesa. Ta ima velik vpliv na njegovo kinetiko vezave [12]. Glikozilacijski profil predstavlja razliko med dvema vrstama protitels, ki jih opisuje model predstavljen v tem diplomskem delu.

## 3.2 Interferometrija z biološkimi plastmi.

Interferometrija z biološkimi plastmi, angl. *Bio-layer interferometry* ali BLI temelji na interakciji med dvema proteinoma, enim pritrjenim na biosenzor in drugim, ki je raztopljen v pufru. Protein, ki je pritrjen na biosenzor imenujemo ligand, tistega, ki je raztopljen v pufru pa analit. Pufer je raztopina šibke kisline in njene konjugirane baze, ki preprečuje spremembe vrednosti pH. Meritev poteka v dveh fazah. V prvi fazi, asociaciji, biosenzor pomočimo v pufer z analitom, v drugi, disociaciji, pa isti senzor pomočimo v prazen pufer. V fazi asociacije se analit veže na ligand, v fazi disociacije se iz njega razveže. Med procesom vezave in razvezave merimo debelino plasti nabrane na senzorju in tako beležimo, kako poteka kinetika vezave [13].

V konkretnem primeru, ki ga modeliramo v naši nalogi, merimo vezavo vzorca protitelesa na receptor Fc $\gamma$ RIIIa. Vzorec vsebuje enaka protitelesa z različnimi glikozilacijskimi profili. Teh različic protitelesa je veliko, a v našem modelu zaradi visokih koncentracij in velikih vplivov na kinetiko vezave upoštevamo samo dve, ostale pa zanemarimo. Protitelesi sta pritrjeni na biosenzor, receptor pa je raztopljen v pufru.

## 3.3 Predstavitev podatkov

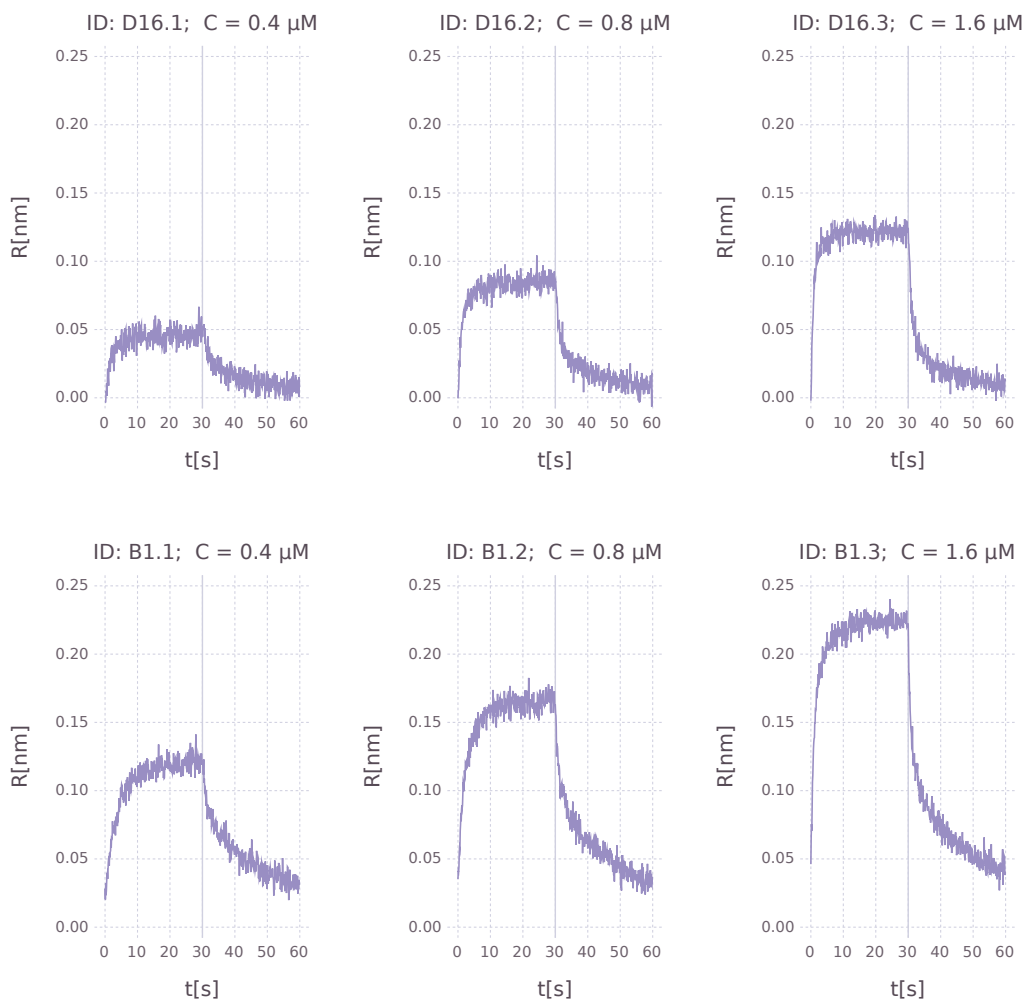
Eksperimentalni podatki, pridobljeni z metodo BLI, so množica meritev odziva biosenzorja v odvisnosti od časa, pomerjenih desetkrat na sekundo. Vsaka meritev vsebuje podatke obeh faz procesa, eno za drugo.

Imamo več vzorcev, ki vsebujejo ista protitelesa z različnimi glikozilacijskimi profili, v različnih razmerjih. Vsak vzorec pomerimo večkrat, pri tem pa vsako meritev izvedemo pri drugačni, vnaprej znani, koncentraciji receptorja.

V tem diplomskem delu bomo za primer izbrali množico podatkov, ki zajema 120 vzorcev. Vsak posamezen vzorec vsebuje protitelesa z enakimi glikozilacijskimi profili. Vsak od njih je pomerjen pri treh različnih koncentracijah receptorja. Vsaka meritev traja 60 sekund, od tega 30 sekund asociacije

in 30 sekund disociacije. Tako imamo skupno 360 različnih meritev, vsaka od njih pa vsebuje 600 podatkovnih točk. Primer meritev je prikazan na sliki 3.2.





Slika 3.2: Primer rezultatov eksperimentalnih meritev z metodo BLI. Disociacijska faza se začne v 30. sekundi. V vsaki vrstici so meritve istega vzorca (npr. D16 v prvi vrstici) pri različnih koncentracijah receptorja. Na posameznem grafu  $R$  predstavlja odziv biosenzorja,  $C$  pa koncentracijo receptorja v pufri.



# Poglavje 4

## Model in metoda ocene parametrov iz podatkov

V tem poglavju bomo najprej predstavili uporabljen matematični model, nato pa še metodo s katero bomo določili parametre tega modela.

### 4.1 Matematični model interakcije

Uporabili smo mehanistični model, ki opisuje vezavo med protitelesi in receptorjem, kot sta ga opisala D. J. O'Shanessy in D. J. Winzort [14]. Model je bil mišljen kot mehanizem za odpravljanje napak v meritvah v vzorcih z več kot enim protitelesom. Ker vemo, da naši vzorci vsebujejo več kot eno vrsto protitelesa, smo model uporabili in prilagodili za napovedovanje razmerja med količinama tistih dveh, ki imata na vezavo največji vpliv.

Model začnemo izpeljevati iz stehiometrijske enačbe, ki opisuje reakcijo med enim protitelesom in receptorjem:



kjer  $A$  predstavlja receptor,  $B$  protitelo,  $AB$  kompleks vezanega protitelesa na receptor, konstanti  $k_a$  in  $k_d$  pa konstanti asociacije in disociacije.

Reakcijo lahko zapišemo kot diferencialno enačbo:

$$\frac{dC_{AB}}{dt} = k_a C_{A(t)} (C_{B(tot)} - C_{AB(t)}) - k_d C_{AB(t)} , \quad (4.2)$$

kjer  $C_{AB}$  predstavlja koncentracijo kompleksa  $AB$ ,  $C_{A(t)}$  in  $C_{AB(t)}$  predstavljata koncentraciji analita  $A$  in kompleksa  $AB$  v času  $t$ ,  $C_{B(tot)}$  pa koncentracijo protitelesa  $B$ , ki je konstantna, saj je  $B$  imobiliziran na senzorju.

Med asociacijo predpostavimo, da se koncentracija analita ne spreminja. Tako dobimo enačbo:

$$\frac{dC_{AB}}{dt} = k_a C_A (C_{B(tot)} - C_{AB(t)}) - k_d C_{AB(t)} . \quad (4.3)$$

Ker je odziv biosenzorja  $R$  premo sorazmeren s koncentracijo  $C_{AB}$ , enačbo zapišemo kot:

$$\frac{dR}{dt} = k_a C_A (R_{max} - R_t) - k_d R_t , \quad (4.4)$$

kjer je  $R_{max}$  kapaciteta biosenzorja,  $R_t$  pa predstavlja odziv v časovni točki  $t$ .

Če enačbo integriramo in poenostavimo, dobimo:

$$R_a(k_a, k_d, R_{max}, C, t) = R_{max} \frac{C k_a}{C k_a + k_d} (1 - e^{-t(C k_a + k_d)}) , \quad (4.5)$$

kjer  $C$  predstavlja koncentracijo receptorja v pufru.

V našem konkretnem primeru imamo dva liganda, kar se obnaša podobno, kot če bi naenkrat opravljali dve meritvi in sešteli njuna odziva.

$$R_{asoc}(\mathbf{k}, R_{max1}, R_{max2}, C, t) = R_{max1} \frac{C k_{a1}}{C k_{a1} + k_{d1}} (1 - e^{-t(C k_{a1} + k_{d1})}) + R_{max2} \frac{C k_{a2}}{C k_{a2} + k_{d2}} (1 - e^{-t(C k_{a2} + k_{d2})}) , \quad (4.6)$$

kjer s  $\mathbf{k}$  označimo vektor parametrov kinetike modela  $[k_{a1}, k_{d1}, k_{a2}, k_{d2}]$ .

Uvedemo novo spremenljivko  $X = \frac{R_{max1}}{R_{max}}$ , ki predstavlja delež liganda  $A_1$ , kjer je  $R_{max} = R_{max1} + R_{max2}$ . Tako dobimo enačbo:

$$R_{asoc}(\mathbf{k}, R_{max}, X, C, t) = R_{max} \left( X \frac{C^{k_{a1}}}{C^{k_{a1}} + k_{d1}} (1 - e^{-t(C^{k_{a1}} + k_{d1})}) + (1 - X) \frac{C^{k_{a2}}}{C^{k_{a2}} + k_{d2}} (1 - e^{-t(C^{k_{a2}} + k_{d2})}) \right). \quad (4.7)$$

Krajše lahko enačbo zapišemo tudi z uporabo funkcije  $R_a(k_a, k_d, R_{max}, C, t)$  iz enačbe (4.6):

$$R_{asoc}(\mathbf{k}, R_{max}, X, C, t) = R_a(k_{a1}, k_{d1}, X R_{max}, C, t) + R_a(k_{a2}, k_{d2}, (1 - X) R_{max}, C, t) \quad (4.8)$$

Enačbo za fazo disociacije dobimo na podoben način, le da predpostavimo, da je koncentracija analita vedno enaka 0. Tako pridemo do enačbe:

$$R_d(\mathbf{k}, R_{max}, C, t) = R_a(k_a, k_d, R_{max}, C, t_1) e^{-(t-t_1)k_d}, \quad (4.9)$$

kjer  $t_1$  predstavlja točko v kateri se konča asociacija in začne disociacija, vrednost  $R_a$  v točki  $t_1$  pa predstavlja vrednost meritve v prvi časovni točki disociacije.

Spet je potrebno vzeti seštevek dveh takih enačb, tako da je celotna enačba oblike:

$$R_{disoc}(\mathbf{k}, R_{max}, X, C, t) = R_a(k_{a1}, k_{d1}, X R_{max}, C, t_1) e^{-(t-t_1)k_{d1}} + R_a(k_{a2}, k_{d2}, (1 - X) R_{max}, C, t_1) e^{-(t-t_1)k_{d2}}. \quad (4.10)$$

Če združimo obe enačbi dobimo celotno funkcijo modela:

$$R(\mathbf{k}, R_{max}, X, C, t) = \begin{cases} R_{asoc}(\mathbf{k}, R_{max}, X, C, t) & ; t < t_1 \\ R_{disoc}(\mathbf{k}, R_{max}, X, C, t) & ; t \geq t_1 \end{cases}. \quad (4.11)$$

Parameter modela  $R_{max}$  predstavlja maksimalno kapaciteto posameznega senzorja. Ker so senzorji nepopolni in ima vsak med njimi nekoliko drugačno maksimalno kapaciteto, ta parameter v sistem prinese nekaj šuma. Čeprav ima parameter pomembno vlogo pri obliki krivulje, je odvisen samo od merilne naprave in zato iz njega ne dobimo nobene informacije o vezavi protiteles na receptor.

S pomočjo modela iz danih eksperimentalnih podatkov poskušamo napovedati konstante kinetike  $k_{a1}$ ,  $k_{d1}$ ,  $k_{a2}$ ,  $k_{d2}$  in deleža protiteles  $X$  in  $1 - X$ .

Določanja parametrov smo se lotili z metodo najmanjših kvadratov. Če z  $n$  označimo število vzorcev in z  $m$  število ponovitev meritev vsakega vzorca pri različnih koncentracijah, z  $\hat{R}_{asoc}(t)$  in  $\hat{R}_{disoc}(t)$  pa označimo vrednosti meritev za asociacijo in disociacijo, je kriterijska funkcija za metodo najmanjših kvadratov enaka:

$$f(\mathbf{p}) = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} \sum_t \begin{cases} \left( \hat{R}_{asoc}(t_a) - R_{asoc}(\mathbf{k}, R_{max}^{(i+j)}, X^{(i)}, C^{(i)}, t_a) \right)^2 & ; t < t_1 \\ \left( \hat{R}_{disoc}(t_d) - R_{disoc}(\mathbf{k}, R_{max}^{(i+j)}, X^{(i)}, C^{(i)}, t_d) \right)^2 & ; t \geq t_1 \end{cases}, \quad (4.12)$$

kjer smo s  $\mathbf{k}$  označili vektor parametrov kinetike modela  $[k_{a1}, k_{d1}, k_{a2}, k_{d2}]$ , s  $\mathbf{p}$  pa smo označili vektor vseh parametrov:

$$\mathbf{p} = [k_{a1}, k_{d1}, k_{a2}, k_{d2}, R_{max}^{(0)}, R_{max}^{(1)}, \dots, R_{max}^{(i+j)}, X^{(0)}, X^{(1)}, \dots, X^{(i)}]. \quad (4.13)$$

Vrednosti  $C^{(i)}$  in  $t$  nismo vključili v  $\mathbf{p}$ , ker so del meritev in jih ni potrebno določiti.

V primeru naših podatkov s 120 vzorci in 360 meritvami tako določamo vrednosti za skupno 484 parametrov, od tega 4 parametre kinetike  $\mathbf{k}$ , 120 razmerij med količinama  $X^{(i)}$  in 360 maksimalnih kapacitet senzorjev  $R_{max}^{(i+j)}$ .

## 4.2 Ocena parametrov iz podatkov

Za oceno vrednosti parametrov iz podatkov smo uporabili metodo nelinearnih najmanjših kvadratov. To je metoda, pri kateri poskušamo minimizirati kvadrirane vrednosti razlik med meritvami in napovedmi modela. Te razlike imenujemo residuali in jih označimo z  $r(\mathbf{p})$ . V konkretnem primeru hočemo minimizirati kriterijsko funkcijo  $f(\mathbf{p})$  iz enačbe (4.12). Ker je kriterijska funkcija lepo odvedljiva in ker je bila Gauss-Newtonova metoda preveč občutljiva na izbiro začetnih parametrov, smo se odločili za uporabo Levenberg-Marquardtovega algoritma.

### 4.2.1 Gauss-Newtonova metoda

Gauss-Newtonov algoritem [15] je iterativna metoda za reševanje problema nelinearnih najmanjših kvadratov. Je približek Newtonovemu algoritmu, metodi za iskanje ničel funkcije.

Z Gauss-Newtonovim algoritmom minimiziramo vsoto kvadratov residualov  $S$ :

$$S(\mathbf{p}) = \sum_{i=1}^m r_i^2(\mathbf{p}) . \quad (4.14)$$

V našem primeru to vsoto predstavlja kriterijska funkcija  $f(\mathbf{p})$ , residuele pa predstavljajo vrednosti

$$r(\mathbf{p})_{(i,j,t)} = \begin{cases} \left( \hat{R}_{asoc}(t_a) - R_{asoc}(\mathbf{k}, R_{max}^{(i+j)}, X^{(i)}, C^{(i)}, t_a) \right)^2 & ; t < t_1 \\ \left( \hat{R}_{disoc}(t_d) - R_{disoc}(\mathbf{k}, R_{max}^{(i+j)}, X^{(i)}, C^{(i)}, t_d) \right)^2 & ; t \geq t_1 \end{cases} . \quad (4.15)$$

Če s  $\mathbf{p}^{(s)}$  označimo vrednosti parametrov v trenutni iteraciji, je vrednost parametrov v naslednji iteraciji algoritma določena z enačbo

$$\mathbf{p}^{(s+1)} = \mathbf{p}^{(s)} - \mathbf{J}^+ r(\mathbf{p}^{(s)}) , \quad (4.16)$$

kjer matrika  $\mathbf{J}$  predstavlja jakobijevo matriko funkcije  $R(\mathbf{p})$  iz enačbe (4.11),  $\mathbf{J}^+$  pa njen Moore-Penroseov inverz. Jakobijeva matrika je sestavljena iz parcialnih odvodov funkcije  $R$ . Njena vrednost v  $u$ -ti vrstici in  $v$ -tem stolpcu je določena kot:

$$\mathbf{J}_{uv} = \frac{\partial R_u(\mathbf{p}^{(s)})}{\partial \mathbf{p}_v}, \quad (4.17)$$

kjer je  $u = 1, 2, \dots, n \cdot m \cdot N_t$ , kjer je  $v = 1, 2, \dots, N_p$  in kjer  $N_t$  in  $N_p$  predstavljata število časovnih točk in število parametrov modela.

Ta iterativen postopek ponavljamo dokler se vrednost kriterijske funkcije  $f(\mathbf{p})$  zmanjšuje, ali pa do kakšnega drugega ustavitvenega pogoja, kot je npr. doseženo določeno število iteracij.

Algoritem za začetek iterativnega postopka potrebuje začetne parametre  $\mathbf{p}^{(0)}$ , ki jih je potrebno določiti ročno. Izbira dobrih začetnih parametrov je pomemben dejavnik, saj lahko ob neprimerni izbiri algoritem konvergira zelo počasi, konvergira v lokalni minimum z visoko vrednostjo kriterijske funkcije, ali pa celo divergira. Ker parametrov modela vnaprej ne poznamo, to lahko predstavlja velik problem. Ena od možnosti, s katero lahko omilimo pomembnost izbire teh parametrov je uporaba Levenberg-Marquardtovega algoritma.

## 4.2.2 Levenberg-Marquardtov algoritem

Levenberg-Marquardtov algoritem [16] je izboljšava Gauss-Newtonovega algoritma, ki občutno zmanjša njegovo občutljivost na izbiro pravih začetnih parametrov, njegova slaba lastnost pa je to, da v primerjavi z Gauss-Newtonovim algoritmom konvergira nekoliko počasneje.

Pri Gauss-Newtonovem algoritmu problem nastane, če je trenutna vrednost parametrov  $\mathbf{p}^{(s)}$  predač od minimuma. Levenberg-Marquardtov algoritem to rešuje z uporabo območja zaupanja (angl. *trust region* [17]), ki velikost koraka omeji na okolico trenutnih vrednosti parametrov. To stori



tako, da v enačbo (4.16) doda člen  $\lambda \mathbf{D}$ , kjer pozitiven skalar  $\lambda$  predstavlja velikost območja zaupanja, matrika  $\mathbf{D}$  pa diagonalo matrike  $\mathbf{J}^\top \mathbf{J}$ :

$$\mathbf{p}^{(s+1)} = \mathbf{p}^{(s)} - (\mathbf{J}^\top \mathbf{J} + \lambda \mathbf{D})^{-1} \cdot \mathbf{J}^\top r(\mathbf{p}^{(s)}) . \quad (4.18)$$

Parameter  $\lambda$  v vsaki iteraciji popravimo glede na to kako dober je bil prejšnji korak. Kvaliteto koraka ocenimo tako, da z linearno aproksimacijo ocenimo velikost koraka ki jo pričakujemo:

$$\Delta f_{pred}^{(s)} = \sum (r(\mathbf{p}^{(s)}) + \mathbf{J} \Delta \mathbf{p}^{(s)})^2 - f(\mathbf{p}^{(s)}) , \quad (4.19)$$

izračunamo dejansko vrednost koraka:

$$\Delta f^{(s)} = f(\mathbf{p}^{(s)} + \Delta \mathbf{p}^{(s)}) - f(\mathbf{p}^{(s)}) \quad (4.20)$$

in iz njiju izračunamo kvaliteto koraka  $\rho^{(s)}$ :

$$\rho^{(s)} = \frac{\Delta f^{(s)}}{\Delta f_{pred}^{(s)}} . \quad (4.21)$$

Večje vrednosti  $\rho^{(s)}$  predstavljajo dober korak, manjše pa slabega. Če je korak zelo dober, parameter  $\lambda$  pomnožimo s faktorjem  $\mu > 1$ , če je zelo slab, pa korak razveljavimo in  $\lambda$  delimo z istim faktorjem.

V naši implementaciji algoritma se za zelo dober korak smatra vrednost  $\rho > 0.75$ , za zelo slab korak pa je uporabljena vrednost  $\rho \leq 10^{-3}$ . Za vrednost faktorja  $\mu$  in začetno vrednost parametra  $\lambda$  smo izbrali 10. Implementacijo Levenberg-Marquardtovega algoritma, smo vzeli iz knjižnice Optim.jl za programski jezik Julia [18]. Algoritem smo tudi rahlo prilagodili našim potrebam.

Intuitivno si lahko Levenberg-Marquardtov algoritem predstavljamo kot mešanico Gauss-Newtonovega algoritma in gradientnega spusta. Algoritem se v primerih, ko ima parameter  $\lambda$  majhne vrednosti, obnaša kot Gauss-Newtonov algoritem, v primerih, ko je parameter  $\lambda$  velik, pa se obnaša kot gradientni spust [19].

### 4.2.3 Napake ocenjenih parametrov

Ko s procesom minimizacije končamo, lahko še ocenimo standardno napako izračunanih parametrov [19]. Najprej izračunamo oceno variance podatkov  $\sigma_\varepsilon^2$ :

$$\sigma_\varepsilon^2 = \frac{f(\mathbf{p})}{n \cdot m \cdot N_t - N_p}, \quad (4.22)$$

kjer  $n \cdot m \cdot N_t - N_p$  predstavlja število prostostnih stopenj,  $f(\mathbf{p})$  pa predstavlja kriterijsko funkcijo.

Ko imamo oceno za  $\sigma_\varepsilon$  izračunamo kovariančno matriko parametrov  $\mathbf{C}$ :

$$\mathbf{C} = \frac{\sigma_\varepsilon^2}{2} (\mathbf{J}^\top \mathbf{J})^{-1}. \quad (4.23)$$

Diagonala te matrike predstavlja variance vsakega posameznega parametra, standardna napaka  $i$ -tega parametra pa je enaka:

$$\sigma_i = \sqrt{(\text{diag}(\mathbf{C}))_i} \quad (4.24)$$

S standardno napako parametrov lahko ocenimo, kako natančno smo določili vrednost parametrov glede na eksperimentalne podatke.

# Poglavje 5

## Rezultati in diskusija

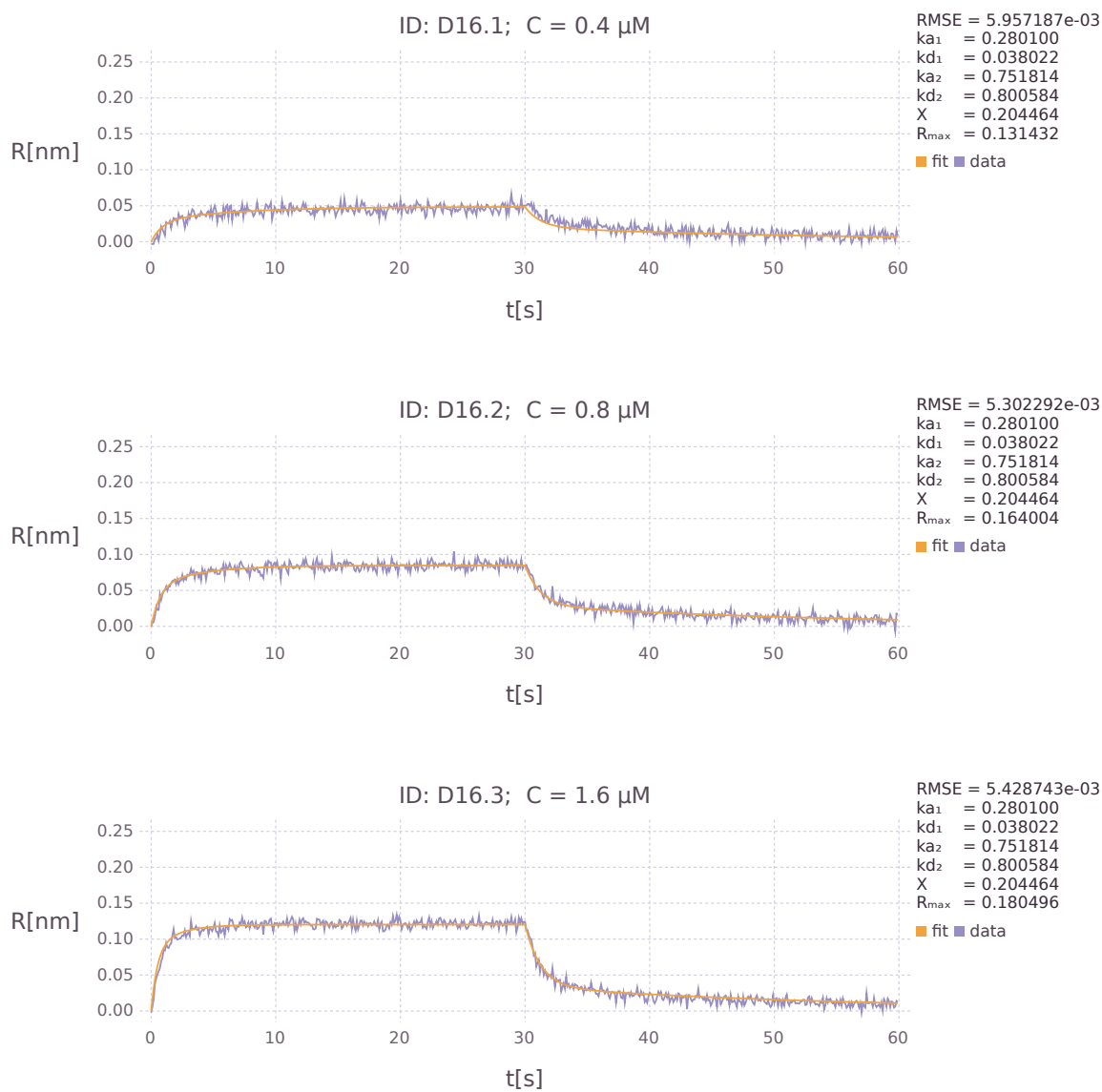
V tem poglavju bomo predstavili rezultate in jih ovrednotili. Ocenili bomo algoritmovo učinkovitost in zanesljivost pri izbiri začetnih parametrov in pri količini podatkov, ki so algoritmu na voljo.

### 5.1 Rezultati

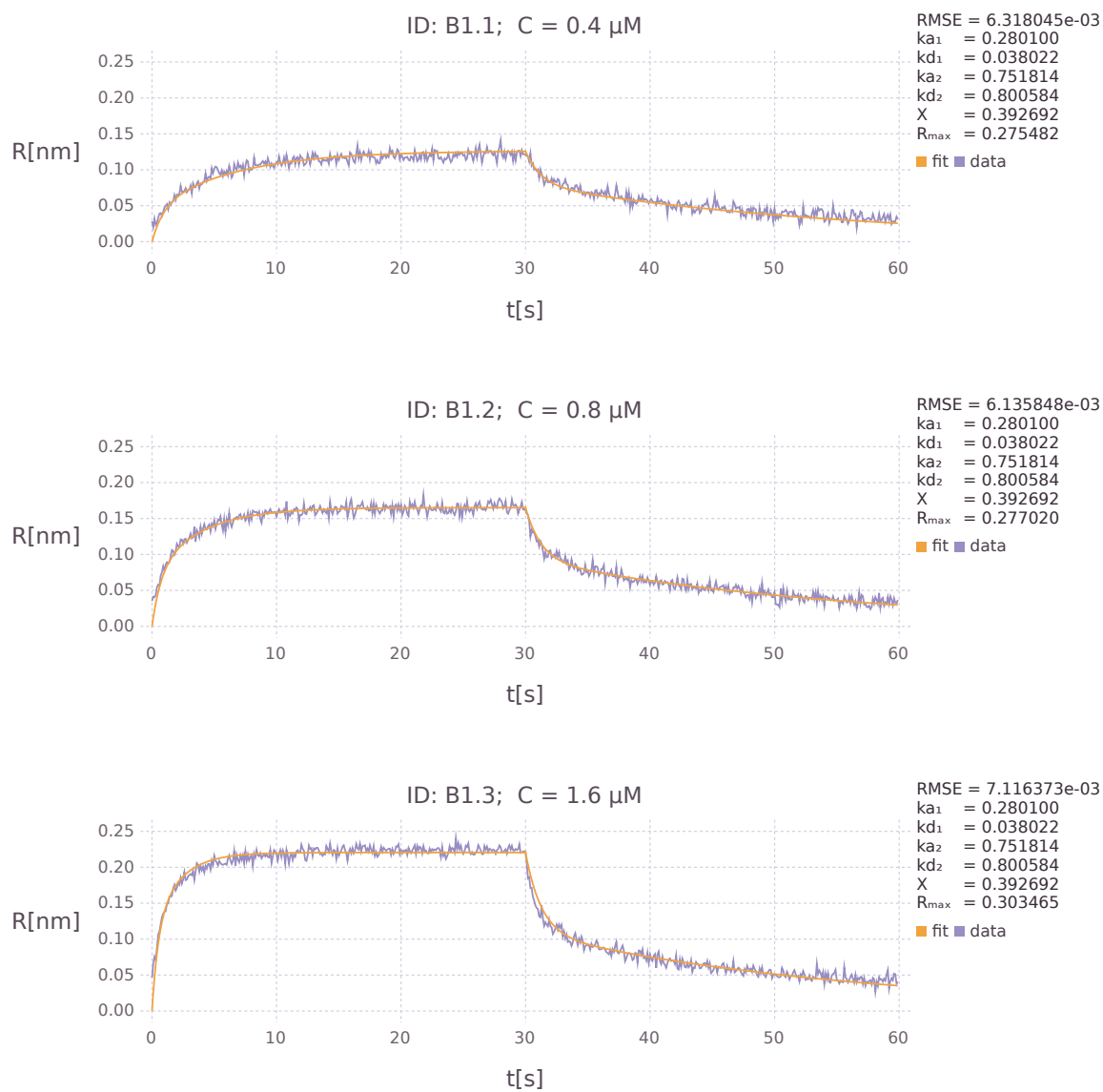
Pri uporabi Levenberg-Marquardtovega algoritma na podatkih smo dobili dobre rezultate, kar vidimo iz slik 5.1, 5.2, 5.3 in 5.4. Na sliki 5.3 je prikazana meritev, ki je imela najvišjo vrednost kriterijske funkcije, na sliki 5.4 pa tista z najnižjo. Vrednost kriterijske funkcije za vse podatke skupaj je bila  $f(\mathbf{p}) = 8.03$ , kvadratni koren povprečne kvadratne napake (root mean square error ali RMSE) pa je bil  $\text{RMSE} = 0.0061$ . Lahko še izračunamo determinacijski koeficient med napovedanimi in izmerjenimi vrednostmi in dobimo  $R^2 = 0.996$ . Korelacijo med izmerjenimi in napovedanimi vrednostmi prikazuje slika 5.5.

Izračunane konstantne kinetike protiteles so bile  $k_{a1} = 0.28$ ,  $k_{d1} = 0.04$ ,  $k_{a2} = 0.75$  in  $k_{d2} = 0.80$ . Iz teh parametrov vidimo, da glavna razlika med protitelesoma tiči v konstantah  $k_d$ .

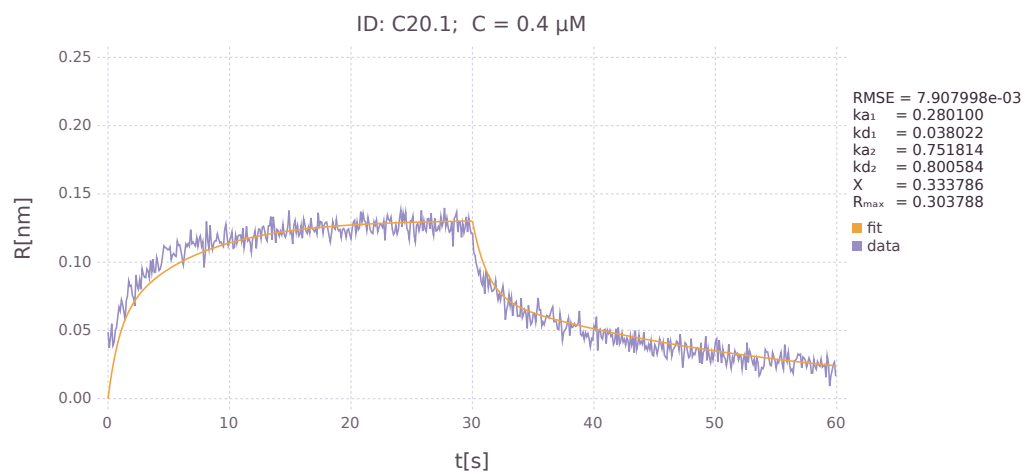
Na grafu 5.6 sta prikazana histograma vrednosti  $R_{max}$  in  $X$ . Parameter  $R_{max}$  izgleda porazdeljen približno normalno, kar smo od naključno



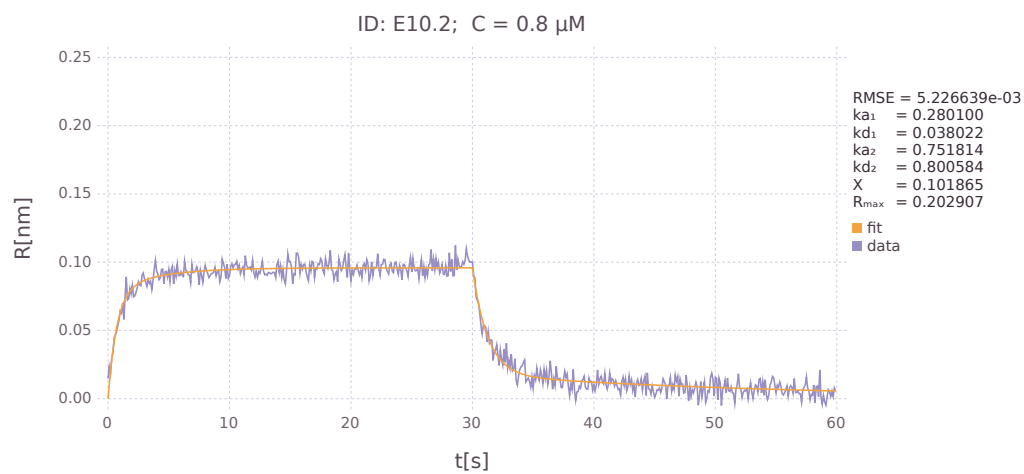
Slika 5.1: Primer prilagoditve funkcije modela na podatke. Uporabljeni so bili isti podatki, kot v prvi vrstici na sliki 3.2.



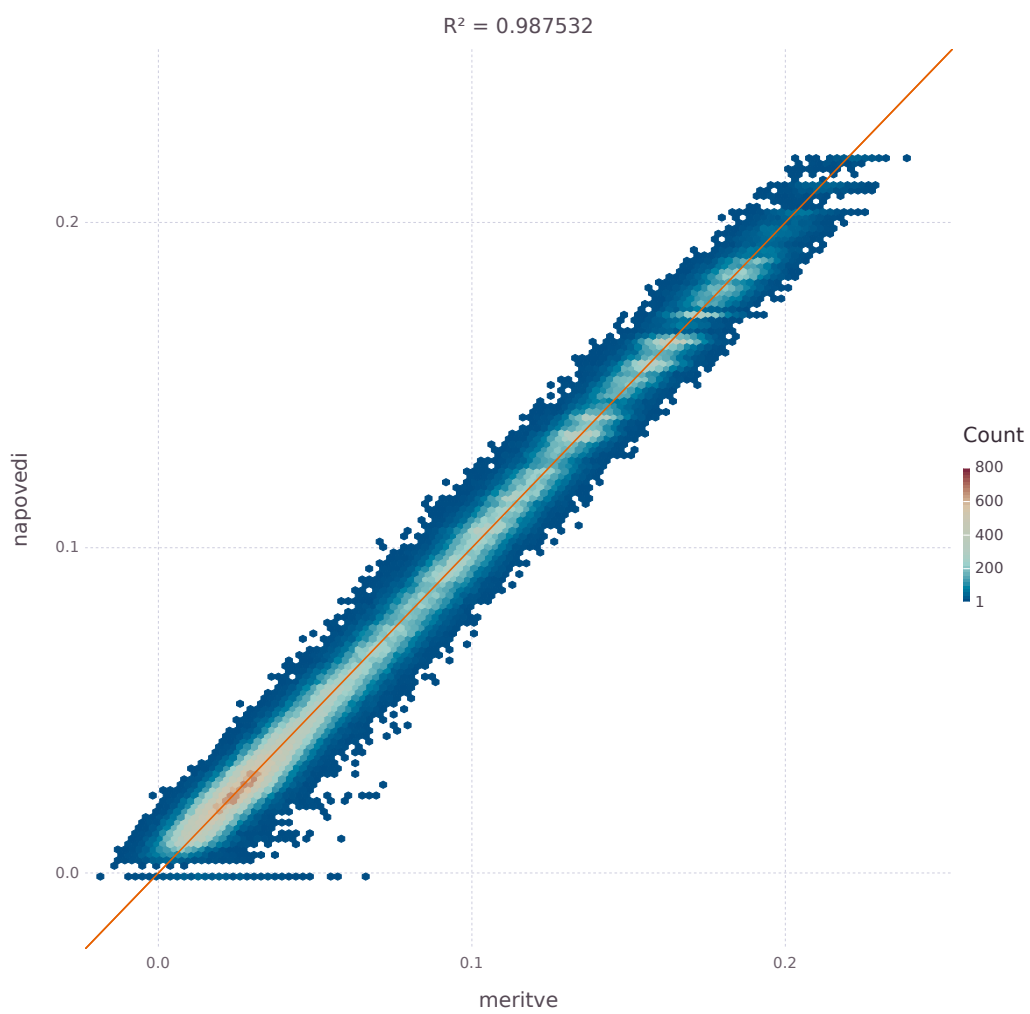
Slika 5.2: Primer prilagoditve funkcije modela na podatke. Uporabljeni so bili isti podatki, kot v drugi vrstici na sliki 3.2.



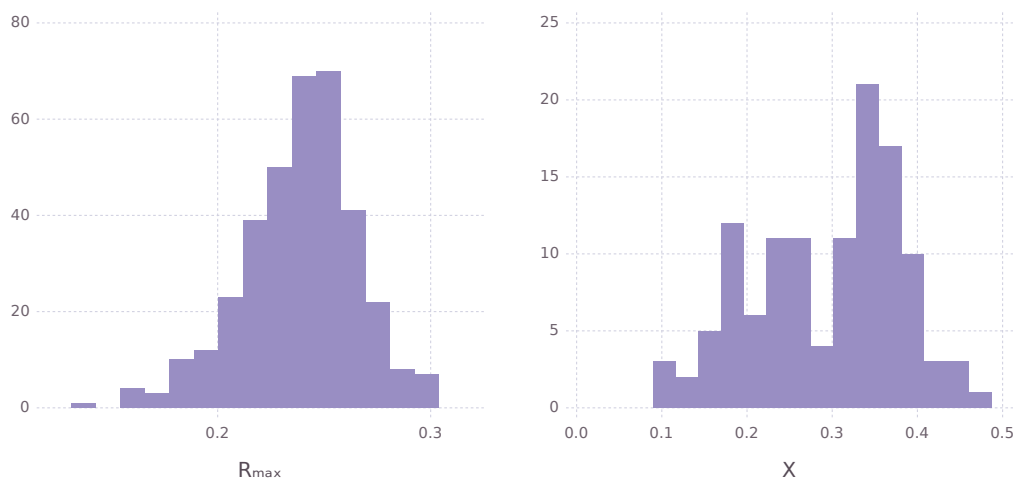
Slika 5.3: Meritev, pri kateri je bilo prilagajanje najmanj uspešno.



Slika 5.4: Meritev, kjer se dobljeni model najbolj prilagaja meritvam.



Slika 5.5: Graf prikazuje razmerje med izmerjenimi in napovedanimi vrednostmi. Zaradi velike količine podatkovnih točk so točke predstavljene kot gostota.



Slika 5.6: Histograma prikazujeta razpore ocenjenih vrednosti parametrov  $R_{max}$  in  $X$ .

zašumljenega parametra pričakovali.

Ker sta vrednosti kriterijske funkcije in RMSE odvisni od merske skale podatkov, z njima ne moremo dobro ovrednotiti rezultatov. Iz tega razloga smo se odločili, da bomo rezultate preverili še na druge načine, ki so predstavljeni v naslednjih razdelkih.

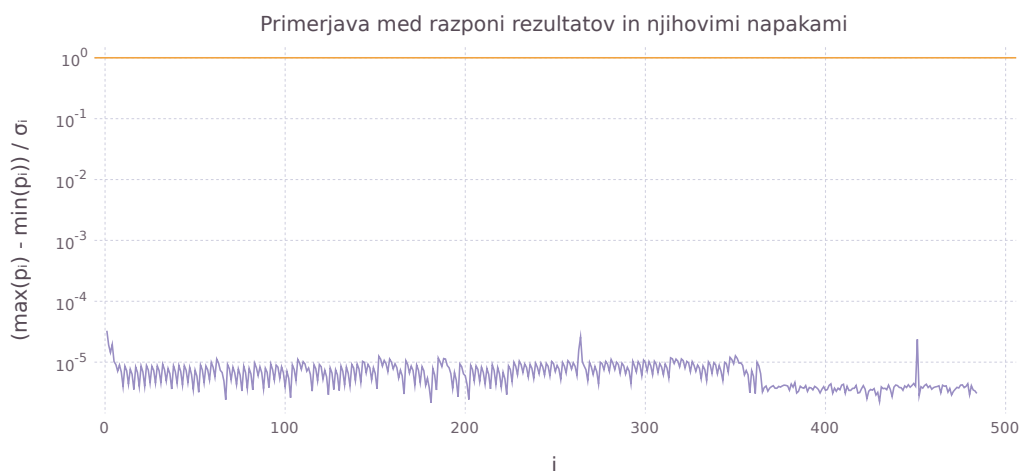
## 5.2 Preverjanje robustnosti metode

### 5.2.1 Občutljivost na izbiro začetnih parametrov

Test občutljivosti metode na izbiro začetnih parametrov smo izvedli tako, da smo algoritem večkrat pognali z različnimi naključnimi začetnimi parametri. Pri tem smo si shranjevali rezultate in število iteracij, ki jih je algoritem potreboval za doseganje minimuma.

Začetne parametre smo omejili na fizikalno smiselne vrednosti. Vsi začetni parametri so bili pozitivni in parameter  $X$  nikoli ni bil večji od 1. Prav tako smo se odločili, da bodo v vsakem vektorju začetnih parametrov vrednosti





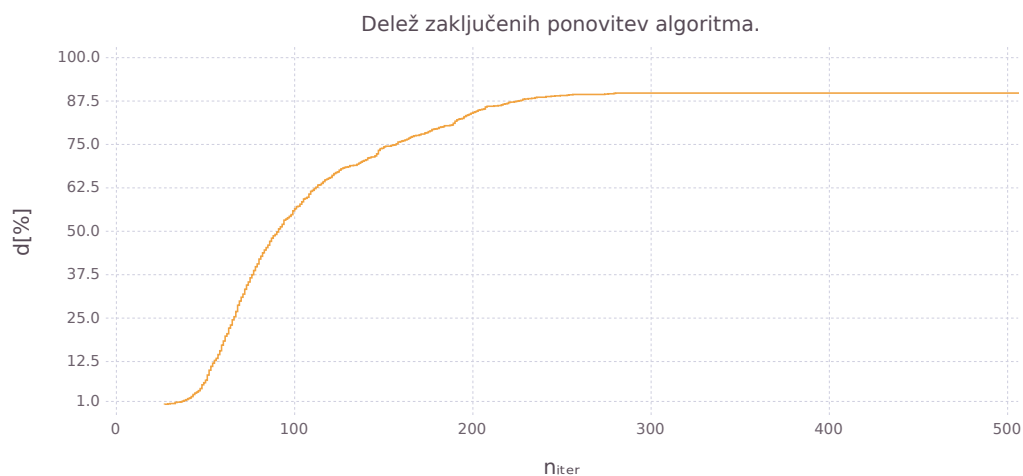
Slika 5.7: Graf prikazuje razpone napovedi posameznih parametrov v enotah njihove standardne napake. Na abscisni osi so posamezni parametri modela, normalizirana razlika med najmanjšo in največjo dobljeno vrednostjo pa je na ordinatni osi.

posameznih  $X^{(i)}$  in posameznih  $R_{max}^{(i+j)}$  med seboj enake. To smo storili, ker pričakujemo, da uporabnik metode ne bo ročno nastavljal parametrov za posamezne vzorce.

Meritev smo ponovili 1000-krat, maksimalno število iteracij pa smo omejili na 500. Minimizacija se je uspešno zaključila v 897 primerih, v ostalih 103 poskusih pa je dosegla največje dovoljeno število iteracij.

Vsi uspešni primeri so konvergirali v isto točko. To smo preverili tako, da smo za vsak parameter primerjali njegovo standardno napako in razliko med največjo in najmanjšo dobljeno vrednostjo. Na grafu 5.7 so prikazani razponi napovedi posameznih parametrov, normalizirani z njihovo standardno napako. Ker so največje vrednosti na grafu globoko pod 1 lahko sklepamo, da res vsi nabori parametrov predstavljajo isti minimum kriterijske funkcije.

Neuspešni primeri so vsi končali v fizikalno nesmiselnih točkah, torej takih, kjer so nekateri parametri negativni, ali pa so vrednosti  $X$  večje od 1. Za preizkus smo 10 neuspešnih primerov ponovili z neomejenim maksimalnim



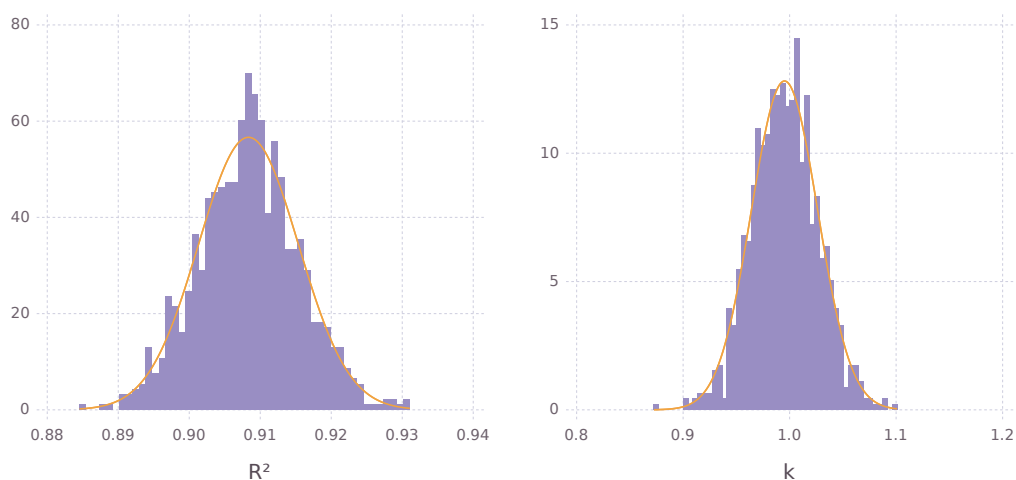
Slika 5.8: Delež zaključenih ponovitev algoritma v odvisnosti od števila iteracij.

številom iteracij. Ti primeri so v povprečju končali v več kot 100000 iteracijah, kljub temu pa še vedno niso dosegli smiselnih rezultatov, med tem, ko so vsi uspešni primeri končali v 280 iteracijah ali prej. Iz grafa 5.8 lahko razberemo kakšen delež ponovitev bi končal v kakšnem številu iteracij.

Iz tega preizkusa lahko sklepamo, da je algoritem relativno neobčutljiv na izbiro začetnih parametrov. Algoritem namreč v relativno kratkem času zanesljivo konvergira v isti lokalni minimum. Ko se to ne zgodi konvergenca traja zelo dolgo in vrne fizikalno nesmiselen rezultat. To pomeni, da lahko v primeru slabe izbire začetnih parametrov to hitro opazimo in algoritem preprosto ponovimo.

## 5.2.2 Občutljivost na količino podatkov

V drugem testu smo preizkušali občutljivost metode na količino podatkov ki so na voljo. To smo preizkušali tako, da smo množico podatkov delili na dva dela, na vsakem delu posebej pognali algoritem in primerjali rezultate za dobljene parametre  $\mathbf{k}$  in  $X^{(i)}$ . Podatke smo najprej delili naključno, nato



Slika 5.9: Porazdelitvi koeficientov  $R^2$  in  $k$ . Na oba histograma je z oranžno črto vpeta normalna porazdelitev.

pa smo jih delili še po koncentracijah.

Pri naključnem deljenju podatkov smo poskrbeli, da so bili v obeh podmnožicah podatkov zastopani vsi vzorci, saj smo tako lažje primerjali dobljene rezultate. Naključni test smo ponovili 1000-krat. Za vsak rezultat smo izračunali determinacijski koeficient  $R^2$  in koeficient premice  $k$ . Pri naključnem testu smo dobili konsistentne rezultate. V veliki večini primerov sta obe podmnožici podatkov napovedali enake vrednosti parametrov. Povprečna vrednost koeficienta  $R^2$  za vseh 1000 poskusov je bila 0.908 s standardno deviacijo 0.007. Povprečje koeficientov premic  $k$  je bilo 0.995 s standardno deviacijo 0.031. Slika 5.9 prikazuje histograma rezultatov  $R^2$  in  $k$ . Ker je  $R^2$  visok in  $k$  blizu 1, lahko sklepamo, da je algoritem dal iste rezultate ne glede na izbiro podatkov.

V drugem delu preizkusa smo podatke delili tako, da so bili v eni skupini podatki z eno vrednostjo koncentracije, v drugi pa vsi ostali.

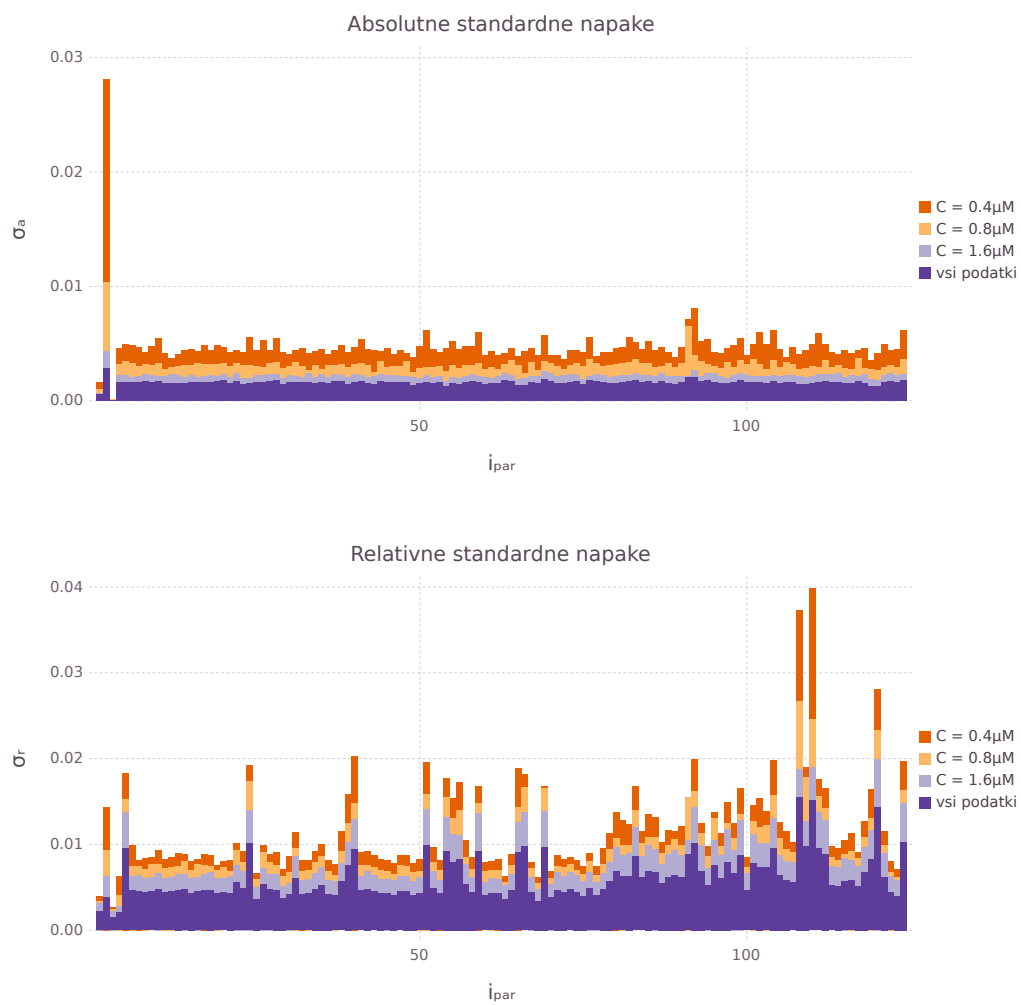
Tabela 5.1 prikazuje vrednosti  $R^2$  in  $k$  za posamezne koncentracije. V njej tudi hitro opazimo, da so rezultati slabši, kot v primeru z naključnim izborom podatkov. Prav tako opazimo, da so najslabši v primeru, kjer je

Tabela 5.1: Dobljene vrednosti  $R^2$  in  $k$  pri delitvah podatkov v podmnožice glede na koncentracijo.

$C[\mu\text{M}]$	$R^2$	$k$
1.6	0.848	0.788
0.8	0.899	1.203
0.4	0.665	1.528

koncentracija enaka  $0.4\mu\text{M}$ . Razlog za to verjetno tiči v tem, da je relativna napaka podatkov manjša pri višjih koncentracijah. Do tega pride, ker meritve z višjimi koncentracijami dosegajo višje vrednosti, količina šuma pa je enaka za vse meritve.

Primerjamo lahko tudi standardne napake parametrov za različne podmnožice podatkov. Na sliki 5.10 so prikazane standardne napake posameznih parametrov  $\sigma_i$ . Opazimo tudi, da je razlika v standardnih napakah med množico vseh podatkov in samo tistih z najvišjimi koncentracijami zelo majhna. Relativne standardne napake vseh parametrov pri meritvah, kjer je  $C = 1.6\mu\text{M}$  so namreč manjše od 2%. To pomeni, da če smo zadovoljni z malenkost grobjo oceno parametrov modela, lahko to oceno dobimo iz manjše količine podatkov. To bi lahko predstavljalo velik prihanek časa pri izvajanju meritev.



Slika 5.10: Absolutne in relativne standardne napake posameznih parametrov pri različnih podmožicah podatkov.



# Poglavje 6

## Zaključek

Cilj diplomskega dela je bil opis eksperimentalnih podatkov z matematičnim modelom, ter določitev metode, ki bo učinkovito in zanesljivo iz podatkov določila njegove parametre. Podatki so bili izmerjeni z interferenco z biološkimi plastmi [2] in prikazujejo dinamiko vezave dveh vrst protiteles in Fc $\gamma$ RIIIa receptorja. Opisati je bilo treba tudi biološko ozadje problema in najti matematični model, ki bi lahko opisal podatke.

Pri izbiri algoritmov smo preizkusili veliko število metod, med drugim Netwon-Raphsonov, Gauss-Newtonov in BFGS algoritem. Odločili smo se za Levenberg-Marquardtovo metodo, saj se je izkazala za najbolj zanesljivo.

Prve preizkuse smo začeli v programskem jeziku R [20], ki pa se je kljub dobro implementiranim in hitrim metodam za optimizacijo izkazal za prepočasnega in pomnilniško požrešnega. Ti algoritmi namreč velik delež časa preživijo med računanjem odvodov kriterijske funkcije, ki jih je bilo potrebno implementirati ročno. Kasneje smo začeli uporabljati programski jezik Julia [18], kar je znatno izboljšalo hitrost delovanja.

Ker imajo dani eksperimentalni podatki več kot 200.000 časovnih točk in ima model 484 parametrov, je Jakobijeva matrika tega sistema zelo velika. Na srečo je velik delež parametrov v modelu medsebojno neodvisnih, kar pomeni, da matriko lahko implementiramo kot redko. To prinaša velik prihranek v porabi pomnilnika. Juliina implementacija Levenberg-Marquardtovega algo-

ritma redkih matrik ne podpira, zato smo algoritem, ki je bil uporabljen v tem diplomskem delu, prilagodili.

Končni program, ki vsebuje funkcije za branje podatkov in določanje parametrov je kratek in preprost, in vsebuje le približno 700 vrstic kode. Če dodamo še funkcije za risanje grafov in vse analize, ki smo jih opravili, je kode nekoliko več, približno 1800 vrstic. Približno enaka količina kode je bila napisana tudi v programskem jeziku R, a smo to kodo zaradi neučinkovitosti zavrgli.

Ugotovili smo, da se model danim podatkom prilega odlično, in da se z Levenberg-Marquardtovim algoritmom zelo zanesljivo izračuna njegove parametre. Žal podatki o dejanski sestavi protiteles trenutno niso na voljo in tako naših rezultatov ne moremo primerjati z resničnimi fizikalnimi količinami. To bomo preizkusili v prihodnosti. Ugotovili smo tudi, da se da parametre dovolj dobro izračunati tudi z manjšo količino podatkov, celo samo z eno tretjino, kar pomeni potencialno pohitritev procesa karakterizacije protiteles.



# Literatura

- [1] B. Ingalls. *Mathematical Modelling in Systems Biology: An Introduction*. Applied Mathematics, University of Waterloo, 2012.
- [2] N. B. Shah and T. M. Duncan. Bio-layer interferometry for measuring kinetics of protein-protein interactions and allosteric ligand effects. *Journal of Visualized Experiments*, 84:e51383, 2014.
- [3] R. Frigg and S. Hartmann. Models in science. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Fall 2012 edition, 2012. <http://plato.stanford.edu/archives/fall2012/entries/models-science/>.
- [4] Sze-Bi Hsu. *Mathematical Modelling in Biological Science*. Tsing-Hua University, Taiwan, 2004.
- [5] F. Hoppensteadt. Getting started in mathematical biology. *Notices of the AMS*, 42(9):969–975, 1995.
- [6] G. Mendel. Versuche über pflanzenhybriden. *Verhandlungen des naturforschenden Vereines in Brünn*, 1865.
- [7] V. Volterra. Variazioni e fluttuazioni del numero d'individui in specie animali conviventi. *Mem. Acad. Lincei Roma*, 2:31–113, 1926.
- [8] F. Hoppensteadt. Predator-prey model. *Scholarpedia*, 1(10):1563, 2006. [http://www.scholarpedia.org/article/Predator-prey\\_model](http://www.scholarpedia.org/article/Predator-prey_model).

- 
- [9] H. Abdi. The method of least squares. In N. Salkind, editor, *Encyclopedia of Measurement and Statistics*. 2007.
- [10] M. L. Johnson and L. M. Faunt. Parameter estimation by least-squares methods. *Methods in enzymology*, 210:1–37, 1992.
- [11] E. M. Tansey and P. P. Catterall. Monoclonal antibodies: a witness seminar in contemporary medical history. *Medical History*, 38:322–327, 1994.
- [12] M. Peipp1and, J. J. Lammerts van Bueren, T. Schneider-Merck, W. W. K. Bleekerand, M. Dechantand, T. Beyerand, R. Repp1and, P. H. C. van Berkeland, T. Vinkand, J. G. J. van de Winkeland, P. W. H. I. Parren, and T. Valerius. Antibody fucosylation differentially impacts cytotoxicity mediated by nk and pmn effector cells. *Blood*, 112:2390–2399, 2008.
- [13] A. Golob. Kinetika vezave rekombinantnih monoklonskih protiteles na receptor  $Fc\gamma RIIIa$ . Master’s thesis, Univerza v Ljubljani, Biotehniška fakulteta, 2015.
- [14] D. J. O’Shanessy and D. J. Winzort. Interpretation of deviations from pseudo-first-order kinetic behavior in the characterization of ligand binding by biosensor technology. *Macromolecular Sciences*, 236(0167):275–283, 1996.
- [15] S. Gratton, A. S. Lawless, and N. K. Nichols. Approximate gauss-newton methods for nonlinear least squares problems. *SIAM Journal on Optimization*, 18:106–132, 2007.
- [16] D. W. Marquardt. An algoirthm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11:431–441, 1962.
- [17] Y. Ya-xiang. A review of trust region algorithms for optimization. Inst. of Computational Mathetmatics, TR 99-038, Beijing, 1999.

- 
- [18] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. Julia: A fresh approach to numerical computing. *arXiv*, 2014, 1411.1607.
- [19] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 2007.
- [20] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016.

,