

CHALMERS



GÖTEBORGS UNIVERSITET

En kvantitativ modell för Cdc42 under påverkan
av ROS i polariseringsfasen hos *S. Cerevisiae*

MVEX01-17-18

Kandidatarbete inom civilingenjörsutbildningen vid Chalmers

Anton Björk

Julia Larsson

Angelo Limeta

Patrik Svedberg

Institutionen för matematiska vetenskaper
Chalmers tekniska högskola
Göteborgs universitet
Göteborg 2011

Populärvetenskaplig presentation

Föreställ dig själv vid 70 års ålder. Prova 100. Tänk om du kunde leva ytterligare 50 år efter det. Det har spekulerats i att den första människan som blir 150 år gammal redan har fötts. Denna spekulering bygger på förväntad förbättring inom sjukvård men även framsteg inom åldersforskning.

Den viktiga frågan är HUR vi vill tillbringa de extra 60 eller 70 åren. Idag lider många gamla av åldersrelaterade sjukdomar. Dessa sjukdomar kan gravt försämra livskvaliteten de sista åren, eller leda till en tidigare död. För att verkligen förstå, och behandla, dessa sjukdomar måste vi hitta mekanismen bakom åldrande.

Forskning på åldrande genomförs med fördel på organismer med betydligt mycket kortare livslängd än människor. Ett ofta förekommande val är bagerijäst, se Figur 2. Människan har använt jäst i tusentals år för att göra bröd eller alkohol. På grund av det är bagerijäst förmodligen en av de organismer vi vet mest om. I rena antal är det världens näst mest studerade organism. Ofta utförs studier ämnade för högre organismer först på bagerijäst, då bagerijäst fungerar som modell-organism. Jästen delar många av de viktigaste cell-egenskaperna som finns i högre organismer och kan därför ge en indikation på hur människoceller kommer att reagera på t.ex. en ny medicin. Eftersom de är encelliga (till skillnad från oss som består av miljarder av celler) är de lätta att hantera och odla. Jästen har en livslängd på ungefär ett dygn, under vilket den hinner dela sig runt 25 gånger. Dessa egenskaper gör bagerijästen till en passande organism för åldrande-studier.



Figur 2: *Bagerijäst, en av de mest undersökta organismerna i världen [2], CC-BYSA*

kallas reaktiva syreföreningar. Detta arbete ämnar undersöka hur syreföreningarna påverkar en viss process i cellen.



Figur 1: *Om vi kan leva till 150, hur kommer de sista 50 åren av våra liv se ut? [1], CC-0*

Den cellulära process som undersöks i detta arbete kallas polarisering. Polarisering avgör "riktningen" hos en cell, det vill säga att det skapas en pol (jämför med syd- och nordpolen på vår planet). Detta är viktigt för att vissa nyckelprocesser i cellen ska fungera, till exempel celledelning och cellförflyttning. Mer specifikt undersöks ett speciellt protein som är viktigt i processen: Cdc42. Proteinet är intressant då det existerar i ett stort antal olika grupper av organismer, från encellig bagerijäst till människor.

I jästen har proteinet Cdc42 en speciell roll inför celledelning. Dess uppgift är att bilda olikheter i koncentration. Dessa olikheter identifieras av andra processer i cellen och bildar den pol som avgör var cellen ska dela sig. Att bilda olikheter är, inom kemin, inte helt enkelt. Ämnen har en tendens att vilja sprida ut sig jämt i en volym, t.ex. om du håller lite salt i ett vattenglas kommer alla delar i vattenglasets smaka lika salt. Cdc42 kan skapa olikheterna genom att byta form när den binder sig till cellens kant. Den nya formen kan locka till sig mer av proteinet och bildar då en så kallad positiv feedback-loop. När tillräckligt mycket av det Cdc42 som finns i cellen har bundit in till en viss pol förhindras andra poler från att bildas, och cellen går vidare med sin delning. En intressant detalj är att Cdc42 visat sig vara sämre på att utföra sin roll inför delningen när cellerna är äldre. Hur proteinet praktiskt beter sig under polariseringsprocessen är svårt att undersöka, men med matematik kan det beskrivas teoretiskt. Det görs med modellering.

Matematisk modellering är ett mycket användbart verktyg som låter oss utvärdera strukturen hos en process. Det görs genom att hitta en uppsättning ekvationer som tillsammans beskriver processen. Med en färdig modell kan processen sedan testas under olika tillstånd. Ofta ligger det svåra i att finna ekvationerna och deras konstanter. Att hitta konstanterna (fasta värden) i en modell kallas parameteruppskattning och är ett viktigt område inom modellering. Modellen jämförs med resultat från verkliga experiment och oftast används datorprogram för att passa modellen mot resultaten vilket leder till att konstanterna kan bestämmas. På samma sätt som en provdocka aldrig kan ge en rättvis bild av hur ett plagg passar på en rörlig, levande människa, kan inte modeller ge en fullständigt korrekt bild av verkligheten. De kan däremot ge insikt i hur processen fungerar eller låta oss förutspå ungefär hur processen beter sig under andra förhållanden.

I detta kandidatarbete används matematisk modellering för att undersöka hur Cdc42 fungerar under jästens polarisering vid närvaro av syre-radikaler.

Resultatet av arbetet har blivit en modell som beskriver processen för normala förhållanden. Cellerna i de testresultat som använts var av obestämd ålder, troligen med normala nivåer av syre-radikaler, och med fungerande polarisering. Tanken är nu att projektet ska kunna gå vidare med laboratorie-försök på äldre celler eller celler som har konstgjort höga nivåer av syre-radikaler. Fungerar modellen när den passas mot dessa nya data, så bör syre-radikalernas påverkan på förloppet kunna utredas. Kan förloppet utredas i jäst bör motsvarande hos människor inte vara långt borta.

Om radikalerna påverkar delningen hos mänskliga celler så kan det vara en av mekanismerna bakom åldrande, och kanske en nyckel till att behandla och förebygga åldersrelaterade sjukdomar. Kanske kan anti-oxidanter verkligen hjälpa mot åldrande. Kanske kan vi bestämma vilka celler som är känsligast, och då punktmarkera dessa med behandling. Istället för att lägga till år på slutet av livet kanske vi kommer kunna dra ut på processen;

Tillbringa 60-års-åldern med en 40-åring hälsa, och lev till 150.

En kvantitativ modell för Cdc42 under påverkan av ROS i polariseringsfasen hos *S. Cerevisiae*

MVEX01-17-18

*Kandidatarbete i matematik inom civilingenjörsprogrammet Bioteknik vid
Chalmers*

Anton Björk
Julia Larsson
Angelo Limeta
Patrik Svedberg

Institutionen för matematiska vetenskaper
Chalmers tekniska högskola
Göteborgs universitet
Göteborg 2011

Handledare: Johannes Borgqvist, Marija Cvijovic
Examinator: Marina Axelson-Fisk, Maria Roginskaya

Sammanfattning

Kan en reduktion av syreradikaler förlänga livet? I detta kandidatarbete studeras påverkan av reaktiva syre-specier, förkortat ROS, på proteinet Cdc42 med hjälp av matematiska modeller. Cdc42 är aktivt under polarisering i jästceller, vilket betyder att en asymmetri i cellen skapas inför celledelning. Forskning indikerar att ROS är en bidragande faktor till åldrande och kan skada både DNA och protein. Det har också visats att polariseringen fungerar sämre för äldre celler. Cdc42 är ett jäst-protein med homologi hos människor. Slutsatser från experiment i jästceller kan ge kunskap om motsvarande mekanismer i människan.

Tre modeller, med olika grad av komplexitet, för Cdc42 under påverkan av ROS bestående av system av differentialekvationer har skapats. Modellerna har sedan parameteruppskattats med hjälp av experimentella data. Bland annat har mängden skadat Cdc42 kunnat förutspås av modellerna.

För att få fördjupad insikt i parameteruppskattning har ett eget program för detta skrivits i MATLAB. Parametrarna har även uppskattats i en programvara kallad Copasi.

Resultaten visar att två av tre producerade modeller kan simulera fördelningen av Cdc42 i polariseringsfasen och att ROS har en försumbar inverkan på Cdc42 under given tidsperiod. Det egenskrivna programmet i MATLAB kan genomföra parameteruppskattningar i modeller med relativt få parametrar men ger missvisande resultat i modeller med många parametrar.

Att ROS har en låg påverkan på Cdc42 var ett förväntat resultat. Däremot finns det en hel del i de fungerande modellerna som kan förbättras för att göra dem mer tillförlitliga. En grundlig undersökning av identifierbarheten hos parametrarna i modellen skulle behövas vid en eventuell vidareutveckling av ODE-systemen.

Anledningen till att det egenskrivna programmet inte kunde parameteruppskatta de mer komplicerade modellerna är på grund av tidsbrist under programmering. Med ett par modifieringar skulle programmet med stor sannolikhet kunna utföra svårare parameteruppskattningar.

Abstract

Can a reduction of oxygen radicals prolong life? In this bachelor thesis, the impact of reactive oxygen species, abbreviated ROS, on the protein Cdc42 is studied using mathematical modelling. Cdc42 is active during polarization in yeast cells, which creates asymmetry in the cell before division. Research indicates that ROS is a contributing factor to ageing and can damage both DNA and proteins. It has also been shown that the polarization process is less efficient in older cells. Cdc42 is a protein in yeast with homology in humans. Conclusions from experiments in yeast can therefore shed light on similar mechanisms in humans.

Three models of increasing complexity, consisting of systems of differential equations, for Cdc42 under the impact of ROS were developed. Existing experimental data have been used in parameter estimation. Among other relevant parameters, the levels of damaged Cdc42 were estimated.

To get a deeper insight regarding parameter estimation, an in-house routine has been created using MATLAB. The parameters have also been estimated in the systems biology software Copasi.

The results show that at least two of the three models are able to simulate the concentration distribution of Cdc42 during polarisation and that ROS has a negligible impact on Cdc42 during this period of time. The in-house routine in MATLAB can perform parameter estimations on models with few parameters but gives misleading results when applied to models with many parameters.

The conclusion that ROS has little impact on Cdc42 does not come as a surprise. However, there is a lot in the developed models that can be improved to make them more reliable. A thorough investigation of the identifiability of the model parameters is needed if the ODE systems should be further developed.

The reason that the in-house program is unable to estimate the parameters of the more complicated models is lack of programming time. With a few modifications, the program should be able to perform more complicated parameter estimations.

Innehåll

1 Inledning	4
2 Teori	5
2.1 Biologi	5
2.1.1 <i>S.Cerevisiae</i>	5
2.1.2 ROS	5
2.1.3 Cdc42	6
2.2 Matematik	7
2.2.1 Matematisk modellering	7
2.2.2 Parameteruppskattning	7
3 Metod	10
3.1 Skapande av modeller utifrån litteraturstudier	10
3.2 Datainsamling	10
3.3 Parameteruppskattning i Copasi	11
3.4 Implementering	11
4 Resultat	13
4.1 Parameteruppskattning	13
4.2 Modeller och deras anpassning till data	16
4.2.1 Modell baserad på litteraturstudier	17
4.2.2 Goryachev - ROS attackerar cytosoliskt Cdc42	18
4.2.3 Goryachev - ROS attackerar cytosoliskt och membranbundet Cdc42	19
5 Diskussion	20
6 Referenser	23
A Härledning av Goryachevs modell med inkorporering av ROS	27
B Data	33
C Kod	38

Förord

Arbetet har varit uppdelat i en teoretisk fas där inläsning och modellskapande legat i fokus samt en mer praktisk fas där programmering och parameteruppskattning dominerat. I den teoretiska fasen arbetade gruppen huvudsakligen tillsammans med bland annat inläsning och modellförslag. Under övergångsfasen blev specifika ämnesområden uppdelade bland gruppmedlemmarna: Anton och Julia härledde modeller, Patrik fokuserade på rollen hos ROS i systemet och Angelo påbörjade programmeringen i MATLAB. I den senare delen av arbetet, den mer praktiska delen, stod Anton och Angelo för MATLAB-programmering, Julia skötte parameteruppskattning i Copasi och Patrik assisterade med teoretiska frågor. I slutet, innan allas fokus flyttades till rapportskrivning, arbetade Anton och Angelo med att få resultat från programmeringen medan Julia och Patrik påbörjade slutrapporten. Möten och diskussion kring problem som uppstått har förts kontinuerligt och med allas medverkan. Arbetsgången är dokumenterad i en veckovis dagbok, samt i individuella tidsloggar. Huvudförfattare för stycken i slutrapporten redovisas i Tabell 1. Notera att vissa stycken har flera huvudförfattare.

Vi vill tacka våra handledare, Johannes Borgqvist och Marija Cvijovic, för enormt stöd och tillgänglighet under arbetet. Vi vill även tacka resten av Cvijovic-labbet som har välkomnat oss med öppna armar under både fredagsmöten och sociala sammankomster. Jacob Leander har dessutom gett oss några mycket värdefulla tips rörande parameteruppskattning under arbetets gång. Avslutningvis vill vi tacka Andrew Goryachev för hans artikel *Dynamics of Cdc42 network embodies a Turing-type mechanism of yeast cell polarity* som legat till grund för modellformuleringen i detta arbete.

Tabell 1: Tabell över huvudförfattare för varje avsnitt i rapporten

Huvudansvarig författare	Avsnitt
A. Björk	1 2.1.3 3.4 5
J. Larsson	3.1 3.3 4.2 4.2.1 4.2.2 4.2.3 5 Bilaga A
A. Limeta	2.2 2.2.1 2.2.2 5 Bilaga C
P. Svedberg	Populärvetenskaplig presentation 1 2 2.1 2.1.1 2.1.2 2.2 2.2.1 3.2 5 Bilaga B

1 Inledning

Åldrande är en oundviklig del av livet för många organismer och med ålder följer en ökad risk för åldersrelaterade åkommor. I människor kan det bland annat röra sig om Alzheimers sjukdom [3]. Flertalet av dessa åldersrelaterade sjukdomar är obotliga och det finns ännu inte nog med kunskap om fenomenet åldrande för att hitta fungerande behandlingar. Därför forskas det för att utreda detaljerna om åldrande och förhoppningen är att det kommer att bidra till framtidens sjuk- och äldreomsorg [4].

I vardagligt tal är åldrande en alltför vag term för forskning och därför används ett par mer väldefinierade termer. Kronologiskt åldrande avser den tid en cell eller organism varit vid liv. Inom forskning på encelliga organismer avses den tid en cell överlever i ett tillstånd då den inte delar sig, alltså i stationär fas [5]. Biologiskt åldrande avser den förändring och försämring av funktion som celler genomgår då de åldras. Det är ett brett begrepp eftersom det inte är definierat vilken funktion hos cellen som mäts utan bara att den måste förändras då cellen åldras. Det kan ofta vara svårt att kvantifiera biologiskt åldrande. Det mest undersökta är hur celler klarar av att föröka sig, vilket beskrivs av ett relaterat men snävare begrepp, replikativ ålder. Den definieras av hur många gånger en cell delat sig och många studier undersöker hur den maximala replikativa åldern hos en organism påverkas av olika genetiska modifieringar [5].

Att undersöka biologiskt åldrande i människor är mycket tidskrävande, då eventuella försök bokstavligen kan ta en hel livstid. För att studera någon specifik del av biologiskt åldrande i detalj och undvika långa försöksprocesser, kan åldrandet studeras i olika modellorganismer med betydligt kortare livslängd än människa. Ett vanligt val av organism är bagerijästen *Saccharomyces Cerevisiae*, en encellig organism som delar många viktiga proteiner och funktioner med mänskliga celler, men med en generationstid på runt 100 minuter [6]. Likheterna mellan jäst och mänskliga celler gör att det går att extrapolera kunskap, om exempelvis hur en jästcell reagerar på en medicin, till mänskliga celler. Specifikt kan jästcellens replikativa ålder jämföras med åldern hos delande celler, som stamceller, och kronologisk ålder jämföras med åldern hos icke-delande celler, som neuroner [5].

Den viktigaste processen i bagerijäst i relation till replikativ ålder är naturligtvis förmågan att dela sig [5]. *S.cerevisiae* delar sig genom knoppning, det vill säga att avkomman är betydligt mindre än modercellen. Inför knoppningen behövs ett säte för knoppbildning, vilket kräver en asymmetri i cellen. Processen då cellen bildar denna asymmetri kallas polarisering och kontrolleras av flera olika proteiner [7]. Det viktigaste proteinet vid jästcellens polarisering är Cdc42. Det är ett konserverat protein, vilket betyder att det finns homologer i många olika arter, bland annat människan [8]. Homologerna är besläktade både i funktion och ursprung [9]. Försök på stamceller har visat att polariseringsprocessen fungerar sämre med ålder, och att Cdc42 kan ha en nyckelroll i denna försämring [10, 11].

En faktor som korrelerar med ökande åldrande är högre nivåer av reaktiva syre-specier (ROS) [12]. Det har genomförts en hel del forskning på kopplingen mellan ROS-nivåer och ålder. En teori, "Free radical theory of Aging", menar att ROS är orsaken bakom biologiskt åldrande [13]. Andra menar att det fullständiga sambandet inte är helt fastställt [14, 15]. Tydligt är dock att ROS har en roll i biologin bakom åldrande, och att den rollen bör utredas. Ett sätt att utreda funktionen hos ROS i någon specifik process kan vara genom matematisk modellering.

Matematisk modellering hjälper till att fördjupa förståelsen för biologiska system eller processer. När hypoteser och idéer om biologiska system ska översättas i ekvationer tvingar det fram konkretisering av dem vilket kan leda till bättre förståelse och nya insikter redan då modellen konstrueras [16]. Modellering är också mycket användbart eftersom system med fler än ett par interagerande komponenter ofta ger upphov till komplexa beteenden som inte går att förutse eller förstå med ren intuition [16]. Matematisk modellering har på senare tid fått ett allt större genomslag inom cell- och molekylärbiologin [16]. Inom det relativt nya forskningsområdet systembiologi läggs stor vikt vid att kombinera modellering och experimentellt arbete [17].

En del av modellering är att uppskatta de parametrar som ingår i modellen. Det görs genom att använda experimentella data från processen som modelleras för att uppskatta okända parametrar under antagandet att modellen ska passa mot datan [18]. Denna process

kallas parameteruppskattning och är ett viktigt område inom modellering.

Syfte

Syftet med kandidatarbetet är att undersöka Cdc42:s dynamik under *S.Cerevisiae*s polariseringsfas vid närvaro av ROS. För att genomföra detta skall minst en matematisk modell för dynamiken härledas och dess parametrar skall uppskattas utifrån dynamisk mätdata. Som stöd för parameteruppskattning kommer en optimeringsalgoritm, en så kallad descent-algoritm baserad på quasi-Newton, implementeras i MATLAB.

2 Teori

Då arbetet rör sig i gränslandet mellan biologi och matematik är teoriavsnittet indelat i två huvudrubriker. Det biologiska presenteras under avsnitt 2.1 och det matematiska under 2.2.

2.1 Biologi

I arbetets syftesformulering ingår tre biologiska entiteter; Jästen *S.Cerevisiae*, reaktiva syrespecier (ROS) och proteinet Cdc42. Den första är den organism som undersöks. De två senare utgör en central roll i modellen som ska tas fram. Nedan följer närmare beskrivning av dessa entiteter.

2.1.1 *S.Cerevisiae*

Saccharomyces cerevisiae, kanske mer känt under den vardagliga benämningen bagerijäst, är en encellig eukaryot. Eukaryoter, till skillnad från prokaryoter, har sin arvs massa i en cellkärna omgiven av ett kärnmembran [19]. I gruppen ingår, förutom jäst och andra svampar, alla högre organismer, så som växter och djur.

S.cerevisiae fungerar som modellorganism för eukaryoter [20], vilket innebär att studier ofta genomförs på modellorganismen först, istället för att direkt undersöka den eller de organismer som är av intresse. Detta förfarande kan tyckas kontraintuitivt, men modellorganismer är, förutom billigare och enklare att arbeta med, bland de bäst kartlagda av alla organismer [21, 22]. Denna förkunskap innebär att studier på modellorganismer kan ge mer detaljerad information om exempelvis mekanismer än om studien genomförts direkt på den intressanta organismen. Informationen ligger sedan till grund för hur och om studien går vidare till andra organismer. De viktigaste modellorganismerna är *E.coli*, *S.cerevisiae*, masken *C.elegans*, fruktflugan *Drosophila melanogaster* samt musen *Mus musculus* [21].

Bagerijäst är en av världens mest studerade organismer [20] och det finns en uppsjö av verktyg för studier i organismen. En detalj som skiljer bagerijästen från till exempel mänskliga celler är dess metod för celledelning. *S.cerevisiae* delar sig genom knoppning, vilket innebär att den nya cell som bildas, avkomman, är betydligt mindre än moder-cellen. Denna detalj gör det relativt enkelt att följa en och samma cell över en hel livstid, för studier av exempelvis åldrande och livscykel. Skillnaden gör däremot själva celledelningsfasen mindre relevant för jämförelse med mänskliga celler. För modellstudier av celledelning väljs därför ofta en annan jäst, *S.pombe*, vars delning mer liknar den som sker i våra celler [23].

2.1.2 ROS

ROS är ett samlingsnamn för en uppsättning reaktiva syrespecier (reactive oxygen species). I biologiska system rör det sig oftast om superoxidjonen ($O_2^{\bullet-}$) och de produkter den kan skapa vid närvaro av protoner; hydroxylradikaler (HO^{\bullet}) och väteperoxid (H_2O_2) [24, 25]. För strukturer se Figur 3.

Som tidigare nämnt kan en ökning av ROS-nivåer ses i celler med ökande ålder [12]. Höga nivåer av ROS i cellerna leder till oxidativ stress och kan orsaka oxidativ skada på makromolekyler, såsom protein eller DNA [26]. Oxidativt skadade protein kan i vissa fall skapa proteinnaggregat, en annan faktor som relaterar till åldrande [27]. Intracellulärt ROS produceras normalt som en biprodukt av mitokondriernas elektrontransportkedja, då syremolekyler ibland

kan ta upp en ensam elektron och bilda superoxidjoner [28]. Superoxidjonen, i sig en form av ROS, kan reagera vidare och bilda andra typer av ROS, t.ex. väteperoxid. Väteperoxid kan också produceras av membranprotein som svar på patogena mikroorganismer eller skador [29]. I mitokondrierna kan ROS orsaka nedsatt funktion över tid, exempelvis genom oxidativ skada på mitokondrie-DNA, men också signalera för biosyntes (nybildning av biologiskt material) [30]. Denna signalerande effekt hos ROS återfinns även i cytosolen, dit ROS kan transporteras genom bland annat akvaporiner [31]. Den signalerande effekten är inte helt utredd, men har visats vara inblandad i allt från stress-respons [32] till tillväxt [33].

I cellen finns det flera system som kan försvara mot höga ROS-nivåer [34, 35, 36]. ROS-produktion i celler hos eukaryoter har med stor sannolikhet pågått sedan mitokondrien blev en organell, och cellen har skapat system både för ROS-signalering och hantering av ROS-nivåer. Det innebär att ROS roll i cellen skulle kunna vara en cellulär klocka, som indikerar ålder, snarare än orsaken bakom biologiskt åldrande [15].

2.1.3 Cdc42

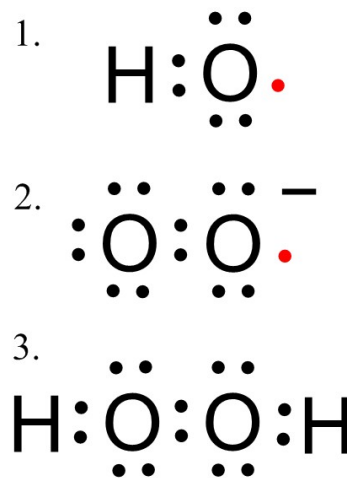
Cdc42 är ett GTPase-protein i Rho-familjen som bland annat är involverat i celldelning och cellens polarisering [37]. Detta protein är intressant att titta närmare på eftersom Cdc42 är ett så kallat konserverat protein. Det betyder att liknande protein med liknande funktion, så kallade homologer, existerar i flera olika organismer, bland annat människan [38].

Cdc42 bildar, via interaktioner med andra proteiner, ett kluster på cellmembranet som bestämmer var den nya knoppen ska växa ut då jästen knoppar sig [39]. Detta kallas att cellen polariseras. Polarisering gör att jästcellens symmetri bryts så att olika riktningar i cellen kan definieras. I vild-typ jäst finns det flera sorters redan existerande asymmetrier såsom gamla knoppningsår som utnyttjas för att bestämma var Cdc42-klustret ska bildas. I jästceller där alla kända existerande asymmetrier tagits bort polariseras cellen trots avsaknad av dessa, så länge Cdc42 och vissa relaterade proteiner fortfarande är funktionella. Detta har lett till en hypotes om att Cdc42 tillsammans med interagerande proteiner självmant kan bryta symmetrin hos jästceller om det skulle behövas [37, 39].

Proteinfamiljen Rho-GTP:aser, där Cdc42 ingår, har gemensamt att de kan skifta mellan en aktiv och en inaktiv form. Den aktiva formen har GTP (guanosintrifosfat) bundet till sig och om GTP:n hydrolyseras till en GDP (guanosindifosfat) blir GTP:aset inaktivt. I inaktiv form är även Cdc42 komplexbunden med en inhibitor, GDI (GDP dissociation inhibitor), som förhindrar Cdc42 att aktiveras spontant. Aktiva GTP:aser binder till och påverkar andra proteiner som kallas deras effektorer. Många effektorer och proteiner som reglerar aktiviteten hos Cdc42 är kända [40]. Cdc24 är ett protein som katalyserar inbindningen av GTP till Cdc42 och därmed aktiverar det. Bem1 är en av Cdc42s effektorer och är speciellt intressant. Aktivt Cdc42 binder till Bem1 som i sin tur binder till Cdc24. Det blir en sluten cykel mellan Cdc24, Cdc42 och Bem1 och det har föreslagits att denna cykel fungerar som en positiv feedback loop och är orsaken till cellens förmåga att självmant polariseras [41].

GTP:ase Activating Proteins (GAP) aktiverar den GTP-hydrolyserande förmågan hos GTP:aser. När GTP hydrolyseras till GDP blir GTP:aset inaktivt [40]. GAP:s inaktiverar alltså indirekt GTP:aser vilket kan tyckas lite motsägelsefullt med tanke på deras namn. Flera GAP:s har hittats men många detaljer kring de enskilda proteinerna är fortfarande okända [39].

I hematopoietiska stamceller (HSC) i möss och människa försämras förmågan till polarisering i gamla celler. Samtidigt finns det en korrelation mellan överaktivitet av Cdc42 och dåligt fungerande polarisering samt tidigt biologiskt åldrande av HSC [10, 11]. Dessutom har



Figur 3: De viktigaste reaktiva syre-specierna (ROS) i cellen, sorterade enligt reaktivitet: 1. Hydroxylradikal 2. Superoxidjon 3. Väteperoxid, författarens egen bild

försök visat på föryngring av HSC då Cdc42 har inhiberats i gamla celler [11].

Det finns modeller som beskriver hur Cdc42 polariserar jästceller inför celledelning [37]. Dessa metoder saknar eventuell påverkan från ROS på Cdc42. Både ROS och Cdc42 har visat sig ha roller i cellers åldrande och Cdc42 är en av de viktigaste komponenterna i jästcellers polarisering.

2.2 Matematik

Huvudsakligen används två matematiska verktyg i arbetet; modellering och parameteruppskattning. Dessa metoder beskrivs i följande teoriavsnitt.

2.2.1 Matematisk modellering

Modellering bygger på att skapa en modell som beskriver en process. En matematisk modell kan bestå av system av kopplade differentialekvationer (DE), som antingen är ordinära (ODE) eller partiella (PDE). Skillnaden mellan ODE:er och PDE:er är att de beror av en respektive flera variabler. Ett vanligt val av variabel vid användning av ODE:er är tid och för PDE:er tid och rum. Både ODE-system och PDE-system är vanliga för att modellera kemiska eller biologiska processer [42].

Kvantitativa modeller för kemiska eller biologiska fenomen baseras ofta på *massverkans lag*. Lagen innebär att en koncentrationsförändring av ett ämne i en reaktion, med avseende på tid, kan beskrivas som koncentrationerna av de ingående substraten multiplicerat med varandra samt en hastighetskonstant [43]. Ett exempel på detta är reaktionen



där produktionshastigheten av ämne C kan beskrivas matematiskt som

$$\frac{dC}{dt} = k \cdot A \cdot B$$

där k är hastighetskonstanten.

För mer komplicerade ODE:er eller kopplade ODE-system med många variabler, existerar det inga explicita analytiska lösningar. Detta kräver i sin tur numeriska metoder för att få ut en approximerad lösning. Exempel på dessa är så kallade Runge-Kuttametoder, vilket MATLAB använder sig av i dess ODE-lösare (ode45, ode15s, ode23tb etc.) [44].

2.2.2 Parameteruppskattning

För att sätta numeriska värden på teoretiska modeller, används det fält inom matematiken som kallas parameteruppskattning [18]. Det går ut på att uppskatta ingående parametrar i en modell, givet mätdata. Genom att dessutom begränsa vilka värden parametrarna kan anta och testa vilka modeller som anpassar data bäst, kan man på så sätt styrka valet av modell. Detta kräver att modellen är konstruerad på rätt sätt för att kunna identifiera parametrarna. Fältet som behandlar denna process kallas för identifierbarhetsanalys [45].

Betrakta en modell av kopplade ordinära differentialekvationer (ODE:er) med N stycken tidsberoende variabler ($x_n, n \in \{1, 2, \dots, N\}$) samt M stycken parametrar ($k_m, m \in \{1, 2, \dots, M\}$). Dessa variabler och parametrar skulle i detta fall kunna motsvaras av olika halter av vissa kemiska föreningar respektive hastighetskonstanter för deras motsvarande reaktioner. Följande system av ODE:er erhålls därefter:

$$\begin{aligned} \frac{d(x_1)}{dt} &= f(x_1, \dots, x_N, k_1, \dots, k_M, t) \\ \frac{d(x_2)}{dt} &= f(x_1, \dots, x_N, k_1, \dots, k_M, t) \\ &\vdots \\ \frac{d(x_N)}{dt} &= f(x_1, \dots, x_N, k_1, \dots, k_M, t) \end{aligned} \tag{1}$$

där vänsterledet beskriver förändringen av halten av en viss förening med avseende på tiden och högerledet hur förändringen kan skrivas som en funktion av de ingående föreningarna och dess hastighetskonstanter. Givet en mängd experimentellt uppmätta datapunkter vid olika tidpunkter gäller det att finna de parametervärden \mathbf{k}^* som minimerar avståndet mellan datan och lösningen till systemet av ODE:er.

Denna studie innefattar två specier av Cdc42: x_1, x_2 vilket motsvarar membranbundet respektive cytosoliskt Cdc42. Deras relativa halt har uppmätts experimentellt i T stycken datapunkter ($y_i, i \in \{1, 2, \dots, T\}$) på följande form:

$$y_i = y(t_i) = \frac{x_1(t_i)}{x_1(t_i) + x_2(t_i)} \quad (2)$$

Formen på datan i (2) motiveras av att i studier där halten av fluorescens-tagat protein mäts, är det oftast endast en relativ skillnad i fluorescens som uppmäts [46, 47]. Detta kan antingen vara andelen av den ena specien relativt den andra eller den ena specien relativt den totala mängden. I detta arbete används huvudsakligen den senare formen. Den simulerade datan $\hat{y}(t_i)$ från lösningen av differentialekvationerna definieras på samma sätt som i ekvation (2).

Istället för att slumpmässigt variera \mathbf{k} (en vektor med alla parametrar) för hitta de parametrar vars simulerad data ligger så nära den experimentella som möjligt, görs problemet om till ett optimeringsproblem genom att först definiera ekvation (3); en kostfunktion ($\mathbf{CF} : \mathbb{R}^M \rightarrow \mathbb{R}_+$) enligt minstakvadratmetoden [48].

$$\mathbf{CF}(\mathbf{k}) = \sum_{i=0}^T \left\{ (y_i - \hat{y}(\mathbf{x}, \mathbf{k}, t_i))^2 \right\} \quad (3)$$

Värdet på ekvation (3) kommer helt att bero på vilka parametrar som finns i argumentet. Detta innebär att då skillnaden mellan experimentell och simulerad data är så liten som möjligt befinner sig kostfunktionen i ett minimum med parametervärden \mathbf{k}^* . En strategi för att hitta ett minimum till ekvation (3) är att gissa ett startvärde på parametrarna och röra sig i den negativa gradientens riktning tills ett minimum nås och $\nabla \mathbf{CF} = \mathbf{0}^M$ [48]. Gradienten fås genom att derivera ekvation (3) med avseende på alla parametrar (k_1, k_2, \dots, k_M). Ett element i gradientvektorn får därför följande utseende:

$$\frac{\partial \mathbf{CF}(\mathbf{k})}{\partial k_m} = -2 \sum_{i=0}^T \left\{ (y_i - \hat{y}(\mathbf{x}, \mathbf{k}, t_i)) \right\} \frac{\partial \hat{y}(\mathbf{x}, k_m, t_i)}{\partial k_m} \quad (4)$$

där $\hat{y}(\mathbf{x}, \mathbf{k}, t_i)$ fås genom att lösa ODE-systemet med hjälp av en godtycklig ODE-lösare i till exempel MATLAB. Dess partiella derivata med avseende på varje parameter är däremot inte lika självklara att få ut. Derivatans för en modell med avseende på en parameter kallas en sensitivitet, vilket är ett mått på hur känslig modellen är med avseende på en viss parameter k_m . En enkel metod är att använda sig av en finittdifferensmetod där derivatan approximeras i varje tidpunkt med en differenskvot:

$$\frac{\partial \hat{y}}{\partial k_m} \approx \frac{\hat{y}(\mathbf{x}, k_1, \dots, k_m + h, \dots, k_M, t) - \hat{y}(\mathbf{x}, k_1, \dots, k_m, \dots, k_M, t)}{h} \quad (5)$$

där h är steglängden. Värdet på h är dock svårt att välja på ett bra sätt. Då parametervärdena kan skilja sig enormt mycket i magnitud, kan ett värde på h som approximerar derivatan bra för en parameter leda till en dålig approximation av en annan. En bättre metod är att använda så kallade sensitivitetsekvationer baserade på kedjeregeln [49]. I detta fall antas en modell baserad på två specier x_1, x_2 som beror av sig själva, alla parametrar k_1, \dots, k_M samt t . Förändringen med tid för varje specie ges av en ODE och betecknas f_n där n är specien med tidsderivatan. Härledningen av sensitivitetsekvationen i (7) görs med avseende på specie

\hat{x}_1 , men görs på samma sätt för \hat{x}_2 . Härledningen är baserad på ODE systemet i (1).

$$\frac{\partial}{\partial k_m} \underbrace{\left(\frac{d\hat{x}_1(\mathbf{x}, \mathbf{k}, t)}{dt} \right)}_{f_1} = \frac{d}{dt} \underbrace{\left(\frac{\partial \hat{x}_1(\mathbf{x}, \mathbf{k}, t)}{\partial k_m} \right)}_{s_{1m}} = \frac{\partial f_1}{\partial k_m} \underbrace{\left(\frac{\partial k_m}{\partial k_m} \right)}_{=1} + \frac{\partial f_1}{\partial \hat{x}_1} \underbrace{\left(\frac{\partial \hat{x}_1}{\partial k_m} \right)}_{s_{1m}} + \frac{\partial f_1}{\partial \hat{x}_2} \underbrace{\left(\frac{\partial \hat{x}_2}{\partial k_m} \right)}_{s_{2m}} \Rightarrow \quad (6)$$

$$\Rightarrow \frac{ds_{1m}}{dt} = \frac{\partial f_1}{\partial k_m} + \frac{\partial f_1}{\partial \hat{x}_1} s_{1m} + \frac{\partial f_1}{\partial \hat{x}_2} s_{2m} \quad (7)$$

Sensitiviteterna betecknas med s_{nm} där första index är specien och andra är param etern. Den resulterande sensitivitets-ODE:n kan i samband med kopplade sensitivitets- och modell-ODE:er lösas med hjälp av en godtycklig ODE-lösare i till exempel MATLAB. Fördelen med en sensitivitets-ODE-baserad metodik är att en steglängd i k -led slipper väljas och därmed undviks problemet med finitdifferensmetoden. Det räcker därför med att endast integrera systemet med avseende på tid. De enskilda sensitiviteterna för en parameter k_m samt lösningarna till modell-ODE-systemet i (1) måste dock kombineras för att ha samma form som den experimentella datan i (2) och användas för att beräkna $\nabla \mathbf{CF}$. I fallet med två specier måste därför de två sensitiviteterna s_{1m}, s_{2m} kombineras för att bilda sensitiviteten för den simulerade datan $\frac{\partial \hat{y}}{\partial k_m}$ vilket erhålls med hjälp av att derivera ekvation (2):

$$\frac{\partial \hat{y}}{\partial k_m} = \frac{\partial \left(\frac{\hat{x}_1}{\hat{x}_1 + \hat{x}_2} \right)}{\partial k_m} = \frac{\frac{\partial \hat{x}_1}{\partial k_m} (\hat{x}_1 + \hat{x}_2) - \hat{x}_1 \left(\frac{\partial \hat{x}_1}{\partial k_m} + \frac{\partial \hat{x}_2}{\partial k_m} \right)}{(\hat{x}_1 + \hat{x}_2)^2} \quad (8)$$

\hat{y} erhålls genom att helt enkelt använda ekvation (2) i samband med lösningarna \hat{x}_1, \hat{x}_2 från ODE-lösaren. $\nabla \mathbf{CF}$ kan nu beräknas, givet en mängd parametrar. Från en initialgissning k_{in} på parametrarna gäller det att iterera sig fram till den optimala lösningen \mathbf{k}^* . En så kallad descent-algoritm har valts för optimeringen. Uppdateringssteget i itereringen görs med hjälp av en så kallad Quasi-Newton approximering [48]. En "vanlig" descent-algoritm med Newtons metod kräver tillgång till Jacobianen samt Hessianen av \mathbf{CF} för att hitta minima. Jacobianen svarar mot lutningen i parameterlandskapet och Hessianen svarar mot kurvaturen, vilket är information som kan effektivisera nedstigningen till minimum. Hessianmatrisen \mathbf{H} är komponentvis definierad som:

$$\mathbf{H}_{\mathbf{a}, \mathbf{b}} = \frac{\partial^2 \mathbf{CF}}{\partial k_a \partial k_b} \quad (9)$$

Den är dock väldigt beräkningskrävande att ta fram på grund av att kostfunktionen måste deriveras flera gånger med avseende på parametrarna, vilket gör andra metoder mer optimala. Quasi-Newton-uppdateringssteget innebär att Hessianen approximeras istället för att beräknas fullständigt [48]. Målet med parameteruppskattning är att hitta en nollpunkt till $\nabla \mathbf{CF}$ vilket kräver tillgång till Hessianen av kostfunktionen (samma som Jacobianen till $\nabla \mathbf{CF}$). Följande optimalitetskrav erhålls därefter:

$$\begin{cases} \frac{\partial \mathbf{CF}(\mathbf{k})}{\partial k_1} = 0 \\ \frac{\partial \mathbf{CF}(\mathbf{k})}{\partial k_2} = 0 \\ \vdots \\ \frac{\partial \mathbf{CF}(\mathbf{k})}{\partial k_M} = 0 \end{cases}$$

Minimum nås när optimalitetskravet är uppfyllt. Hela parameteruppskattningsalgoritmen presenteras i Algoritm 1 [48].

En annan sak överväga kring en modell och motsvarande parameteruppskattning är identifierbarheten hos dess parametrar givet modellens simulerade data. En modell kallas strukturellt identifierbar om det är möjligt att, givet den teoretiskt bästa (icke-stokastiska) datan från processen som modelleras, få ut endast ett unikt värde på alla parametrar, oavsett startgissning [45]. Vidare kallas en modell praktiskt identifierbar om det även går att få ut ett

Algorithm 1 Parameteruppskattning för att hitta optimala parametrar \mathbf{k}^*

- 1: **Utmatningar:**
 $\mathbf{k}^* \in \mathbb{R}^M$
 - 2: **Inmatningar:**
 $\mathbf{k}_{in} \in \mathbb{R}^M$
 - 3: **Initialisering:**
 \mathbf{k}_0 & $\mathbf{k}^* \leftarrow \mathbf{0} \in \mathbb{R}^M$
 - 4: **while** $\mathbf{CF} > \epsilon$
 - 5: (ϵ är ett brytvillkor som är olika lågt beroende på variationen hos den experimentella datan) **do**
 - 6: *Kostfunktion och gradientberäkning:* Beräkna $\mathbf{CF}(\mathbf{k}_{in}) \in \mathbb{R}_+$ och $\nabla \mathbf{CF}(\mathbf{k}_{in}) \in \mathbb{R}^M$
 - 7: *Hessianapproximation:* Definiera matrisen $H \in \mathbb{M}_{M \times M}$ enligt följande:
$$H \leftarrow \left(\mathbf{CF}(\mathbf{k}_{in}) \cdot \mathbf{CF}(\mathbf{k}_{in})^T \right)$$
 - 8: *Beräkna stegriktning:* Stegriktningen $\mathbf{p} \in \mathbb{R}^M$ genom att lösa följande ekvation:
$$H \cdot \mathbf{p} = -\nabla \mathbf{CF}(\mathbf{k}_{in})$$
 - 9: *Beräkna steglängd:* Beräkna steglängden $\alpha \in \mathbb{R}_+$ definierad som:
$$\alpha := \frac{\mathbf{CF}(\mathbf{k}_{in})}{\|\nabla \mathbf{CF}(\mathbf{k}_{in})\|}$$
 - 10: *Uppdatera parametrar:* Sätt
$$\mathbf{k}^* \leftarrow \mathbf{k}_{in} + \alpha \mathbf{p}$$
 - 11: **end while**
 - 12: **return** \mathbf{k}^*
-

entydigt värde på alla parametrar från en realistiskt datamängd med underliggande brus [45]. Identifierbarhet beror därför på hur många parametrar som ingår, modellens uppbyggnad, samt formen och stokasticiteten hos datan. Det finns färdiga program som till exempel STRIKE-GOLDD för att studera den strukturella identifierbarheten hos en modell [50].

3 Metod

3.1 Skapande av modeller utifrån litteraturstudier

Initialt utfördes en litteraturstudie för att lägga grunden till de modeller som ligger i fokus för rapporten. Studien var uppdelad i tre huvudområden; Cdc42 och dess roll i polarisering, reaktiva syre-specier (ROS) och hur ROS påverkar protein samt redan existerande modeller. Målet med studien var att föreslå en eller flera modeller som kan beskriva Cdc42 under polarisering i närvaro av ROS.

Då potentiella modeller identifierats bearbetades dessa för att uppnå passande nivåer av komplexitet och olika grader av ROS-påverkan infördes. Med hjälp av den bakomliggande biologin, både rörande ROS och polarisering, kunde en del antaganden göras och ODE-systemen som modellerna bestod av förenklas. Systemen bearbetades tills de endast bestod av två variabler och undersöktes sedan i parameteruppskattningen. Mer detaljer om härledning och modellerna presenteras i avsnitt 4.2 samt Bilaga A.

3.2 Datainsamling

Initialt tillhandahölls testdata för koden. Testdatan beskriver ett närliggande fenomen, en FRAP-studie. FRAP står för Fluorescence Recovery After Photobleaching och är en metod för att undersöka rörlighet hos molekyler [51]. Efter viss diskussion ansågs polariseringen

inte följa samma mekanismer som FRAP och testdatan ansågs därför inte kunna ge biologisk relevans hos uppskattade parametrar.

För att genomföra parameteruppskattningen korrekt krävdes annan data som beskriver de biologiska processer som modelleras. Då projektet i sig inte inkluderar produktion av sådan data behövde detta inhämtas. Viss sökning efter passande data genomfördes därför. Den inhämtade datan bearbetades för att få korrekt form för modellerna. Bearbetningen innebar dock att datan blev enhetslös, något som påverkar enheten för parametrarna i modellerna. Testdata, rådata, bearbetad data samt metod för bearbetning av data presenteras i Bilaga B.

3.3 Parameteruppskattning i Copasi

Parameteruppskattningen av modellerna genomfördes i två skilda program; MATLAB och Copasi. Copasi är en programvara skapat för att testa matematiska modeller av biologiska fenomen. Programvaran har förmågan att göra flera olika tester på de modeller som önskas prövas och i detta arbete användes verktyget parameteruppskattning. Genom att ladda ner den data som modellen ska anpassas till, välja i vilket storleksintervall parametrarna kan tänkas befinna sig inom, samt välja initialgissning för varje parameter, kunde en uppskattning göras. Resultatet från uppskattningen illustrerades av figurer samt tabeller som beskrev storleken på parametrarna. Resultatet från parameteruppskattningen i Copasi presenteras i avsnitt 4.2.

Nackdelen med Copasi är att det inte går att analysera hur programmet arbetar, vilket gör eventuell felsökning väldigt komplicerad. Av denna anledning skrevs en egen kod för parameteruppskattning i MATLAB med syftet att få en djupare förståelse för hur uppskattningen fungerar och för att underlätta eventuell felsökning.

3.4 Implementering

Den matematiska teorin beskriven i avsnitt 2.2.2 implementerades i ett parameteruppskattningsprogram skrivet i MATLAB. För en översikt av de olika stegen i programmet, se Figur 4. Programmet är i sin nuvarande form byggt för en ODE-modell med två ekvationer, vilket motsvarar aktiv och passiv form av Cdc42-proteinet. Datan består av en tidsserie med observationer av det membranlokaliserade aktiva Cdc42 dividerat med den totala mängden Cdc42. Programmet är relativt lätt att anpassa till uppskattning av ett annat antal parametrar och till ändringar i modellens ekvationer. Att ändra till en modell med ett annat antal ekvationer eller till data på en annan form kräver mer omskrivning av programmet. Hela koden presenteras i bilaga C. Programmets olika delar och funktion beskrivs i detta avsnitt och notationen följer programmets för att underlätta för användaren. Det enda som skiljer programmets notation från teorins i de bitar som nämns nedan är; $\nabla\mathbf{CF}=\text{gradCF}$, $\hat{y}=yhat$, $Cdc42_m=x1$ och $Cdc42_c=x2$

Programmet tar in ODE:er, data och initialgissningar till parametrar. Dessa anges i main.m av användaren av programmet, se Figur 4. Filen main.m anropar parameterestimation.m (PE.m) vilket är den .m-fil som resten av koden utgår från. I PE.m deriveras ODE:erna med avseende på de olika parametrarna för att få sensitivitetsekvationerna för de parametrar som ska uppskattas. Detta görs symboliskt med hjälp av ett par olika funktioner från MATLAB's symboliska toolbox. När en ny modell ska användas i programmet, eller ett annat antal parametrar ska uppskattas krävs anpassning av koden i PE.m, genom att ändra sensitivitetsekvationerna. Se avsnitt 2.2.2 för mer information om sensitivitetsekvationerna. De framtagna sensitivitetsekvationerna bildar tillsammans med modell-ODE:erna som användaren gett som input till programmet ett större system av kopplade ODE:er. Ekvationssystemet skickas vidare till sensitivities.m där hela systemet löses numeriskt med hjälp av ODE45. Ibland har ODE23tb gett bättre resultat, så det kan vara värt att testa med båda. Lösningarna i form av tidsserier av de två specierna används för att beräkna ett output $yhat = \frac{x1}{x1+x2}$, det vill säga på samma form som datan den ska jämföras med. På liknande sätt omvandlas även sensitiviteterna till samma form som datan. Detta görs enligt ekvation (8) i avsnitt 2.2.2. Därefter interpoleras $yhat$ och dess sensitiviteter till samma tidpunkter som i datan.

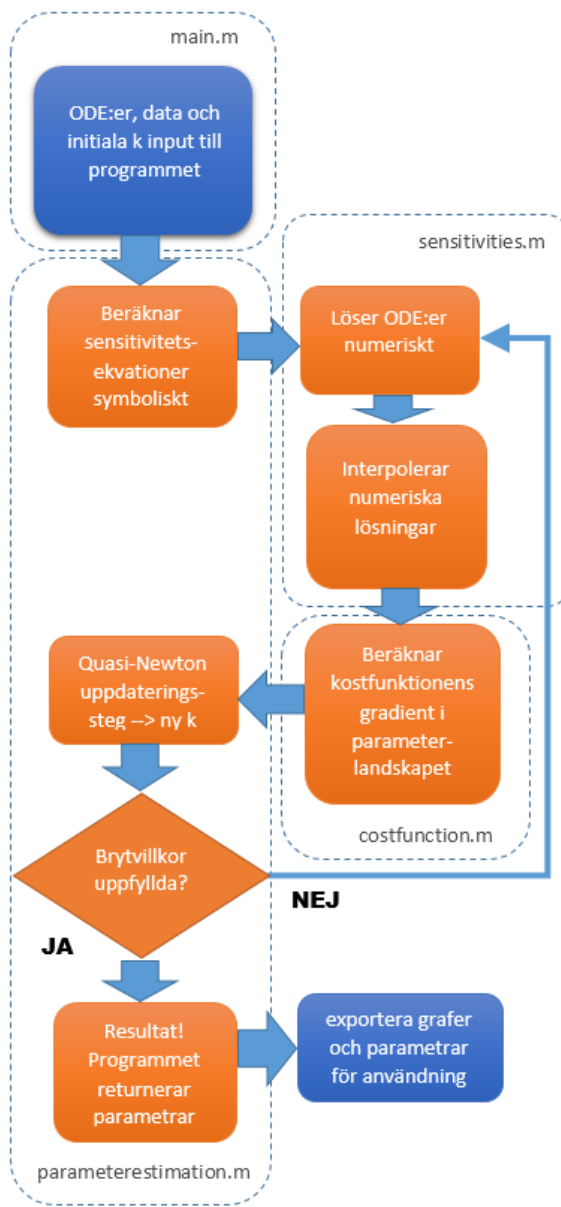
Detta görs för att förbereda en jämförelse mellan data (y) och y_{hat} . Filen sensitivities.m anropar sedan costfunction.m där kostfunktionens (CF, Cost Function) värde beräknas tillsammans med kostfunktionens gradient i parameterområdet (gradCF).

I programmet används minstakvadratmetoden för att definiera kostfunktionen, se avsnitt 2.2.2. CF och gradCF returneras till parameterestimation.m (via sensitivities.m) och används i descent-algoritmens uppdateringssteg. Även descent-algoritmen presenteras i avsnitt 2.2.2 i Algoritm 1. Uppdateringssteget stegar i parameterlandskapet mot minskande kostfunktion och ger en ny uppsättning parametrar som ska uppskattas. CF jämförs med ett toleransvillkor som användaren sätter, se diamantformade boxen i Figur 4. Om CF:s värde är mindre än den angivna toleransen returnerar programmet värdena på den sista uppsättningen parametrar och avslutar. Lämplig toleransnivå varierar beroende på mängden brus i datan och användaren rekommenderas laborera med toleransnivån tills en lämplig nivå hittas. Enligt vår erfarenhet är det rättfram. Om $CF > \text{toleransnivån}$ loopar PE.m tillbaka och anropar sensitivities.m igen, se Figur 4. Nästa loop börjar med att ODE-systemet löses igen med de nya parametrarna och fortsätter därifrån, se Figur 4.

För varje loop ritas programmet ut den senaste lösningen i en graf där även datan är utritad så att programmets framsteg kan följas visuellt. Ibland blir det ganska många linjer totalt så användaren kan välja att rita ut var n :te lösning istället, där n är valfritt heltal. Detta ställs in i PE.m. I koden, bilaga C, finns mer detaljerade anvisningar om var användaren ska ändra, både för modifieringar av grafritandet och eventuella andra inställningar som behöver göras.

När parametrarna motsvarar kinetik-konstanter, som i detta fall, ska de vara positiva. För att begränsa parameteruppskattningen till positiva parametrar kan logaritmerade parametrar användas enligt $k1 \equiv e^{\ln(k1)}$ samt $k2 = \ln(k1)$, där $k1$ är den ursprungliga parametern i ekvationerna och $k2$ är en ny parameter som uppskattas istället. $k1$ kan sedan beräknas som e^{k2} . Detta begränsar den uppskattade $k1$ till positiva tal eftersom $e^{k2} > 0$ för alla $k2$. Koden som presenteras i Bilaga C har logaritmerade parametrar eftersom det förväntas att parametrarna ska vara positiva.

Förutom de MATLAB-filer som nämns ovan och i Figur 4 används två m.-filer till i



Figur 4: Flödesschema som visar hur programmet jobbar och i vilka MATLAB-filer de olika delarna av programmet ligger. Rutorna (orange eller blå) visar de viktigaste stegen i programmets beräkningar. Pilar visar flödesriktning. Streckade rutor visar i vilken MATLAB-fil de olika beräkningsstegen ligger. Namnet på matlabfilen står i den streckade rutan. Tillsammans ger rutorna en översikt över programmets viktigaste delar.

programmet (totalt sex stycken .m-filer). Den första, data.m är en mycket enkel funktion som anropas från main och returnerar data som en variabel. Dess syfte är endast att hålla långa dataset ute ur de script som ofta görs ändringar i. Användaren rekommenderas att lägga in sin data enligt exemplet i data.m men det går också att lösa på andra sätt, så länge det skapas en variabel med datan i main.m enligt rätt format. Den andra m-filen är en fil som skapas automatiskt av en av funktionerna som hanterar de symboliska uttrycken i PE.m och kräver ingen input från användaren. Slutligen finns datagenerering.m vilken är en fil som genererar data med brus från valfri ODE-modell. Denna används inte vid normal körning av programmet men är mycket smidig att ha vid utveckling och ersätter då riktig data från data.m.

4 Resultat

De resultat som producerats kommer att delas in i två delar: "Parameteruppskattning" och "Modeller och dess anpassning till data". I avsnittet "Parameteruppskattning" redogörs resultatet från MATLAB-koden och hur väl programmet fungerar för att utföra en parameteruppskattning. I "Modeller och dess anpassning till data" kommer de modeller som tagits fram, samt de antaganden som gjorts kring varje modell, att presenteras tillsammans med resultatet från parameteruppskattningen som gjordes i Copasi.

4.1 Parameteruppskattning

polariseringsfenomenet MATLAB-programmet som skrevs för detta arbete presenteras i detalj i implementeringsavsnittet 3.4. Här presenteras främst under vilka villkor programmet fungerar och när det inte gör det. Med villkor menas saker som modell, hur många parametrar som uppskattas, datans form och olika inställningar i programmet, såsom hurvida parametrarna är logaritmerade eller inte. Oavsett villkor stegar programmet oftast närmre och närmre en bra anpassning till data från en iteration till nästa. Programmet har dock problem där det oväntat divergerar från data och tar stora hopp åt fel håll vid vissa iterationer. När det hoppar snett är det också vanligt att det inte återhämtar sig utan fortsätter åt något felaktigt håll i ett par mycket stora hopp innan det kör fast vid en uppsättning parametrar som är helt orimliga. Problemet med hoppen verkar vara relaterat till numeriska problem. Approximationen av hessianen (\bar{H}) som används i quasi-Newton algoritmens uppdateringssteg är oftast nära singulär när programmet körs vilket kan leda till dålig precision i beräkningarna. Detta skulle kunna leda till att algoritmen stegar år fel håll. Problemet med numeriken verkar bli värre ju närmare en bra anpassning till data programmet kommer genom att risken för hopp i felaktig riktning blir större. En möjlig anledning till de numeriska problemen skulle kunna vara att modellerna inte är praktiskt identifierbara. De enda fall då programmet gett övertygande resultat är i de fall då endast en parameter uppskattats åt gången. Det sammanfaller med att de modeller där bara en parameter uppskattas har störst chans att vara praktiskt identifierbara. I detta avsnitt presenteras ett antal situationer programmet har testats i och hur väl programmet fungerat i dessa. Först beskrivs vilken data och vilka modeller som programmet har testats med.

Programmet har utvecklats med FRAP-data som testdata, se tabell 6 i Bilaga B. FRAP-data användes eftersom den antogs ha en form som kvalitativt liknande polariseringsfenomenet och då data från polarisering saknades när programmet började utvecklas. Förutom FRAP-datan har data simulerats som ett verktyg i utvecklingen. Den simulerade datan består av numeriska lösningar av ODE-modellen med normalfördelat brus tillagt. Fördelen med simulerad data är att de underliggande parametrarna i modellen är fullständigt kända. Parametrarna i den simulerade datan valdes som de parametrar som gav en bra anpassning till FRAP-data i Copasi. Senare i projektet blev data som beskriver polariseringsfenomenet tillgänglig. Då visade det sig att formen på den var sigmoidal snarare än logaritmisk som FRAP-datan. I avsnitt 4.2 som behandlar de olika modellerna presenteras hur de olika modeller som föreslagits i detta arbete beter sig då de anpassas till den sigmoidala datan.

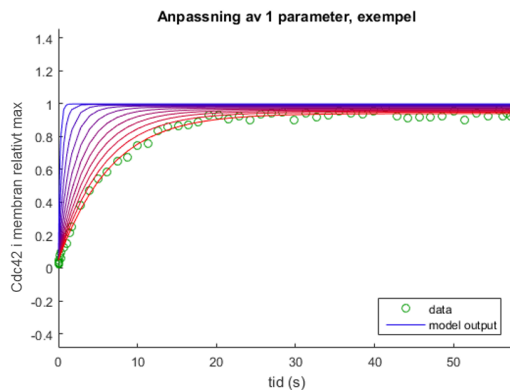
Det visade sig svårt att få parameteruppskattningsprogrammet att fungera med den sigmoidala datan. För en diskussion av detta med möjliga förklaringar till varför, se avsnitt 5. På grund av ovanstående kommer resultaten angående programmet presenteras med FRAP-data eller simulerad data. En till följd av att inga resultat har uppnåtts med den sigmoidala datan är att inga biologiskt relevanta parameteruppskattningar har kunnat göras med programmet. För att få fram biologiskt relevanta parameteruppskattningar har ett annat program, Copasi, använts. De parameteruppskattningar som tagits fram med Copasi presenteras också i avsnitt 4.2

Vid utveckling av programmet har framförallt en av de modeller som presenteras i 4.2 använts. Det exakta utseendet av den modellen beskrivs i avsnitt 4.2.1. Det är en modell bestående av två ODE:er med totalt fyra parametrar. Förutom den har programmet testats med en lite mer komplex modell med två ODE:er och totalt sju parametrar, vilken beskrivs i avsnitt 4.2.2. Inga resultat uppnåddes med den mer komplexa modellen. I de testningar som gjordes med den uppskattades alla sju parametrar samtidigt och så många parametrar leder antagligen till allvarliga problem med identifierbarheten. Mer om det nedan och i diskussionen, avsnitt 5. Resterande del av detta avsnitt kommer att fokusera på den mindre komplexa modellen, härnäst bara "modellen".

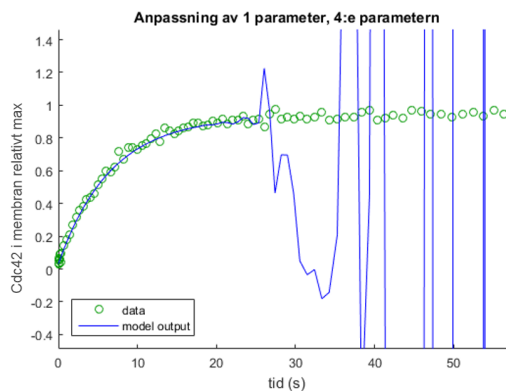
Innan resultaten presenteras är det en sista sak att nämna. I de resultat som presenteras nedan används en variant av sensitivitetsekvationerna där den sista termen i ekvation (7) inte är med. Denna version användes under större delen av projektet innan det upptäcktes att den antagligen inte stämde med teorin. När sensitivitetsekvationerna enligt ekvation (7) testades var det svårt att få programmet att konvergera till en anpassning av data. Vad detta beror på är oklart och det rekommenderas varmt att undersöka detta i framtiden. Notera att när den sista termen inte är med är varje sensitivitet oberoende av alla andra sensitiviteter, möjligtvis kan det vara en del av förklaringen.

Programmet har testats vid parameteruppskattning av olika antal parametrar, och det tydliga mönstret är att programmet i hög utsträckning klarar av att utföra parameteruppskattningen med gott resultat då endast en parameter uppskattas åt gången. När flera parametrar uppskattas samtidigt har programmet mycket svårare att konvergera till en bra anpassning till data och divergerar istället. Trots det har vissa resultat uppnåtts även då två eller fler parametrar uppskattats samtidigt. Nedan följer resultat, först då en parameter uppskattas åt gången och sedan då flera parametrar uppskattas samtidigt.

Programmet klarar av att uppskatta tre av fyra parametrar i modellen då de uppskattas



(a) Figuren visar en lyckad parameteruppskattning då bara en parameter uppskattas. Liknande resultat uppnås med tre av de fyra parametrarna i modellen. Den genererade datan visas som gröna cirklar medan linjerna visar modellens output från olika iterationer, där färgen går från blå till röd med ökande antal iterationer. Programmet når på sista iterationen en bra anpassning till data.

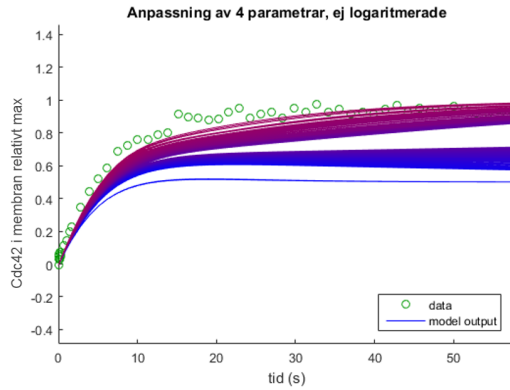


(b) Figuren visar ett typiskt resultat där den fjärde parametern i modellen uppskattas. Den genererade datan visas som gröna cirklar medan linjen i blått visar modellens output. Den numeriska lösningen av ODE:erna divergerar redan under första iterationen.

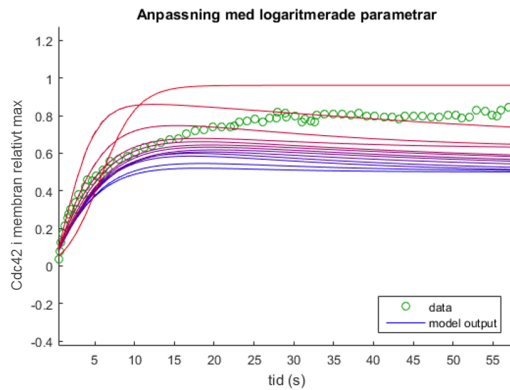
Figur 5: Parameteruppskattningar för enskilda parametrar.

var och en för sig. Det är oklart varför den sista parametern, låt kan kallas k_4 , inte fungerar. I Figurerna 5a och 5b kan man se typiska resultat då en parameter i modellen uppskattas. De andra tre har i dessa fall satts till de "rätta" parametervärdena, det vill säga värdena som användes för att generera datan. Formen på den simulerade datan liknar i hög grad FRAP-datan. Figur 5a visar hur programmet lyckas göra en bra uppskattning av parametern och kommer nära data efter ett antal iterationer. Detta trots att initialgissningen ligger långt ifrån det rätta parametervärdet. Liknande resultat fås för de andra två parametrar som fungerar bra att uppskatta. Figur 5b visar hur programmet inte klarar att uppskatta den fjärde parametern k_4 i modellen. Istället syns ett kantigt, oregelbundet oscillerande beteende hos lösningen av ODE:erna. Det är intressant att den numeriska lösningen med de initiala parametrarna följer datan mycket nära innan den börjar divergera. Detta kan tyda på att k_4 inte heller påverkar modellens output nämnvärt. I så fall är den parametern oidentifierbar per definition. Det är dock oklart varför det uppstår problem med den numeriska lösningen av ODE:erna.

När flera parametrar uppskattas samtidigt är risken att parameteruppskattningen ska divergera från data mycket större än då bara en uppskattas. Trots det har vissa resultat erhållits. I Figur 6a visas en någorlunda lyckad parameteruppskattning då alla fyra parametrar i modellen uppskattats samtidigt. För att åstadkomma det krävs ett visst laborerande med initialgissningarna till parametrarna och toleransnivån för parameteruppskattningen samt ett flertal körningar av programmet. Notera dock att parametrarna skiljer sig från de som användes för att generera datan. Uppsättningen parametrar som programmet kommer fram till är $k_1=0.1618$, $k_2=0.1061$, $k_3=0.1191$, $k_4=-0.0618$ medan uppsättningen som använts för att generera datan är $k_1=0.019$, $k_2=0.01$, $k_3=0.14$, $k_4=0.0001$. Att flera uppsättningar parametrar kan ge samma output från en modell är ett typiskt tecken på att modellen inte är identifierbar. Som nämnt i inledningen av detta avsnitt är hessianen (\bar{H}) nästan singular och detta problem har funnits för alla parameteruppskattningar då två eller fler parametrar har uppskattats. När endast en parameter uppskattas har \bar{H} bara ett element och parameteruppskattningen går mycket bättre. Detta kan vara en indikation på att modellerna ej är identifierbara när fler än en parameter uppskattas åt gången och att de numeriska problemen orsakas av identifierbarhetsproblem.



(a) Figuren visar en anpassning av 4 parametrar, där parametrarna inte är logaritmerade. Den genererade datan visas som gröna cirklar medan linjerna visar modellens output från olika iterationer, där färgen går från blå till röd med ökande antal iterationer. Anpassningen kommer hyfsat nära data men är långt ifrån perfekt. Vidare skiljer sig parametrarna programmet kommer fram till markant från de som datan genererats med. Uppskattade parametrar är $k_1=0.1618$, $k_2=0.1061$, $k_3=0.1191$, $k_4=-0.0618$, medan datan genererats med $k_1=0.019$, $k_2=0.01$, $k_3=0.14$, $k_4=0.0001$. Detta är typiskt för när modellen ej är identifierbar.



(b) Figuren visar ett typiskt resultat då alla fyra parametrar anpassas till data och parametrarna är logaritmerade. Datans visas som gröna cirklar medan linjerna visar modellens output från olika iterationer, där färgen går från blå till röd med ökande antal iterationer. Programmet närmar sig först data men lyckas inte hitta en bra anpassning. Efter den sista utritade iterationen blir de nya parametrarna fullständigt orimliga och den numeriska lösningen av ODE:erna falerar.

Figur 6: Parameteruppskattningar av alla fyra parametrar samtidigt.

Den strukturella identifierbarheten har undersökts med MATLAB-programmet STRIKE-GOLDD [50]. Flera varianter av modellen, både när en och två parametrar uppskattas, är strukturellt identifierbara. Strukturell identifierbarhet har dock endast att göra med identifierbarhet i fallet då perfekt data är tillgänglig, vilket sällan är fallet inom biologin. Flera av de varianter som är strukturellt identifierbara behöver alltså inte vara praktiskt identifierbara givet det dataset som är tillgängligt i detta arbete. Det finns metoder för att hantera praktisk identifierbarhet vilket diskuteras i avsnitt 5 men ingen sådan har applicerats i detta arbete. Därför förblir det osäkert exakt i vilken grad de strukturellt identifierbara modellerna även är praktiskt identifierbara. Chansen att en modell ska vara praktiskt identifierbar är dock större ju färre parametrar som uppskattas. Att programmet hanterar varianter av modellen där endast en parameter uppskattas så mycket bättre kan tyda på att det finns problem med den praktiska identifierbarheten i de varianter där två eller fler parametrar uppskattas.

Som beskrivet i implementeringsavsnittet, avsnitt 3.4, kan parametrarna i modellen logaritmeras vilket påverkar programmet på flera sätt. Fördelen är att uppskattningen begränsas till positiva parametrar. I detta fall är det mycket bra då modellen är konstruerad så att enbart positiva parametrar förväntas. Vidare har de flesta av de modeller som testats problem med identifierbarheten. Det leder till att det inte finns en unik uppsättning parametrar som producerar ett visst output. Istället kan flera uppsättningar parametrar ge samma output. Genom att tvinga fram positiva parametrar undviks de lösningar som innehåller negativa parametrar vilket ger större chanser att hitta relevanta parametrar vid en uppsättning. Den stora nackdelen med logaritmerade parametrar är att programmet har haft mycket svårare att konvergera. Överlag har inga övertygande anpassningar till data uppnåtts med logaritmerade parametrar. Figur 6b visar en typisk körning där alla parametrar logaritmerats och programmet inte lyckas anpassa modellen till data. Jämför med 6a där alla fyra parametrar uppskattas och där parametrarna inte logaritmerats i modellen. Notera dock att en parameter i Figur 6a är svagt negativ vilket inte förväntas i ett biologiskt sammanhang och att parametrarna skiljer sig från de som användes för att generera datan, som nämnt ovan.

4.2 Modeller och deras anpassning till data

Utifrån den litteraturstudie som gjorts har tre olika modeller genererats. Anledningen till varför tre modeller har undersökts är på grund av att modellerna har en varierande komplexitet och baseras på olika antaganden. Skillnader och likheter kommer att tas upp mer i detalj vid beskrivningen av respektive modell.

Den första modellen baseras på de litteraturstudier som gjorts kring biologin angående polarisering samt påverkan av ROS, och ett ODE-system har skapats utifrån detta. De två andra är baserade på en redan befintlig modell som sägs beskriva polariseringen i jäst. Modellen är tagen från en artikel, skriven av Andrew Goryachev från 2008, där ODE-systemen från artikeln används som bas och inverkan av ROS på Cdc42 har inkorporerats [37]. De två modellerna behandlar fallen där ROS reagerar med endast cytosoliskt Cdc42 respektive cytosoliskt och membranbundet Cdc42. Anledningen till varför båda undersöks är på grund av att modellen där ROS reagerar med cytosolsikt Cdc42 är matematiskt mer simpel och innehåller färre okända parametrar. Det kan därför vara bra att testa båda, utifall att den mer komplicerade modellen är för svår att implementera i ett program som hanterar parameteruppskattning.

I alla modeller har koncentrationen av ROS antagits vara konstant, med motiveringen att ROS kan antas befinna sig i stationärtillstånd under det korta tidsintervall som polariseringen upptar. Vid stationärtillstånd sker inga förändringar av koncentrationer utan ROS håller samma koncentration. Det finns tecken i litteraturen på att ROS kan variera med cellcykelprogression [52], men polariseringen upptar bara en liten del av cellcykeln. Koncentrationen av ROS antas även vara jämt fördelad över hela cellens volym, då modellen inte hanterar diffusion.

En uppskattning av ROS-koncentrationen i celler existerar [53]. Däremot finns det inget som säger hur mycket ROS relativt Cdc42 som finns, ett förhållande som skulle behövas för att få mängden ROS på samma form som den data som används. Eftersom andelen ROS är okänd kommer en parameteruppskattning endast kunna avgöra storleken på produkten av koncentrationen och hastighetskonstanten, men inte vardera faktor separat. Av denna

anledning kommer hastighetskonstanten och ROS-koncentrationen kombineras till en enda parameter och benämnas som k_{ROS} . Ett ytterligare antagande är att k_{ROS} är densamma oavsett modell och om den reagerar med cytosoliskt eller membranbundet Cdc42. Detta är på grund av ROS-molekylernas reaktivitet [54]. Då ROS är betydligt mer reaktivt än proteinet det reagerar med antas bidraget från ROS till hastighetskonstanten dominera över proteinets. Skillnaden mellan struktur hos membranbundet respektive cytosoliskt Cdc42 antas även vara liten, vilket bör leda till likartade, om inte identiska, hastighetskonstanter för de olika specierna.

4.2.1 Modell baserad på litteraturstudier

Framtagandet av denna modell baseras på artiklar som beskriver polariseringen ur ett biologiskt perspektiv. Modellen tar i beaktning att det sker en kontinuerlig skyttel mellan inaktivt cytosoliskt Cdc42 ($Cdc42_c$) och dess aktiva membranbundna form ($Cdc42_m$). En skyttel som vid cellytan, där polariseringen kommer äga rum, gynnar $Cdc42_m$ på grund av den positiva feedback-loop som presenterades under avsnitt 2.1.3 [37, 55, 56]. Däremot inkluderas inte Bem1 och Cdc24 i modellen. $Cdc42_m$ antas alltid vara inbundet till komplexet och bidrar därmed till feedback-loopen. Modellen tar dock hänsyn till att ROS reagerar med både $Cdc42_m$ och $Cdc42_c$ och bildar oxidativt skadat Cdc42 ($Cdc42_{os}$). Ett reaktionsschema för de ingående reaktionerna presenteras i Figur 7.

Utifrån de reaktioner som presenteras i Figur 7, samt antagandet att reaktionerna kan beskrivas med hjälp av *massverkans lag* [43], har ett ODE-system skapats enligt ekvation (10)

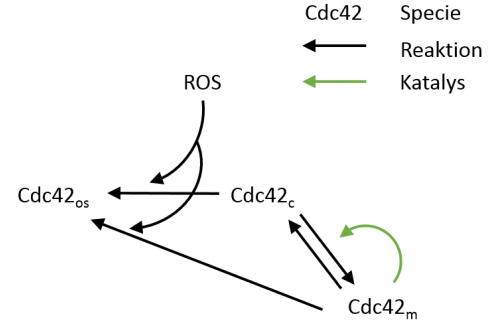
$$\begin{aligned} \frac{d(Cdc42_c)}{dt} &= -k_c \cdot Cdc42_c + k_m \cdot Cdc42_m - k_{kat} \cdot Cdc42_c \cdot Cdc42_m - k_{ROS} \cdot Cdc42_c \\ \frac{d(Cdc42_m)}{dt} &= k_c \cdot Cdc42_c - k_m \cdot Cdc42_m + k_{kat} \cdot Cdc42_c \cdot Cdc42_m - k_{ROS} \cdot Cdc42_m \end{aligned} \quad (10)$$

där k_i är olika parametrar och koncentrationen ROS , som tidigare nämnt, är inkluderad i parametern k_{ROS} .

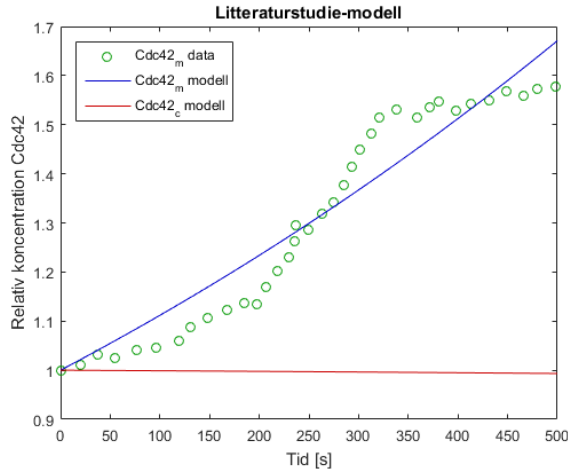
En parameteruppskattning i Copasi gjordes på ODE-systemet och de parametrar som gav bästa passform till datan utvanns. Värdet på parametrarna användes i en MATLAB-kod, som löser de olika differentialekvationerna som modellerna består av, och figurer som illustrerar resultatet producerades. Hur väl modellen passade data uppskattades sedan genom att beräkna avståndet mellan datapunkterna och modellen med hjälp av minstakvadratmetoden.

I Copasi finns möjligheten att bestämma i vilken del av cellen en specie ska förekomma. Därför definierades en cytosolvolymer för $Cdc42_c$ samt en membranvolymer för $Cdc42_m$. Detta gav dock en förändring i system (10) som resulterade i att alla termer, förutom de där k_{ROS} ingår, divideras med respektive volym. Termerna i ekvationen för $Cdc42_c$ divideras alltså med cytosolvolymer och de i ekvationen för $Cdc42_m$ med membranvolymer. Cytosolens volym sattes till $50\mu m^3$, enligt litteratur om haploida celler under celdelning [57], och membranets till $0,5\mu m^3$ då membranvolymer antas vara hundra gånger mindre än cytosolvolymer [37]. Resultatet från parameteruppskattningen illustreras i Figur 8 och parametrarnas värde respektive det beräknade minstakvadrat-avståndet presenteras i Tabell 2. Notera att parametrarna i Tabell 2 saknar enhet. Då datan som används är enhetslös saknar flera av parametrarnas enheter relevans. Detta gäller för alla modeller framöver.

Ringarna motsvarar datapunkter för $Cdc42_m$, den blå linjen är $Cdc42_m$ enligt modell och den röda är $Cdc42_c$ enligt modell. Datat som används är tagen från Tabell 8 för organism 2, se Bilaga B. Denna data kommer att användas i samtliga parameteruppskattningar i Copasi. Notera att det inte finns någon enhet på y-axeln då mängden Cdc42 anges i andelar. Notera även att cytosolvolymer är hundra gånger större än membranvolymer och graden av koncentrationsförändringar därmed blir av motsvarande karaktär.



Figur 7: Reaktionsschema för $Cdc42$ under påverkan av ROS enligt ODE-system (10)



Figur 8: Resultatet av parameteruppskattningen i Copasi av system (10)

Då modellen inte har lyckats anpassas till data, vilket syns tydligt i Figur 8, kan heller inga slutsatser dras från detta resultat.

4.2.2 Goryachev - ROS attackerar cytosoliskt Cdc42

Här presenteras varianten av Goryachev där ROS endast reagerar med cytosoliskt Cdc42. Det initiala ODE-systemet består av åtta differentialekvationer som beskriver alla mekanismer som ingår i aktiveringen och inaktiveringen av Cdc42: Från att inaktivt Cdc42 komplexbundet med GDI transporteras till membranet och aktiveras av Cdc24-Bem1-komplexet, till att Cdc42 inaktiveras av GAP och transporteras ut i cytosolen igen [37]. Detta ODE-system reduceras sedan, med hjälp av ett antal antaganden, till ett system med endast två differentialekvationer samt en hjälpfunktion. Systemet blir därmed

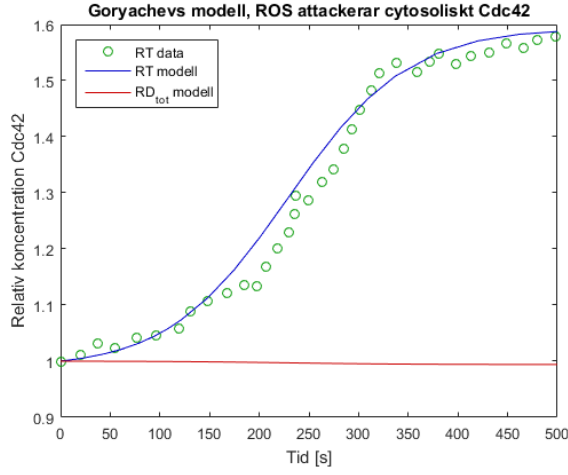
$$\begin{aligned} \frac{d(RT)}{dt} &= E_c RT \cdot RD_{tot} (A_1 RT + A_2) - A_3 RT \\ \frac{d(RD_{tot})}{dt} &= A_3 RT - E_c RT \cdot RD_{tot} (A_1 RT + A_2) - A_4 RD_{tot} \end{aligned} \quad (11)$$

$$E_c = \frac{1}{1 + A_5 RT^2 + A_6 RT + A_7}$$

där RT är koncentrationen membranbundet aktivt Cdc42, E_c är den cytosoliska koncentrationen av Cdc24-Bem1-komplexet, RD_{tot} är den totala koncentrationen inaktivt Cdc42 och A_{1-7} är de olika parametrarna. Den benämning av olika specier som används är densamma som Goryachev använder i sin artikel. För fullständig härledning, se Bilaga A. I detta ODE-system är det termen $A_4 RD_{tot}$ som beskriver hur cytosoliskt inaktivt Cdc42 reagerar med ROS där k_{ROS} ingår i parametern A_4 . Likt modellen baserad på litteraturstudier tas hänsyn till att RT och RD_{tot} tillhör membran- respektive cytosolvolymer. Skillnaden mellan ODE-system (11) och det system som används i Copasi blir att alla termer i ekvationen för RT , förutom $A_3 RT$, divideras med cytosolvolymer och alla termer utom $A_4 RD_{tot}$ i ekvationen för RD_{tot} divideras med membranvolymer. Det korrigerade systemet används sedan för parameteruppskattning i Copasi. Resultatet från uppskattningen illustreras i Figur 9 och värdena på parametrarna samt det beräknade minstakvadratavståndet i Tabell 3. De olika graferna motsvarar samma sak som i Figur 8, fast $Cdc42_m$ är omnämnt till RT och $Cdc42_c$ till RD_{tot} . Även här saknas enheter på parametrarna i Tabell 3, av samma anledning som nämndes i avsnitt 4.2.1. Dessutom består parametrarna egentligen av än flera parametrar multiplicerade och dividerade med varandra, vilket syns tydligt i Bilaga A, vilket leder till icke-relevanta enheter. Mängden oxidativt skadat Cdc42 uppskattas också med hjälp av denna modell men antar ett så lågt värde att de exkluderas ur Figur 9. De högsta värdet för oxidativt skadat Cdc42 ligger i storleksordningen 10^{-8} .

Tabell 2: Värden för de uppskattade parametrarna i Copasi från ekvation (10) samt det beräknade minstakvadrat-avståndet

Parameter	Värde
k_c	9.11296e-20
k_m	3.22335e-30
k_{kat}	4.43939e-19
k_{ROS}	3.90555e-22
Minstakvadrat-avstånd	0,1271



Figur 9: Resultatet av parameteruppskattningen i Copasi av system (11)

Tabell 3: Värden för de uppskattade parametrarna i Copasi från system (11) samt det beräknade minstakvadrat-avståndet

Parameter	Värde
A_1	5,65107e-17
A_2	1,04089e-19
A_3	3,49432e-17
A_4	3,91429e-11
A_5	0,616288
A_6	1,23528e-14
A_7	1,03082e-5
Minstakvadrat-avstånd	0,0419

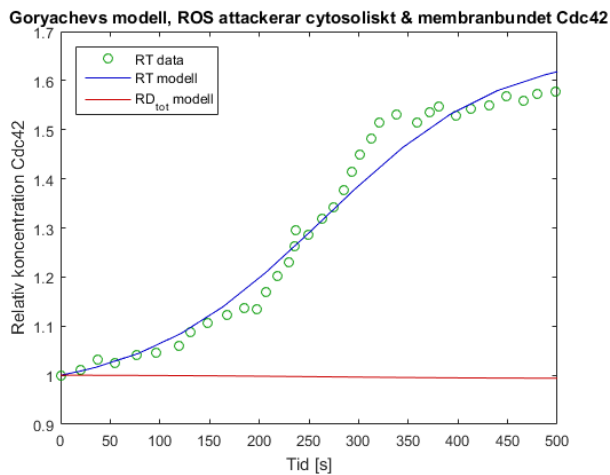
En undersökning av de olika parametrarnas betydelse på ODE-system (11) gjordes genom att systematiskt ändra parametrarnas värde, en i taget, i MATLAB. Efter varje förändring av värde löstes ODE-systemet och den eventuella förändringen noterades. Slutsatsen av undersökningen var att de termer i ODE-system (11) innehållandes RT^2 är de som ger upphov till den sigmoidala formen på grafen för RT beräknad från modellen. Samma undersökning gjordes även för ODE-system (10) och (12).

4.2.3 Goryachev - ROS attackerar cytosoliskt och membranbundet Cdc42

Modellen där ROS reagerar med både cytosoliskt och membranbundet Cdc42, ger liknande resultat som ODE-system (11). Likt system (11) består det initiala ODE-systemet av åtta differentialekvationer där de extra termerna för hur ROS reagerar med cytosoliskt och membranbundet Cdc42 lagts till. På samma sätt reduceras systemet och kvar fås

$$\begin{aligned}
 \frac{d(RT)}{dt} &= E_c RT \cdot RD_{tot} (B_1 RT + B_2) - B_3 RT - RT (B_4 + E_c (B_5 RT + B_6)) \\
 \frac{d(RD_{tot})}{dt} &= B_3 RT - E_c RT \cdot RD_{tot} (B_1 RT + B_2) - B_7 RD_{tot} \\
 E_c &= \frac{1}{1 + B_8 RT^2 + B_9 RT + B_{10}}
 \end{aligned} \tag{12}$$

där alla variabler är desamma som i system (11) men parametrarna B_{1-10} ser olika ut. De största skillnaderna är antalet termer i systemet samt att k_{ROS} ingår i alla utom en av parametrarna, se Bilaga A. Även här definieras membran- och cytosolvolymer som olika avdelningar i Copasi och likt ODE-system (11) divideras de flesta av termerna i ekvationerna med cytosol- respektive membranvolymer. De termer som inte divideras med någon volym är $RT (B_4 + E_c (B_5 RT + B_6))$ respektive $B_7 RD_{tot}$. Resultatet från parameteruppskattningen illustreras i Figur 10 och parametrarna samt minstakvadrat-avståndet presenteras i Tabell 4. Än en gång har parametrarnas enheter exkluderats från av samma anledning som nämns i avsnitt 4.2.2. Figur 10 och 9 är väldigt lika varandra men inte helt identiska. Andelen oxidativt skadat Cdc42 i denna modell ligger runt 10^{-8} , samma värde som ODE-system (11) förutspår.



Figur 10: Resultatet av parameteruppskattningen i Copasi av ODE-system (12)

Tabell 4: Värden för de uppskattade parametrarna från system (12) i Copasi samt det beräknade minstakvadrat-avståndet

Parameter	Värde
B_1	3.25753e-17
B_2	6.67347e-22
B_3	2.00745e-17
B_4	1.46589e-25
B_5	2.08033e-8
B_6	1.7045e-25
B_7	8.02671e-21
B_8	0.607535
B_9	5.85904e-18
B_{10}	3.3561e-16
Minstakvadrat-avstånd	0,0448

5 Diskussion

ODE-system (11) och (12) visar att ROS har en försumbar påverkan på Cdc42 under polariseringsfasen. Det är ett förväntat resultat. Eukaryota organismer har förmodligen levt med ROS ända sedan mitokondrien blev en del av cellen. De bör därför ha anpassat sig så att normala nivåer av ROS inte påverkar cellens maskineri nämnvärt. Speciellt inte under så korta tidsperioder som en polarisering, vilken endast pågår under cirka åtta minuter. I studien varifrån datan är hämtad har cellerna inte heller utsatts för förhöjda nivåer av ROS. Med det som bakgrund är det inte överraskande att påverkan av ROS visat sig vara mycket liten. Över ett längre tidsintervall kan det däremot vara intressant att undersöka hur ROS påverkar mängden Cdc42. Oxidativ nedbrytning av Cdc42 må vara en långsam process, men den kan samtidigt vara irreversibel. Om modellerna skulle behandla hela jästcellens livstid kanske andra slutsatser dragits. För att simulera påverkan av ROS under en jästcells liv behöver varje celldelning tas i beaktning vilket kräver en utökning av de modeller som används i detta arbete.

Goryachevs modell från 2008 gav en bra grund till de modeller som presenterats i detta arbete. Värt att nämna är dock att alla forskare inom ämnet inte är överens om vilken modell som bäst återspeglar polarisering som fenomen [55, 42]. Undersökningar kring redan befintliga modeller gjordes tidigt i arbetet och Goryachevs modell var den som ansågs vara bäst lämpad för detta arbete. Om en annan redan existerande modell skulle använts istället för Goryachevs hade kanske arbetet och dess resultat sett väldigt annorlunda ut. Att rimliga resultat erhållits från ODE-system (11) och (12) tyder på att Goryachevs typ av modell fungerar. Båda modeller förutsäger väldigt låga mängder oxidativt skadat Cdc42 (10^{-8}) vilket vilket är precis vad som förväntades. En jämförelse mellan de beräknade minstakvadrat-avstånden, vars värde finns i Tabell 2, 3 respektive 4, visar att ODE-system (11) är den modell som passar data bäst. Det relativt höga värdet på avståndet för system (10) tyder på att modellen inte kan simulera koncentrationsförändringen av Cdc42 under polarisering. Anledningen till att system (10) inte är kapabel att passa data lika bra som (11) och (12) är på grund av avsaknaden av kvadratisk term, vilket beskrevs i avsnitt 4.2.2. Som undersökningen av de olika parametrarnas påverkan på modellerna visade, är det den kvadratiske termen som ger upphov till den sigmoidala formen. Då ODE-system (10) inte har någon sådan term kan inte heller modellen anpassas till data.

En del av de antaganden som gjorts skulle behövt mer underlag från litteraturen. Ett exempel rör införandet av ROS-terminen. Eventuellt har införandet av ROS gjorts i en för förenklad form; kanske är nivån av ROS i cellen mer dynamisk under polariseringen än vad som antagits, eller så sker kanske interaktionen mellan Cdc42 och ROS över flera steg. Samtidigt är flera av antagandena nödvändiga för att modellen inte ska bli för komplicerad. För vissa

av de antaganden som gjorts finns inte heller någon information att hitta i litteraturen, då det helt enkelt inte är någon som har studerat detta ännu.

Ytterligare testning och validering av modellerna i arbetet behövs. Att bara få modellen att passa data är inte ett nog starkt bevis för att dra slutsatsen att den kan beskriva rollen hos Cdc42 under polarisering. Till exempel är det svårt att avgöra om ekvationssystem (12) verkligen är en bra modell. Antingen passar den data för att det är en bra modell eller så kan det lika gärna bero på att det relativt stora antalet parametrar ger modellen förmågan att passa vilken data som helst. För att verkligen validera en modell krävs att den testas med data från flera olika situationer och att den då konsekvent visar sig beskriva verkligheten. Denna möjlighet har inte funnits i detta arbete eftersom sådan data saknas, och detta är något som kan förbättras i framtiden.

Mycket av den parameteruppskattning som låg till grund för slutsatserna om det biologiska systemet utfördes i Copasi. Det finns vissa potentiella felkällor i detta som bör tas i beaktning. Till exempel har Copasi svårigheter med att ta hänsyn till enhetslös data. Problemet har lösts på ett sätt som tros vara korrekt men det är långt ifrån säkert. Ett annat av de större problemen är att det är svårt att veta hur programmet arbetar och felsökningar är överlag krångliga att åstadkomma. Därför är fortsatt förbättring av det egenskrivna MATLAB-programmet en angelägen uppgift i detta arbete.

Numeriska problem orsakade många av parameteruppskattningsprogrammets divergeringar. I stort sett varje iterationsloop visade ett felmeddelande om dåligt skalade matriser $\det(\overline{\mathbf{H}}) \approx 0$. Dessa problem med skalningen förstärktes ju närmre en bra anpassning till datan programmet kom. Om brytvillkoret i koden sattes för lågt (det vill säga en bra anpassning till data krävdes) lyckades programmet inte nå en bra anpassning efter att ha först kommit nära data. Ofta jobbade programmet sig först nära en bra anpassning innan problemen med $\det(\overline{\mathbf{H}})$ blev för stora, varpå det tog ett stort hopp bort från en bra anpassning och sedan divergerade helt. Detta är bland annat anledningen till den halvbra anpassningen till data i Figur 6a. Lägre toleransnivå för anpassningen kunde inte sättas utan divergering som följd. Det är ett intressant beteende som kanske skulle kunna ge ledtrådar till vad som behöver ändras i programmet eller modellerna, och skulle därför kunna studeras i mer detalj. Det skulle även kunna förklara varför den datan som användes vid parameteruppskattningen i Copas, inte fungerade. Där ökade andelen membranbunden Cdc42 mindre i y -led över tidsintervallet om man jämför med FRAP-datan [46]. Detta krävde därför en lägre toleransnivå för brytvillkoret.

Den kanske absolut viktigaste frågan att svara på för fortsatt utveckling av programmet är vad som orsakar de numeriska problemen. Hypotesen i nuläget är att de flesta av dem orsakas av problem med identifierbarheten i modellerna. Detta på grund av att chansen att en parameter ska vara identifierbar är som störst då endast en parameter i en modell uppskattas åt gången. Samtidigt kördes programmet mycket bättre när endast en parameter uppskattades, se Figur 5a och jämför med Figur 6a.

En strategi behövs för att pröva identifierbarheten hos parametrarna i modellerna. Detta hade gjort resultaten mer tillförlitliga. Ett försök att utreda parametrarnas strukturella identifierbarhet i STRIKE-GOLDD gjordes men för två av de tre modellerna lyckades programmet inte köra klart och resultat uteblev. För den modell där STRIKE-GOLDD lyckades köras klart var en av de fyra parametrarna inte strukturellt identifierbar. Även om en modell är strukturellt identifierbar behöver den inte vara praktiskt identifierbar. Troligen är inte någon av modellerna praktiskt identifierbar vid uppskattning av fler än en variabel med aktuell data. Det fanns dock ingen tid att undersöka identifierbarheten hos parametrar ytterligare. Det rekommenderas varmt att undersöka detta och det är ett av de viktigaste områdena att kolla närmare på för att utveckla detta arbete vidare. Trots problemen med identifierbarhet verkar Copasi ändå kunna finna rimliga parametervärden. Detta tyder på att Copasi har mekanismer för att kringgå icke-identifierbarhet. Identifierbarhetsproblem i biologiska modeller är vanligt [45]. Det finns metoder för att kringgå problematiken och ändå få ut parametrar med åtminstone viss trovärdighet där det dessutom är kvantifierat hur troliga de är genom konfidensintervall [45]. En sådan är Profile Likelihood-metoden (PLH) [58]. PLH eller andra metoder med samma mål har inte applicerats i detta arbete, men det vore mycket intressant att se vilken effekt de skulle kunna ha. Implementering av PLH blir därför ett av de mest angelägna områdena för framtida utveckling. Generellt är identifierbarhetsanalys [58] ett

område som bör appliceras mer framöver.

Även om vissa försök gjorts är det mycket möjligt att ett mer sofistikerat sätt att välja steglängd då parametrarna uppdateras hade förbättrat programmets prestanda. Sättet som steglängden bestäms på i detta arbete är ett vanligt förekommande sätt, se teoriavsnittet 2.2.2, men innehåller ingen analys av olika punkter i parameterlandskapet i stegriktningen. Genom att utvärdera kostfunktionen på flera ställen i samma riktning och sedan ta det mest fördelaktiga steget skulle möjligtvis effektiviteten och robustheten hos programmet förbättras. Ett primitivt sätt att göra detta testades. Då jämfördes kostfunktionen innan och efter ett steg och ifall dess värde ökade testades en hälften så lång steglängd. Problemet var att programmet snabbt stötte på uppförsbackar i parameterlandskapet där det fastnade genom att steglängden halverades igen och igen tills den blev ≈ 0 . Därför visade det sig mer framgångsrikt att inte ha en sådan funktion i programmet. Uppförsbackarna kan dock ha orsakats av att stegriktningen var felaktig på grund av de numeriska problemen. Om dessa problem löstes hade steglängds-ansatsen kanske fungerat. Oavsett rekommenderas en mer sofistikerad ansats för varierande steglängd för att undvika risken att köra fast på kullar i parameterlandskapet.

Det hade även varit intressant att testa andra optimerings-algoritmer än quasi-Newton för att se ifall de producerar bättre resultat. Ett exempel på alternativ är så kallade genetiska algoritmer som har fått sitt namn för att de hämtar inspiration från biologisk evolution. De använder sig av koncept såsom selektion, mutation och korsning för att välja bort parametrar som ger dålig anpassning till data och hitta de som ger bra anpassning[59]. Genetiska algoritmer skulle eventuellt lyckas bättre än Newton-algoritmen eftersom de inte använder sig av $\det(\mathbf{H})$ vilken har varit inblandad i de numeriska problemen i detta arbete. Dessutom är alla varianter av Newton-algoritmen för optimering kända för att vara typiskt lokala metoder. Då är risken stor att optimeringen fastnar i ett lokalt minima. Genetiska algoritmer löper mindre risk för detta. En kombination av de två, där genetiska algoritmer används för att göra en första avsökning av parameterlandskapet innan quasi-Newton används för att förbättra lösningarna, skulle också kunna visa sig framgångsrik.

Slutsats

Slutsatsen från arbetet är att ROS påverkan på Cdc42 under polariseringsfasen i *S. Cerevisiae* är försumbar, givet icke-förhöjda nivåer av ROS. Dock bör beaktas att modellen endast hanterar en polarisering i sin nuvarande form, och beskriver därmed inte påverkan över ett längre tidsintervall. För att undersöka ROS inverkan på Cdc42 i jästceller över längre tid vore det intressant att i framtiden använda en modell som kan ta hänsyn till hela cellens livlängd med dess tillväxtfaser och celldelningar.

De två ODE-systemen baserade på Goryachevs modell från 2008 visar rimliga resultat, vilket tyder på att de har förmågan att simulera koncentrationsfördelningen av Cdc42 under polarisering. Däremot bör de antaganden som gjorts, samt identifierbarheten av parametrarna, undersökas ytterligare. En grundligt gjord validering av modellerna bör även utföras för att göra modellerna mer tillförlitliga.

Vad gäller programmet i MATLAB är det på god väg. Det behöver dock utvecklas för att bli mer robust och klara av parameteruppskattning i fler situationer. Första prioritet är tillägg av metoder för att hantera situationer då parametrar i modellen inte är praktiskt identifierbara, såsom PLH. Vidare vore det intressant att testa alternativa sätt att minimera kostfunktionen, såsom genetiska algoritmer. Det skulle kunna lösa de tidigare diskuterade numeriska problemen. Dock skulle det inte lösa underliggande identifierbarhetsproblem och mer identifierbarhetsanalys behöver appliceras framöver. Det hade gett en bättre bild av hur mycket av svårigheterna med parameteruppskattningen som beror av programmet och hur mycket som har att göra med de underliggande modellerna.

6 Referenser

- [1] Okänd, Woman old senior female elderly retired, 2015 [Elektronisk bild] Tillgänglig från: <https://pixabay.com/en/woman-old-senior-female-elderly-1031000/>, licens: <https://creativecommons.org/publicdomain/zero/1.0/legalcode.sv> Hämtad: 2017-05-10.
- [2] Hellahulla, Compressed fresh yeast - 1, 2007 [Elektronisk bild] Tillgänglig från: https://sv.wikipedia.org/wiki/Fil:Compressed_fresh_yeast_-_1.jpg, Licens: <https://creativecommons.org/licenses/by-sa/4.0/legalcode.sv> Hämtad: 2017-05-10.
- [3] Stephen Bondy och Edward H Sharman. Melatonin, Oxidative Stress, and the Aging Brain. I: Maiese K. Bondy, S., redaktör, *Aging and Age-Related Disorders*, kapitel 17, ss 341–342. Humana Press, New York, 2010.
- [4] Robert L. Nussbaum och Christopher E. Ellis. Alzheimer’s disease and Parkinson’s disease. *The New England Journal of Medicine*, 348(14):1356–1364, 2003.
- [5] Annina Denoth Lippuner, Thomas Julou och Yves Barral. Budding yeast as a model organism to study the effects of age. *FEMS Microbiology Reviews*, 38:300–325, 2014.
- [6] Ira Herskowitz. Life Cycle of the Budding Yeast *Saccharomyces cerevisiae*. *Microbiological Reviews*, 52(4):536–553, 1988.
- [7] Andrew B. Goryachev och Marcin Leda. Many roads to symmetry breaking: molecular mechanisms and theoretical models of yeast cell polarity. *Molecular Biology of the Cell*, 28(3):370–380, 2017.
- [8] Sophie Cotteret och Jonathan Chernoff. The evolutionary history of effectors downstream of Cdc42 and Rac. *Genome Biology*, 3(2):1–8, 2002.
- [9] Roman L Tatusov, Michael Y Galperin, Darren A Natale och Eugene V Koonin. The COG database: a tool for genome-scale analysis of protein functions and evolution. *Nucleic Acids Research*, 28(1):33–36, 2000.
- [10] Hartmut Geiger, Anja Koehler och Matthias Gunzer. Stem cells, aging, niche, adhesion and Cdc42: A model for changes in cell-cell interactions and hematopoietic stem cell aging. *Cell Cycle*, 6(8):884–887, 2007.
- [11] Hartmut Geiger, Yi Zheng och Correspondence To. Cdc42 and aging of hematopoietic stem cells. *Curr Opin Hematol*, 20(4):295–300, 2013.
- [12] S Hekimi, J Lapointe och Y Wen. Taking a “good” look at free radicals in the aging process. *Trends in cell biology*, 2011.
- [13] D. Harman. Origin and evolution of the free radical theory of aging: A brief personal history, 1954-2009. *Biogerontology*, 10(6), 2009.
- [14] Kenneth Hensley och Robert A Floyd. MINIREVIEW Reactive Oxygen Species and Protein Oxidation in Aging : A Look Back , A Look Ahead. *Archives of Biochemistry and Biophysics*, 397(2):377–383, 2002.
- [15] Robert S. Balaban, Shino Nemoto och Toren Finkel. Mitochondria, oxidants, and aging. *Cell*, 120(4):483–495, 2005.
- [16] Alex Mogilner, Roy Wollman och Wallace F Marshall. Review Quantitative Modeling in Cell Biology: What Is It Good for? *Developmental Cell*, 11:279–287, 2006.
- [17] Hiroaki Kitano. Systems Biology: A Brief Overview. *Science*, 295(5560):1662–1664, 2002.
- [18] Jay L Devore. *Probability and Statistics for Engineering and the Sciences*. Cengage Learning, 8 utgåvan, 2015.

- [19] Bruce Alberts, Alexander Johnsson och Julian Lewis. *Molecular Biology of the Cell*. Garland Publishing Inc, 6 utgåvan, 2014.
- [20] David Botstein, Steven A Chervitz och J Michael Cherry. Yeast as a Model Organism. *Source: Science, New Series*, 277(5330):1259–1260, 1997.
- [21] Stanley Fields och Mark Johnston. Whither Model Organism Research? *Science*, 307(5717):1885–1886, 2005.
- [22] S. Blair Hedges. The origin and evolution of model organisms. *Nature Reviews Genetics*, 3(11):838–849, 2002.
- [23] Mitsuhiro Yanagida. The model unicellular eukaryote, *Schizosaccharomyces pombe*. *Genome Biology*, 3(34):2003–1, 2002.
- [24] Maan Hayyan, Mohd Ali Hashim och Inas M. Alnashef. Superoxide Ion: Generation and Chemical Implications. *Chemical Reviews*, 116(5):3029–3085, 2016.
- [25] Michael P Murphy. How mitochondria produce reactive oxygen species. *Biochem. J*, 417:1–13, 2009.
- [26] Elisa Cabiscol, Jordi Tamarit och Joaquim Ros. Oxidative stress in bacteria and protein damage by reactive oxygen species. *International Microbiology*, 3(1):3–8, 2000.
- [27] Ariel B. Lindner och Alice Demarez. Protein aggregation as a paradigm of aging, 2009.
- [28] J Lee, N Koo och D B Min. Reactive oxygen species, aging, and antioxidative nutraceuticals. *Comprehensive Reviews in Food Science and Food Safety*, 3(1):21–33, 2004.
- [29] Steven Neill, Radhika Desikan och John Hancock. Hydrogen peroxide signalling. *Current Opinion in Plant Biology*, 5(5):388–395, 2002.
- [30] Enrique Cadenas och Kelvin J A Davies. Mitochondrial Free Radical Generation, Oxidative Stress, and Aging. *Free Radical Biology & Medicine*, 29(3/4):222–230, 2000.
- [31] Gerd P Bienert, Jan K Schjoerring och Thomas P Jahn. Membrane transport of hydrogen peroxide. *Biochimica et Biophysica Acta - Biomembranes*, 1758(8):994–1003, 2006.
- [32] Ian M. Møller och Lee J. Sweetlove. ROS signalling - specificity is required. *Trends in Plant Science*, 15(7):370–374, 2010.
- [33] Mikko O Laukkanen, Francesca Cammarota, Tiziana Esposito, Marco Salvatore och Maria D Castellone. Extracellular Superoxide Dismutase Regulates the Expression of Small GTPase Regulatory Proteins GEFs, GAPs, and GDI. *PLoS one*, 10(3):e0121441, 2015.
- [34] Katharine Brown, Stephanie Xie, Xiaolei Qiu, Mary Mohrin, Jiyung Shin, Yufei Liu, Dan Zhang, David T. Scadden och Danica Chen. SIRT3 Reverses Aging-Associated Degeneration. *Cell Reports*, 3(2):319–327, 2013.
- [35] Sandrine Ragu, Michéle Dardalhon, Sushma Sharma, Ismail Iraqui, Géraldine Buhagiar-Labarchède, Virginie Grondin, Guy Kienda, Laurence Vernis, Roland Chanet, Richard D. Kolodner, Meng Er Huang och Gérard Faye. Loss of the thioredoxin reductase Trr1 suppresses the genomic instability of peroxiredoxin TSA1 mutants. *PLoS ONE*, 9(9):1–15, 2014.
- [36] Lewis Elwood Tomalin, Alison Michelle Day, Zoe Elizabeth Underwood, Graham Robert Smith, Piero Dalle Pezze, Charalampos Rallis, Waseema Patel, Bryan Craig Dickinson, Jürg Bähler, Thomas Francis Brewer, Christopher Joh Leung Chang, Daryl Pierson Shanley och Elizabeth Ann Veal. Increasing extracellular H₂O₂ produces a bi-phasic response in intracellular H₂O₂, with peroxiredoxin hyperoxidation only triggered once the cellular H₂O₂-buffering capacity is overwhelmed. *Free Radical Biology and Medicine*, 95:333–348, 2016.

- [37] Andrew B. Goryachev och Alexandra V. Pokhilko. Dynamics of Cdc42 network embodies a Turing-type mechanism of yeast cell polarity. *FEBS Letters*, 582:1437–1443, 2008.
- [38] Franz Meitinger, Anton Khmelinskii, Sandrine Morlot, Bahtiyar Kurtulmus, Saravanan Palani, Amparo Andres-Pons, Birgit Hub, Michael Knop, Gilles Charvin och Gislene Pereira. A memory system of negative polarity cues prevents replicative aging. *Cell*, 159:1056–1069, 2014.
- [39] Hay-Oak Park och Erfei Bi. Central Roles of Small GTPases in the Development of Cell Polarity in Yeast and Beyond. *Microbiology and Molecular Biology Reviews*, 71(1):48–96, 2007.
- [40] Aron B Jaffe och Alan Hall. Rho GTPases: Biochemistry and Biology. *Annu. Rev. Cell Dev. Biol.*, 21:247–69, 2005.
- [41] Javier E Irazoqui, Amy S Gladfelter och Daniel J Lew. Scaffold-mediated symmetry breaking by Cdc42p. *Nature Cell Biology*, 5(12):1062–1070, 2003.
- [42] Alex Mogilner, Jun Allard och Roy Wollman. Cell Polarity: Quantitative Modeling as a Tool in Cell Biology. *Science*, 336(6078):175–179, 2012.
- [43] Mika Järvinen, Ville-Valtteri Visuri, Eetu-Pekka Heikkinen, Aki Kärnä, Petri Sulasalmi, Cataldo De Blasio och Timo Fabritius. Law of Mass Action Based Kinetic Approach for the Modelling of Parallell Mass Transfer Limited Reactions: Application to Metallurgical Systems. *ISIJ International*, 56(9):1543–1552, 2016.
- [44] Lawrence F Shampine och Mark W Reichelt. The matlab ode suite. *SIAM journal on scientific computing*, 18(1):1–22, 1997.
- [45] Gunnar Cedersund. *Prediction uncertainty estimation despite unidentifiability: an overview of recent developments*. Springer International, 2016.
- [46] Felipe O. Bendezú, Vincent Vincenzetti, Dimitrios Vavylonis, Romain Wyss, Horst Vogel och Sophie G. Martin. Spontaneous Cdc42 Polarization Independent of GDI-Mediated Extraction and Actin-Based Trafficking. *PLoS Biology*, 13(4), 2015.
- [47] Roland Wedlich-Soldner, Stephanie C. Wai, Thomas Schmidt och Rong Li. Robust cell polarity is a dynamic state established by coupling transport and GTPase signaling. *Journal of Cell Biology*, 2004.
- [48] Michael Patriksson, Niclas Andréasson, Anton Evgrafov, Emil Gustavsson och Magnus Önnheim. *Introduction to Continuous Optimization*. Studentlitteratur, 2013.
- [49] B. Sengupta, K. J. Friston och W. D. Penny. Efficient gradient computation for dynamical models. *NeuroImage*, 2014.
- [50] Alejandro F. Villaverde, Antonio Barreiro och Antonis Papachristodoulou. Structural Identifiability of Dynamic Systems Biology Models. *PLoS Computational Biology*, 2016.
- [51] Charles A Day, Lewis J Kraft, Minchul Kang och Anne K Kenworthy. Analysis of protein and lipid dynamics using confocal fluorescence recovery after photobleaching (FRAP). *Current Protocols in Cytometry*, CHAPTER: Unit 2.19, 2012.
- [52] J E Conour, W V Graham och H R Gaskins. A combined in vitro/bioinformatic investigation of redox regulatory mechanisms governing cell cycle progression. *Physiological Genomics*, 18(2):196–205, 2004.
- [53] Fan-Guo Meng och Zhong-Yin Zhang. Redox Regulation of Protein Tyrosine Phosphatase Activity by Hydroxyl Radical. *Biochim Biophys Acta*, 1834(1):464–469, 2013.
- [54] Benoît D’Autréaux och Michel B. Toledano. ROS as signalling molecules: mechanisms that generate specificity in ROS homeostasis. *Nature reviews Molecular cell biology*, 8(10):813–824, 2007.

- [55] Tina Freisinger, Ben Klünder, Jared Johnson, Nikola Müller, Garwin Pichler, Gisela Beck, Michael Costanzo, Charles Boone, Richard A Cerione, Erwin Frey och Roland Wedlich-Söldner. Establishment of a robust single axis of cell polarity by coupling multiple positive feedback loops. *Nature Communications*, 4, 2013.
- [56] Alexandra Jilkine, A. F M Marée och Leah Edelstein-Keshet. Mathematical model for spatial segregation of the Rho-family GTPases based on inhibitory crosstalk. *Bulletin of Mathematical Biology*, 2007.
- [57] Andrea K Bryan, Alexi Goranov, Angelika Amon, Scott R Manalis och Marc W Kirschner. Measurement of mass, density, and volume during the cell cycle of yeast. *Proceedings of the National Academy of Sciences USA*, 107(3):999–1004, 2010.
- [58] Andreas Raue, C. Kreutz, T. Maiwald, J. Bachmann, M. Schilling, U. Klingmuller och J. Timmer. Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. *Bioinformatics*, 25(15):1923–1929, 2009.
- [59] L. J. Park, C. H. Park, C. Park och T. Lee. Application of genetic algorithms to parameter estimation of bioprocesses. *Medical and biological engineering and computing*, 35(1):47–49, 1997.

A Härledning av Goryachevs modell med inkorporering av ROS

Denna bilaga behandlar i detalj härledningarna av Goryachevs modell samt hur effekten av ROS inkorporeras i modellen [37]. Alla antaganden som nämns i härledningen är enligt Goryachevs artikel om inget annat nämns. De termer som kommer att användas i härledningen introduceras i Tabell 5 nedan.

I denna bilaga kommer två modeller härledas: En där ROS endast reagerar med cytosoliskt Cdc42 och en där ROS reagerar med både cytosoliskt och membranbundet Cdc42. Då de olika stegen i härledningarna är snarlika varandra kommer antaganden kring varje steg i huvudsak presenteras i första härledningen. I andra härledningen presenteras främst det som skiljer de båda åt.

Tabell 5: Förkortningar av termer som används i härledningen samt en kort beskrivning av varje term

Förkortning	Beskrivning
RT	Aktivt membranbundet Cdc42
M	Aktivt membranbundet Cdc42 komplexbundet med Cdc24 · Bem1
E_m	Membranbundet Cdc24 · Bem1
E_c	Cytosoliskt Cdc24 · Bem1
RD	Inaktivt membranbundet Cdc42
RDI_m	Inaktivt membranbundet Cdc42 komplexbundet med GDI
RDI_c	Inaktivt cytosoliskt Cdc42 komplexbundet med GDI
I	Cytosoliskt GDI

Den första modellen behandlar fallet där ROS reagerar med cytosoliskt Cdc42. ODE-systemet som härledningen utgår från är väldigt likt Goryachevs. De enda skillnaderna är att diffusionstermerna ansätts lika med noll då detta system endast beror av tid, samt att i den ekvation som berör cytosoliskt Cdc42 (RDI_c) finns det en extra term som beskriver hur ROS påverkar koncentrationen RDI_c . Det initiala ODE-systemet är därmed

$$\frac{d(RT)}{dt} = (k_2 E_m + k_3 M)RD - k_{-2}RT - k_4 E_m RT + k_{-4}M - k_7 E_c RT \quad (13)$$

$$\frac{d(M)}{dt} = k_4 E_m RT - k_{-4}M + k_7 E_c RT \quad (14)$$

$$\frac{d(E_m)}{dt} = k_1 E_c - k_{-1}E_m - k_4 E_m RT + k_{-4}M \quad (15)$$

$$\frac{d(E_c)}{dt} = \eta(k_{-1}E_m - (k_1 + k_7 RT)E_c) \quad (16)$$

$$\frac{d(RD)}{dt} = k_{-2}RT - (k_2 E_m + k_3 M)RD + k_{-6}RDI_m - k_6 I \cdot RD \quad (17)$$

$$\frac{d(RDI_m)}{dt} = k_6 I \cdot RD - k_{-6}RDI_m + k_5 RDI_c - k_{-5}RDI_m \quad (18)$$

$$\frac{d(RDI_c)}{dt} = \eta(k_{-5}RDI_m - k_5 RDI_c - k_{ROS} \cdot RDI_c) \quad (19)$$

$$\frac{d(I)}{dt} = \eta(k_{-6}RDI_m - k_6 I \cdot RD) \quad (20)$$

där k_i är de olika hastighetskonstanterna och η är en skalfaktor för att ta hänsyn till att den cytosoliska volymen är större än membranvolymen. Lagg märke till att ROS antas reagera med hela Cdc42·GDI-komplexet. Alla hastighetskonstanter bestäms entydigt av dess värde k_i förutom k_{-2} som tar hänsyn till att reaktionen är beroende av enzymen GAP enligt

$$k_{-2} = k_{-2.0} + k_{-2.1}A_0$$

där A_0 är koncentrationen GAP. E_c kan även beskrivas med hjälp av lagen av massans bevarande enligt

$$E_c = E_c^0 - \int_S (M + E_m) dS \quad (21)$$

där E_c^0 är initialkoncentrationen av E_c och S är membranytan. För att genomföra härledningen krävs ett antal antaganden. Det första är att reaktionstermen k_2E_m är försumbart liten i jämförelse med k_3M enligt Goryachevs artikel, samt att koncentrationen ROS, som är en del av hastighetskonstanten k_{ROS} , är konstant under hela tidsperioden, se delavsnitt 4.2.

För det andra antas ett flertal ekvationer ske under stationärt tillstånd, det vill säga att ackumulationstermen är lika med noll. Detta antagande gäller för ekvation (14), (15), (16), (18) och (20). Vad gäller I antas totala koncentrationen I vara konstant och kan därmed betecknas $I = I_0$. Det sista antagandet är att det sker en snabb jämvikt mellan RD och RDI_c . Det betyder att RDI_m kan beskrivas med hjälp av både RD och RDI_c enligt

$$K_1 = \frac{k_6}{k_{-6}} = \frac{RDI_m}{I_0RD} \quad (22)$$

$$K_2 = \frac{k_5}{k_{-5}} = \frac{RDI_m}{RDI_c} \quad (23)$$

där K_i är jämviktskonstanten för respektive reaktion. Om RDI_m löses ut ur både ekvation (22) och (23) fås sambandet

$$\frac{k_6}{k_{-6}} I_0RD = \frac{k_5}{k_{-5}} RDI_c \quad (24)$$

ett samband som kommer att utnyttjas senare i härledningen. Målet är att reducera ODE-systemet från att bero av åtta till att bara bero av två ekvationer. De variabler som det slutgiltiga systemet kommer att bestå av är RT och RD_{tot} , där RD_{tot} definieras som

$$RD_{tot} = RD + RDI_c \quad (25)$$

Det betyder att alla andra specier behöver beskrivas i termer av de två ovannämnda variablerna. Notera att RDI_m inte ingår i definitionen av RD_{tot} . Detta på grund av att koncentrationen RDI_m är försumbar i detta sammanhang.

Antagandet att ackumulationstermen i ekvation (14) och (16) är lika med noll utnyttjas och M respektive E_m kan skrivas som

$$M = \frac{1}{k_{-4}} (k_4E_mRT + k_7E_cRT) \quad (26)$$

$$E_m = E_c \left(\frac{k_1}{k_{-1}} + \frac{k_7}{k_{-1}} RT \right) \quad (27)$$

och om ekvation (27) kombineras med (26) fås

$$M = E_cRT \left(\frac{k_1k_4}{k_{-4}k_{-1}} + \frac{k_4k_7}{k_{-4}k_{-1}} RT + \frac{k_7}{k_{-4}} \right) \quad (28)$$

Med hjälp av ovannämnda omskrivningar kan $\frac{d(RT)}{dt}$ formuleras om. Först utnyttjas att $k_{-4}M = k_4E_mRT + k_7E_cRT$ enligt ekvation (14) så att flertalet termer i ekvation (13) tar ut varandra. Tillsammans med antagandet att $k_2E_m \ll k_3M$ kan ekvation (13) skrivas som

$$\frac{d(RT)}{dt} = k_3M \cdot RD - k_{-2}RT \quad (29)$$

där resultatet från ekvation (28) kan ansättas. För att skapa materialbalansen som beskriver koncentrationen RD_{tot} adderas ekvation (17) och (19) enligt

$$\begin{aligned}\frac{d(RD_{tot})}{dt} &= \frac{d(RD)}{dt} + \frac{d(RDI_c)}{dt} = \\ &= k_{-2}RT - (k_2E_m + k_3M)RD + k_{-6}RDI_m - k_6I_0RD + \eta \cdot k_{ROS} \cdot RDI_c\end{aligned}\quad (30)$$

Antalet termer i ekvation (30) reduceras med hjälp av att ackumulationstermen är lika med noll i ekvationerna (20) och (18). Detta ger upphov till sambanden

$$k_{-6}RDI_m - k_6I_0RD = 0 \quad (31)$$

$$k_5RDI_c = k_{-5}RDI_m \quad (32)$$

och ekvation (30) kan, med hjälp av (31) och (32), samt att $k_2E_m \ll k_3M$, förenklas till

$$\frac{d(RD_{tot})}{dt} = k_{-2}RT - k_3MRD - \eta \cdot k_{ROS} \cdot RDI_c \quad (33)$$

Det som är kvar att göra är att skriva RD , RDI_m och RDI_c i termer av RT och RD_{tot} . Detta kan göras med hjälp av sambanden i ekvation (22), (24) samt (25). Först skrivs RDI_m i termer av RDI_c enligt

$$RDI_m = \frac{k_6I_0}{k_{-6}}RD \quad (34)$$

och därefter skrivs RDI_c i termer av RD , baserat från resultatet i ekvation (24):

$$RDI_c = \frac{k_{-5}k_6I_0}{k_5k_{-6}}RD \quad (35)$$

Omskrivningen av RDI_c används sedan i ekvation (25) och RD löses ut. Resultatet blir

$$RD = RD_{tot} \frac{k_5/k_{-5}}{\frac{k_5}{k_{-5}} + \frac{k_6}{k_{-6}}I_0} \quad (36)$$

och detta samband kan sedan användas för att skriva (34) och (35) i termer av RD_{tot} . Även E_c beskriven i ekvation (21) kan omformuleras med hjälp av ekvationerna (27), (28). Ansättning av dessa i (21) ger:

$$E_c = E_c^0 - \int_S \left(E_c \left(\left(\frac{k_4k_7}{k_{-4}k_{-1}} \right) RT^2 + \left(\frac{k_4k_1 + k_7k_{-1} + k_7k_{-4}}{k_{-4}k_{-1}} \right) RT + \left(\frac{k_1}{k_{-1}} \right) \right) \right) dS \quad (37)$$

Då varken E_c eller RT beror av S kan E_c lösas ut ur högerled i ekvation (37). Kvar fås

$$E_c = \frac{E_c^0}{1 + \left(\left(\frac{k_4k_7}{k_{-4}k_{-1}} \right) RT^2 + \left(\frac{k_4k_1 + k_7k_{-1} + k_7k_{-4}}{k_{-4}k_{-1}} \right) RT + \left(\frac{k_1}{k_{-1}} \right) \right) \int_S dS} \quad (38)$$

där $\int_S dS$ är cellens area. Innan alla samband ansätts i ekvation (29) och (33) införs nya parametrar för att förenkla notationen. De nya parametrarna definieras som

$$\begin{aligned}A_1 &= E_c^0 \cdot k_3 \cdot \left(\frac{k_4k_7}{k_{-4}k_{-1}} \right) \left(\frac{k_5/k_{-5}}{\frac{k_5}{k_{-5}} + \frac{k_6}{k_{-6}}I_0} \right) & A_2 &= E_c^0 \cdot k_3 \cdot \left(\frac{k_1k_4}{k_{-4}k_{-1}} + \frac{k_7}{k_{-4}} \right) \left(\frac{k_5/k_{-5}}{\frac{k_5}{k_{-5}} + \frac{k_6}{k_{-6}}I_0} \right) \\ A_3 &= k_{-2} & A_4 &= \eta \cdot k_{ROS} \cdot \left(\frac{k_{-5}k_6I_0}{k_5k_{-6}} \right) \left(\frac{k_5/k_{-5}}{\frac{k_5}{k_{-5}} + \frac{k_6}{k_{-6}}I_0} \right) \\ A_5 &= \left(\frac{k_4k_7}{k_{-4}k_{-1}} \right) \int_S dS & A_6 &= \left(\frac{k_4k_1 + k_7k_{-1} + k_7k_{-4}}{k_{-4}k_{-1}} \right) \int_S dS \\ A_7 &= \left(\frac{k_1}{k_{-1}} \right) \int_S dS\end{aligned}$$

Med hjälp av ekvationerna (28), (34), (35), (36) samt de definierade parametrarna ovan kan ekvation (29), (33) och (38) skrivas som

$$\begin{aligned}\frac{d(RT)}{dt} &= E_c RT \cdot RD_{tot}(A_1 RT + A_2) - A_3 RT \\ \frac{d(RD_{tot})}{dt} &= A_3 RT - E_c RT \cdot RD_{tot}(A_1 RT + A_2) - A_4 RD_{tot} \\ E_c &= \frac{1}{1 + A_5 RT^2 + A_6 RT + A_7}\end{aligned}$$

vilket är det slutliga ODE-systemet.

Den andra modellen är varianten av Goryachev där ROS reagerar med både cytosoliskt och membranbundet Cdc42. Denna skillnad leder till att i varje ekvation som berör någon specie av Cdc42 (RT , M , RD , RDI_m och RDI_c) finns det en extra term som beskriver hur ROS påverkar koncentrationen av respektive specie. Det initiala ODE-systemet blir därmed

$$\frac{d(RT)}{dt} = (k_2 E_m + k_3 M)RD - k_{-2}RT - k_4 E_m RT + k_{-4}M - k_7 E_c RT - k_{ROS} \cdot RT \quad (39)$$

$$\frac{d(M)}{dt} = k_4 E_m RT - k_{-4}M + k_7 E_c RT - k_{ROS} \cdot M \quad (40)$$

$$\frac{d(E_m)}{dt} = k_1 E_c - k_{-1}E_m - k_4 E_m RT + k_{-4}M \quad (41)$$

$$\frac{d(E_c)}{dt} = \eta(k_{-1}E_m - (k_1 + k_7 RT) \cdot E_c) \quad (42)$$

$$\frac{d(RD)}{dt} = k_{-2}RT - (k_2 E_m + k_3 M)RD + k_{-6}RDI_m - k_6 I \cdot RD - k_{ROS} \cdot RD \quad (43)$$

$$\frac{d(RDI_m)}{dt} = k_6 I \cdot RD - k_{-6}RDI_m + k_5 RDI_c - k_{-5}RDI_m - k_{ROS} \cdot RDI_m \quad (44)$$

$$\frac{d(RDI_c)}{dt} = \eta(k_{-5}RDI_m - k_5 RDI_c - k_{ROS} \cdot RDI_c) \quad (45)$$

$$\frac{d(I)}{dt} = \eta(k_{-6}RDI_m - k_6 I \cdot RD) \quad (46)$$

där alla hastighetskonstanter är demsamma som i ODE-systemet för den första härledningen och E_c kan skrivas som

$$E_c = E_c^0 - \int_S (M + E_m) dS \quad (47)$$

Alla antaganden som gäller för varianten av Goryachev där ROS attackerar cytosoliskt Cdc42 gäller även för denna variant, vilket betyder att ekvationerna (22), (23), (24) och (25) är exakt demsamma. Antagandet att ackumulationstermen i ekvation (40) och (42) utnyttjas och leder till att M och E_m kan skrivas som:

$$M = \frac{1}{k_{ROS} + k_{-4}}(k_4 E_m RT + k_7 E_c RT) \quad (48)$$

$$E_m = E_c \left(\frac{k_1}{k_{-1}} + \frac{k_7}{k_{-1}} RT \right) \quad (49)$$

Om ekvation (49) kombineras med (48) fås

$$M = E_c RT \left(\frac{k_1 k_4}{(k_{ROS} + k_{-4})k_{-1}} + \frac{k_4 k_7}{(k_{ROS} + k_{-4})k_{-1}} RT + \frac{k_7}{k_{ROS} + k_{-4}} \right) \quad (50)$$

$\frac{d(RT)}{dt}$ formuleras om genom att utnyttja att $k_{-4}M + k_{ROS}M = k_4 E_m RT + k_7 E_c RT$ enligt ekvation (40), samt att $k_2 E_m \ll k_3 M$. Ekvation (39) kan därmed skrivas som

$$\frac{d(RT)}{dt} = k_3 M \cdot RD - k_{-2}RT - k_{ROS}(M + RT) \quad (51)$$

där resultatet från ekvation (50) kan ansättas. Ekvation (43) och (45) adderas för att få ekvationen för $\frac{d(RD_{tot})}{dt}$ enligt

$$\begin{aligned}\frac{d(RD_{tot})}{dt} &= \frac{d(RD)}{dt} + \frac{d(RDI_c)}{dt} = \\ &= k_{-2}RT - (k_2E_m + k_3M)RD + k_{-6}RDI_m - k_6I_0RD - k_{ROS} \cdot RD + \\ &\eta(k_{-5}RDI_m - k_5RDI_c - k_{ROS} \cdot RDI_c)\end{aligned}\quad (52)$$

och antalet termer i ekvation (52) kan därefter reduceras, med hjälp av att ackumulations- termen är lika med noll i ekvationerna (46) och (44), vilket ger sambanden

$$k_{-6}RDI_m - k_6I_0RD = 0 \quad (53)$$

$$k_{ROS} \cdot RDI_m = k_5RDI_c - k_{-5}RDI_m \quad (54)$$

Ekvation (52) kan nu, med hjälp av ekvationerna (53) och (54) samt att $k_2E_m \ll k_3M$, förenklas till

$$\frac{d(RD_{tot})}{dt} = k_{-2}RT - k_3MRD - k_{ROS}(RD + \eta(RDI_m + RDI_c)) \quad (55)$$

RD , RDI_m och RDI_c skrivs sedan i termer av RT och RD_{tot} enligt ekvationerna (34), (35) tillsammans med ekvation (36), precis som i varianten där ROS attackerar cytosoliskt Cdc42. Resultatet från ekvation (48) och (49) ansätts i ekvation (47) och ger uttrycket

$$E_c = E_c^0 - \int_S \left(E_c \left(\left(\frac{k_4k_7}{(k_{ROS} + k_{-4})k_{-1}} \right) RT^2 + \left(\frac{k_1k_4 + k_7k_{-1}}{(k_{ROS} + k_{-4})k_{-1}} + \frac{k_7}{k_{-7}} \right) RT + \left(\frac{k_1}{k_{-1}} \right) \right) \right) dS$$

som vid en omkastning av termer ger

$$E_c = \frac{E_c^0}{1 + \left(\left(\frac{k_4k_7}{(k_{ROS} + k_{-4})k_{-1}} \right) RT^2 + \left(\frac{k_1k_4 + k_7k_{-1}}{(k_{ROS} + k_{-4})k_{-1}} + \frac{k_7}{k_{-7}} \right) RT + \left(\frac{k_1}{k_{-1}} \right) \right) \int_S dS} \quad (56)$$

Nya parametrar definieras enligt

$$\begin{aligned}B_1 &= E_c^0 \cdot k_3 \cdot \left(\frac{k_5/k_{-5}}{\frac{k_5}{k_{-5}} + \frac{k_6}{k_{-6}}I_0} \right) \left(\frac{k_4k_7}{(k_{ROS} + k_{-4})k_{-1}} \right) \\ B_2 &= E_c^0 \cdot k_3 \cdot \left(\frac{k_5/k_{-5}}{\frac{k_5}{k_{-5}} + \frac{k_6}{k_{-6}}I_0} \right) \left(\frac{k_1k_4}{(k_{ROS} + k_{-4})k_{-1}} + \frac{k_7}{k_{ROS} + k_{-4}} \right) \\ B_3 &= k_{-2} \\ B_4 &= k_{ROS} \\ B_5 &= E_c^0 \cdot k_{ROS} \cdot k_3 \cdot \left(\frac{k_5/k_{-5}}{\frac{k_5}{k_{-5}} + \frac{k_6}{k_{-6}}I_0} \right) \left(\frac{k_4k_7}{(k_{ROS} + k_{-4})k_{-1}} \right) \\ B_6 &= E_c^0 \cdot k_{ROS} \cdot \left(\frac{k_5/k_{-5}}{\frac{k_5}{k_{-5}} + \frac{k_6}{k_{-6}}I_0} \right) \left(\frac{k_1k_4}{(k_{ROS} + k_{-4})k_{-1}} + \frac{k_7}{k_{ROS} + k_{-4}} \right) \\ B_7 &= k_{ROS} \cdot \left(\frac{k_5/k_{-5}}{\frac{k_5}{k_{-5}} + \frac{k_6}{k_{-6}}I_0} \right) \left(1 + \frac{k_{-5}k_6I_0}{k_5k_{-6}}\eta \left(1 + \frac{k_5}{k_{-5}} \right) \right) \\ B_8 &= \left(\frac{k_4k_7}{(k_{ROS} + k_{-4})k_{-1}} \right) \int_S dS \\ B_9 &= \left(\frac{k_1k_4 + k_7k_{-1}}{(k_{ROS} + k_{-4})k_{-1}} + \frac{k_7}{k_{-7}} \right) \int_S dS \\ B_{10} &= \left(\frac{k_1}{k_{-1}} \right) \int_S dS\end{aligned}$$

och det slutgiltiga ODE-systemet blir:

$$\begin{aligned}\frac{d(RT)}{dt} &= E_c RT \cdot RD_{tot}(B_1 RT + B_2) - B_3 RT - RT(B_4 + E_c(B_5 RT + B_6)) \\ \frac{d(RD_{tot})}{dt} &= B_3 RT - E_c RT \cdot RD_{tot}(B_1 RT + B_2) - B_7 RD_{tot} \\ E_c &= \frac{1}{1 + B_8 RT^2 + B_9 RT + B_{10}}\end{aligned}$$

B Data

I Tabell 6 presenteras den data som använts som testdata vid MATLAB-programmeringen. Den är inhämtad och tolkad från en artikel skriven av Felipe O. Bendezú från 2015 [46]. Den beskriver koncentration av GFP-tagat Cdc42 i membranet under FRAP.

I Tabell 7 presenteras den rådata som använts i parameteruppskattningen. Datan är uppdelad i två organismer och är inhämtad från en artikel skriven av Roland Wedlich-Soldner från 2004 [47]. Datan beskriver intensitet av ljussignal från inmärkt Cdc42 i membranet hos enskilda celler under polarisering. Intensiteten är angiven relativt den maxintensitet som nås när polariseringen är fullbordad.

I tabell 8 presenteras data från tabell 7 i bearbetad form. Bearbetningen utfördes enligt följande antaganden:

1. Intensitet i ljusstyrka för en fluorescent protein-tag varierar linjärt med koncentration hos proteinet
2. Den initiala koncentrationen av proteinet i membranet är uniform över hela cellen
3. Cytosolisk volym är 100 gånger större än membranvolym

Vidare beräknades ett förhållande mellan proteinkoncentration i membranet och proteinkoncentration i cytosolen, M/C , samt koncentrationen i normaliserad form.

Tabell 6: Testdata för MATLAB-kod angiven som koncentration Cdc42 i membranet angiven i molar mot tid angiven i sekunder [46].

Tid [s]	Koncentration membranbundet Cdc42 [M]	Tid [s]	Koncentration membranbundet Cdc42 [M]
0.4320496329	0.0370867027	⋮	⋮
0.5799570757	0.0781201322	26.03743811	0.7671758858
0.7771669995	0.1244785106	26.92824084	0.7880792436
0.9250744423	0.1721524645	27.91700889	0.8172439171
1.287592684	0.214044519	27.81957779	0.7882177252
1.615309175	0.2543132932	28.9781861	0.8124220383
2.02270336	0.3006634287	29.03734907	0.794357078
1.763216618	0.2738903111	30.16144564	0.7991808552
2.552056313	0.3412352527	31.00737242	0.7683851757
2.910148017	0.3821038231	31.4186589	0.7838324732
2.650661275	0.3031253245	32.08424239	0.7712335746
3.82633582	0.4234707752	32.88605643	0.8042090577
3.513454691	0.3794495916	32.62656968	0.7687973235
4.099395715	0.4580323438	33.90843419	0.8107704498
5.139262694	0.4781859484	34.99308877	0.8083784942
4.622760513	0.454312108	36.07774335	0.8067838571
6.047602212	0.5128865456	37.16239793	0.803082021
5.707415093	0.4731060452	38.20268028	0.8149677625
6.935046868	0.556335344	39.1788694	0.7991182088
6.644162231	0.5138262424	40.18464001	0.7962364718
7.975329216	0.5684995142	41.30380633	0.79186261
9.059983797	0.5864961329	42.32618409	0.7906906117
10.12116101	0.6070382045	43.1773006	0.7804652261
11.15416537	0.6308160154	44.00448667	0.7975450866
12.21534258	0.6405998914	45.07427225	0.7966004178
13.29999716	0.6570588243	46.13544946	0.7993801849
14.3600005	0.6710062037	47.22010404	0.7950518842
15.46930632	0.6794976462	48.30475862	0.794311517
16.5539609	0.7023920788	49.3894132	0.8016012865
17.63861548	0.7189649143	50.38162563	0.8031745669
18.7380608	0.7235039349	51.05953474	0.7861034073
19.80792464	0.7388409266	52.05174717	0.794311517
20.89257922	0.7431692273	52.98356406	0.8281772155
21.9772338	0.7414606876	53.97947418	0.8245902583
23.08536575	0.7676040576	54.9556633	0.800740056
22.56886357	0.7462445989	56.04031788	0.8296270335
24.14654296	0.7728978189	55.53578916	0.8005381951
25.16892073	0.7802715167	57.036228	0.8444115974
⋮	⋮	57.67223	0.8502377179

Tabell 7: Rådata för parameteruppskattning. Intensitet av ljussignal från MG-taggar på Cdc42 i procent av maximal intensitet mot tid i sekunder [47].

Organism 1		Organism 2	
Tid [s]	Intensitet membranbundet Cdc42 [% av max]	Tid [s]	Intensitet membranbundet Cdc42 [% av max]
0.27247956403269313	69.13043478260869	0.27247956403273577	63.04347826086956
16.07629427792918	69.34782608695653	20.435967302452326	63.69565217391304
29.15531335149865	70.65217391304348	37.87465940054497	65
40.054495912806544	69.8913043478261	54.223433242506815	64.56521739130434
53.13351498637604	71.19565217391305	77.11171662125344	65.65217391304348
69.48228882833789	72.28260869565219	96.73024523160765	65.97826086956522
95.64032697547688	71.84782608695653	119.61852861035422	66.73913043478261
117.43869209809264	71.52173913043478	130.51771117166214	68.58695652173913
131.60762942779292	73.04347826086956	147.95640326975476	69.78260869565217
150.13623978201636	72.28260869565219	167.574931880109	70.76086956521739
159.94550408719348	73.47826086956522	185.01362397820165	71.6304347826087
168.66485013623978	73.15217391304347	198.09264305177115	71.52173913043478
180.65395095367845	73.80434782608697	206.81198910081744	73.69565217391305
190.46321525885563	73.15217391304347	218.80108991825617	75.76086956521739
199.18256130790192	74.34782608695653	229.70027247956406	77.5
204.63215258855584	76.08695652173913	235.14986376021798	79.56521739130434
213.3514986376022	77.6086956521739	236.23978201634876	81.63043478260869
223.16076294277931	79.34782608695653	249.31880108991825	81.08695652173913
232.97002724795638	80.97826086956522	263.4877384196185	83.15217391304348
242.7792915531335	82.6086956521739	274.38692098092645	84.56521739130434
254.76839237057223	84.02173913043478	285.2861035422343	86.84782608695653
264.5776566757494	85.65217391304348	292.9155313351499	89.13043478260869
280.9264305177112	88.47826086956522	301.6348773841961	91.30434782608695
288.5558583106267	90.86956521739131	312.53405994550405	93.47826086956522
313.62397820163494	91.41304347826087	321.25340599455046	95.43478260869566
334.33242506811996	93.15217391304347	338.69209809264305	96.52173913043478
358.3106267029973	93.69565217391305	359.40054495912807	95.54347826086956
382.28882833787463	94.78260869565217	371.3896457765668	96.73913043478261
394.27792915531336	94.45652173913044	381.19891008174386	97.6086956521739
400.8174386920981	95.43478260869566	397.5476839237057	96.41304347826087
412.8065395095367	94.56521739130434	412.8065395095367	97.28260869565217
431.3351498637602	96.30434782608695	431.3351498637602	97.71739130434783
443.3242506811989	96.41304347826087	448.7738419618528	98.80434782608695
453.133514986376	97.28260869565217	466.2125340599455	98.26086956521739
464.03269754768394	96.63043478260869	480.38147138964575	99.13043478260869
479.291553133515	97.6086956521739	497.82016348773845	99.45652173913044
498.9100817438691	99.45652173913044		

Tabell 8: Behandlad data från tabell 7. M indikerar membranbundet Cdc42, C indikerar cytosoliskt Cdc42, M/C kvoten dem emellan. Normaliserad form av intensiteten presenteras även för båda specierna

Organism 1					
M intensitet	M normaliserad	C intensitet	C normaliserad	M/C ratio	Tid [s]
69.13043478	1.00	69.13043478	1	1	0.27247956403269313
69.34782609	1.003144654	69.12826087	0.999968553	1.003176201	16.07629427792918
70.65217391	1.022012579	69.11521739	0.999779874	1.022237599	29.15531335149865
69.89130435	1.011006289	69.12282609	0.999889937	1.011117576	40.054495912806544
71.19565217	1.029874214	69.10978261	0.999701258	1.030181973	53.13351498637604
72.2826087	1.045597484	69.09891304	0.999544025	1.046074468	69.48228882833789
71.84782609	1.039308176	69.10326087	0.999606918	1.03971687	95.64032697547688
71.52173913	1.034591195	69.10652174	0.999654088	1.034949196	117.43869209809264
73.04347826	1.056603774	69.09130435	0.999433962	1.05720219	131.60762942779292
72.2826087	1.045597484	69.09891304	0.999544025	1.046074468	150.13623978201636
73.47826087	1.062893082	69.08695652	0.999371069	1.063561989	159.94550408719348
73.15217391	1.058176101	69.09021739	0.999418239	1.058792065	168.66485013623978
73.80434783	1.067610063	69.08369565	0.999323899	1.068332363	180.65395095367845
73.15217391	1.058176101	69.09021739	0.999418239	1.058792065	190.46321525885563
74.34782609	1.075471698	69.07826087	0.999245283	1.076283988	199.18256130790192
76.08695652	1.100628931	69.06086957	0.998993711	1.101737598	204.63215258855584
77.60869565	1.122641509	69.04565217	0.998773585	1.124020025	213.3514986376022
79.34782609	1.147798742	69.02826087	0.998522013	1.149497685	223.16076294277931
80.97826087	1.171383648	69.01195652	0.998286164	1.173394654	232.97002724795638
82.60869565	1.194968553	68.99565217	0.998050314	1.197302918	242.7792915531335
84.02173913	1.215408805	68.98152174	0.997845912	1.218032554	254.76839237057223
85.65217391	1.238993711	68.96521739	0.997610063	1.241961922	264.5776566757494
88.47826087	1.279874214	68.93695652	0.997201258	1.283466305	280.9264305177112
90.86956522	1.314465409	68.91304348	0.996855346	1.318611987	288.5558583106267
91.41304348	1.322327044	68.9076087	0.99677673	1.326603044	313.62397820163494
93.15217391	1.347484277	68.89021739	0.996525157	1.3521829	334.33242506811996
93.69565217	1.355345912	68.88478261	0.996446541	1.360179253	358.3106267029973
94.7826087	1.371069182	68.87391304	0.996289308	1.376175746	382.28882833787463
94.45652174	1.366352201	68.87717391	0.996336478	1.371376268	394.27792915531336
95.43478261	1.380503145	68.8673913	0.996194969	1.385776066	400.8174386920981
94.56521739	1.367924528	68.87608696	0.996320755	1.372976044	412.8065395095367
96.30434783	1.393081761	68.85869565	0.996069182	1.398579321	431.3351498637602
96.41304348	1.394654088	68.8576087	0.996053459	1.400179955	443.3242506811989
97.2826087	1.407232704	68.84891304	0.995927673	1.412986849	453.133514986376
96.63043478	1.397798742	68.85543478	0.996022013	1.403381376	464.03269754768394
97.60869565	1.411949686	68.84565217	0.995880503	1.417790268	479.291553133515
99.45652174	1.438679245	68.82717391	0.995613208	1.44501824	498.9100817438691

Organism 2					
M Intensity	M normalized	C intensity	C normalised	M/C intensity	Time [s]
63.04347826	1	63.04347826	1	1	0.272479564
63.69565217	1.010344828	63.03695652	0.999896552	1.010449357	20.4359673
65	1.031034483	63.02391304	0.999689655	1.031354558	37.8746594
64.56521739	1.024137931	63.02826087	0.999758621	1.024385196	54.22343324
65.65217391	1.04137931	63.0173913	0.999586207	1.041810404	77.11171662
65.97826087	1.046551724	63.01413043	0.999534483	1.047039139	96.73024523
66.73913043	1.05862069	63.00652174	0.999413793	1.059241624	119.6185286
68.58695652	1.087931034	62.98804348	0.99912069	1.088888505	130.5177112
69.7826087	1.106896552	62.97608696	0.998931034	1.108081052	147.9564033
70.76086957	1.122413793	62.96630435	0.998775862	1.123789466	167.5749319
71.63043478	1.136206897	62.9576087	0.998637931	1.1377566	185.013624
71.52173913	1.134482759	62.95869565	0.998655172	1.136010497	198.0926431
73.69565217	1.168965517	62.93695652	0.998310345	1.170944009	206.8119891
75.76086957	1.201724138	62.91630435	0.997982759	1.204153206	218.8010899
77.5	1.229310345	62.89891304	0.997706897	1.23213576	229.7002725
79.56521739	1.262068966	62.87826087	0.99737931	1.265385147	235.1498638
81.63043478	1.294827586	62.8576087	0.997051724	1.298656383	236.239782
81.08695652	1.286206897	62.86304348	0.997137931	1.289898676	249.3188011
83.15217391	1.318965517	62.8423913	0.996810345	1.323186024	263.4877384
84.56521739	1.34137931	62.82826087	0.996586207	1.345974188	274.386921
86.84782609	1.377586207	62.80543478	0.996224138	1.382807497	285.2861035
89.13043478	1.413793103	62.7826087	0.995862069	1.41966759	292.9155313
91.30434783	1.448275862	62.76086957	0.995517241	1.454797368	301.6348774
93.47826087	1.482758621	62.73913043	0.995172414	1.48995149	312.5340599
95.43478261	1.513793103	62.71956522	0.994862069	1.521611036	321.253406
96.52173913	1.531034483	62.70869565	0.994689655	1.539208209	338.6920981
95.54347826	1.515517241	62.71847826	0.994844828	1.523370479	359.400545
96.73913043	1.534482759	62.70652174	0.994655172	1.542728376	371.3896458
97.60869565	1.548275862	62.69782609	0.994517241	1.556811484	381.1989101
96.41304348	1.529310345	62.70978261	0.994706897	1.537448217	397.5476839
97.2826087	1.543103448	62.70108696	0.994568966	1.55152986	412.8065395
97.7173913	1.55	62.69673913	0.9945	1.558572147	431.3351499
98.80434783	1.567241379	62.68586957	0.994327586	1.576182137	448.773842
98.26086957	1.55862069	62.69130435	0.994413793	1.567376378	466.2125341
99.13043478	1.572413793	62.6826087	0.994275862	1.581466324	480.3814714
99.45652174	1.577586207	62.67934783	0.994224138	1.586751062	497.8201635

C Kod

main.m

```
% Instructions for parameter estimation program:
% Where the comment says "USER:" the script needs to be edited by the user
% of this program during normal operation. Anywhere is doesn't say that it
% does not, unless the user want to do something out of the ordinary.

%In main.m USER required inputs are data, ODEs, and initial guesses for
%parameters

clc, clear, close all, format compact

% USER: Insert initial guesses for parameters in model

in(1)=log(0.1);
in(2)=log(0.1);
in(3)=log(0.1);
in(4)=log(0.1);

in=in';

% USER: Insert your data into function file "datafile"
data = datafile; % Retrieves the data from function "datafile".
%data = Datagenerator(); % For development
% See function "datafile" for more information.
h=figure('Position',[1100 550 600 400]); hold on % Create figure for plot
plot(data(:,1),data(:,2),'o','color',[0 0.8 0]), hold on % Plots datapoints
title('Title') %USER: add plot details
legend('type of data','Location','east')
xlabel('time')
ylabel('dataform')
axis([min(data(:,1)) max(data(:,1)) -0.5*max(data(:,2)) 1.5*max(data(:,2))])

syms x1(t) x2(t) f1 f2
k=sym('k', [1 length(in)]);

% k(1)=kcat , k(2)=kc , k(3)=km , k(4)=kROS = kR*[ROS]
f1 = exp(k(1))*x1*x2-exp(k(2))*x1+exp(k(3))*x2-exp(k(4))*x1; %dxm/dt
f2 = -exp(k(1))*x1*x2+exp(k(2))*x1-exp(k(3))*x2-exp(k(4))*x2; %dxc/dt

%Suppress warnings about nearly singular matrix
warning('off','MATLAB:nearlySingularMatrix')
warning('off','MATLAB:illConditionedMatrix')

[CF,gradCF,k,gradCFfinal,CFfinal]=parameterestimation(f1,f2,in,data);

gradCFfinal;
CFfinal;
k;
```

parameterestimation.m

```
function [CF,gradCF,in,gradCFfinal,CFfinal]=parameterestimation(f1,f2,in,data)
%This function computes sensitivity equations and performs the descent algorithm for the parameters
%In parameterestimation.m USER required inputs are the ammounts of
%sensitivities and parameters along with plot details.

k=sym('k', [1 length(in)]);
syms x1(t) x2(t)
%USER: insert sensitivities depending on no. parameters
syms s11(t) s12(t) s13(t) s14(t) s21(t) s22(t) s23(t) s24(t)
syms x2C x1C

%sens. eq. (index1:specie index2: parameter)
% remeber: x1=xm , x2=xc
% f1 = dxm/dt , f2 = dxc/dt --> f index corresponds to specie
f1x1C = subs(f1,x1(t),x1C);
f1x2C = subs(f1,x2(t),x2C);
f2x1C = subs(f2,x1(t),x1C);
f2x2C = subs(f2,x2(t),x2C);
df1dx1=subs(diff(f1x1C,x1C),x1C,x1(t));
df1dx2=subs(diff(f1x2C,x2C),x2C,x2(t));
df2dx1=subs(diff(f2x1C,x1C),x1C,x1(t));
df2dx2=subs(diff(f2x2C,x2C),x2C,x2(t));

%USER: comment sensitivity eq. that are not needed.
eqs = [diff(x1(t),t) == f1 %model membrane
diff(x2(t),t) == f2 %model cytosole
diff(s11(t),t) == diff(f1,k(1))+df1dx1*s11(t)+df1dx2*s21(t)
diff(s12(t),t) == diff(f1,k(2))+df1dx1*s12(t)+df1dx2*s22(t)
diff(s13(t),t) == diff(f1,k(3))+df1dx1*s13(t)+df1dx2*s23(t)
diff(s14(t),t) == diff(f1,k(4))+df1dx1*s14(t)+df1dx2*s24(t)
diff(s21(t),t) == diff(f2,k(1))+df2dx1*s11(t)+df2dx2*s21(t)
diff(s22(t),t) == diff(f2,k(2))+df2dx1*s12(t)+df2dx2*s22(t)
diff(s23(t),t) == diff(f2,k(3))+df2dx1*s13(t)+df2dx2*s23(t)
diff(s24(t),t) == diff(f2,k(4))+df2dx1*s14(t)+df2dx2*s24(t)];
%USER: insert all species and sensitivities
vars = [x1(t) x2(t) s11(t) s12(t) s13(t) s14(t) s21(t) s22(t) s23(t) s24(t)];
[M,F1] = massMatrixForm(eqs,vars);
F=F1;
%USER: insert all parameters next to vars
F = odeFunction(F,vars,k(1),k(2),k(3),k(4), 'File', 'Eqsys');

% Preparation step for the iteration
% Initial and temporary CF larger than any resonable updated CF from loop below.
CF=1000;
gradCF=1000;
% This is to make the plot window appear before the heavy calculations start
pause(0.01);
n=linspace(0,100,101);
```

```

for i=1:10000
    disp(' ')
    i
    CF0=CF;
    gradCF0=gradCF;
    [CF, gradCF, t, yhat]=sensitivities(in, data, F);
    in;
    parameters=exp(in)
    CF
    gradCF

%USER: choose colorvalue change between iterations or comment if not needed.
colorvalue=[0 0 1]+[0.07*i 0 -0.07*i];

% USER: if only every tenth iteration needs to be plotted, uncomment this
% section
% if mod(i,10)==1
plot(t, yhat, 'color', colorvalue)
% end

%Break requirements
if CF<0.2
    break
end

% %uncomment if NaN issues
% if CF>35
%     disp('CF>35')
%     break
% elseif isnan(CF)
% disp('The costfunction is NaN!')
%     break
% end

% % If only the latest simulated data needs to be plotted, uncomment
% hold off
% plot(data(:,1), data(:,2), 'ob'), hold on % Plots datapoints
% title('Title')
% axis([min(data(:,1)) max(data(:,1)) -0.5*max(data(:,2)) 1.5*max(data(:,2))])
% plot(t, yhat)

H=gradCF*gradCF'
format long, detH = det(H), format short
Hinv=inv(H);
I=H*Hinv
format long, detI=det(I), format short

P=H\(-gradCF)
in=in+(CF/norm(gradCF))*P;

pause(0.1) %This is to make the plot update on every iteration,
           %try removing and see what happens.

end

CFfinal=CF;
gradCFfinal=gradCF;
end

```

sensitivities.m

```
function [CF,gradCF,t,yhat]=sensitivities(guess,data,F)
%This function calculates the sensitivities
%In sensitivities.m USER required inputs are the ammount of parameters,
%form of the data along with the derivative of this form.
kparam=guess;

t=data(:,1);
y=data(:,2);

param1 = kparam(1); %USER: insert ammount of parameters
param2 = kparam(2);
param3 = kparam(3);
param4 = kparam(4);
tspan = [0 data(end,1)];
y0 = zeros(10,1);
y0(1)=data(1,2);
y0(2)=1-data(1,2);

%USER: insert ammount of parameters in the arguments
[T, SENS]=ode23tb(@(t,in2)F(t,in2,param1,param2,param3,param4), tspan, y0);

yhat=SENS(:,1)./(SENS(:,1)+SENS(:,2)); %USER: insert the shape of the data

dydk=ones(length(T),length(kparam));

    for i=1:length(kparam)

        xmt=SENS(:,1);
        xct=SENS(:,2);
        xmdki=SENS(:,i+2);
        xcdki=SENS(:,i+2+length(kparam));

        %USER: insert derivative of the dataform expression
        dydk(:,i)=(xmdki.*(xct+xmt)-(xmdki+xcdki).*xmt)./(xct+xmt).^2;

    end

dydk;

dydk=interp1(T,dydk,t);
yhat=interp1(T,yhat,t);

[CF,gradCF]=costfunction(length(kparam),t,y,yhat,dydk);

end
```

costfunction.m

```
function [CF,gradCF]=costfunction(nk,t,y,yhat,dydk)
%This function computes CF and gradCF

gradCF=zeros(nk,1); %creates gradient of CF

for i=1:nk
gradCF(i)=(-2)*sum((y-yhat).*dydk(:,i));

end

CF=(y-yhat) '*(y-yhat);
%gradCF(nk+1:n)=-((y-yhat).^2)./(eparam.^2);

end
```


datagenerator.m

```
function dataSim = Datagenerator
%This function generates data according to our model for debugging purposes

data = datafile;
k(1)=0.019;
k(2)=0.01;
k(3)=0.14;
k(4)=0.0001;
tspan=[0 data(end,1)];
y0(1)=data(1,2);
y0(2)=1-data(1,2);

F = @(t,x,k) [k(1)*x(1)*x(2)-k(2)*x(1)+k(3)*x(2)-k(4)*x(1); %dxm/dt
             -k(1)*x(1)*x(2)+k(2)*x(1)-k(3)*x(2)-k(4)*x(2)]; %dxc/dt

[t,SENS]=ode45(@(t,x)F(t,x,k), tspan, y0);

yhat=SENS(:,1)/(SENS(:,1)+SENS(:,2));
figure('Position',[1100 50 600 400]); hold on
plot(t,yhat)

brus=normrnd(0,0.02,[length(yhat),1]);
yhatBrus=yhat+brus;
plot(t,yhatBrus,'LineStyle','none','Marker','o','Color',[1 0.5 0])
title('data plotted from "datagenerator"')
legend('data from ODEs','with normrnd noise','Location','east')
axis([min(tspan) max(tspan) -0.5*max(yhatBrus) 1.5*max(yhatBrus)])

dataSim=[t yhatBrus];

end

% Parameters used for generating data
% k(1)=0.019;
% k(2)=0.01;
% k(3)=0.14;
% k(4)=0.0001;
```

datafile.m

```
function data = datafile()

% This function is just to keep long data matrices out of the main script
% of the program.

% Insert data for parameter estimation below
% (data=[Time vector, y-vector], px2 matrix, p=number of timepoints)
% (Do not change variable name "data" in this function, it will break it.)

data=[0.4320496329408101, 0.03708670269684511
0.5799570757499328, 0.07812013220095615
0.777166999495428, 0.12447851057201276
0.9250744423045507, 0.17215246454549127
1.2875926844837728, 0.21404451902261146
1.6153091754137883, 0.2543132932396006
2.0227033599932973, 0.3006634286557548
1.7632166182229092, 0.2738903111327833
2.5520563132048935, 0.3412352526853317
2.91014801684803, 0.3821038230913949
2.650661275077642, 0.3031253245199361
3.826335820483486, 0.4234707752323257
3.5134546914641867, 0.37944959161282443
4.099395714900325, 0.4580323438058411
5.139262693770807, 0.47818594841461026
4.622760512532604, 0.45431210803635513
6.047602211593812, 0.5128865455727578
5.707415093132834, 0.4731060452138105
6.935046868448545, 0.5563353440400456
6.644162230923939, 0.5138262424316306
7.975329216206038, 0.5684995142205916
9.059983796806268, 0.5864961328511248
10.12116100553203, 0.6070382045025466
11.154165368008442, 0.6308160153907644
12.215342576734205, 0.6405998913922843
13.299997157334435, 0.6570588242537528
14.360000497466476, 0.6710062036924159
15.46930631853489, 0.679497646217139
16.55396089913512, 0.7023920787787664
17.63861547973535, 0.7189649142897954
18.738060804616488, 0.7235039348747747
19.80792464093581, 0.7388409266380739
20.89257922153604, 0.7431692273213667
21.97723380213626, 0.7414606875779617
23.085365754610955, 0.7676040576199054
22.568863573372752, 0.7462445988594958
24.14654296333672, 0.7728978188566142
25.168920725912052, 0.7802715166965728
26.03743811454354, 0.7671758857551716
26.928240835115297, 0.7880792435801192
27.917008886375612, 0.8172439170854384
27.819577793096585, 0.7882177252224795
28.97818609510138, 0.8124220382540508
29.037349072225027, 0.7943570780337815
30.161445637574353, 0.7991808552426618
31.007372415745827, 0.7683851757273226
31.41865890145189, 0.7838324732144065
32.08424239409294, 0.7712335745880383
32.88605642616345, 0.8042090577331213
32.626569684393054, 0.7687973234724422
33.90843418873878, 0.810770449835426
34.99308876933901, 0.8083784941946589
36.07774334993924, 0.8067838571008141
37.162397930539456, 0.8030820209901033
38.20268027829695, 0.8149677624717244
39.17886940083717, 0.7991182087854036
40.18464001193918, 0.7962364717515271
41.30380632919488, 0.7918626100084103
42.32618409177021, 0.7906906116931973
```

43.1773006047771, 0.780465226136779
44.004486673820715, 0.7975450866364759
45.07427225223377, 0.7966004178365509
46.135449460959556, 0.7993801848793922
47.220104041559786, 0.7950518841960996
48.30475862216001, 0.7943115169739573
49.38941320276024, 0.8016012865458189
50.38162563160476, 0.8031745668928711
51.059534744479905, 0.7861034072900157
52.05174717332444, 0.7943115169739574
52.9835640630219, 0.8281772154593072
53.97947417793665, 0.8245902583065816
54.95566330047686, 0.8007400559566432
56.04031788107709, 0.8296270334701396
55.53578915949487, 0.8005381951499224
57.03622799599184, 0.8444115973830711
57.67223000007107, 0.8502377179080823];

end